ΑΤΕΙ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΠΟΛΥΜΕΣΩΝ

Πτυχιακή Εργασία

Δημιουργία εφαρμογής οικονομικής διαχείρισης συστημάτων

Σπουδαστές:

Μπικουβάρης Γιώργος
Χατζηδάκης Δημήτρης

Επιβλέπων Καθηγητής:
Γεώργιος Μαμάκης

Ηράκλειο 2010

Περιεχόμενα

Εισαγωγή

Κεφάλαιο 1: JBoss
- **1.1: Admin**
- **1.2: Hotelier**
- **1.3: Customer**

Κεφάλαιο 2: Web Services
- **2.1: hoteldbAdd**
- **2.2: hoteldbFill**
- **2.3: hoteldbDelete**
- **2.4: hoteldbOther**

Κεφάλαιο 3: Βαση Δεδομένων
- **3.1 Stored Procedures**

Εισαγωγή

Η συγκεκριμένη πτυχιακη εργασία έχει ως θέμα της την ανάπτυξη ενός ολοκληρομένου online συστήματος διαχείρησης ξενοδοχείων, κρατήσεων, πωλήσεων και πληρωμών μεσα από ένα φιλικό και εύχρηστο περιβάλλον διεπαφής χρησιμοποιόντας τις τελευταιες τεχνολογίες στην διαχείρηση και απεικόνηση διαδικτυακού περιεχομένου.

Συνδιαζεί και υποστηρίζει πληρως ολες τις διαδικασίες ενός Front-Office και Back-Office ταυτόχρονα. Front-Office γιατι αποτελλεί ένα ολοκληρωμένο σύστημα με το οποίο έρχονται σε επαφή οι πελάτες, πραγματοποιούν κρατήσεις και πληρωμές μεσα από απλουστευμένες και ταχύτατες διαδικασίες. Back-Office διότι περιλαμβάνει ένα πλήρες σύστημα διαχείρησης κρατήσεων, πληρωμών, διαθεσιμοτήτων, τιμών και περιεχομένου των ξενοδοχείων μέσα από εκτενώς αναλυτικές διαδικασίες που καλύπτουν ακομα και τις πιο απαιτητικες περιπτώσεις.

Σκοπός της εργασίας είναι, οι σπουδαστές, να γνωρίσουν και να εξοικιωθούν με μια σειρά από τεχνολογίες όπου αλλες χρησιμοποιούνται ευρέως και άλλες είναι νέες που βρίσκονται στο στάδιο της εξέλιξης με μεγάλη όμως αποδοχή και συνεχή βελτίωση.

Αναφορικά, οι σπουδαστες χρησιμοποίησαν αντικειμενοστραφή προγραμματισμό με γλώσσα προγραμματισμού τη Java.

Χρησιμοποιήθηκε το JDK 6 update 14 και από IDE’s χρησιποποιήθηκαν τα Eclipse και NetBeans. Η εφαρμογή που αναπτύχθηκε μεσω Eclipse είναι ένα web application portlet το οποίο στήθηκε πάνω στο JBoss Portal 2.4 με server τον JBoss 4.2.3 GA, Application Server.

Επίσης γίνεται χρήση web services οι οποίες υλοποιήθηκαν σε γλώσσα java μέσω NetBeans και στήθηκαν μέσω του Apache Axis 2 Core Engine στον Apache Tomcat 6.0.18 server.

Γίνεται χρήση της τεχνολογίας Java Servlet και JavaServer Pages (JSP). Σαν επέκταση της JSP τεχνολογίας υλοποιήθηκε και η JavaServer Faces (JSF) μεσω του RichFaces API 3.3.1 GA.

Για την δυναμικη αποθήκευση των δεδομένων χρησιμοποιήθηκε η βάση δεδομένων MySQL με τον MySQL Community Server 5.1.35 και το γραφικό περιβάλλον εργασίας MySQL GUI Tools 5.0 r17.

Επίσης χρησιμοποιήθηκαν οι τεχνολογίες Extensible MarkUp Lanfuage (XM) και το πρωτοκολλο Simple Object Access Protocol 1.2 (SOAP) . Για κωδικοποίηση χρησιμοποιήθηκε το Base64 πλάνο κωδικοποίησης.

Αναλυτικότερα οι τεχνολογίες περιγράφονται παρακατω.

JBOSS

Ο JBoss ειναι ενας application server ο οποίος ειναι ο πιο διαδεδομένος για Java εφαρμογες. Η έκδοση JBoss 4.2.3 GA που χρησιμοποιήσαμε ειναι φτιαγμένη απο τη κοινότητα, απο τη συνεργασία δηλαδη των χρηστών του internet που μοιράζονται το ίδιο ενδιαφέρον για εναν freeware Java Application Server. Το μεγαλύτερο πλεονέκτημα του server ειναι οτι επειδή ειναι βασισμένος σε Java μπορει να χρησιμοποιηθεί cross-platform επεκτείνοντας έτσι τις δυνατοτητες του σε οποιοδήποτε λειτουργικό σύστημα υποστηρίζει java.

JBOSS Portal 2.4

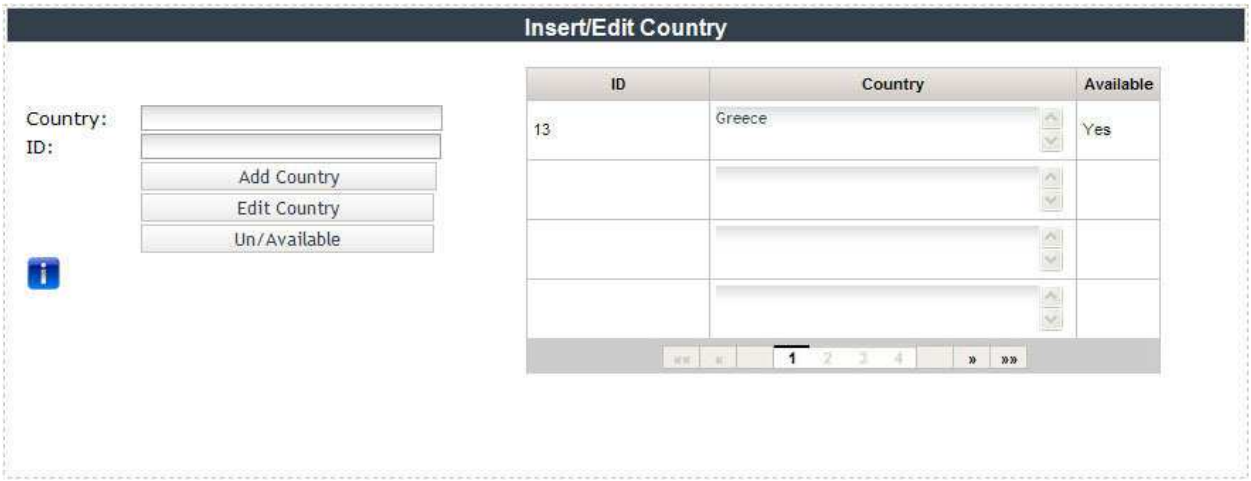
Η τεχνολογια JBoss Portal παρέχει μια open source πλατφορμα για την φιλοξενια και εξυπηρέτηση του web interface ενος portal και την έκδοση, προσαρμογή και διαχείρηση του περιεχομενου του.

Κεφάλαιο 1 - JBoss

1.1 Admin

1.1.1 Admin Management

Panel: Add/Edit Country



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής, διόρθωσης, προβολής, ενεργοποίησης και απενεργοποίησης των χωρών που είναι διαθέσιμες στο σύστημα.

Αποτελείται από

- ✓ Το πεδίο κειμένου **"Country"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Add Country"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το κουμπί **"Edit Country"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το πεδίο κειμένου **"ID"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Un/Available"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Τον **πίνακα** με τις χώρες ο οποίος είναι **"rich:dataTable"**
- ✓ Τον **scroller** ο οποίος είναι τύπου **"rich:dataScroller"**
- ✓ Το **tooltip** επεξήγησης το οποίο είναι τύπου **"rich:tooltip"**

Επεξήγηση Ρόλων

Το κουμπί **"Add/Edit Country"** είναι υπεύθυνο να παίρνει το κείμενο που έχει εισάγει ο χρήστης στο πεδίο **"Country"** και μετά από τους απαραίτητους ελέγχους να καταχωρεί τη χώρα στο σύστημα.

Το κουμπί **"Un/Available"** είναι υπεύθυνο να παίρνει τον αριθμό που έχει εισάγει ο χρήστης στο πεδίο **"ID"**, ο οποίος αντιστοιχεί στην αυτόματη αρίθμηση από τη βάση και είναι μοναδικός για κάθε χώρα και μετά από τους απαραίτητους ελέγχους να ενεργοποιεί ή να απενεργοποιεί τη χώρα από το σύστημα.

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τις χώρες του συστήματος. Οι τιμές που εμφανίζει είναι ο αύξων αριθμός ID, μοναδικός για κάθε χώρα, το όνομα της χώρας και τέλος το αν είναι διαθέσιμη ή όχι. Οι χώρες χωρίζονται σε σελίδες με τέσσερεις χώρες ανά σελίδα.

Ο **scroller** είναι υπεύθυνος στο να αλλάζει ο χρήστης τις σελίδες με τις χώρες που φαίνονται μέσα στο πίνακα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Κουμπί Add Country

Το κουμπί **"Add/Edit Country"** καλεί τη μέθοδο **"addCountry"** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τη μεταβλητή τύπου String με όνομα **"countryAdmin"**, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **"Country"**, και να επικοινωνήσει με τη μέθοδο **"addCountry", της web service "hoteldbAdd",** αποστέλλοντας της την τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
</root>
```

Όπου country_name το όνομα της χώρας που θέλει ο χρήστης να προστεθεί στη βάση.
Μετα από μια επιτυχημένη προσθήκη θα καλεστεί η μέθοδος **fillCountry** και θα ενεργοποιηθεί η διαδικασία για την ανανέωση των περιεχομένων του πίνακα.

Διαδικασία fillCountry

Καλείται η μέθοδος **"fillCountry"** από το bean. Αυτή η μέθοδος δεν δέχεται ορίσματα και σαν αποτέλεσμα δεν στέλνει κάποιο ωφέλημο μήνυμα. Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **"fillCountry", της web service "hoteldbFill",** με σκοπό να παραλάβει ένα μήνυμα με τις χώρες που είναι καταχωρημένες στη βάση και συγκεκριμένα στον πίνακα **"country".**
Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>7</id>
<country_name>Germany</country_name>
<flag>1</flag>
<id>8</id>
<country_name>Greece</country_name>
<flag>1</flag>
<id>11</id>
<country_name>Italy</country_name>
<flag>1</flag>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε χώρα στον πίνακα "**country"**. Το πεδίο **country_name** το οποίο αντιπροσωπεύει το όνομα της χώρας και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν η συγκεκριμένη χώρα είναι

ενεργοποιημένη ή όχι. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης

Κουμπί Edit Country

Το κουμπί **“Edit Country”** καλεί τη μέθοδο **“editCountry”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“countryAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Country”**
- **“selectedCountryAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“editCountry”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>NewValue</country_name>
<countryID>9</countryID>
</root>
```

Όπου **country_name** το νέο όνομα της χώρας την οποία διορθώνουμε και **countryID** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων που αντιστοιχεί στην χώρα.
Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **“fillCountry”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Un/Available

Το κουμπί **“Un/Available”** καλεί τη μέθοδο **“deleteCountry”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

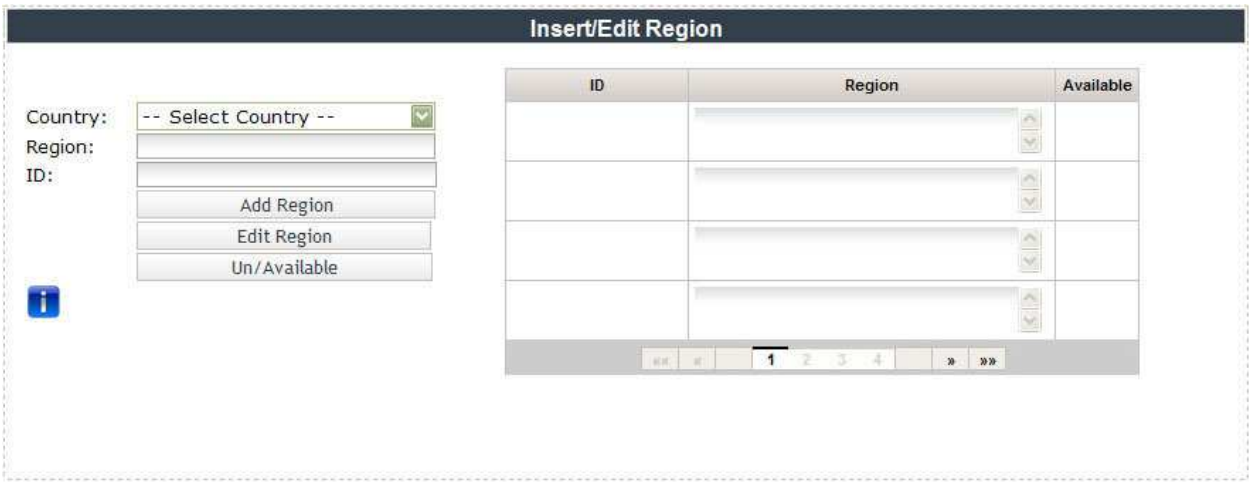
- **“selectedCountryAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“deleteCountry”, της web service “hoteldbDelete”,** αποστέλλοντας τη τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>9</country_name>
</root>
```

Όπου **country_name** το id της χώρας την οποία θέλουμε να απενεργοποιήσουμε / ενεργοποιήσουμε.
Μετά από μια επιτυχημένη αλλαγή θα καλεστεί η μέθοδος **“fillCountry”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Panel: Add/Edit Region



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής, διόρθωσης, προβολής, ενεργοποίησης και απενεργοποίησης των επαρχιών που είναι διαθέσιμες στο σύστημα.

Αποτελείται από

- ✓ Το μενού επιλογής **"Country"** το οποίο είναι ένα **"h:selectOneMenu"**
- ✓ Το πεδίο κειμένου **"Region"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Add Region"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το πεδίο κειμένου **"ID"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Un/Available"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το κουμπί **"Edit Region"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Τον **πίνακα** με τις επαρχίες ο οποίος είναι **"rich:dataTable"**
- ✓ Τον **scroller** ο οποίος είναι τύπου **"rich:dataScroller"**
- ✓ Το **tooltip** επεξήγησης το οποίο είναι τύπου **"rich:tooltip"**

Επεξήγηση Ρόλων

Το μενού επιλογής **"Country"** είναι υπεύθυνο να εμφανίζει τις χώρες που έχουν εισαχθεί στο σύστημα και όταν ο χρήστης επιλέξει μια τιμή από τη λίστα επιλογής τότε γεμίζει τον πίνακα με τις επαρχίες που έχουν έως τώρα εισαχθεί για την συγκεκριμένη χώρα.

Το κουμπί **"Add Region"** είναι υπεύθυνο να παίρνει το κείμενο που έχει εισάγει ο χρήστης στο πεδίο **"Region"** και μετά από τους απαραίτητους ελέγχους να καταχωρεί την επαρχία, κάτω από την επιλεγμένη χώρα, στο σύστημα.

Το κουμπί **"Un/Available"** είναι υπεύθυνο να παίρνει τον αριθμό που έχει εισάγει ο χρήστης στο πεδίο **"ID"**, ο οποίος αντιστοιχεί στην αυτόματη αρίθμηση από τη βάση και είναι μοναδικός για κάθε επαρχία και μετά από τους απαραίτητους ελέγχους να ενεργοποιεί ή να απενεργοποιεί την επαρχία από το σύστημα.

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τις επαρχίες του συστήματος, που αντιστοιχούν σε μια επιλεγμένη χώρα. Οι τιμές που εμφανίζει είναι ο αύξων αριθμός ID, μοναδικός για κάθε επαρχία, το όνομα της επαρχίας και τέλος το αν είναι διαθέσιμη ή όχι. Οι επαρχίες χωρίζονται σε σελίδες με τέσσερεις επαρχίες ανά σελίδα.

Ο **scroller** είναι υπεύθυνος στο να αλλάζει ο χρήστης τις σελίδες με τις επαρχίες που φαίνονται μέσα στον πίνακα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Μενού Επιλογής Country

Το μενού επιλογής **"Country"** καλεί τη μέθοδο **"fillRegion"** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event selectedIndexChanged() του μενού επιλογής. Είναι υπεύθυνη στο να διαβάσει τη μεταβλητή τύπου String με όνομα **"countryRegionAdmin"**, η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή του μενού επιλογής, και να επικοινωνήσει με τη μέθοδο **"fillRegion", της web service "hoteldbFill",** αποστέλλοντας της την τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
</root>
```

Όπου **country_name** το όνομα της χώρας της οποίας, έχει επιλέξει ο χρήστης να πάρει, τις επαρχίες. Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>22</id>
<region_name>Attica</region_name>
<flag>1</flag>
<id>25</id>
<region_name>Crete</region_name>
<flag>1</flag>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε επαρχία ανεξαρτήτου χώρας στο πίνακα **region** της βάσης. Το πεδίο **region_name** το οποίο αντιπροσωπεύει το όνομα της επαρχίας και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν η συγκεκριμένη επαρχία είναι διαθέσιμη ή όχι. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης

Κουμπί Add Region

Το κουμπί **"Add Region"** καλεί τη μέθοδο **"addRegion"** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **"regionAdmin"** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **"Region"**
- **"countryRegionAdmin"** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή από το μενού επιλογής **"Country".**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“addRegion”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<region_name>Crete</region_name>
</root>
```

Όπου **country_name** το όνομα της χώρας στην οποία θα προστεθεί η επαρχία και **region_name** το όνομα της επαρχίας προς προσθήκη.
Μετά από μια επιτυχημένη προσθήκη θα καλεστεί η μέθοδος **“fillRegion”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Edit Region

Το κουμπί **“Edit Region”** καλεί τη μέθοδο **“editRegion”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“regionAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Region”**
- **“countryRegionAdmin”** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή από το μενού επιλογής **“Country”**
- **“selectedRegionAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“editRegion”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<region_name>Creta</region_name>
<regionID>60</regionID>
</root>
```

Όπου **country_name** το όνομα της χώρας της οποίας την επαρχία διορθώνουμε, **region_name** το καινούργιο όνομα που θα πάρει η επαρχία μετά την διόρθωση και **regionID** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων που αντιστοιχεί στην επαρχία.
Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **“fillRegion”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Un/Available

Το κουμπί **“Un/Available”** καλεί τη μέθοδο **“deleteRegion”** από το bean με σκοπό να ενεργοποιήσει ή να απενεργοποιήσει μια επαρχία από τη βάση δεδομένων. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“selectedRegionAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“deleteRegion”, της web service “hoteldbDelete”,** αποστέλλοντας τη τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<region_name>22</region_name>
</root>
```

Όπου **region_name** το id της επαρχίας την οποία θέλουμε να απενεργοποιήσουμε / ενεργοποιήσουμε.
Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **“fillRegion”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Panel: Add/Edit City



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής, διόρθωσης, προβολής, ενεργοποίησης και απενεργοποίησης των πόλεων που είναι διαθέσιμες στο σύστημα.

Αποτελείται από

- ✓ Το μενού επιλογής **“Country”** το οποίο είναι ένα **“h:selectOneMenu”**
- ✓ Το μενού επιλογής **“Region”** το οποίο είναι ένα **“h:selectOneMenu”**
- ✓ Το πεδίο κειμένου **“City”** το οποίο είναι ένα **“h:inputText”**
- ✓ Το κουμπί **“Add City”** το οποίο είναι ένα **“a4j:commandButton”**
- ✓ Το πεδίο κειμένου **“ID”** το οποίο είναι ένα **“h:inputText”**
- ✓ Το κουμπί **“Un/Available”** το οποίο είναι ένα **“a4j:commandButton”**
- ✓ Το κουμπί **“Edit City”** το οποίο είναι ένα **“a4j:commandButton”**
- ✓ Τον **πίνακα** με τις πόλεις ο οποίος είναι **“rich:dataTable”**

- ✓ Τον **scroller** ο οποίος είναι τύπου **“rich:dataScroller”**
- ✓ Το **tooltip** επεξήγησης το οποίο είναι τύπου **“rich:tooltip”**

Επεξήγηση Ρόλων

Το μενού επιλογής **“Country”** είναι υπεύθυνο να εμφανίζει τις χώρες που έχουν εισαχθεί στο σύστημα και όταν ο χρήστης επιλέξει μια τιμή από τη λίστα επιλογής τότε το επόμενο μενού επιλογής, **“Region”,** γεμίζει με τις επαρχίες της επιλεγμένης χώρας. Στη συνέχεια όταν ο χρήστης επιλέξει μια επαρχία από τη λίστα επιλογής τότε γεμίζει ο πίνακας με τις πόλεις που έχουν έως τώρα εισαχθεί για την συγκεκριμένη επαρχία.

Το κουμπί **“Add City”** είναι υπεύθυνο να παίρνει το κείμενο που έχει εισάγει ο χρήστης στο πεδίο **“City”** και μετά από τους απαραίτητους ελέγχους να καταχωρεί την πόλη, κάτω από την επιλεγμένη επαρχία, στο σύστημα.

Το κουμπί **“Un/Available”** είναι υπεύθυνο να παίρνει τον αριθμό που έχει εισάγει ο χρήστης στο πεδίο **“ID”**, ο οποίος αντιστοιχεί στην αυτόματη αρίθμηση από τη βάση και είναι μοναδικός για κάθε πόλη και μετά από τους απαραίτητους ελέγχους να ενεργοποιεί ή να απενεργοποιεί την πόλη από το σύστημα.

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τις πόλεις του συστήματος, που αντιστοιχούν σε μια επιλεγμένη επαρχία. Οι τιμές που εμφανίζει είναι ο αύξων αριθμός ID, μοναδικός για κάθε πόλη, το όνομα της πόλης και τέλος το αν είναι διαθέσιμη ή όχι. Οι πόλεις χωρίζονται σε σελίδες με τέσσερεις πόλεις ανά σελίδα.

Ο **scroller** είναι υπεύθυνος στο να αλλάζει ο χρήστης τις σελίδες με τις πόλεις που φαίνονται μέσα στον πίνακα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Μενού Επιλογής Country

Το μενού επιλογής **“Country”** καλεί τη μέθοδο **“fillCbxRegion”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event selectedIndexChanged() του μενού επιλογής. Είναι υπεύθυνη στο να διαβάσει τη μεταβλητή τύπου String με όνομα **“countryCityAdmin”**, η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή του μενού επιλογής, και να επικοινωνήσει με τη μέθοδο **“fillRegion”, της web service “hoteldbFill”,** αποστέλλοντας της την τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
</root>
```

Όπου **country_name** το όνομα της χώρας της οποίας, έχει επιλέξει ο χρήστης να πάρει, τις επαρχίες. Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>22</id>
<region_name>Attica</region_name>
<flag>1</flag>
<id>25</id>
<region_name>Crete</region_name>
```

<flag>1</flag>

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε επαρχία ανεξαρτήτου χώρας στο πίνακα **region** της βάσης. Το πεδίο **region_name** το οποίο αντιπροσωπεύει το όνομα της επαρχίας και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν η συγκεκριμένη επαρχία είναι διαθέσιμη ή όχι. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης

Μενού Επιλογής Region

Το μενού επιλογής **“Region”** καλεί τη μέθοδο **“fillCity”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event selectedIndexChanged() του μενού επιλογής. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

1. String με όνομα **“countryCityAdmin”**, η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή του μενού επιλογής **country**
2. String με όνομα **“regionCityAdmin”**, η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή του μενού επιλογής **region**

και να επικοινωνήσει με τη μέθοδο **“fillCity”, της web service “hoteldbFill”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<region_name>Crete</region_name>
</root>
```

Όπου **country_name** το όνομα της χώρας και **region_name** το όνομα της επαρχίας, της οποίας έχει επιλέξει ο χρήστης να πάρει τις πόλεις. Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>27</id>
<city_name>Chania</city_name>
<flag>1</flag>
<id>29</id>
<city _name>Rethymno</city _name>
<flag>1</flag>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε πόλη στο πίνακα **city** της βάσης. Το πεδίο **city_name** το οποίο αντιπροσωπεύει το όνομα της πόλης και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν η συγκεκριμένη πόλη είναι διαθέσιμη ή όχι. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης

Κουμπί Add City

Το κουμπί **“Add City”** καλεί τη μέθοδο **“addCity”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“cityAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“City”**
- **“countryCityAdmin”** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή από το μενού επιλογής **“Country”**
- **“regionCityAdmin”** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή από το μενού επιλογής **“Region”**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“addCity”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<region_name>Crete</region_name>
<city_name>Chania</city_name>
</root>
```

Όπου **country_name** το όνομα της χώρας, **region_name** το όνομα της επαρχίας στην οποία θα προστεθεί η πόλη και **city_name** το όνομα της πόλης προς προσθήκη.
Μετά από μια επιτυχημένη προσθήκη θα καλεστεί η μέθοδος **“fillCity”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Edit City

Το κουμπί **“Edit City”** καλεί τη μέθοδο **“editCity”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“cityAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“City”**
- **“countryCityAdmin”** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή από το μενού επιλογής **“Country”**
- **“regionCityAdmin”** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή από το μενού επιλογής **“Region”**
- **“selectedCityAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“editCity”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<region_name>Creta</region_name>
<city_name>Khania</city_name>
<cityID>54</cityID>
</root>
```

Όπου **country_name** το όνομα της χώρας, **region_name** το όνομα της επαρχίας, **city_name** το νέο όνομα της πόλης μετά την διόρθωση και **cityID** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων που αντιστοιχεί στην πόλη.
Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **"fillCity",** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Un/Available

Το κουμπί **"Un/Available"** καλεί τη μέθοδο **"deleteCity"** από το bean με σκοπό να ενεργοποιήσει ή να απενεργοποιήσει μια πόλη από τη βάση δεδομένων. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **"selectedCityAdmin"** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **"ID".**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **"deleteCity", της web service "hoteldbDelete",** αποστέλλοντας τη τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>22</id >
</root>
```

Όπου **id** ο αύξων αριθμός που αντιστοιχεί στη πόλη την οποία θέλουμε να απενεργοποιήσουμε / ενεργοποιήσουμε.
Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **"fillCity",** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Panel: Add/Edit Location



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής, διόρθωσης, προβολής, ενεργοποίησης και απενεργοποίησης των τοπεθεσιών που είναι διαθέσιμες στο σύστημα.

Αποτελείται από

- ✓ Το μενού επιλογής **"Country"** το οποίο είναι ένα **"h:selectOneMenu"**
- ✓ Το μενού επιλογής **"Region"** το οποίο είναι ένα **"h:selectOneMenu"**
- ✓ Το μενού επιλογής **"City"** το οποίο είναι ένα **"h:selectOneMenu"**
- ✓ Το πεδίο κειμένου **"Location"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Add Location"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το πεδίο κειμένου **"ID"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Un/Available"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το κουμπί **"Edit Location"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Τον **πίνακα** με τις πόλεις ο οποίος είναι **"rich:dataTable"**
- ✓ Τον **scroller** ο οποίος είναι τύπου **"rich:dataScroller"**
- ✓ Το **tooltip** επεξήγησης το οποίο είναι τύπου **"rich:tooltip"**

Επεξήγηση Ρόλων

Το μενού επιλογής **"Country"** είναι υπεύθυνο να εμφανίζει τις χώρες που έχουν εισαχθεί στο σύστημα και όταν ο χρήστης επιλέξει μια τιμή από τη λίστα επιλογής τότε το επόμενο μενού επιλογής, **"Region",** γεμίζει με τις επαρχίες της επιλεγμένης χώρας. Στη συνέχεια όταν ο χρήστης επιλέξει μια επαρχία από τη λίστα επιλογής τότε γεμίζει το επόμενο μενού επιλογής, **"City"** με τις πόλεις της επιλεγμένης επαρχίας. Μετά που θα επιλέξει μια πόλη ο χρήστης θα γεμίσει ο πίνακας με τις τοποθεσίες που έχουν έως τώρα εισαχθεί για την συγκεκριμένη πόλη.

Το κουμπί **"Add Location"** είναι υπεύθυνο να παίρνει το κείμενο που έχει εισάγει ο χρήστης στο πεδίο **"Location"** και μετά από τους απαραίτητους ελέγχους να καταχωρεί την τοποθεσία, κάτω από την επιλεγμένη πόλη, στο σύστημα.

Το κουμπί **"Un/Available"** είναι υπεύθυνο να παίρνει τον αριθμό που έχει εισάγει ο χρήστης στο πεδίο **"ID"**, ο οποίος αντιστοιχεί στην αυτόματη αρίθμηση από τη βάση και είναι μοναδικός για κάθε τοποθεσία και μετά από τους απαραίτητους ελέγχους να ενεργοποιεί ή να απενεργοποιεί την τοποθεσία από το σύστημα.

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τις τοποθεσίες του συστήματος, που αντιστοιχούν σε μια επιλεγμένη πόλη. Οι τιμές που εμφανίζει είναι ο αύξων αριθμός ID, μοναδικός για κάθε τοποθεσία, το όνομα της τοποθεσίας και τέλος το αν είναι διαθέσιμη ή όχι. Οι τοποθεσίες χωρίζονται σε σελίδες με τέσσερεις τοποθεσίες ανά σελίδα.

Ο **scroller** είναι υπεύθυνος στο να αλλάζει ο χρήστης τις σελίδες με τις τοποθεσίες που φαίνονται μέσα στον πίνακα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Μενού Επιλογής Country

Το μενού επιλογής **"Country"** καλεί τη μέθοδο **"fillCbxRegionLoc"** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event selectedIndexChanged() του μενού επιλογής. Είναι υπεύθυνη στο να διαβάσει τη μεταβλητή τύπου String με όνομα **"countryLocationAdmin"**, η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή του μενού επιλογής, και να επικοινωνήσει με τη μέθοδο **"fillRegion", της web service "hoteldbFill",** αποστέλλοντας της την τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
</root>
```

Όπου **country_name** το όνομα της χώρας της οποίας, έχει επιλέξει ο χρήστης να πάρει, τις επαρχίες. Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>22</id>
<region_name>Attica</region_name>
<flag>1</flag>
<id>25</id>
<region_name>Crete</region_name>
<flag>1</flag>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε επαρχία ανεξαρτήτου χώρας στο πίνακα **region** της βάσης. Το πεδίο **region_name** το οποίο αντιπροσωπεύει το όνομα της επαρχίας και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν η συγκεκριμένη επαρχία είναι διαθέσιμη ή όχι. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης

Μενού Επιλογής Region

Το μενού επιλογής **"Region"** καλεί τη μέθοδο **"fillCbxCity"** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event selectedIndexChanged() του μενού επιλογής. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

1. String με όνομα **"countryLocationAdmin"**, η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή του μενού επιλογής **country**
2. String με όνομα **"regionLocationAdmin"**, η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή του μενού επιλογής **region**

και να επικοινωνήσει με τη μέθοδο **“fillCity”, της web service “hoteldbFill”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<region_name>Crete</region_name>
</root>
```

Όπου **country_name** το όνομα της χώρας και **region_name** το όνομα της επαρχίας, της οποίας έχει επιλέξει ο χρήστης να πάρει τις πόλεις. Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>27</id>
<city_name>Chania</city_name>
<flag>1</flag>
<id>29</id>
<city _name>Rethymno</city _name>
<flag>1</flag>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε πόλη στο πίνακα **city** της βάσης. Το πεδίο **city_name** το οποίο αντιπροσωπεύει το όνομα της πόλης και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν η συγκεκριμένη πόλη είναι διαθέσιμη ή όχι. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης

Μενού Επιλογής City

Το μενού επιλογής **“City”** καλεί τη μέθοδο **“fillLocation”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event selectedIndexChanged() του μενού επιλογής. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

1. String με όνομα **“countryLocationAdmin”**, η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή του μενού επιλογής **country**
2. String με όνομα **“regionLocationAdmin”**, η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή του μενού επιλογής **region**
3. String με όνομα **“cityLocationAdmin”**, η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή του μενού επιλογής **city**

και να επικοινωνήσει με τη μέθοδο **“fillLocation”, της web service “hoteldbFill”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<region_name>Crete</region_name>
<city_name>Chania</city_name>
</root>
```

Όπου **country_name** το όνομα της χώρας, **region_name** το όνομα της επαρχίας και **city_name** το όνομα της πόλης της οποίας έχει επιλέξει ο χρήστης να πάρει τις τοποθεσίες. Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>27</id>
<location>Akrotiri</location>
<flag>1</flag>
<id>29</id>
<location>Chalepa</location>
<flag>1</flag>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε τοποθεσία στο πίνακα **location** της βάσης. Το πεδίο **location** το οποίο αντιπροσωπεύει το όνομα της τοποθεσίας και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν η συγκεκριμένη τοποθεσία είναι διαθέσιμη ή όχι. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης

Κουμπί Add Location

Το κουμπί **“Add Location”** καλεί τη μέθοδο **“addLocation”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“locationAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Location”**
- **“countryLocationAdmin”** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή από το μενού επιλογής **“Country”**
- **“regionLocationAdmin”** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή από το μενού επιλογής **“Region”**
- **“cityLocationAdmin”** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή από το μενού επιλογής **“City”**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“addLocation”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<region_name>Crete</region_name>
<city_name>Chania</city_name>
<location>Akrotiri</location>
</root>
```

Όπου **country_name** το όνομα της χώρας, **region_name** το όνομα της επαρχίας, **city_name** το όνομα της πόλης κάτω από την οποία θα προστεθεί η τοποθεσία και **location** το όνομα της τοποθεσίας προς προσθήκη.

Μετά από μια επιτυχημένη προσθήκη θα καλεστεί η μέθοδος **“fillLocation”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Edit Location

Το κουμπί **“Edit Location”** καλεί τη μέθοδο **“editLocation”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“locationAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Location”**
- **“countryLocationAdmin”** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή από το μενού επιλογής **“Country”**
- **“regionLocationAdmin”** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή από το μενού επιλογής **“Region”**
- **“cityLocationAdmin”** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή από το μενού επιλογής **“City”**
- **“selectedLocationAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“editLocaiton”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<region_name>Creta</region_name>
<city_name>Khania</city_name>
<location>Akrotirion</location>
<locationID>84</locationID>
</root>
```

Όπου **country_name** το όνομα της χώρας, **region_name** το όνομα της επαρχίας, **city_name** το όνομα της πόλης, **location** το νέο όνομα της τοποθεσίας μετά την διόρθωση και **locationID** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων που αντιστοιχεί στην τοποθεσία.
Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **“fillLocation”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Un/Available

Το κουμπί **“Un/Available”** καλεί τη μέθοδο **“deleteLocation”** από το bean με σκοπό να ενεργοποιήσει ή να απενεργοποιήσει μια τοποθεσία από τη βάση δεδομένων. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

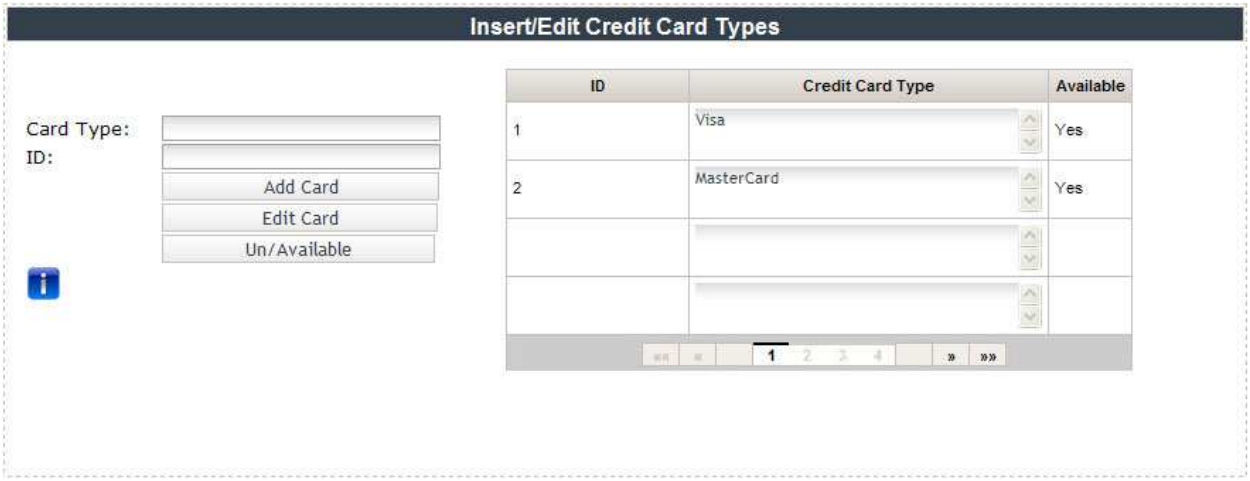
- **“selectedLocationAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“deleteLocation”, της web service “hoteldbDelete”,** αποστέλλοντας τη τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>22</id >
</root>
```

Όπου **id** ο αύξων αριθμός που αντιστοιχεί στη τοποθεσία την οποία θέλουμε να απενεργοποιήσουμε / ενεργοποιήσουμε.
Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **“fillLocation”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Panel: Add/Edit Credit Card



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής, διόρθωσης, προβολής, ενεργοποίησης και απενεργοποίησης των τύπων πιστωτικών καρτών που είναι διαθέσιμες στο σύστημα.

Αποτελείται από

- ✓ Το πεδίο κειμένου **“Card Type”** το οποίο είναι ένα **“h:inputText”**
- ✓ Το κουμπί **“Add Card”** το οποίο είναι ένα **“a4j:commandButton”**
- ✓ Το κουμπί **“Edit Card”** το οποίο είναι ένα **“a4j:commandButton”**
- ✓ Το πεδίο κειμένου **“ID”** το οποίο είναι ένα **“h:inputText”**
- ✓ Το κουμπί **“Un/Available”** το οποίο είναι ένα **“a4j:commandButton”**
- ✓ Τον **πίνακα** με τις κάρτες ο οποίος είναι **“rich:dataTable”**

- ✓ Τον **scroller** ο οποίος είναι τύπου **“rich:dataScroller”**
- ✓ Το **tooltip** επεξήγησης το οποίο είναι τύπου **“rich:tooltip”**

Επεξήγηση Ρόλων

Το κουμπί **“Add Card”** είναι υπεύθυνο να παίρνει το κείμενο που έχει εισάγει ο χρήστης στο πεδίο **“Card Type”** και μετά από τους απαραίτητους ελέγχους να καταχωρεί τον τύπο πιστωτικής κάρτας στο σύστημα.

Το κουμπί **“Un/Available”** είναι υπεύθυνο να παίρνει τον αριθμό που έχει εισάγει ο χρήστης στο πεδίο **“ID”**, ο οποίος αντιστοιχεί στην αυτόματη αρίθμηση από τη βάση και είναι μοναδικός για κάθε τύπο κάρτας και μετά από τους απαραίτητους ελέγχους να ενεργοποιεί ή να απενεργοποιεί τον τύπο από το σύστημα.

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τους τύπους πιστωτικών καρτών του συστήματος. Οι τιμές που εμφανίζει είναι ο αύξων αριθμός ID, μοναδικός για κάθε τύπο, το όνομα του τύπου και τέλος το αν είναι διαθέσιμος ή όχι. Οι τύποι χωρίζονται σε σελίδες με τέσσερεις ανά σελίδα.

Ο **scroller** είναι υπεύθυνος στο να αλλάζει ο χρήστης τις σελίδες με τους τύπους καρτών που φαίνονται μέσα στο πίνακα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Κουμπί Add Card

Το κουμπί **“Add Card”** καλεί τη μέθοδο **“addCardType”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τη μεταβλητή τύπου String με όνομα **“cardTypeAdmin”**, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Card Type”**, και να επικοινωνήσει με τη μέθοδο **“addCardType”, της web service “hoteldbAdd”,** αποστέλλοντας της την τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<cardType>Greece</ cardType>
</root>
```

Όπου cardType το όνομα του τύπου που θέλει ο χρήστης να προστεθεί στη βάση.
Μετα από μια επιτυχημένη προσθήκη θα καλεστεί η μέθοδος **fillCardType** και θα ενεργοποιηθεί η διαδικασία για την ανανέωση των περιεχομένων του πίνακα.

Διαδικασία fillCardType

Καλείται η μέθοδος **“fillCardType”** από το bean. Αυτή η μέθοδος δεν δέχεται ορίσματα και σαν αποτέλεσμα δεν στέλνει κάποιο ωφέλημο μήνυμα. Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **“fillCardType”, της web service “hoteldbFill”,** με σκοπό να παραλάβει ένα μήνυμα με τους τύπους πιστωτικών καρτών που είναι καταχωρημένοι στη βάση και συγκεκριμένα στον πίνακα **“cardtype”.**
Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>7</id>
<cardtype>Visa</ cardtype>
```

<flag>1</flag>
<id>8</id>
< cardtype>Mastercard</ cardtype>
<flag>1</flag>

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε τύπο στον πίνακα "**cardtype"**. Το πεδίο **cardtype** το οποίο αντιπροσωπεύει το όνομα του τύπου και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν ο συγκεκριμένος τύπος είναι ενεργοποιημένος ή όχι. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης

Κουμπί Edit Card

Το κουμπί **"Edit Card"** καλεί τη μέθοδο **"editCardType"** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **"cardTypeAdmin"** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **"Card Type"**
- **"selectedCardType"** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **"ID".**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **"editCardType", της web service "hoteldbAdd",** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

<?xml version="1.0" encoding="UTF-8"?>
<root>
<cardType>NewValue</cardType>
<cardTypeID>9</cardTypeID>
</root>

Όπου **cardType** το νέο όνομα του τύπου τον οποίο διορθώνουμε και **cardTypeID** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων που αντιστοιχεί στον τύπο.
Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **"fillCardType",** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Un/Available

Το κουμπί **"Un/Available"** καλεί τη μέθοδο **"deleteCardType"** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **"selectedCardType"** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **"ID".**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **"deleteCardType", της web service "hoteldbDelete",** αποστέλλοντας τη τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

<?xml version="1.0" encoding="UTF-8"?>
<root>
<cardType>9</ cardType>
</root>

Όπου **cardType** το id του τύπου που θέλουμε να απενεργοποιήσουμε / ενεργοποιήσουμε.

Μετά από μια επιτυχημένη αλλαγή θα καλεστεί η μέθοδος **“fillCardType”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Panel: Add/Edit Currency



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής, διόρθωσης, προβολής, ενεργοποίησης και απενεργοποίησης των νομισμάτων που είναι διαθέσιμα στο σύστημα.

Αποτελείται από

- ✓ Το μενού επιλογής **“Country”** το οποίο είναι ένα **“h:selectOneMenu”**
- ✓ Το πεδίο κειμένου **“Currency”** το οποίο είναι ένα **“h:inputText”**
- ✓ Το κουμπί **“Add Currency”** το οποίο είναι ένα **“a4j:commandButton”**
- ✓ Το πεδίο κειμένου **“ID”** το οποίο είναι ένα **“h:inputText”**
- ✓ Το κουμπί **“Un/Available”** το οποίο είναι ένα **“a4j:commandButton”**
- ✓ Το κουμπί **“Edit Currency”** το οποίο είναι ένα **“a4j:commandButton”**
- ✓ Τον **πίνακα** με τις επαρχίες ο οποίος είναι **“rich:dataTable”**
- ✓ Τον **scroller** ο οποίος είναι τύπου **“rich:dataScroller”**
- ✓ Το **tooltip** επεξήγησης το οποίο είναι τύπου **“rich:tooltip”**

Επεξήγηση Ρόλων

Το μενού επιλογής **“Country”** είναι υπεύθυνο να εμφανίζει τις χώρες που έχουν εισαχθεί στο σύστημα και που θα γίνει αντιστοίχηση, ένα νόμισμα ανα χώρα.

Το κουμπί **“Add Currency”** είναι υπεύθυνο να παίρνει το κείμενο που έχει εισάγει ο χρήστης στο πεδίο **“Currency”** και μετά από τους απαραίτητους ελέγχους να καταχωρεί στο σύστημα το νόμισμα για την επιλεγμένη χώρα από το μενού επιλογής.

Το κουμπί **“Un/Available”** είναι υπεύθυνο να παίρνει τον αριθμό που έχει εισάγει ο χρήστης στο πεδίο **“ID”**, ο οποίος αντιστοιχεί στην αυτόματη αρίθμηση από τη βάση και είναι μοναδικός για κάθε χώρα-νόμισμα και μετά από τους απαραίτητους ελέγχους να ενεργοποιεί ή να απενεργοποιεί την καταχώρηση από το σύστημα.

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τα νομίσμα του συστήματος. Οι τιμές που εμφανίζει είναι ο αύξων αριθμός ID, μοναδικός για κάθε νόμισμα-χώρα, το όνομα της χώρας, το όνομα του νομίσματος και τέλος το αν είναι διαθέσιμο ή όχι. Τα νομίσματα χωρίζονται σε σελίδες με τέσσερα ανά σελίδα.

Ο **scroller** είναι υπεύθυνος στο να αλλάζει ο χρήστης τις σελίδες με τα νομίσματα που φαίνονται μέσα στον πίνακα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Μενού Επιλογής Country

Το μενού επιλογής **“Country”** δεν καλεί κάποια μέθοδο στην αλλαγή επιλογής του, απλώς είναι συνδεδεμένο με την μεταβλητή **“countryExchangeAdmin”** του bean, η οποία χρησιμοποιείται από τις επόμενες διαδικασίες της διεπαφής.

Διαδικασία fillCurrency

Καλείται η μέθοδος **“fillCurrency”** από το bean. Αυτή η μέθοδος δεν δέχεται ορίσματα και σαν αποτέλεσμα δεν στέλνει κάποιο ωφέλημο μήνυμα. Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **“fillCurrency”, της web service “hoteldbFill”,** με σκοπό να παραλάβει ένα μήνυμα με τα νομίσματα που είναι καταχωρημένα στη βάση και συγκεκριμένα στον πίνακα **“exchangetype”.**

Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>3</id>
<country_name>Germany</country_name>
<currency>Euro</currency>
<flag>1</flag>
<id>4</id>
<country_name>Greece</country_name>
<currency>Euro</currency>
<flag>1</flag>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε χώρα-νόμισμα στον πίνακα “**exchangetype”**. Το πεδίο **country_name** το οποίο αντιπροσωπεύει το όνομα της χώρας, το πεδίο **currency** το οποίο περιλαμβάνει το όνομα του νομίσματος και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν το συγκεκριμένο νόμισμα είναι ενεργοποιημένο ή

όχι. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης

Κουμπί Add Currency

Το κουμπί **“Add Currency”** καλεί τη μέθοδο **“addExchangeType”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“exchangeAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Currency”**
- **“countryExchangeAdmin”** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή από το μενού επιλογής **“Country”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“addExchangeType”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<currency>Euro</currency>
</root>
```

Όπου **country_name** το όνομα της χώρας και **currency** το όνομα του νομίσματος που εισάγουμε για την επιλεγμένη χώρα.
Μετά από μια επιτυχημένη προσθήκη θα καλεστεί η μέθοδος **“fillExchangeType”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Edit Currency

Το κουμπί **“Edit Currency”** καλεί τη μέθοδο **“editExchangeType”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“exchangeAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Currency”**
- **“countryExchangeAdmin”** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή από το μενού επιλογής **“Country”**
- **“selectedExchangeAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“editExchangeType”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<currency>Drachmas</currency>
<currencyID>4</currencyID>
</root>
```

Όπου **currencyID** η θέση στον πίνακα **exhcangetype** και **currency** το καινούργιο όνομα του νομίσματος μετά την διόρθωση.
Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **“fillExchangeType”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Un/Available

Το κουμπί **“Un/Available”** καλεί τη μέθοδο **“deleteExchangeType”** από το bean με σκοπό να ενεργοποιήσει ή να απενεργοποιήσει ένα νόμισμα από τη βάση δεδομένων. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“selectedExchangeAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“deleteExchangeType”, της web service “hoteldbDelete”,** αποστέλλοντας τη τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<currency>4</currency>
</root>
```

Όπου **currency** το id του νομίσματος το οποίο θέλουμε να απενεργοποιήσουμε / ενεργοποιήσουμε.
Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **“fillExchangeType”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Panel: Add/Edit Place of Interest



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής, διόρθωσης, προβολής, ενεργοποίησης και απενεργοποίησης των σημείων ενδιαφέροντος (POI) που είναι διαθέσιμα στο σύστημα.

Αποτελείται από

- ✓ Το πεδίο κειμένου **"Station"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Add Station"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το κουμπί **"Edit Station"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το πεδίο κειμένου **"ID"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Un/Available"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Τον **πίνακα** με τις κάρτες ο οποίος είναι **"rich:dataTable"**
- ✓ Τον **scroller** ο οποίος είναι τύπου **"rich:dataScroller"**
- ✓ Το **tooltip** επεξήγησης το οποίο είναι τύπου **"rich:tooltip"**

Επεξήγηση Ρόλων

Το κουμπί **"Add Station"** είναι υπεύθυνο να παίρνει το κείμενο που έχει εισάγει ο χρήστης στο πεδίο **"Staion"** και μετά από τους απαραίτητους ελέγχους να καταχωρεί το POI στο σύστημα.

Το κουμπί **"Un/Available"** είναι υπεύθυνο να παίρνει τον αριθμό που έχει εισάγει ο χρήστης στο πεδίο **"ID"**, ο οποίος αντιστοιχεί στην αυτόματη αρίθμηση από τη βάση και είναι μοναδικός για κάθε POI και μετά από τους απαραίτητους ελέγχους να το ενεργοποιεί ή να απενεργοποιεί από το σύστημα.

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τα POI του συστήματος. Οι τιμές που εμφανίζει είναι ο αύξων αριθμός ID, μοναδικός για κάθε POI, το όνομα του και τέλος το αν είναι διαθέσιμο ή όχι. Τα POIs χωρίζονται σε σελίδες με τέσσερα ανά σελίδα.

Ο **scroller** είναι υπεύθυνος στο να αλλάζει ο χρήστης τις σελίδες με τα POI που φαίνονται μέσα στο πίνακα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Κουμπί Add Station

Το κουμπί **"Add Station"** καλεί τη μέθοδο **"addDistanceStation"** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τη μεταβλητή τύπου String με όνομα **"distanceFromAdmin"**, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **"Station"**, και να επικοινωνήσει με τη μέθοδο **"addDistanceStation", της web service "hoteldbAdd",** αποστέλλοντας της την τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<station>Airport</ station>
</root>
```

Όπου station το όνομα του POI που θέλει ο χρήστης να προστεθεί στη βάση.
Μετα από μια επιτυχημένη προσθήκη θα καλεστεί η μέθοδος **fillDistanceFrom** και θα ενεργοποιηθεί η διαδικασία για την ανανέωση των περιεχομένων του πίνακα.

Διαδικασία fillDistanceFrom

Καλείται η μέθοδος **“fillDistanceFrom”** από το bean. Αυτή η μέθοδος δεν δέχεται ορίσματα και σαν αποτέλεσμα δεν στέλνει κάποιο ωφέλημο μήνυμα. Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **“fillDistanceFrom”, της web service “hoteldbFill”,** με σκοπό να παραλάβει ένα μήνυμα με τα POI που είναι καταχωρημένα στη βάση και συγκεκριμένα στον πίνακα **“distancefrom”.**
Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>14</id>
<station>Airport</ cardtype>
<flag>1</flag>
<id>24</id>
<station>Ancient Stadium</staion>
<flag>1</flag>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε τύπο στον πίνακα “**distancefrom”**. Το πεδίο **station** το οποίο αντιπροσωπεύει το όνομα του POI και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν το συγκεκριμένος POI είναι ενεργοποιημένο ή όχι. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης

Κουμπί Edit Station

Το κουμπί **“Edit Station”** καλεί τη μέθοδο **“editDistanceStation”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“distanceFromAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Station”**
- **“selectedDistanceFrom”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“editDistanceStation”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<station>Airport</station>
<stationID>14</stationID>
</root>
```

Όπου **station** το νέο όνομα του POI τον οποίο διορθώνουμε και **stationID** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων που αντιστοιχεί στο POI.
Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **“fillDistanceFrom”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Un/Available

Το κουμπί **“Un/Available”** καλεί τη μέθοδο **“deleteDistanceFrom”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

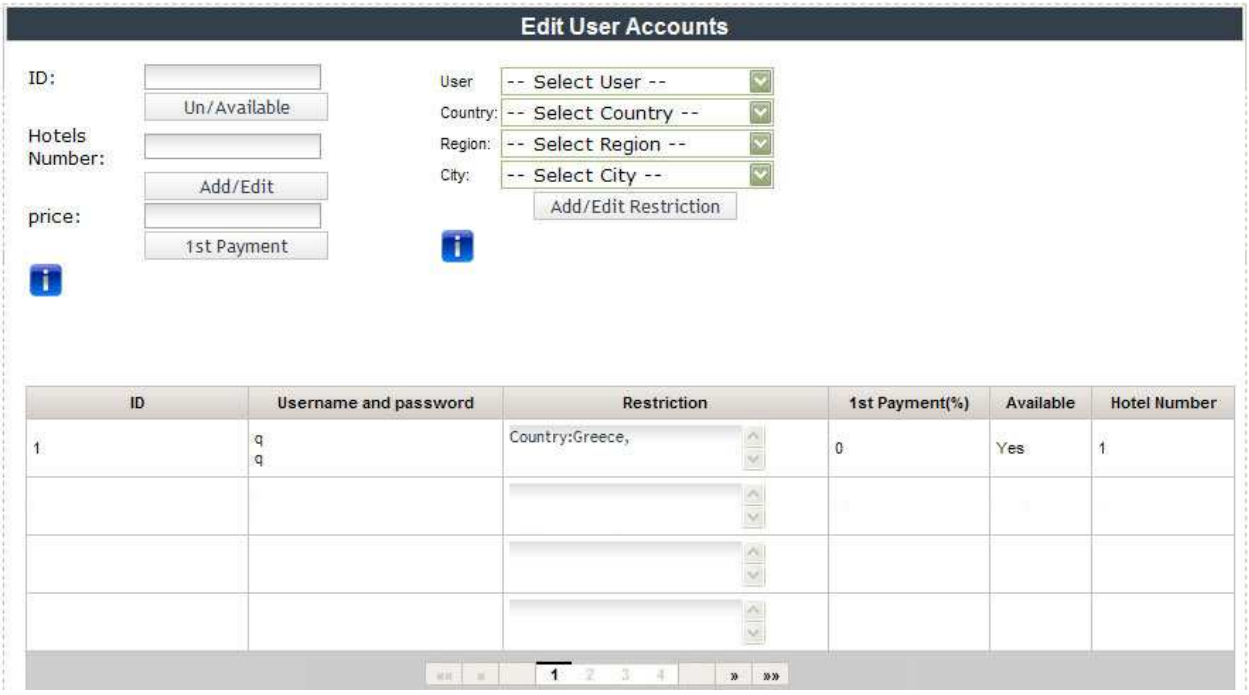
- **“selectedDistanceFrom”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“deleteDistanceFrom”, της web service “hoteldbDelete”,** αποστέλλοντας τη τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<station>9</station>
</root>
```

Όπου **station** το id του POI που θέλουμε να απενεργοποιήσουμε / ενεργοποιήσουμε.
Μετά από μια επιτυχημένη αλλαγή θα καλεστεί η μέθοδος **“fillDistanceFrom”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Panel: Add/Edit User Accounts



Λειτουργικότητα

Εμφανίζει τη διεπαφή της τροποποίησης των πληροφοριών για τους πράκτορες που είναι καταχωρημένοι στο σύστημα και συγκεκριμένα στον πίνακα "**userhotel"** της βάσης δεδομένων. Επιστρέπει την εισαγωγή του μέγιστου αριθμού ξενοδοχείων που μπορεί να εισάγει ένας πράκτορας, την εισαγωγή του ποσοστού που έχει συμφωνηθεί να καταβάλλεται στον πράκτορα ως προκαταβολλή για κάθε κράτηση, τον περιορισμό του πράκτορα σε μια συγκεκριμένη περιοχή του συστήματος έτσι ώστε να μην μπορεί να βάλει ξενοδοχεία ή δεδομένα εκτός αυτής της περιοχής.

Αποτελείται από

- ✓ Το πεδίο κειμένου **"ID"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Un/Available"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το πεδίο κειμένου **"Hotels Number"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Add / Edit"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το πεδίο κειμένου **"Price"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"1st Payment"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το μενού επιλογής **"User"** το οποίο είναι ένα **"h:selectOneMenu"**
- ✓ Το μενού επιλογής **"Country"** το οποίο είναι ένα **"h:selectOneMenu"**
- ✓ Το μενού επιλογής **"City"** το οποίο είναι ένα **"h:selectOneMenu"**
- ✓ Το κουμπί **"Add/Edit Restriction"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Τον **πίνακα** με τις πόλεις ο οποίος είναι **"rich:dataTable"**
- ✓ Τον **scroller** ο οποίος είναι τύπου **"rich:dataScroller"**
- ✓ Τα δύο **tooltip** επεξήγησης τα οποία είναι τύπου **"rich:tooltip"**

Επεξήγηση Ρόλων

Το κουμπί **Un/Available** είναι υπεύθυνο να παίρνει τον αριθμό που εισάγει ο χρήστης στο πεδίο **ID** ο οποίος αντιστοιχεί στον αύξων αριθμό της βάσης για έναν πράκτορα και μετά τους απαραίτητους ελέγχους να ενεργοποιεί ή να απενεργοποιεί τον πράκτορα από το σύστημα.

Το κουμπί **Add/Edit** είναι υπεύθυνο να παίρνει τον αριθμό του πράκτορα από το πεδίο **ID** και τον αριθμό από το πεδίο **Hotels Number,** ο οποίος αντιστοιχεί στον μέγιστο αριθμό ξενοδοχείων που μπορεί να εισάγει ένας πράκτορας στο σύστημα και μετά τους απαραίτητους ελέγχους να καταχωρεί τον αριθμό των ξενοδοχείων στον συγκεκριμένο χρήστη.

Το κουμπί **1st Payment** είναι υπεύθυνο να παίρνει τον αριθμό του πράκτορα από το πεδίο **ID** και τον αριθμό από το πεδίο **Price**, ο οποίος αντιστοιχεί στο ποσοστό της προκαταβολής επι της τιμής της κράτησης που έχει συμφωνηθεί να καταβληθεί στον πράκτορα και μετα τους απαραίτητους ελέγχους να καταχωρεί το ποσοστό στον πράκτορα.

Το μενού επιλογής **User** είναι υπεύθυνο να εμφανίζει τη λίστα με τους πράκτορες που είναι καταχωρημένοι στο σύστημα.

Το μενού επιλογής **"Country"** είναι υπεύθυνο να εμφανίζει τις χώρες που έχουν εισαχθεί στο σύστημα και όταν ο χρήστης επιλέξει μια τιμή από τη λίστα επιλογής τότε το επόμενο μενού επιλογής, **"Region",** γεμίζει με τις επαρχίες της επιλεγμένης χώρας. Στη συνέχεια όταν ο χρήστης επιλέξει μια επαρχία από τη λίστα επιλογής τότε γεμίζει το επόμενο μενού επιλογής, **"City"** με τις πόλεις της επιλεγμένης επαρχίας.

Το κουμπί **"Add/Edit Restriction"** είναι υπεύθυνο να παίρνει τον χρήστη που έχει επιλεχθεί από τον μενού επιλογής **User,** την επιλογή της περιοχής που έχει δομηθεί από τα μενού επιλογής και να εισάγει τον περιορισμό στη βάση δεδομένων για τον συγκεκριμένο χρήστη. Εάν έχει καταχωρηθεί παλιότερα διαφορετικός περιορισμός για τον επιλεγμένο χρήστη τότε επαναγράφεται και ισχύει ο καινούργιος.

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τους χρήστες του συστήματος. Οι τιμές που εμφανίζει είναι ο αύξων αριθμός ID, μοναδικός για κάθε χρήστη, το συνθηματικό και ο κωδικός του χρήστη σύμφωνα με τα οποία εισέρχεται στο σύστημα, ο περιορισμός του χρήστη σε μια συγκεκριμένη τοποθεσία, το ποσοστό της προκαταβολής για τον χρήστη, το αν είναι ενεργοποιημένος ή όχι και τέλος τον μέγιστο αριθμό των ξενοδοχείων που μπορεί να εισάγει στο σύστημα. Οι χρήστες χωρίζονται σε σελίδες με τέσσερεις ανά σελίδα.

Ο **scroller** είναι υπεύθυνος στο να αλλάζει ο χρήστης τις σελίδες με τους χρήστε που φαίνονται μέσα στον πίνακα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Διαδικασία fillUserHotel

Καλείται η μέθοδος **“fillUserHotel”** από το bean. Αυτή η μέθοδος δεν δέχεται ορίσματα και σαν αποτέλεσμα δεν στέλνει κάποιο ωφέλημο μήνυμα. Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **“fillUserHotel”, της web service “hoteldbFill”,** με σκοπό να παραλάβει ένα μήνυμα με τους χρήστες που είναι καταχωρημένοι στη βάση και συγκεκριμένα στον πίνακα **“userhotel”.**
Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>14</id>
<user_name>Airport</user_name>
<pass_word>1</pass_word>
<rest>24</rest>
<flag>Ancient Stadium</flag>
<forward>1</forward>
<hotelNum>1</hotelNum>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε χρήστη στον πίνακα “**userhotel”**.
Το πεδίο **user_name** το οποίο αντιπροσωπεύει το συνθηματικό του χρήστη.
Το πεδίο **pass_word** το οποίο περιλαμβάνει τον κωδικό του χρήστη.
Το πεδίο rest
Το πεδίο **flag** το οποίο μας δείχνει αν ο χρήστης είναι διαθέσιμος ή όχι στο σύστημα
Το πεδίο **forward** και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν το συγκεκριμένος POI είναι ενεργοποιημένο ή όχι. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης

Μενού Επιλογής Country

Το μενού επιλογής **“Country”** καλεί τη μέθοδο **“cbxCountryAdmin”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event selectedIndexChanged() του μενού επιλογής. Είναι υπεύθυνη στο να διαβάσει τη μεταβλητή τύπου String με όνομα **“adminRestCountry”**, η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή του μενού επιλογής, Στη συνέχεια καλείται η **fillRegionHotel** και δίνεται η τιμή της μεταβλητής σαν όρισμα. Αυτή η μέθοδος είναι υπεύθυνη να κάνει την επικοινωνία με την μέθοδο **“fillRegionHotel”, της web service “hoteldbFill”,** αποστέλλοντας της την τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
</root>
```

Όπου **country_name** το όνομα της χώρας της οποίας, έχει επιλέξει ο χρήστης να πάρει, τις επαρχίες. Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<region_name>Attica</region_name>
<region_name>Crete</region_name>
</root>
```

Όπου **region_name** το όνομα της επαρχίας που ανήκει στην χώρα που στείλαμε. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του επόμενου μενού επιλογής με τις επαρχίες.

Μενού Επιλογής Region

Το μενού επιλογής **“Region”** καλεί τη μέθοδο **“cbxRegionAdmin”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event selectedIndexChanged() του μενού επιλογής. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

1. String με όνομα **“adminRestCountry”**, η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή του μενού επιλογής **country**
2. String με όνομα **“adminRestRegion”**, η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή του μενού επιλογής **region**

και να καλέσει με τη μέθοδο **“fillCityHotel”** δίνοντας της μεταβλητές σαν ορίσματα. Αυτή με τη σειρά της είναι υπεύθυνη να επικοινωνήσει με τη μέθοδο **“fillCityHotel” της web service “hoteldbFill”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<region_name>Crete</region_name>
</root>
```

Όπου **country_name** το όνομα της χώρας και **region_name** το όνομα της επαρχίας, της οποίας έχει επιλέξει ο χρήστης να πάρει τις πόλεις. Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<city_name>Chania</city_name>
<city _name>Rethymno</city _name>
</root>
```

Όπου **city_name** τα πεδία με τα ονόματα των πόλεων που ανήκουν στην επαρχία που επιλέξαμε. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης

Μενού Επιλογής City

Το μενού επιλογής **"City"** δεν καλεί κάποια μέθοδο, είναι συνδεδεμένη όμως με τη μεταβλητή **adminRestCity** από το bean στην οποία δίνει την επιλεγμένη της τιμή κάθε φορά.

Κουμπί Add/Edit Restrinction

Το κουμπί **"Add/Edit Restrinction"** καλεί τη μέθοδο **"usernameRestrDefine"** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **"adminRestUser"** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **"User"** και αντιπροσωπεύει τον επιλεγμένο χρήστη.
- **"adminRestCountry"** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή από το μενού επιλογής **"Country"**
- **"adminRestRegion"** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή από το μενού επιλογής **"Region"**
- **"adminRestCity"** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή από το μενού επιλογής **"City"**

Στη συνέχεια αναλόγως με το ποιες μεταβλητές έχουν τιμή καλεί την αντίστοιχη μέθοδο. Συγκεκριμένα εξετάζονται οι εξής περιπτώσεις:

1. Εφόσον έχει επιλεχθεί μόνον χώρα.
   Καλείται η μέθοδος **UserNameRestrCountry,** αυτή με τη σειρά της είναι υπεύθυνη να επικοινωνήσει την μέθοδο **"addUserNameRestrCountry"** της webservice **"hoteldbAdd",** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<md5username>Greece</md5username>
<country>Crete</country>
</root>
```

Όπου **country** το όνομα της χώρας στην οποία θα περιοριστεί ο χρήστης και **md5username** το συνθηματικο του χρήστη σε κωδικοποιημένη μορφή **MD5 Hash**. Στη συνέχεια καλείται η fillUserHotel και ανανεώνονται τα περιεχόμενα του πίνακα με τη διαδικασία που αναλύσαμε παραπάνω.

2. Εφόσον έχει επιλεχθεί χώρα και επαρχία.
   Καλείται η μέθοδος **UserNameRestrRegion,** αυτή με τη σειρά της είναι υπεύθυνα να επικοινωνήσει με την μέθοδο **"addUserNameRestrRegion"** της webservice **"hoteldbAdd",** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<md5username>Greece</md5username>
<country>Crete</country>
<region>Crete</region>
</root>
```

Όπου **country** το όνομα της χώρας, **region** το όνομα της επαρχίας στην οποία θα περιοριστεί ο χρήστης και **md5username** το συνθηματικο του χρήστη σε κωδικοποιημένη μορφή **MD5 Hash**. Στη συνέχεια καλείται η fillUserHotel και ανανεώνονται τα περιεχόμενα του πίνακα με τη διαδικασία που αναλύσαμε παραπάνω.

3. Εφόσον έχει επιλεχθεί χώρα, επαρχία και πόλη.
   Καλείται η μέθοδος **UserNameRestrCity,** αυτή με τη σειρά της είναι υπεύθυνα να επικοινωνήσει με την μέθοδο **“addUserNameRestrCity”** της webservice **“hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<md5username>Greece</md5username>
<country>Crete</country>
<region>Crete</region>
<city>Crete</city>
</root>
```

Όπου **country** το όνομα της χώρας, **region** το όνομα της επαρχίας, **city** το όνομα της πόλης στην οποία θα περιοριστεί ο χρήστης και **md5username** το συνθηματικο του χρήστη σε κωδικοποιημένη μορφή **MD5 Hash**. Στη συνέχεια καλείται η fillUserHotel και ανανεώνονται τα περιεχόμενα του πίνακα με τη διαδικασία που αναλύσαμε παραπάνω.

<u>Κουμπί Un/Available</u>

Το κουμπί **“Un/Available”** καλεί τη μέθοδο **“deleteUserHotel”** από το bean με σκοπό να ενεργοποιήσει ή να απενεργοποιήσει έναν χρήστη από τη βάση δεδομένων. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“selectedUserAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“deleteUserHotel”, της web service “hoteldbDelete”,** αποστέλλοντας τη τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<User>2</User>
</root>
```

Όπου **id** ο αύξων αριθμός που αντιστοιχεί στον χρήστη τον οποίο θέλουμε να απενεργοποιήσουμε / ενεργοποιήσουμε.

Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **“fillUserHotel”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Add/Edit

Το κουμπί **“Add/Edit”** καλεί τη μέθοδο **“addHotelNum”** από το bean με σκοπό να εισάγει ή να αλλάξει τον αριθμό τον ξενοδοχείων που μπορεί να καταχωρήσει ένας χρήστης στο σύστημα. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“selectedUserAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**
- **“hotelNum”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Hotels Number”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“addHotelNum”, της web service “hoteldbAdd”,** αποστέλλοντας τη τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<forward>5</forward>
</root>
```

Όπου **id** ο αύξων αριθμός που αντιστοιχεί στον χρήστη και **forward** ο αριθμός των ξενοδοχειων που επιτρέπουμε στο χρήστη να καταχωρήσει στη βάση. Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **“fillUserHotel”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί 1st Payment

Το κουμπί **“1st Payment”** καλεί τη μέθοδο **“frontMoney”** από το bean με σκοπό να εισάγει ή να αλλάξει το ποσοστό που πρέπει να καταβληθεί σαν προκαταβολή για κάθε κράτηση στον πράκτορα. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“selectedUserAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**
- **“forwardMoney”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Price”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“addMoney”, της web service “hoteldbAdd”,** αποστέλλοντας τη τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<forward>5</forward>
</root>
```

Όπου **id** ο αύξων αριθμός που αντιστοιχεί στον χρήστη και **forward** ο αριθμός του ποσοστού για την προκαταβολή του πράκτορα. Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **"fillUserHotel",** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Panel: Add/Edit Hotel Type



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής, διόρθωσης, προβολής, ενεργοποίησης και απενεργοποίησης των τύπων ξενοδοχείων που είναι διαθέσιμοι στο σύστημα.

Αποτελείται από

- ✓ Το πεδίο κειμένου **"Hotel Type"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Add Type"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το κουμπί **"Edit Type"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το πεδίο κειμένου **"ID"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Un/Available"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Τον **πίνακα** με τις κάρτες ο οποίος είναι **"rich:dataTable"**
- ✓ Τον **scroller** ο οποίος είναι τύπου **"rich:dataScroller"**
- ✓ Το **tooltip** επεξήγησης το οποίο είναι τύπου **"rich:tooltip"**

Επεξήγηση Ρόλων

Το κουμπί **"Add Type"** είναι υπεύθυνο να παίρνει το κείμενο που έχει εισάγει ο χρήστης στο πεδίο **"Hotel Type"** και μετά από τους απαραίτητους ελέγχους να καταχωρεί το τύπο στο σύστημα.

Το κουμπί **"Un/Available"** είναι υπεύθυνο να παίρνει τον αριθμό που έχει εισάγει ο χρήστης στο πεδίο **"ID"**, ο οποίος αντιστοιχεί στην αυτόματη αρίθμηση από τη βάση που είναι μοναδικός για κάθε τύπο ξενοδοχείου και μετά από τους απαραίτητους ελέγχους να το ενεργοποιεί ή να απενεργοποιεί από το σύστημα.

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τους τύπους ξενοδοχείων του συστήματος. Οι τιμές που εμφανίζει είναι ο αύξων αριθμός ID, μοναδικός για κάθε τύπο, το όνομα του και τέλος το αν είναι διαθέσιμος ή όχι. Οι τύποι χωρίζονται σε σελίδες με τέσσερεις ανά σελίδα.

Ο **scroller** είναι υπεύθυνος στο να αλλάζει ο χρήστης τις σελίδες με τους τύπους που φαίνονται μέσα στο πίνακα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Κουμπί Add Type

Το κουμπί **"Add Type"** καλεί τη μέθοδο **"addStayingType"** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τη μεταβλητή τύπου String με όνομα **"stayingTypeAdmin"**, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **"Hotel Type"**, και να επικοινωνήσει με τη μέθοδο **"addStayingType", της web service "hoteldbAdd",** αποστέλλοντας της την τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<stayingtype>Airport</stayingtype>
</root>
```

Όπου stayingtype το όνομα του τύπου ξενοδοχείου που θέλει ο χρήστης να προστεθεί στη βάση. Μετα από μια επιτυχημένη προσθήκη θα καλεστεί η μέθοδος **fillStayingType** και θα ενεργοποιηθεί η διαδικασία για την ανανέωση των περιεχομένων του πίνακα.

Διαδικασία fillStayingType

Καλείται η μέθοδος **"fillStayingType"** από το bean. Αυτή η μέθοδος δεν δέχεται ορίσματα και σαν αποτέλεσμα δεν στέλνει κάποιο ωφέλημο μήνυμα. Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **"fillStayingType", της web service "hoteldbFill",** με σκοπό να παραλάβει ένα μήνυμα με τους τύπους ξενοδοχείων που είναι καταχωρημένοι στη βάση και συγκεκριμένα στον πίνακα **"stayingtype".**

Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>1</id>
<stayingtype>Hotel</stayingtype>
<flag>1</flag>
<id>2</id>
<stayingtype>Villa</stayingtype>
<flag>1</flag>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε τύπο στον πίνακα "**stayingtype"**. Το πεδίο **stayingtype** αντιπροσωπεύει το όνομα του τύπου και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν το συγκεκριμένος τύπος είναι ενεργοποιημένος ή όχι. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης

Κουμπί Edit Type

Το κουμπί **“Edit Type”** καλεί τη μέθοδο **“editStayingType”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“stayingTypeAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Hotel Type”**
- **“selectedStayingType”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“editStayingType”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<stayingtype>Hotel</stayingtype>
<stayingtypeID>1</stayingtypeID>
</root>
```

Όπου **stayingtype** το νέο όνομα του τύπου ξενοδοχείου τον οποίο διορθώνουμε και **stayingtypeID** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων που αντιστοιχεί στον τύπο.

Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **“fillStayingType”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Un/Available

Το κουμπί **“Un/Available”** καλεί τη μέθοδο **“deleteStayingType”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“selectedStayingType”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**

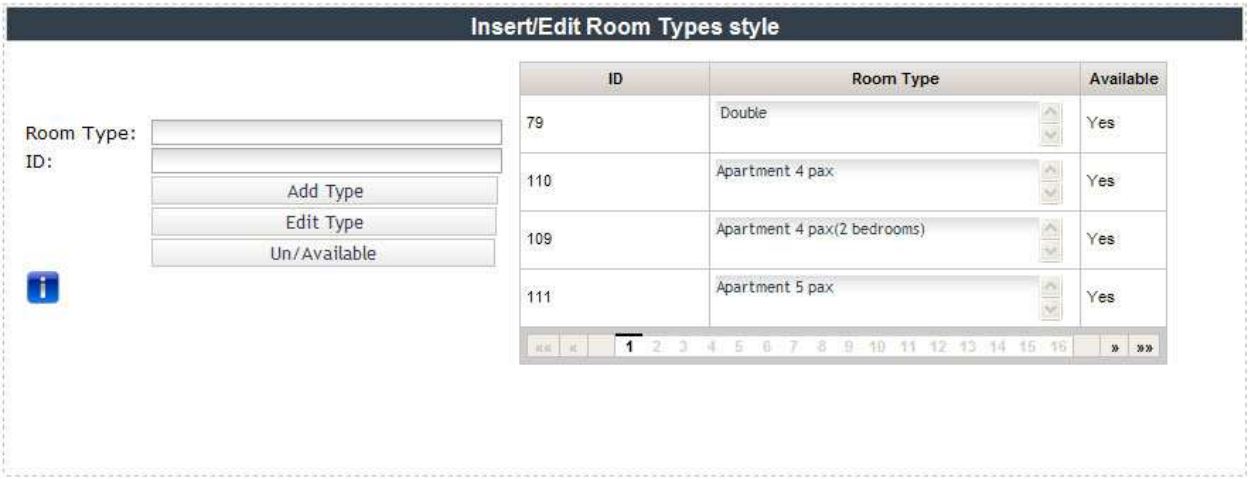
Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“deleteStayingType”, της web service “hoteldbDelete”,** αποστέλλοντας τη τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<stayingtype>9</stayingtype>
</root>
```

Όπου **stayingtype** το id του τύπου που θέλουμε να απενεργοποιήσουμε / ενεργοποιήσουμε.

Μετά από μια επιτυχημένη αλλαγή θα καλεστεί η μέθοδος **“fillStayingType”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Panel: Add/Edit Room Type



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής, διόρθωσης, προβολής, ενεργοποίησης και απενεργοποίησης των τύπων ξενοδοχείων που είναι διαθέσιμοι στο σύστημα.

Αποτελείται από

- ✓ Το πεδίο κειμένου **"Room Type"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Add Type"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το κουμπί **"Edit Type"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το πεδίο κειμένου **"ID"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Un/Available"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Τον **πίνακα** με τις κάρτες ο οποίος είναι **"rich:dataTable"**
- ✓ Τον **scroller** ο οποίος είναι τύπου **"rich:dataScroller"**
- ✓ Το **tooltip** επεξήγησης το οποίο είναι τύπου **"rich:tooltip"**

Επεξήγηση Ρόλων

Το κουμπί **"Add Type"** είναι υπεύθυνο να παίρνει το κείμενο που έχει εισάγει ο χρήστης στο πεδίο **"Room Type"** και μετά από τους απαραίτητους ελέγχους να καταχωρεί το τύπο στο σύστημα.

Το κουμπί **"Un/Available"** είναι υπεύθυνο να παίρνει τον αριθμό που έχει εισάγει ο χρήστης στο πεδίο **"ID"**, ο οποίος αντιστοιχεί στην αυτόματη αρίθμηση από τη βάση που είναι μοναδικός για κάθε τύπο δωματίου και μετά από τους απαραίτητους ελέγχους να το ενεργοποιεί ή να απενεργοποιεί από το σύστημα.

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τους τύπους δωματίων του συστήματος. Οι τιμές που εμφανίζει είναι ο αύξων αριθμός ID, μοναδικός για κάθε τύπο, το όνομα του και τέλος το αν είναι διαθέσιμος ή όχι. Οι τύποι χωρίζονται σε σελίδες με τέσσερεις ανά σελίδα.

Ο **scroller** είναι υπεύθυνος στο να αλλάζει ο χρήστης τις σελίδες με τους τύπους που φαίνονται μέσα στο πίνακα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Κουμπί Add Type

Το κουμπί **“Add Type”** καλεί τη μέθοδο **“addRoomSize”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τη μεταβλητή τύπου String με όνομα **“roomSizeAdmin”**, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Room Type”**, και να επικοινωνήσει με τη μέθοδο **“addRoomSize”, της web service “hoteldbAdd”,** αποστέλλοντας της την τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<room>Airport</room>
</root>
```

Όπου room το όνομα του τύπου δωματίου που θέλει ο χρήστης να προστεθεί στη βάση.
Μετα από μια επιτυχημένη προσθήκη θα καλεστεί η μέθοδος **fillRoomSize** και θα ενεργοποιηθεί η διαδικασία για την ανανέωση των περιεχομένων του πίνακα.

Διαδικασία fillRoomSize
Καλείται η μέθοδος **“fillRoomSize”** από το bean. Αυτή η μέθοδος δεν δέχεται ορίσματα και σαν αποτέλεσμα δεν στέλνει κάποιο ωφέλημο μήνυμα. Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **“fillRoomSize”, της web service “hoteldbFill”,** με σκοπό να παραλάβει ένα μήνυμα με τους τύπους ξενοδοχείων που είναι καταχωρημένοι στη βάση και συγκεκριμένα στον πίνακα **“roomsize”.**
Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<room>Single</room>
<flag>1</flag>
<id>3</id>
<room>Double</room>
<flag>1</flag>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε τύπο στον πίνακα “**roomsize”**. Το πεδίο **room** αντιπροσωπεύει το όνομα του τύπου και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν το συγκεκριμένος τύπος είναι ενεργοποιημένος ή όχι. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης

Κουμπί Edit Type
Το κουμπί **“Edit Type”** καλεί τη μέθοδο **“editRoomSize”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“roomSizeAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Room Type”**
- **“selectedRoomSizeAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“editRoomSize”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<room>Hotel</room>
<roomID>1</roomID>
</root>
```

Όπου **room** το νέο όνομα του τύπου δωματίου τον οποίο διορθώνουμε και **roomID** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων που αντιστοιχεί στον τύπο.
Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **“fillRoomSize”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Un/Available

Το κουμπί **“Un/Available”** καλεί τη μέθοδο **“deleteRoomSize”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

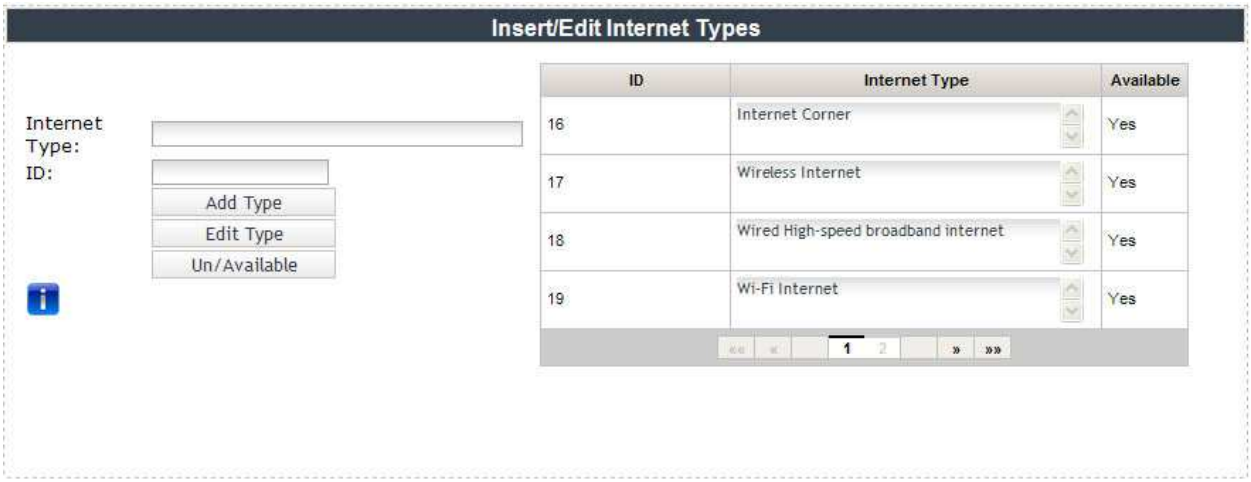
- **“selectedRoomSizeAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“deleteRoomSize”, της web service “hoteldbDelete”,** αποστέλλοντας τη τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

Όπου **id** ο αύξων αριθμός που αντιστοιχεί στον τύπο δωματίου που θέλουμε να απενεργοποιήσουμε / ενεργοποιήσουμε.
Μετά από μια επιτυχημένη αλλαγή θα καλεστεί η μέθοδος **“fillRoomSize”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Panel: Add/Edit Internet Type



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής, διόρθωσης, προβολής, ενεργοποίησης και απενεργοποίησης των τύπων interenet που είναι διαθέσιμοι στο σύστημα.

Αποτελείται από

- ✓ Το πεδίο κειμένου **"Internet Type"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Add Type"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το κουμπί **"Edit Type"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το πεδίο κειμένου **"ID"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Un/Available"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Τον **πίνακα** με τις κάρτες ο οποίος είναι **"rich:dataTable"**
- ✓ Τον **scroller** ο οποίος είναι τύπου **"rich:dataScroller"**
- ✓ Το **tooltip** επεξήγησης το οποίο είναι τύπου **"rich:tooltip"**

Επεξήγηση Ρόλων

Το κουμπί **"Add Type"** είναι υπεύθυνο να παίρνει το κείμενο που έχει εισάγει ο χρήστης στο πεδίο **"Internet Type"** και μετά από τους απαραίτητους ελέγχους να καταχωρεί το τύπο στο σύστημα.

Το κουμπί **"Un/Available"** είναι υπεύθυνο να παίρνει τον αριθμό που έχει εισάγει ο χρήστης στο πεδίο **"ID"**, ο οποίος αντιστοιχεί στην αυτόματη αρίθμηση από τη βάση που είναι μοναδικός για κάθε τύπο internet και μετά από τους απαραίτητους ελέγχους να το ενεργοποιεί ή να απενεργοποιεί από το σύστημα.

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τους τύπους internet του συστήματος. Οι τιμές που εμφανίζει είναι ο αύξων αριθμός ID, μοναδικός για κάθε τύπο, το όνομα του και τέλος το αν είναι διαθέσιμος ή όχι. Οι τύποι χωρίζονται σε σελίδες με τέσσερεις ανά σελίδα.

Ο **scroller** είναι υπεύθυνος στο να αλλάζει ο χρήστης τις σελίδες με τους τύπους που φαίνονται μέσα στο πίνακα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Κουμπί Add Type

Το κουμπί **"Add Type"** καλεί τη μέθοδο **"addInternetType"** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τη μεταβλητή τύπου String με όνομα

“internetTypeAdmin”, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Internet Type”**, και να επικοινωνήσει με τη μέθοδο **“addInternetType”, της web service “hoteldbAdd”,** αποστέλλοντας της την τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<internet>Airport</internet>
</root>
```

Όπου internet το όνομα του τύπου internet που θέλει ο χρήστης να προστεθεί στη βάση.
Μετα από μια επιτυχημένη προσθήκη θα καλεστεί η μέθοδος **fillInternetType** και θα ενεργοποιηθεί η διαδικασία για την ανανέωση των περιεχομένων του πίνακα.

Διαδικασία fillInternetType
Καλείται η μέθοδος **“fillInternetType”** από το bean. Αυτή η μέθοδος δεν δέχεται ορίσματα και σαν αποτέλεσμα δεν στέλνει κάποιο ωφέλημο μήνυμα. Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **“fillInternetType”, της web service “hoteldbFill”,** με σκοπό να παραλάβει ένα μήνυμα με τους τύπους ξενοδοχείων που είναι καταχωρημένοι στη βάση και συγκεκριμένα στον πίνακα **“internet”.**
Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<internet>Wireless</internet>
<flag>1</flag>
<id>3</id>
<internet>BroadBand</internet>
<flag>1</flag>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε τύπο στον πίνακα “**internet”**. Το πεδίο **internet** αντιπροσωπεύει το όνομα του τύπου και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν το συγκεκριμένος τύπος είναι ενεργοποιημένος ή όχι. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης

Κουμπί Edit Type
Το κουμπί **“Edit Type”** καλεί τη μέθοδο **“editInternetType”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“internetTypeAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Internet Type”**
- **“selectedInternetType”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“editInternetType”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<internet>Hotel</internet>
<internetID>1</internetID>
</root>
```

Όπου **internet** το νέο όνομα του τύπου internet τον οποίο διορθώνουμε και **internetID** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων που αντιστοιχεί στον τύπο.
Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **“fillInternetType”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Un/Available

Το κουμπί **“Un/Available”** καλεί τη μέθοδο **“deleteInternetType”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“selectedInternetType”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“ deleteInternetType”, της web service “hoteldbDelete”,** αποστέλλοντας τη τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<internet>2</internet>
</root>
```

Όπου **internet** ο αύξων αριθμός που αντιστοιχεί στον τύπο internet που θέλουμε να απενεργοποιήσουμε / ενεργοποιήσουμε.
Μετά από μια επιτυχημένη αλλαγή θα καλεστεί η μέθοδος **“fillInternetType”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Panel: Add/Edit Hotel Facilities



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής, διόρθωσης, προβολής, ενεργοποίησης και απενεργοποίησης των τύπων παροχών ξενοδοχείου που είναι διαθέσιμοι στο σύστημα.

Αποτελείται από

- ✓ Το πεδίο κειμένου **"Facility"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Add Facility"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το κουμπί **"Edit Facility"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το πεδίο κειμένου **"ID"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Un/Available"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Τον **πίνακα** με τις κάρτες ο οποίος είναι **"rich:dataTable"**
- ✓ Τον **scroller** ο οποίος είναι τύπου **"rich:dataScroller"**
- ✓ Το **tooltip** επεξήγησης το οποίο είναι τύπου **"rich:tooltip"**

Επεξήγηση Ρόλων

Το κουμπί **"Add Facility"** είναι υπεύθυνο να παίρνει το κείμενο που έχει εισάγει ο χρήστης στο πεδίο **"Facility"** και μετά από τους απαραίτητους ελέγχους να καταχωρεί το τύπο στο σύστημα.

Το κουμπί **"Edit Facility"** είναι υπεύθυνο να παίρνει το κείμενο που έχει εισάγει ο χρήστης στο πεδίο **"ID"** και το πεδίο **"Facility"** και μετά από τους απαραίτητους ελέγχους να διορθώνει το τύπο στο σύστημα.

Το κουμπί **"Un/Available"** είναι υπεύθυνο να παίρνει τον αριθμό που έχει εισάγει ο χρήστης στο πεδίο **"ID"**, ο οποίος αντιστοιχεί στην αυτόματη αρίθμηση από τη βάση που είναι μοναδικός για κάθε τύπο facility και μετά από τους απαραίτητους ελέγχους να τον ενεργοποιεί ή να απενεργοποιεί από το σύστημα.

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τους τύπους hotel facilities του συστήματος. Οι τιμές που εμφανίζει είναι ο αύξων αριθμός ID, μοναδικός για κάθε τύπο, το όνομα του και τέλος το αν είναι διαθέσιμος ή όχι. Οι τύποι χωρίζονται σε σελίδες με τέσσερεις ανά σελίδα.

Ο **scroller** είναι υπεύθυνος στο να αλλάζει ο χρήστης τις σελίδες με τους τύπους που φαίνονται μέσα στο πίνακα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Κουμπί Add Facility

Το κουμπί **“Add Facility”** καλεί τη μέθοδο **“addFacility”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τη μεταβλητή τύπου String με όνομα **“facilityAdmin”**, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Facility”**, και να επικοινωνήσει με τη μέθοδο **“addFacility”, της web service “hoteldbAdd”,** αποστέλλοντας της την τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<facility>Bar</facility>
</root>
```

Όπου facility το όνομα του τύπου facility που θέλει ο χρήστης να προστεθεί στη βάση.
Μετα από μια επιτυχημένη προσθήκη θα καλεστεί η μέθοδος **fillFacilities** και θα ενεργοποιηθεί η διαδικασία για την ανανέωση των περιεχομένων του πίνακα.

Διαδικασία fillFacilities

Καλείται η μέθοδος **“fillFacilities”** από το bean. Αυτή η μέθοδος δεν δέχεται ορίσματα και σαν αποτέλεσμα δεν στέλνει κάποιο ωφέλημο μήνυμα. Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **“fillFacilities”, της web service “hoteldbFill”,** με σκοπό να παραλάβει ένα μήνυμα με τους τύπους ξενοδοχείων που είναι καταχωρημένοι στη βάση και συγκεκριμένα στον πίνακα **“internet”.**
Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<facility_name>Bar</facility_name>
<flag>1</flag>
<id>3</id>
<facility_name>Beach Bar</facility_name>
<flag>1</flag>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε τύπο στον πίνακα “**facilities”**. Το πεδίο **facility_name** αντιπροσωπεύει το όνομα του τύπου και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν το συγκεκριμένος τύπος είναι ενεργοποιημένος ή όχι. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης

Κουμπί Edit Facility

Το κουμπί **“Edit Facility”** καλεί τη μέθοδο **“editFacility”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“facilityAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Facility”**

- **“selectedFacilityAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“editFacility”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<facility>Bar</facility>
<facilityID>2</facilityID>
</root>
```

Όπου **facility** το νέο όνομα του τύπου hotel facility τον οποίο διορθώνουμε και **facilityID** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων που αντιστοιχεί στον τύπο.
Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **“fillFacilities”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Un/Available

Το κουμπί **“Un/Available”** καλεί τη μέθοδο **“deleteFacility”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“selectedFacilityAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“ deleteFacility”, της web service “hoteldbDelete”,** αποστέλλοντας τη τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<facility_id>2</facility_id>
</root>
```

Όπου **facility_id** ο αύξων αριθμός που αντιστοιχεί στον τύπο hotel facility που θέλουμε να απενεργοποιήσουμε / ενεργοποιήσουμε.
Μετά από μια επιτυχημένη αλλαγή θα καλεστεί η μέθοδος **“fillFacilities”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Panel: Add/Edit Room Facilities



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής, διόρθωσης, προβολής, ενεργοποίησης και απενεργοποίησης των τύπων παροχών δωματίων που είναι διαθέσιμοι στο σύστημα.

Αποτελείται από

- ✓ Το πεδίο κειμένου **"Facility"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Add Facility"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το κουμπί **"Edit Facility"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το πεδίο κειμένου **"ID"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Un/Available"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Τον **πίνακα** με τις κάρτες ο οποίος είναι **"rich:dataTable"**
- ✓ Τον **scroller** ο οποίος είναι τύπου **"rich:dataScroller"**
- ✓ Το **tooltip** επεξήγησης το οποίο είναι τύπου **"rich:tooltip"**

Επεξήγηση Ρόλων

Το κουμπί **"Add Facility"** είναι υπεύθυνο να παίρνει το κείμενο που έχει εισάγει ο χρήστης στο πεδίο **"Facility"** και μετά από τους απαραίτητους ελέγχους να καταχωρεί το τύπο στο σύστημα.

Το κουμπί **"Edit Facility"** είναι υπεύθυνο να παίρνει το κείμενο που έχει εισάγει ο χρήστης στο πεδίο **"ID"** και το πεδίο **"Facility"** και μετά από τους απαραίτητους ελέγχους να διορθώνει το τύπο στο σύστημα.

Το κουμπί **"Un/Available"** είναι υπεύθυνο να παίρνει τον αριθμό που έχει εισάγει ο χρήστης στο πεδίο **"ID"**, ο οποίος αντιστοιχεί στην αυτόματη αρίθμηση από τη βάση που είναι μοναδικός για κάθε τύπο room facility και μετά από τους απαραίτητους ελέγχους να τον ενεργοποιεί ή να απενεργοποιεί από το σύστημα.

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τους τύπους room facilities του συστήματος. Οι τιμές που εμφανίζει είναι ο αύξων αριθμός ID, μοναδικός για κάθε τύπο, το όνομα του και τέλος το αν είναι διαθέσιμος ή όχι. Οι τύποι χωρίζονται σε σελίδες με τέσσερεις ανά σελίδα.

Ο **scroller** είναι υπεύθυνος στο να αλλάζει ο χρήστης τις σελίδες με τους τύπους που φαίνονται μέσα στο πίνακα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Κουμπί Add Facility

Το κουμπί **“Add Facility”** καλεί τη μέθοδο **“addRoomFacility”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τη μεταβλητή τύπου String με όνομα **“roomfacilityAdmin”**, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Facility”**, και να επικοινωνήσει με τη μέθοδο **“addRoomFacility”, της web service “hoteldbAdd”,** αποστέλλοντας της την τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<RoomFacility>Bar</RoomFacility>
</root>
```

Όπου **RoomFacility** το όνομα του τύπου facility που θέλει ο χρήστης να προστεθεί στη βάση.
Μετα από μια επιτυχημένη προσθήκη θα καλεστεί η μέθοδος **fillRoomFacilities** και θα ενεργοποιηθεί η διαδικασία για την ανανέωση των περιεχομένων του πίνακα.

Διαδικασία fillRoomFacilities

Καλείται η μέθοδος **“fillRoomFacilities”** από το bean. Αυτή η μέθοδος δεν δέχεται ορίσματα και σαν αποτέλεσμα δεν στέλνει κάποιο ωφέλημο μήνυμα. Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **“fillRoomFacilities”, της web service “hoteldbFill”,** με σκοπό να παραλάβει ένα μήνυμα με τους τύπους ξενοδοχείων που είναι καταχωρημένοι στη βάση και συγκεκριμένα στον πίνακα **“roomfacilities”.**
Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<RoomFacility>Mini Bar</RoomFacility>
<flag>1</flag>
<id>3</id>
<RoomFacility>Shower</RoomFacility>
<flag>1</flag>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε τύπο στον πίνακα “**roomfacilities”**. Το πεδίο **RoomFacility** αντιπροσωπεύει το όνομα του τύπου και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν το συγκεκριμένος τύπος είναι ενεργοποιημένος ή όχι. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης

Κουμπί Edit Facility

Το κουμπί **“Edit Facility”** καλεί τη μέθοδο **“editRoomFacility”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“roomfacilityAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Facility”**
- **“selectedRoomFacilityAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“editRoomFacility”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<RoomFacility>Mini Bar</RoomFacility>
<RoomFacilityID>2</RoomFacilityID>
</root>
```

Όπου **RoomFacility** το νέο όνομα του τύπου room facility τον οποίο διορθώνουμε και **RoomFacilityID** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων που αντιστοιχεί στον τύπο.
Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **“fillRoomFacilities”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Un/Available

Το κουμπί **“Un/Available”** καλεί τη μέθοδο **“deleteRoomFac”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

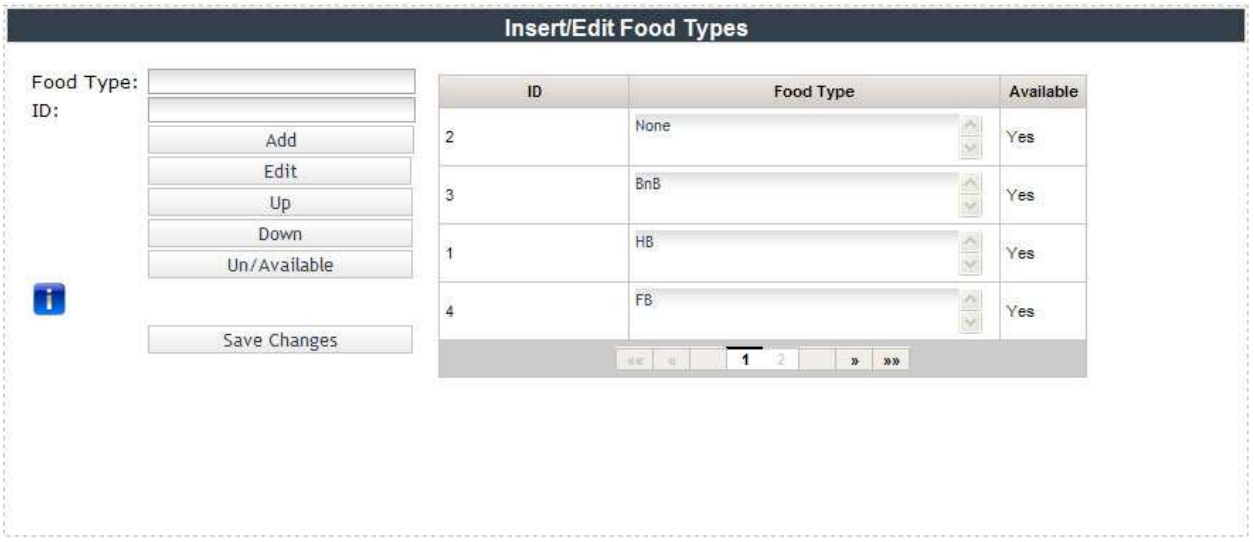
- **“selectedRoomFacilityAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“deleteRoomFac”, της web service “hoteldbDelete”,** αποστέλλοντας τη τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

Όπου **id** ο αύξων αριθμός που αντιστοιχεί στον τύπο room facility που θέλουμε να απενεργοποιήσουμε / ενεργοποιήσουμε.
Μετά από μια επιτυχημένη αλλαγή θα καλεστεί η μέθοδος **“fillRoomFacilities”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Panel: Add/Edit Food Types



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής, διόρθωσης, προβολής, ενεργοποίησης και απενεργοποίησης των τύπων φαγητού που είναι διαθέσιμοι στο σύστημα.

Αποτελείται από

- ✓ Το πεδίο κειμένου **"Food Type"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το πεδίο κειμένου **"ID"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Add"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το κουμπί **"Edit"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το κουμπί **"Up"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το κουμπί **"Down"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το κουμπί **"Un/Available"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το κουμπί **"Save Changes"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Τον **πίνακα** με τις κάρτες ο οποίος είναι **"rich:dataTable"**
- ✓ Τον **scroller** ο οποίος είναι τύπου **"rich:dataScroller"**
- ✓ Το **tooltip** επεξήγησης το οποίο είναι τύπου **"rich:tooltip"**

Επεξήγηση Ρόλων

Οι τύποι των φαγητών που είναι διαθέσιμοι στο σύστημα αποθηκεύονται προσωρινά σε μια λίστα με το καλεστεί η διαδικασια και γεμίσει ο πίνακας. Σε αυτή τη λίστα πάνω κάνει ο χρήστης όλες τις αλλαγές του με τα κουμπιά που είχει στην επιλογή του και όταν τελειώσει με τις αλλαγές πατώντας το κουμπί **Save Changes** αποθηκεύονται οι αλλαγές που έχει κάνει στο σύστημα.

Το κουμπί **“Add”** είναι υπεύθυνο να παίρνει το κείμενο που έχει εισάγει ο χρήστης στο πεδίο **“Food Type”** και να προσθέτει τον τύπο φαγητού στη προσωρινή λίστα.

Το κουμπί **“Edit”** είναι υπεύθυνο να παίρνει το κείμενο που έχει εισάγει ο χρήστης στο πεδίο **“ID”** και το πεδίο **“Food Type”** και να βρίσκει στη λίστα τον τύπο φαγητού με το συγκεκριμένο id και να τον αλλάζει.

Το κουμπί **“Un/Available”** είναι υπεύθυνο να παίρνει τον αριθμό που έχει εισάγει ο χρήστης στο πεδίο **“ID”**, ο οποίος αντιστοιχεί στην αυτόματη αρίθμηση από τη βάση που είναι μοναδικός για κάθε τύπο φαγητού και να τον ενεργοποιεί ή να απενεργοποιεί στη προσωρινή λίστα

Το κουμπί **“Up”** είναι υπεύθυνο να παίρνει το κείμενο που έχει εισάγει ο χρήστης στο πεδίο **“ID”** και να βρίσκει στη λίστα τον τύπο φαγητού με το συγκεκριμένο id και να τον ανεβάζει μια θέση στη λίστα.

Το κουμπί **“Down”** είναι υπεύθυνο να παίρνει το κείμενο που έχει εισάγει ο χρήστης στο πεδίο **“ID”** και να βρίσκει στη λίστα τον τύπο φαγητού με το συγκεκριμένο id και να τον κατεβάζει μια θέση στη λίστα.

Το κουμπί **“Save Changes”** είναι υπεύθυνο να αποθηκεύει τη λίστα με τις αλλαγές στη βάση αντικαθηστώντας τη παλιά λίστα με τους τύπους φαγητών.

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τους τύπους room facilities του συστήματος. Οι τιμές που εμφανίζει είναι ο αύξων αριθμός ID, μοναδικός για κάθε τύπο, το όνομα του και τέλος το αν είναι διαθέσιμος ή όχι. Οι τύποι χωρίζονται σε σελίδες με τέσσερεις ανά σελίδα.

Ο **scroller** είναι υπεύθυνος στο να αλλάζει ο χρήστης τις σελίδες με τους τύπους που φαίνονται μέσα στο πίνακα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Διαδικασία fillFoodType

Καλείται η μέθοδος **“fillFoodType”** από το bean. Αυτή η μέθοδος δεν δέχεται ορίσματα και σαν αποτέλεσμα δεν στέλνει κάποιο ωφέλημο μήνυμα. Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **“fillFoodType”, της web service “hoteldbFill”,** με σκοπό να παραλάβει ένα μήνυμα με τους τύπους ξενοδοχείων που είναι καταχωρημένοι στη βάση και συγκεκριμένα στον πίνακα **“food”.**

Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>3</id>
<food>BnB</food>
<flag>1</flag>
<id>1</id>
<food>HB</food>
<flag>1</flag>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε τύπο στον πίνακα “**food”**. Το πεδίο **food** αντιπροσωπεύει το όνομα του τύπου φαγητού και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν το συγκεκριμένος τύπος είναι ενεργοποιημένος ή όχι. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης και θα αποθηκευτεί εκ νέου η προσωρινή λίστα.

Κουμπί Add

Το κουμπί **“Add”** καλεί τη μέθοδο **“foodTypeAdd”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τη μεταβλητή τύπου String με όνομα **“foodTypeAdmin”**, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Food”** και μετα τους απαραίτητους ελέγχους να προσθέσει τον τύπο φαγητού στη προσωρινή λίστα δίνοντας του και έναν αύξων αριθμό μοναδικό για κάθε τύπο.

Κουμπί Edit

Το κουμπί **“Edit”** καλεί τη μέθοδο **“foodTypeEdit”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“foodTypeAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Food”**
- **“selectedFoodTypeAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**

Στη συνέχεια θα ψάξει τη προσωρινή λίστα για να βρεί τον τύπο φαγητού με το συγκεκριμένο **ID** και θα του αλλάξει το όνομα σύμφωνα με τη μεταβλητή **foodTypeAdmin**.

Κουμπί Un/Available

Το κουμπί **“Un/Available”** καλεί τη μέθοδο **“foodTypeDelete”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“selectedFoodTypeAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**

Στη συνέχεια θα ψάξει τη προσωρινή λίστα για να βρεί τον τύπο φαγητού με το συγκεκριμένο **ID** και θα του αλλάξει την τιμή του πεδίου **flag** έτσι ώστε να τον ενεργοποιήσει ή να τον απενεργοποιήσει.

Κουμπί Up

Το κουμπί **“Up”** καλεί τη μέθοδο **“foodTypeUp”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“selectedFoodTypeAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**

Στη συνέχεια θα ψάξει τη προσωρινή λίστα για να βρεί τον τύπο φαγητού με το συγκεκριμένο **ID** και θα τον ανεβάσει μια θέση στη λίστα έως ότου να φτάσει στη πρώτη.

Κουμπί Down

Το κουμπί **“Down”** καλεί τη μέθοδο **“foodTypeDown”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“selectedFoodTypeAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID”.**

Στη συνέχεια θα ψάξει τη προσωρινή λίστα για να βρεί τον τύπο φαγητού με το συγκεκριμένο **ID** και θα τον κατεβάσει μια θέση στη λίστα έως ότου να φτάσει στην τελευταία.

Κουμπί Save Changes

Το κουμπί **“Save Changes”** καλεί τη μέθοδο **“saveFoodType”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να περάσει την προσωρινή λίστα με τους τύπους φαγητών σε ένα μήνυμα και να επικοινωνήσει με τη μέθοδο **“addFood”** της webservice **“hoteldbAdd”** στέλνοντας το μήνυμα. Το μήνυμα είναι ένα SOAP, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

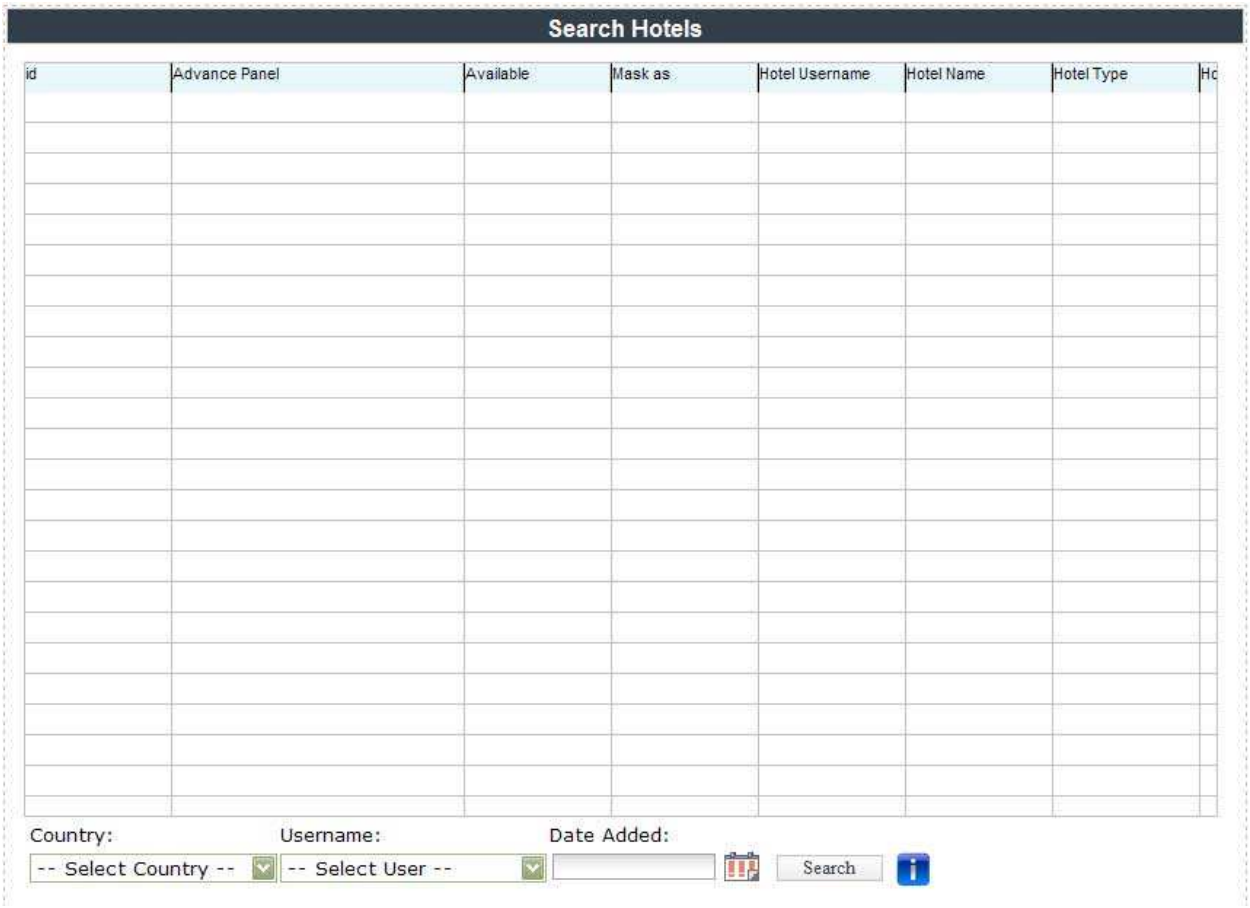
```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>1</id>
<food>None</food>
<flag>1</flag>
<id>2</id>
<food>BnB</food>
<flag>1</flag>
<id>3</id>
<food>HB</food>
<flag>1</flag>
</root>
```

Όπου **id** ο αύξων αριθμός που αντιστοιχεί στον τύπο φαγητού, **food** το όνομα του τύπου και **flag** το πεδίο που περιλαμβάνει το αν ο τύπος είναι ενεργοποιημένος ή όχι.

Μετά από μια επιτυχημένη καταχώρηση θα καλεστεί η μέθοδος **“fillFoodType”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

1.1.2 Search Hotels / Set Prices

Panel: Search Hotels



Λειτουργικότητα

Εμφανίζει τη διεπαφή της προβολής, ενεργοποίησης και απενεργοποίησης των ξενοδοχείων που είναι διαθέσιμα στο σύστημα.

Αποτελείται από

- ✓ Το μενού επιλογής **"Country"** το οποίο είναι ένα **"h:selectOneMenu"**
- ✓ Το μενού επιλογής **"Username"** το οποίο είναι ένα **"h:selectOneMenu"**
- ✓ Το μενού επιλογής ημερομηνίας **"Date Added"** το οποίο είναι ένα **"rich:calendar"**
- ✓ Το κουμπί **"Search"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Τον **πίνακα** με τα ξενοδοχεία οποίος είναι **"rich:dataTable"**
- ✓ Το **tooltip** επεξήγησης το οποίο είναι τύπου **"rich:tooltip"**

Επεξήγηση Ρόλων

Το μενού επιλογής **"Country"** είναι υπεύθυνο να παίρνει την επιλογή του χρήστη από τη λίστα η οποία γεμίζει με τις διαθέσιμες χώρες του συστήματος. Η χώρα θα χρησιμοποιηθεί για την αναζήτηση του ξενοδοχείου.

Το μενου επιλογής **“Username”** είναι υπεύθυνο να παίρνει την επιλογή του χρήστη από τη λίστα η οποία γεμίζει με τους διαθέσιμους χρήστες του συστήματος. Ο επιλεγμένος χρήστης θα χρησιμοποιηθεί για την αναζήτηση του ξενοδοχείου.

Το μενου επιλογής ημερομηνίας **“Date added”** είναι υπεύθυνο να παίρνει την επιλογή του χρήστη για την ημερομηνία εισαγωγής ενός ξενοδοχείου. Η επιλεγμένη ημερομηνία θα χρησιμοποιηθεί για την αναζήτηση του ξενοδοχείου.

Το κουμπί **“Search”** είναι υπεύθυνο να παίρνει τις επιλογές του χρήστη και μετα τους απαραίτητους ελέγχους να γεμίζει τον πίνακα με τα ξενοδοχεία που βρέθηκαν.

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τα ξενοδοχεία που προέκυψαν από την αναζήτηση. Κάθε γραμμή του πίνακα θα κατασκευαστεί για μια γραμμή του πίνακα **“hoteltable”** της βάσης δεδομένων που πληρεί τα κριτήρια αναζήτησης.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Κουμπί Search

Το κουμπί **“Search”** καλεί τη μέθοδο **“fillSearchAdmin”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη να διαβάσει τις επιλογές του χρήστη και να καλέσει την αντίστοιχη μέθοδο ανάλογα τον συνδυασμό των επιλογών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που ανάλογα την μέθοδο περιέχει ένα xml έγγραφο.

i) fillHotelAdmin()

Δεν παίρνει κάποιο όρισμα και το xml που στέλνει δεν έχει κάποια ωφέλιμη μορφή. Επικοινωνεί με την μέθοδο **“fillHotelAdmin”** της webservice **hoteldbFill.**

ii) fillHotelAdminCountry()

Παίρνει σαν όριμα το όνομα της χώρας κάτω από την οποία αναζητούμε τα ξενοδοχεία. Επικοινωνεί με την μέθοδο **“fillHotelAdminCountry”** της webservice **hoteldbFill.**

Το xml που στέλνει είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
</root>
```

iii) fillHotelAdminName()

Παίρνει σαν όρισμα το username του χρήστη του οποίου ψάχνουμε τα ξενοδοχεία. Επικοινωνεί με την μέθοδο **“fillHotelAdminName”** της webservice **hoteldbFill.**

Το xml που στέλνει είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<user_name>username</user_name>
```

</root>

iv) fillHotelAdminCountryName()

Παίρνει σαν όρισμα το username του χρήστη και το όνομα της χώρας. Τα ξενοδοχεία θα πρέπει να πληρούν όλα τα κριτήρια. Επικοινωνεί με την μέθοδο **“fillHotelAdminCountryName”** της webservice **hoteldbFill.**

Το xml που στέλνει είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<user_name>username</user_name>
</root>
```

v) fillHotelAdminDate()

Παίρνει σαν όρισμα την ημερομηνία που επέλεξε ο χρήστης. Τα ξενοδοχεία θα πρέπει να έχουν ημερομηνία εισαγωγής στο σύστημα την επιλεγμένη. Επικοινωνεί με την μέθοδο **“fillHotelAdminDate”** της webservice **hoteldbFill.**

Το xml που στέλνει είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<dateadd>2010-03-02</dateadd>
</root>
```

vi) fillHotelAdminCountryDate()

Παίρνει σαν όρισμα την ημερομηνία και την χώρα που επέλεξε ο χρήστης. Τα ξενοδοχεία θα πρέπει να πληρούν όλα τα κριτήρια. Επικοινωνεί με την μέθοδο **“fillHotelAdminCountryDate”** της webservice **hoteldbFill.**

Το xml που στέλνει είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<dateadd>2010-03-02</dateadd>
</root>
```

vii) fillHotelAdminNameDate()

Παίρνει σαν όρισμα την ημερομηνία και τον χρήστη που επέλεξε ο χρήστης. Τα ξενοδοχεία θα πρέπει να πληρούν όλα τα κριτήρια. Επικοινωνεί με την μέθοδο **“fillHotelAdminNameDate”** της webservice **hoteldbFill.**

Το xml που στέλνει είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<user_name>username</user_name>
<dateadd>2010-03-02</dateadd>
</root>
```

viii) fillHotelAdminCountryNameDate()

Παίρνει σαν όρισμα την ημερομηνία, τον χρήστη και τη χώρα που επέλεξε ο χρήστης. Τα ξενοδοχεία θα πρέπει να πληρούν όλα τα κριτήρια. Επικοινωνεί με την μέθοδο **"fillHotelAdminCountryNameDate"** της webservice **hoteldbFill.**

Το xml που στέλνει είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<user_name>username</user_name>
<dateadd>2010-03-02</dateadd>
</root>
```

Διαδικασία fillHotelAdmin

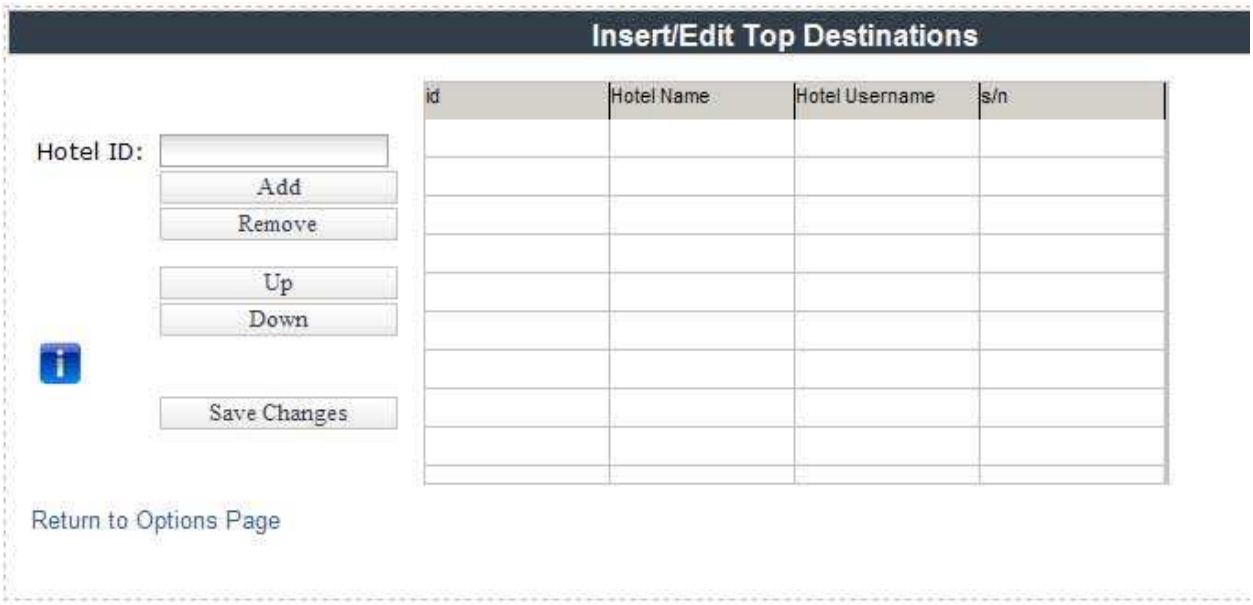
Από όλες τις μεθόδους, ανάλογα τις επιλογές του χρήστη, που κάλεστηκαν θα επιστρέψει το παρακάτω xml που περιλαμβάνει τις πληροφορίες για τα ξενοδοχεία που βρέθηκαν.

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<name>Grand Hotel</name>
<stars>4</stars>
<email>info@grandhotel.com</email>
<phone>302101231234</phone>
<hoteladdress>Kentro A8inwn</hoteladdress>
<totalrooms>233</totalrooms>
<hotelchain>Grand Hotels</hotelchain>
<fax>302101231234</fax>
<stayingtype>Hotel</stayingtype>
<city_name>Athens</city_name>
<location>Athens Centre</location>
<region_name>Attica</region_name>
<country_name>Greece</country_name>
<dateadd>2010-01-23</dateadd>
<checkin>10:00</checkin>
<checkout>12:00</checkout>
<flag>1</flag>
<user_name>username</user_name>
<rating>9</rating>
```

id – Ο αύξων αριθμός από την αυτόματη αρίθμιση της βάσης και είναι μοναδικός για κάθε ξενοδοχείο.
name – Το όνομα του ξενοδοχείου
stars – Τα αστέρια του ξενοδοχείου
email – Το email του ξενοδοχείου
phone – Το τήλεφωνο του ξενοδοχείου
hoteladdress – Η διεύθυνση του ξενοδοχείου
totalrooms – Ο συνολικός αριθμός δωματίοων του ξενοδοχείου
hotelchain – Η αλυσίδα που ανήκει το ξενοδοχείο
fax – Το fax του ξενοδοχείου
stayingtype – Ο τύπος του ξενοδοχείου
city_name – Το όνομα της πόλης του ξενοδοχείου
location – Το όνομα της τοποθεσίας του ξενοδοχείου
region_name – Το όνομα της επαρχίας του ξενοδοχείου
country_name – Το όνομα της χώρας του ξενοδοχείου
dateadd – Η ημερομηνία που καταχώρηθηκε το ξενοδοχείο στο σύστημα
checkin – Η ώρα του checkin του ξενοδοχείου
checkout – Η ώρα του checkout του ξενοδοχείου
flag – Η μεταβλητή που δείχνει αν το ξενοδοχείο είναι ενεργό ή όχι
user_name – Ο χρήστης που είναι υπεύθυνος για το ξενοδοχείο
rating – Η βαθμολογία του ξενοδοχείου από τους πελάτες.

Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης

Panel: Add/Edit Top Destinations



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής, διόρθωσης, προβολής, ενεργοποίησης και απενεργοποίησης των προτεινόμενων ξενοδοχείων που θα φαίνονται στην αρχική σελίδα.

Αποτελείται από

- ✓ Το πεδίο κειμένου **"Hotel ID"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Add"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το κουμπί **"Up"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το κουμπί **"Down"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το κουμπί **"Remove"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το κουμπί **"Save Changes"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Τον **πίνακα** με τις κάρτες ο οποίος είναι **"rich:dataTable"**
- ✓ Το **tooltip** επεξήγησης το οποίο είναι τύπου **"rich:tooltip"**

Επεξήγηση Ρόλων

Τα προτεινόμενα ξενοδοχεία αποθηκεύονται προσωρινά σε μια λίστα με το που καλεστεί η διαδικασια και γεμίσει ο πίνακας. Σε αυτή τη λίστα πάνω κάνει ο χρήστης όλες τις αλλαγές του με τα κουμπιά που είχει στην επιλογή του και όταν τελειώσει με τις αλλαγές πατώντας το κουμπί **Save Changes** αποθηκεύονται οι αλλαγές που έχει κάνει στο σύστημα.

Το κουμπί **"Add"** είναι υπεύθυνο να παίρνει το id που έχει εισάγει ο χρήστης στο πεδίο **"Hotel ID"** και να προσθέτει το ξενοδοχείο στη προσωρινή λίστα.

Το κουμπί **"Remove"** είναι υπεύθυνο να παίρνει τον αριθμό που έχει εισάγει ο χρήστης στο πεδίο **"Hotel ID"** και να αφαιρεί το ξενοδοχείο από τη λίστα.

Το κουμπί **"Up"** είναι υπεύθυνο να παίρνει το id που έχει εισάγει ο χρήστης και να βρίσκει στη λίστα το ξενοδοχείο με το συγκεκριμένο id και να τον ανεβάζει μια θέση στη λίστα.

Το κουμπί **"Down"** είναι υπεύθυνο να παίρνει το κείμενο που έχει εισάγει ο χρήστης και να βρίσκει στη λίστα το ξενοδοχείο με το συγκεκριμένο id και να τον κατεβάζει μια θέση στη λίστα.

Το κουμπί **"Save Changes"** είναι υπεύθυνο να αποθηκεύει τη λίστα με τις αλλαγές στη βάση αντικαθηστώντας τη παλιά λίστα με τα προτεινόμενα ξενοδοχεία.

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τα προτεινόμενα ξενοδοχεία. Οι τιμές που εμφανίζει είναι ο αύξων αριθμός ID, μοναδικός για κάθε ξενοδοχείο, το όνομα του, το

username του χρήστη που είναι υπευθυνος για το ξενοδοχείο και τέλος ο αύξων αριθμός του κάθε ξενοδοχείου στην προσωρινή λίστα. Η λίστα δέχεται έως επτά ξενοδοχεία

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Διαδικασία fillFoodType

Καλείται η μέθοδος **"fillTopDestinationsAdmin"** από το bean. Αυτή η μέθοδος δεν δέχεται ορίσματα και σαν αποτέλεσμα δεν στέλνει κάποιο ωφέλημο μήνυμα. Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **"fillTopDestinationsAdmin", της web service "hoteldbFill",** με σκοπό να παραλάβει ένα μήνυμα με τα προτεινόμενα ξενοδοχεία που είναι καταχωρημένα στη βάση και συγκεκριμένα στον πίνακα **"topdestinations".**
Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<name>Grand Hotel</name>
<user_name>Username</user_name>
</root>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε ξενοδοχείο. Το πεδίο **name** αντιπροσωπεύει το όνομα του ξενοδοχείου και τέλος το πεδίο **user_name** στο οποίο φαίνεται ο χρήστης που είναι υπεύθυνος για το ξενοδοχείο. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης και θα αποθηκευτεί εκ νέου η προσωρινή λίστα.

Κουμπί Add

Το κουμπί **"Add"** καλεί τη μέθοδο **"topHotelAdd"** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τη μεταβλητή τύπου String με όνομα **"selectedTopHotelAdmin"**, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **"Hotel ID"** και μετα τους απαραίτητους ελέγχους να προσθέσει το ξενοδοχείο στη προσωρινή λίστα δίνοντας του και έναν αύξων αριθμό μοναδικό για κάθε ξενοδοχείο.

Κουμπί Remove

Το κουμπί **"Remove"** καλεί τη μέθοδο **"topHotelDelete"** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **"selectedTopHotelAdmin"** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **"Hotel ID".**

Στη συνέχεια θα ψάξει τη προσωρινή λίστα για να βρεί το ξενοδοχείο με το συγκεκριμένο **ID** και θα το αφαιρέσει από τη λίστα.

Κουμπί Up

Το κουμπί **“Up”** καλεί τη μέθοδο **“topHotelUp”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“selectedTopHotelAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Hotel ID”.**

Στη συνέχεια θα ψάξει τη προσωρινή λίστα για να βρεί το ξενοδοχείο με το συγκεκριμένο **ID** και θα το ανεβάσει μια θέση στη λίστα έως ότου να φτάσει στη πρώτη.

Κουμπί Down

Το κουμπί **“Down”** καλεί τη μέθοδο **“topHotelDown”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“selectedTopHotelAdmin”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Hotel ID”.**

Στη συνέχεια θα ψάξει τη προσωρινή λίστα για να βρεί το ξενοδοχείο με το συγκεκριμένο **ID** και θα το κατεβάσει μια θέση στη λίστα έως ότου να φτάσει στην τελευταία.

Κουμπί Save Changes

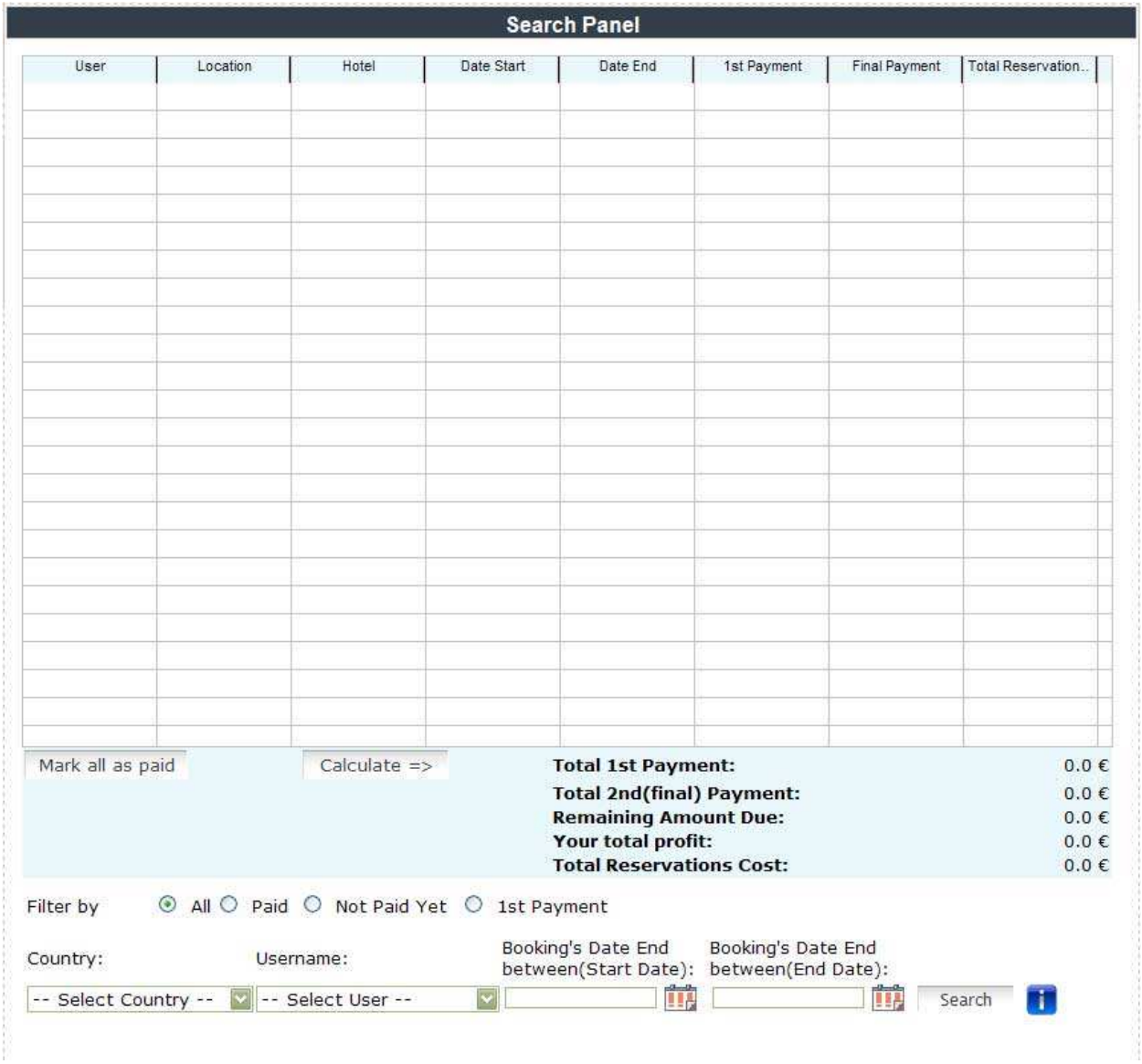
Το κουμπί **“Save Changes”** καλεί τη μέθοδο **“saveTopDestinations”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να περάσει την προσωρινή λίστα με τα προτεινόμενα ξενοδοχεία σε ένα μήνυμα και να επικοινωνήσει με τη μέθοδο **“addTopDestinations”** της webservice **“hoteldbAdd”** στέλνοντας το μήνυμα. Το μήνυμα είναι ένα SOAP, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<id>4</id>
</root>
```

Όπου **id** ο αύξων αριθμός που αντιστοιχεί στο ξενοδοχείο. Μετά από μια επιτυχημένη καταχώρηση θα καλεστεί η μέθοδος **“fillTopDestinationsAdmin”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

1.1.2 Statistics

Panel: Statistics



Λειτουργικότητα

Εμφανίζει τη διεπαφή της αναζήτησης, προβολής, ανάλυσης και πληρωμής των ποσών από τις κρατήσεις που έχουν γίνει στο σύστημα.

Αποτελείται από

- ✓ Το μενού επιλογής **"Country"** το οποίο είναι ένα **"h:selectOneMenu"**
- ✓ Το μενού επιλογής **"Username"** το οποίο είναι ένα **"h:selectOneMenu"**
- ✓ Το μενού επιλογής ημερομηνίας **"Start Date"** το οποίο είναι ένα **"rich:calendar"**
- ✓ Το μενού επιλογής ημερομηνίας **"End Date"** το οποίο είναι ένα **"rich:calendar"**
- ✓ Το κουμπί **"Search"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Τον **πίνακα** με τις κρατήσεις οποίος είναι **"rich:scrollableDataTable"**
- ✓ Το **tooltip** επεξήγησης το οποίο είναι τύπου **"rich:tooltip"**

Επεξήγηση Ρόλων

Το μενου επιλογής **“Country”** είναι υπεύθυνο να παίρνει την επιλογή του χρήστη από τη λίστα η οποία γεμίζει με τις διαθέσιμες χώρες του συστήματος. Η χώρα θα χρησιμοποιηθεί για την αναζήτηση των κρατησεων.

Το μενου επιλογής **“Username”** είναι υπεύθυνο να παίρνει την επιλογή του χρήστη από τη λίστα η οποία γεμίζει με τους διαθέσιμους χρήστες του συστήματος. Ο επιλεγμένος χρήστης θα χρησιμοποιηθεί για την αναζήτηση των κρατήσεων.

Τα μενου επιλογής ημερομηνίας **“Start Date”** και **“End Date”** είναι υπεύθυνα να ορίζουν ένα διάστημα ημερομηνιών στο οποίο μεσα θα πρέπει να βρίσκεται το check out των κρατήσεων που αναζητούμε.

Το κουμπί **“Search”** είναι υπεύθυνο να παίρνει τις επιλογές του χρήστη και μετα τους απαραίτητους ελέγχους να γεμίζει τον πίνακα με τις κρατήσεις που βρέθηκαν.

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τις κρατήσεις που προέκυψαν από την αναζήτηση. Κάθε γραμμή του πίνακα θα κατασκευαστεί για μια γραμμή του πίνακα **“booking”** της βάσης δεδομένων που πληρεί τα κριτήρια αναζήτησης.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Κουμπί Search

Το κουμπί **“Search”** καλεί τη μέθοδο **“fillStatsAdmin”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη να διαβάσει τις επιλογές του χρήστη και να καλέσει την αντίστοιχη μέθοδο ανάλογα τον συνδυασμό των επιλογών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που ανάλογα την μέθοδο περιέχει ένα xml έγγραφο. Όλες οι μεθόδοι διαβάζουν και συμπεριλαμβάνουν στο xml που στέλνουν τη μεταβλητή **adminStatFilterBy** η οποία αντιπροσωπεύει το φιλτράρισμα των αποτελεσμάτων σύμφωνα με το ποιες κρατήσεις έχουν πληρωθεί ή έχει πληρωθεί μόνο η προκαταβολή.

i) fillHotelStatsAdmin()

Δεν παίρνει κάποιο όρισμα. Επικοινωνεί με την μέθοδο **“fillHotelStatsAdmin”** της webservice **hoteldbFill.**
Το xml που στέλνει είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<filter>all</filter>
</root>
```

ii) fillHotelStatsAdminCountry()

Παίρνει σαν όριμα το όνομα της χώρας κάτω από την οποία αναζητούμε τα ξενοδοχεία στα οποία έχουν γίνει οι κρατήσεις. Επικοινωνεί με την μέθοδο **“fillHotelStatsAdminCountry”** της webservice **hoteldbFill.**

Το xml που στέλνει είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<filter>all</filter>
```

<country_name>Greece</country_name>
</root>

iii) fillHotelStatsAdminName()

Παίρνει σαν όρισμα το username του χρήστη του οποίου ψάχνουμε τα ξενοδοχεία στα οποία εχουν γίνει οι κρατησεις. Επικοινωνεί με την μέθοδο **“fillHotelStatsAdminName”** της webservice **hoteldbFill.**

Το xml που στέλνει είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<filter>all</filter>
<user_name>username</user_name>
</root>
```

iv) fillHotelStatsAdminNameCountry()

Παίρνει σαν όρισμα το username του χρήστη και το όνομα της χώρας. Τα ξενοδοχεία των κρατήσεων θα πρέπει να πληρούν όλα τα κριτήρια. Επικοινωνεί με την μέθοδο **“fillHotelStatsAdminCountryName”** της webservice **hoteldbFill.**

Το xml που στέλνει είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<filter>all</filter>
<country_name>Greece</country_name>
<user_name>username</user_name>
</root>
```

v) fillHotelStatsAdminDate()

Παίρνει σαν όρισμα την ημερομηνία που επέλεξε ο χρήστης. Τα ξενοδοχεία των κρατήσεων θα πρέπει να έχουν ημερομηνία εισαγωγής στο σύστημα την επιλεγμένη. Επικοινωνεί με την μέθοδο **“fillHotelStatsAdminDate”** της webservice **hoteldbFill.**

Το xml που στέλνει είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<filter>all</filter>
<dateadd>2010-03-02</dateadd>
</root>
```

vi) fillHotelStatsAdminCountryDate()

Παίρνει σαν όρισμα την ημερομηνία και την χώρα που επέλεξε ο χρήστης. Τα ξενοδοχεία των κρατήσεων θα πρέπει να πληρούν όλα τα κριτήρια. Επικοινωνεί με την μέθοδο **“fillHotelStatsAdminCountryDate”** της webservice **hoteldbFill.**

Το xml που στέλνει είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<filter>all</filter>
<country_name>Greece</country_name>
<dateadd>2010-03-02</dateadd>
</root>
```

vii) fillHotelStatsAdminNameDate()

Παίρνει σαν όρισμα την ημερομηνία και τον χρήστη που επέλεξε ο χρήστης. Τα ξενοδοχεία των κρατησεων θα πρέπει να πληρούν όλα τα κριτήρια. Επικοινωνεί με την μέθοδο **“fillHotelStatsAdminNameDate”** της webservice **hoteldbFill.**

Το xml που στέλνει είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<filter>all</filter>
<user_name>username</user_name>
<dateadd>2010-03-02</dateadd>
</root>
```

viii) fillHotelStatsAdminCountryNameDate()

Παίρνει σαν όρισμα την ημερομηνία, τον χρήστη και τη χώρα που επέλεξε ο χρήστης. Τα ξενοδοχεία των κρατήσεων θα πρέπει να πληρούν όλα τα κριτήρια. Επικοινωνεί με την μέθοδο **“fillHotelStatsAdminCountryNameDate”** της webservice **hoteldbFill.**

Το xml που στέλνει είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<filter>all</filter>
<country_name>Greece</country_name>
<user_name>username</user_name>
<dateadd>2010-03-02</dateadd>
</root>
```

Διαδικασία fillHotelStatsAdmin

Από όλες τις μεθόδους, ανάλογα τις επιλογές του χρήστη, που κάλεστηκαν θα επιστρέψει το παρακάτω xml που περιλαμβάνει τις πληροφορίες για τις κρατήσεις που βρέθηκαν.

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<name>Grand Hotel</name>
```

<user>username</user >
<date_start>username</date_start>
<date_end>username</date_end>
<days>username</days>
<hotelextra>username</hotelextra>
<hotelfood>username</hotelfood>
<roomNum>username</roomNum>
<hotelprofit>username</hotelprofit>
<adminprofit>username</adminprofit>
<completed>username</completed>
<id>username</id>
<country>username</country>
<forward>username</forward>
</root>

id – Ο αύξων αριθμός από την αυτόματη αρίθμιση της βάσης και είναι μοναδικός για κάθε κράτηση.
name – Το όνομα του ξενοδοχείου
user – Ο χρήστης που είναι υπεύθυνος για το ξενοδοχείο
date_start – Η ημερομηνία του check in
date_end – Η ημερομηνία του check out
days – Ο συνολικός αριθμός των ημερών της κράτησης
hotelextra – Το κείμενο από τις πρόσθετες παροχές του ξενοδοχείου που περιλαμβάνονται στη κράτηση
hotelfood – Ο τύπος της διατροφής της κράτησης
adminprofit – Το ποσό του κερδους του διαχειρηστή από την κράτηση
hotelprofit – Το ποσό του κέρδους του πράκτορα από τη κράτηση
roomNum – Ο αριθμός των δωματίων που έχει κλείσει ο πελάτης
completed – Δείχνει ποια και αν έχουν καταβληθεί τα μερίσματα από το κέρδος της κράτησης
country – Η χώρα του ξενοδοχείου της κράτησης.
forward – Το ποσό της προκαταβολής της κράτησης.

Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης

Κουμπι Mark all as Paid
Καλείται η μέθοδος **paidAll** από το bean η οποί παίρνει σαν όρισμα ένα actionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη να διαβάσει όλες τις κρατήσεις που βρίσκονται στον πίνακα, να επιλέξει αυτές που δεν έχουν πληρωθεί και να τις μαρκαρει σαν πληρωμένες. Θα επικοινωνήσει με τη μέθοδο **adminPayAll** της **webservice hoteldbOther** και θα της στείλει ένα xml εγγραφο κωδικοποιημένο σε Base64 format με την παρακάτω μορφή:

<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>4</id>

```
<id>6</id>
</root>
```

Οπου **id** ο αύξων αριθμός της κράτησης από τη βάση δεδομένων. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις επιλογές αναζήτης που έχει κανει είδη ο χρήστης και θα εμφανιστούν οι αλλαγές που έκανε ο χρήστης.

Κουμπι Calculate

Καλείται η μέθοδος **summarise** από το bean η οποία παίρνει σαν όρισμα ένα actionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη να διαβάσει την επιλεγμένη κράτηση από τον πίνακα και να υπολογίσει τα ποσοστά της πληρωμής. Συγκεκριμένα υπολογίζει το ποσό της προκαταβολής, το ποσό της συνολικής πληρωμής (εξόφλησης), το ποσό που απομένει να πληρωθεί, το συνολικό ποσό του κέρδους και το συνολικό ποσό της κράτησης

Κουμπι Change

Μεσα στα αποτελέσματα του πίνακα, για κάθε γραμμή υπάρχει ένα πεδίο **Change Status** το οποίο περιλαμβάνει ένα κουμπί **Change** το οποίο καλεί τη μέθοδο **bookPay** του bean. Η μέθοδος αυτή παίρνει σαν όρισμα ένα ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη να διαβάσει τη μεταβλητή **selectedBookId** η οποία αντιπροσωπεύει το id της κράτησης και να επικοινωνήσει με τη μέθοδο **adminPay** της **webservice hoteldbOther** απόστέλλοντας της την τιμη της μεταβλητης. Η επικοινωνία περιλαμβάνει ένα xml εγγραφο κωδικοποιημένο σε Base64 format της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>4</id>
</root>
```

Οπου id ο αυξων αριθμός της κράτησης. Στη συνέχεια θα καλεστει η μέθοδος **fillStatsAdmin** με τις επιλογές αναζήτησης που είχε κανει ο χρήστης για να ανανεωθουν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Panel: Change Password



Λειτουργικότητα

Εμφανίζει τη διεπαφή της αλλαγής κωδικού πρόσβασης ενός χρήστη ή του administrator.

Αποτελείται από

- ✓ Το πεδίο κειμένου **"Username"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το πεδίο κειμένου **"Old Password"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το πεδίο κειμένου **"New Password"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το πεδίο κειμένου **"Retype Password"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Submit"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το **tooltip** επεξήγησης το οποίο είναι τύπου **"rich:tooltip"**

Επεξήγηση Ρόλων

Στο πεδίο κειμένου **"Username"** ο χρήστης εισάγει το συνθηματικό πρόσβασης.

Στο πεδίο κειμένου **"Old Password"** ο χρήστης εισάγει το παλίο κωδικό πρόσβασης.

Στο πεδίο κειμένου **"New Password"** ο χρήστης εισάγει τον νέο κωδικό πρόσβασης

Στο πεδίο κειμένου **"Retype Password"** ο χρήστης εισάγει ξανά τον κωδικό πρόσβασης προς αποφυγή λάθους.

Το κουμπί **"Submit"** παίρνει τα στοιχεία που έχει εισάγει ο χρήστης και μετα τους απαραίτητους ελέγχους αλλάζει τον κωδικό πρόσβασης του χρήστη

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Κουμπί Submit

Το κουμπί **"Submit"** καλεί τη μέθοδο **"changePasswordAdmin"** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές

1. String με όνομα **"retypepasswordadmin"**, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **"Retype Password"**
2. String με όνομα **"usernameChangeHotel"**, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **"Username"**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **"changePasswordAdmin"** της webservice **hoteldbOther** στέλνοντας ένα μήνυμα SOAP, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<root>
<id>4</id>
<md5>3c130ea5d8d2d3daca7f6808cdf0f148</md5>
<pass_word>password</pass_word>
</root>
```

Όπου **id** ο αύξων αριθμός που αντιστοιχεί στον χρήστη, **md5** το κωδικοποιημένο username και password του χρήστη. Εφόσον η διαδικασία είναι επιτυχής θα αλλάξει ο κωδικός πρόσβασης του χρήστη.

Panel: Change Admin Email



Λειτουργικότητα

Εμφανίζει τη διεπαφή της αλλαγής του email του administrator.

Αποτελείται από

- ✓ Το πεδίο κειμένου **"Current Email"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το πεδίο κειμένου **"New Email"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το πεδίο κειμένου **"Retype New Email"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το πεδίο κειμένου **"Email Password"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Submit"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το **tooltip** επεξήγησης το οποίο είναι τύπου **"rich:tooltip"**

Επεξήγηση Ρόλων

Στο πεδίο κειμένου **"Current Email"** ο χρήστης εισάγει το τωρινό email του.
Στο πεδίο κειμένου **"New Email"** ο χρήστης εισάγει το νέο email.
Στο πεδίο κειμένου **"Retype New Email"** ο χρήστης εισάγει τον ξανά το νέο email.

Στο πεδίο κειμένου **“Email Password”** ο χρήστης εισάγει το κωδικό πρόσβασης στο email του. Αυτό χρειάζεται για την αποστολή email από το σύστημα.

Το κουμπί **“Submit”** παίρνει τα στοιχεία που έχει εισάγει ο χρήστης και μετα τους απαραίτητους ελέγχους αλλάζει τα στοιχεία του email του administrator

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Κουμπί Submit

Το κουμπί **“Submit”** καλεί τη μέθοδο **“addAdminMail”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τη μεταβλητή

1. String με όνομα **“changeemailAdmin”**, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“New Email”**
2. String με όνομα **“changeemailAdminUser”**, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Retype New Email”**
3. String με όνομα **“changeemailAdminPass”**, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Email Password”**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“addAdminMail”** της webservice **hoteldbOther** στέλνοντας ένα μήνυμα SOAP, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>email@email.com</country_name>
<user>username</user>
<pass>password</pass >
</root>
```

Όπου **country_name** το νέο email του admin, **user** το συνθηματικό πρόσβασης του χρήστη στο email του και **pass** ο κωδικός πρόσβασης στο email. Εφόσον η διαδικασία είναι επιτυχής θα αλλάξει ο κωδικός πρόσβασης του χρήστη.

1.2 Hotelier

1.1.1 Set/Edit Hotel’s Data

Panel: Insert Hotel's Distances



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής, διόρθωσης, προβολής και διαγραφής των αποστάσεων ενός ξενοδοχείου από τα σημεία ενδιαφέροντος.

Αποτελείται από

- ✓ Το μενού επιλογής **"Station"** το οποίο είναι ένα **"h:selectOneMenu"**
- ✓ Το πεδίο κειμένου **"Km"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Add/Edit Distance"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το κουμπί **"Delete Distance"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Τον **πίνακα** με τις αποστάσεις ο οποίος είναι **"rich:dataTable"**
- ✓ Τον **scroller** ο οποίος είναι τύπου **"rich:dataScroller"**
- ✓ Το **tooltip** επεξήγησης το οποίο είναι τύπου **"rich:tooltip"**

Επεξήγηση Ρόλων

Το μενού επιλογής **"Station"** είναι υπεύθυνο να εμφανίζει τα POI που έχουν εισαχθεί στο σύστημα.

Το κουμπί **"Add/Edit Distance"** είναι υπεύθυνο να παίρνει το κείμενο που έχει εισάγει ο χρήστης στο πεδίο **"Km"** και μετά από τους απαραίτητους ελέγχους να καταχωρεί την απόσταση, κάτω από το επιλεγμένο POI, στο σύστημα.

Το κουμπί **"Delete Distance"** είναι υπεύθυνο να παίρνει το επιλεγμένο POI και εφόσον έχει εισαχθεί στο σύστημα απόσταση γι'αυτό να την διαγράφει.

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τις αποστάσεις του ξενοδοχείου από τα δίαφορα POI's. Οι τιμές που εμφανίζει είναι ο το όνομα του POI και την απόσταση σε χιλιόμετρα του ξενοδοχείου από αυτό.

Ο **scroller** είναι υπεύθυνος στο να αλλάζει ο χρήστης τις σελίδες με τις αποστάσεις που φαίνονται μέσα στον πίνακα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Διαδικασία fillHotelDistanceFinal

Καλείται η μέθοδος **“fillHotelDistanceFinal”** από το bean. Αυτή η μέθοδος δεν δέχεται ορίσματα. Στέλνει ένα xml της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

Όπου **id** ο αριθμός του ξενοδοχείου του οποίου αναζητούμε τις αποστάσεις.
Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **“fillHotelDistanceFinal”, της web service “hoteldbFill”,** με σκοπό να παραλάβει ένα μήνυμα με τις αποστάσεις του ξενοδοχείου από τα σημεία ενδιαφέροντος που είναι καταχωρημένα στη βάση και συγκεκριμένα στον πίνακα **“distances”.**
Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<station>Airport</station>
<distance>22.3</distance>
</root>
```

Όπου **station** το σημείο ενδιαφέροντος και **distance** η απόσταση σε χιλιόμετρα. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης.

Κουμπί Add Distance

Το κουμπί **“Add Distance”** καλεί τη μέθοδο **“addHotelDistance”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“distanceValue”** τύπου String, η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Km”**
- **“selectedDistance”** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή από το μενού επιλογής **“Station”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“addHotelDistance”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<distance>Greece</distance>
<station>Crete</station>
<id>Crete</id>
</root>
```

Όπου **distance** η απόσταση σε χιλιόμετρα, **station** το σημείο ενδιαφέροντος και **id** ο αυξων αριθμος το ξενοδοχείου.
Μετά από μια επιτυχημένη προσθήκη θα καλεστεί η μέθοδος **“fillHotelDistanceFinal”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Delete Distance

Το κουμπί **“Delete Distance”** καλεί τη μέθοδο **“deleteHotelDistance”** από το bean με σκοπό να διαγράψει μια απόσταση ξενοδοχείου από τη βάση δεδομένων. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“selectedDistance”** τύπου String, η οποία είναι συνδεδεμένη με το μενου επιλογής **“Station”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“deleteHotelDistance”, της web service “hoteldbDelete”,** αποστέλλοντας τη τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<station>Airport</station>
</root>
```

Όπου **id** το id του ξενοδοχείου και **station** το όνομα του σημείου ενδιαφέροντος.
Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **“fillHotelDistanceFinal”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Panel: Insert Hotel’s Facilities



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής, διόρθωσης, προβολής και διαγραφής των παροχών ενός ξενοδοχείου.

Αποτελείται από

- ✓ Το μενού επιλογής **“Hotel Facility”** το οποίο είναι ένα **“h:selectOneMenu”**
- ✓ Το κουμπί **“Add Hotel Facility”** το οποίο είναι ένα **“a4j:commandButton”**
- ✓ Το κουμπί **“Delete Facility”** το οποίο είναι ένα **“a4j:commandButton”**
- ✓ Τον **πίνακα** με τα facilities ο οποίος είναι **“rich:dataTable”**
- ✓ Τον **scroller** ο οποίος είναι τύπου **“rich:dataScroller”**
- ✓ Το **tooltip** επεξήγησης το οποίο είναι τύπου **“rich:tooltip”**

Επεξήγηση Ρόλων

Το μενού επιλογής **“Hotel Facility”** είναι υπεύθυνο να εμφανίζει τα facilities που έχουν εισαχθεί στο σύστημα για ένα ξενοδοχείο.

Το κουμπί **“Add Hotel Facility ”** είναι υπεύθυνο να παίρνει το επιλεγμένο facility και να το εισάγει για το ξενοδοχείο στη βάση.

Το κουμπί **“Delete Facility”** είναι υπεύθυνο να παίρνει το επιλεγμένο facility και να το διαγράφει από το ξενοδοχείο

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τα facilities που εχουν εισαχθεί για το συγκεκριμένο ξενοδοχείο. Οι τιμές που εμφανίζει είναι ο το όνομα του facility.

Ο **scroller** είναι υπεύθυνος στο να αλλάζει ο χρήστης τις σελίδες με τα facilities που φαίνονται μέσα στον πίνακα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Διαδικασία fillSelectedFacilitiesFinal

Καλείται η μέθοδος **“fillSelectedFacilitiesFinal”** από το bean. Αυτή η μέθοδος δεν δέχεται ορίσματα. Στέλνει ένα xml της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

Όπου **id** ο αριθμός του ξενοδοχείου του οποίου αναζητούμε τα facilities.
Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **“fillSelectedFacilitiesFinal”, της web service “hoteldbFill”,** με σκοπό να παραλάβει ένα μήνυμα με τα facilities του ξενοδοχείου που είναι καταχωρημένα στη βάση και συγκεκριμένα στον πίνακα **“hotelfacilities”.**
Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<facility_name>Airport</facility_name>
</root>
```

Όπου **facility_name** το όνομα του facility. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης.

Κουμπί Add Hotel Facility

Το κουμπί **“Add Hotel Facility”** καλεί τη μέθοδο **“addHotelFacility”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“hotelFacSelected”** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή από το μενού επιλογής **“Hotel Facility”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“addHotelFacility”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<facility>Bat</facility>
<id>3</id>
</root>
```

Όπου **facility** το όνομα του facility και **id** ο αυξων αριθμος το ξενοδοχείου.
Μετά από μια επιτυχημένη προσθήκη θα καλεστεί η μέθοδος **“fillSelectedFacilitiesFinal”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Delete Facility

Το κουμπί **“Delete Facility”** καλεί τη μέθοδο **“deleteHotelFacEdit”** από το bean με σκοπό να διαγράψει μια απόσταση ξενοδοχείου από τη βάση δεδομένων. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“hotelFacSelected”** τύπου String, η οποία είναι συνδεδεμένη με το μενου επιλογής **“Hotel Facility”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“deleteHotelFacEdit”, της web service “hoteldbDelete”,** αποστέλλοντας τη τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<facility>Bar</facility>
</root>
```

Όπου **id** το id του ξενοδοχείου και **faciltiy** το όνομα του facility.
Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **“fillSelectedFacilitiesFinal”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Panel: Insert Hotel's Available Rooms



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής, διόρθωσης, προβολής και διαγραφής των δωματίων που υποστηρίζει ένα ξενοδοχείο καθώς και το πλήθος τους.

Αποτελείται από

- ✓ Το μενού επιλογής **"Room Size"** το οποίο είναι ένα **"h:selectOneMenu"**
- ✓ Το κουμπί **"Add/Edit Total Rooms"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το κουμπί **"Add Room Type"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το κουμπί **"Delete Room Type"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το πλαίσιο επιλογής **"New Room Type"** το οποίο είναι ένα **"h:selectBooleanCheckbox"**
- ✓ Τον **πίνακα** με τα δωμάτια ο οποίος είναι **"rich:dataTable"**
- ✓ Τον **scroller** ο οποίος είναι τύπου **"rich:dataScroller"**
- ✓ Τα **tooltip** επεξήγησης τα οποία είναι τύπου **"rich:tooltip"**

Επεξήγηση Ρόλων

Το μενού επιλογής **"Room Size"** είναι υπεύθυνο να εμφανίζει τους τύπους δωματίων που έχουν εισαχθεί στο σύστημα.

Το κουμπί **"Add/Edit Total Rooms"** είναι υπεύθυνο να παίρνει το επιλεγμένο δωμάτιο και τον αριθμο από το πεδίο κειμένου και να το εισάγει στο ξενοδοχείο στη βάση.

Το κουμπί **"Add Room Type"** είναι υπεύθυνο να παίρνει το επιθυμητό όνομα από το πεδίο κειμένου και να το εισάγει ένα καινούργιο τύπο δωματίου στη βάση.

Το κουμπί **"Delete Room type"** είναι υπεύθυνο να παίρνει το επιλεγμένο δωμάτιο και να το διαγράφει από το ξενοδοχείο

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τα δωμάτια που εχουν εισαχθεί για το συγκεκριμένο ξενοδοχείο. Οι τιμές που εμφανίζει είναι ο το όνομα του δωματίου, ο αύξων αριθμός του και το πλήθος που διαθέτει το ξενοδοχείο.

Ο **scroller** είναι υπεύθυνος στο να αλλάζει ο χρήστης τις σελίδες με τα δωμάτια που φαίνονται μέσα στον πίνακα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Διαδικασία fillRoomNumber

Καλείται η μέθοδος **“fillRoomNumber”** από το bean. Αυτή η μέθοδος δεν δέχεται ορίσματα. Στέλνει ένα xml της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

Όπου **id** ο αριθμός του ξενοδοχείου του οποίου αναζητούμε τα δωμάτια.
Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **“fillRoomNumber”, της web service “hoteldbFill”,** με σκοπό να παραλάβει ένα μήνυμα με τα δωμάτια του ξενοδοχείου που είναι καταχωρημένα στη βάση και συγκεκριμένα στον πίνακα **“numberofrooms”.**
Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<number>22</number>
<room>Double</room>
<smoke>Yes</smoke>
<id>23</id>
</root>
```

Όπου **room** το όνομα του δωματίου, **number** το πλήθος που περιέχει το ξενοδοχείο, **smoke** εάν είναι δωμάτιο καπνιστών και τέλος **id** ο αύξων αριθμός του δωματίου. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης.

Κουμπί Add/Edit Total Rooms

Το κουμπί **“Add/Edit Total Rooms”** καλεί τη μέθοδο **“addRoomNum”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“selectedRoomSizeAvailRoomsNum”** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή από το μενού επιλογής **“Room Size”.**
- **“maxRoomNumber”** τύπου String, η οποία είναι συνδεδεμενη με το πεδίο κειμένου στο οποίο ο χρήστης εισάγει το πλήθος των δωματίων.

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“addRoomNum”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>Bat</id>
<roomsize>3</roomsize>
<smoke>3</smoke>
<number>3</number>
</root>
```

Όπου **roomsize** το όνομα του δωματίου, **id** ο αυξων αριθμος του δωματίου, **smoke** η ενδειξη για δωμάτιο καπνιστών και **number** το πλήθος των δωματίων.
Μετά από μια επιτυχημένη προσθήκη θα καλεστεί η μέθοδος **“fillRoomNumber”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Delete Room Type

Το κουμπί **“Delete Room Type”** καλεί τη μέθοδο **“deleteRoomNumberNum”** από το bean με σκοπό να διαγράψει το δωματιο από το ξενοδοχείο. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“selectedRoomSizeAvailRoomsNum”** τύπου String, η οποία είναι συνδεδεμένη με το μενου επιλογής **“Room Size”.**
- **“smoke”** τύπου String, η οποία είναι συνδεδεμένη με το πλαίσιο επιλογής **“New Room Type”.**
- **“hotelEditID”** τύπου String, η οποία αντιπροσωπευει το id του ξενοδοχείου**.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“deleteRoomNumberNum”, της web service “hoteldbDelete”,** αποστέλλοντας τη τιμή των μεταβλητων. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<smoke>Yes</smoke>
<room>Double</room>
</root>
```

Όπου **id** το id του ξενοδοχείου, **smoke** η ένδειξη για το αν πρόκειται για δωμάτιο καπνιστών και **room** το όνομα του δωματίου.
Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **“fillRoomNumber”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Add Room Type

Το κουμπί **“Add Room Type”** καλεί τη μέθοδο **“addRoomSize1”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“roomSizeAdmin”** τύπου String, η οποία είναι συνδεδεμενη με το πεδίο κειμένου στο οποίο ο χρήστης εισάγει το όνομα του δωματίου.

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“addRoomSize”, της web service “hoteldbAdd”,** αποστέλλοντας τη τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<room>Double</room>
</root>
```

Όπου **room** το όνομα του δωματίου.

Panel: Insert Hotel's Room Facilities



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής, διόρθωσης, προβολής και διαγραφής των παροχών δωματίων που υποστηρίζει ένα ξενοδοχείο.

Αποτελείται από

- ✓ Το μενού επιλογής **"Room Size"** το οποίο είναι ένα **"h:selectOneMenu"**
- ✓ Το μενού επιλογής **"Room Facility"** το οποίο είναι ένα **"h:selectOneMenu"**
- ✓ Το κουμπί **"Add Room Facility "** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το κουμπί **"Delete Room Facility"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Τον **πίνακα** με τα δωμάτια ο οποίος είναι **"rich:dataTable"**
- ✓ Τον **scroller** ο οποίος είναι τύπου **"rich:dataScroller"**
- ✓ Τα **tooltip** επεξήγησης τα οποία είναι τύπου **"rich:tooltip"**

Επεξήγηση Ρόλων

Το μενού επιλογής **"Room Size"** είναι υπεύθυνο να εμφανίζει τους τύπους δωματίων που έχουν εισαχθεί στο σύστημα.

Το μενού επιλογής **"Room Facility"** είναι υπεύθυνο να εμφανίζει τους τύπους παροχών που έχουν εισαχθεί στο σύστημα.

Το κουμπί **"Add Room Facility"** είναι υπεύθυνο να παίρνει το επιλεγμένο δωμάτιο και επιλεγμένο τύπο παροχής και να τον εισάγει στο ξενοδοχείο στη βάση.

Το κουμπί **“Delete Room Facility”** είναι υπεύθυνο να παίρνει το επιλεγμένο δωμάτιο και τον επιλεγμένο τύπο παροχής και να το διαγράφει από το ξενοδοχείο

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τις παροχές που έχουν εισαχθεί για το επιλεγμένο τύπο δωματίου. Οι τιμές που εμφανίζει είναι ο το όνομα της παροχής.

Ο **scroller** είναι υπεύθυνος στο να αλλάζει ο χρήστης τις σελίδες με τις παροχές που φαίνονται μέσα στον πίνακα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Διαδικασία fillSelectedRoomFacilities

Καλείται η μέθοδος **“fillSelectedRoomFacilities”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event selectedIndexChanges() του μενου επιλογής. Στέλνει ένα xml της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<RoomFacility>Double</RoomFacility>
</root>
```

Όπου **id** ο αριθμός του ξενοδοχείου του οποίου αναζητούμε τα δωμάτια και **RoomFacility** ο επιλεγμένος τύπος δωματίου.

Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **“fillSelectedRoomFacilities”, της web service “hoteldbFill”,** με σκοπό να παραλάβει ένα μήνυμα με τις παροχές του δωματίου που είναι καταχωρημένες στη βάση και συγκεκριμένα στον πίνακα **“hotelroomfacilities”.**

Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<RoomFacility>Razor electrical socket</RoomFacility>
</root>
```

Όπου **RoomFacility** το όνομα της παροχής. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης.

Κουμπί Add Room Facility

Το κουμπί **“Add Room Facility”** καλεί τη μέθοδο **“addRoomFac”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“roomSizeRoomFacSelected”** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή από το μενού επιλογής **“Room Size”.**
- **“roomFacSelected”** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή από το μενού επιλογής **“Room Facility”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“addRoomFac”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>Bat</id>
<roomsize>Double</roomsize>
<RoomFacility>Mini Bar</RoomFacility>
</root>
```

Όπου **roomsize** το όνομα του δωματίου, **id** ο αυξων αριθμος του ξενοδοχειου και **RoomFacility** ο επιλεγμένος τύπος παροχής.
Μετά από μια επιτυχημένη προσθήκη θα καλεστεί η μέθοδος **“fillSelectedRoomFacility”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Delete Room Facility

Το κουμπί **“Delete Room Facility”** καλεί τη μέθοδο **“deleteRoomFacHotel”** από το bean με σκοπό να διαγράψει τη παροχή από το δωματιο του ξενοδοχείου. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“roomSizeRoomFacSelected”** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή από το μενού επιλογής **“Room Size”.**
- **“roomFacSelected”** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή από το μενού επιλογής **“Room Facility”.**
- **“hotelEditID”** τύπου String, η οποία αντιπροσωπευει το id του ξενοδοχείου**.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“deleteRoomFacHotel”, της web service “hoteldbDelete”,** αποστέλλοντας τη τιμή των μεταβλητων. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>Bat</id>
<roomsize>Double</roomsize>
<RoomFacility>Mini Bar</RoomFacility>
</root>
```

Όπου **roomsize** το όνομα του δωματίου, **id** ο αυξων αριθμος του ξενοδοχειου και **RoomFacility** ο επιλεγμένος τύπος παροχής.
Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **“fillSelectedRoomFacility”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Panel: Insert Hotel’s Stop Sales Dates



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής, διόρθωσης, προβολής και διαγραφής των stop sales ένος ξενοδοχείου.

Αποτελείται από

- ✓ Το ημερολόγιο **"Date Start"** το οποίο είναι ένα **"rich:calendar"**
- ✓ Το ημερολόγιο **"Date End"** το οποίο είναι ένα **"rich:calendar"**
- ✓ Το κουμπί **"Add Stop Sales "** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το κουμπί **"Delete Stop Sales"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Τον **πίνακα** με τα stop sales ο οποίος είναι **"rich:dataTable"**
- ✓ Τον **scroller** ο οποίος είναι τύπου **"rich:dataScroller"**
- ✓ Τα **tooltip** επεξήγησης τα οποία είναι τύπου **"rich:tooltip"**

Επεξήγηση Ρόλων

Το ημερολογιο **"Date Start"** εμφανίζει και δίνει την επιλογή στο χρήστη να επιλέξει την ημερομηνία έναρξης του stop sale.

Το ημερολογιο **"Date End"** εμφανίζει και δίνει την επιλογή στο χρήστη να επιλέξει την ημερομηνία λήξης του stop sale.

Το κουμπί **"Add Stop Sale"** είναι υπεύθυνο να παίρνει τις επιλεγμένες ημερομηνίες και να εισάγει το stop sale για το ξενοδοχείο.

Το κουμπί **"Delete Stop Sale"** είναι υπεύθυνο να παίρνει το μοναδικό Id ενός stop sale και να το διαγράφει από το ξενοδοχείο

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τα stop sales που έχουν εισαχθεί για το ξενοδοχειο. Οι τιμές που εμφανίζει είναι ο το id του stop sale, η ημερομηνία έναρξης και η ημερομηνία λήξης.

Ο **scroller** είναι υπεύθυνος στο να αλλάζει ο χρήστης τις σελίδες με τα stop sales που φαίνονται μέσα στον πίνακα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Διαδικασία fillStopSales

Καλείται η μέθοδος **"fillStopSales"** από το bean. Αυτή η μέθοδος δεν δέχεται κάποιο όρισμα. Στέλνει ένα xml της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

Όπου **id** ο αριθμός του ξενοδοχείου του οποίου αναζητούμε τα stop sales.
Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **“fillStopSalesHotel”, της web service “hoteldbFill”,** με σκοπό να παραλάβει ένα μήνυμα με τα stop sales που είναι καταχωρημένες στη βάση και συγκεκριμένα στον πίνακα **“stopsales”.**
Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<date_start>2010-02-19</date_start>
<date_end>2010-02-20</date_end>
</root>
```

Όπου **date_start** η ημερομηνία έναρξης και **date_end** η ημερομηνία λήξης του stop sale. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης.

Κουμπί Add Stop Sales

Το κουμπί **“Add Stop Sales”** καλεί τη μέθοδο **“addStopSales”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“stopSalesDateSrt”,** η οποία είναι συνδεδεμένη με την επιλεγμένη ημερομηνια από το ημερολόγιο **“Date Start”.**
- **“stopSalesDateEnd”,** η οποία είναι συνδεδεμένη με την επιλεγμένη ημερομηνια από το ημερολόγιο **“Date End”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“addStopSales”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<date_start>2010-02-19</date_start>
<date_end>2010-02-20</date_end>
</root>
```

Όπου **id** ο αυξων αριθμος του ξενοδοχειου, **date_start** η ημερομηνία έναρξης και **date_end** η ημερομηνία λήξης του stop sale..
Μετά από μια επιτυχημένη προσθήκη θα καλεστεί η μέθοδος **“fillStopSales”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Delete Stop Sales

Το κουμπί **“Delete Stop Sales”** καλεί τη μέθοδο **“deleteStopSales”** από το bean με σκοπό να διαγράψει τη παροχή από το δωματιο του ξενοδοχείου. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“stopSalesSelectedId”** τύπου String**,** η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“ID to delete”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“deleteStopSales”, της web service “hoteldbDelete”,** αποστέλλοντας τη τιμή της μεταβλητης. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

Όπου **id** ο αυξων αριθμος του stop sale.
Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **“fillStopSales”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Panel: Insert Hotel’s Release Policy



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής, διόρθωσης, προβολής και διαγραφής της πολιτικής κρατήσεων τελευταίας στιγμής ένος ξενοδοχείου.

Αποτελείται από

- ✓ Το ημερολόγιο **“Date Start”** το οποίο είναι ένα **“rich:calendar”**
- ✓ Το ημερολόγιο **“Date End”** το οποίο είναι ένα **“rich:calendar”**
- ✓ Το κουμπί **“Add”** το οποίο είναι ένα **“a4j:commandButton”**
- ✓ Τον **πίνακα** με τις πολιτικές ο οποίος είναι **“rich:dataTable”**
- ✓ Τον **scroller** ο οποίος είναι τύπου **“rich:dataScroller”**
- ✓ Τα **tooltip** επεξήγησης τα οποία είναι τύπου **“rich:tooltip”**

Επεξήγηση Ρόλων

Το ημερολογιο **“Date Start”** εμφανίζει και δίνει την επιλογή στο χρήστη να επιλέξει την ημερομηνία έναρξης.

Το ημερολογιο **“Date End”** εμφανίζει και δίνει την επιλογή στο χρήστη να επιλέξει την ημερομηνία λήξης.

Το κουμπί **“Add”** είναι υπεύθυνο να παίρνει τις επιλεγμένες ημερομηνίες και τον ελάχιστο αριθμό ημερών και να εισάγει τη πολιτική στο ξενοδοχείο.

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τις πολιτικές που έχουν εισαχθεί στο ξενοδοχειο. Οι τιμές που εμφανίζει είναι η ημερομηνία έναρξης η ημερομηνία λήξης και το πλήθος των ελάχιστων ημερών.

Ο **scroller** είναι υπεύθυνος στο να αλλάζει ο χρήστης τις σελίδες με τις πολιτικές που φαίνονται μέσα στον πίνακα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Διαδικασία fillRelease

Καλείται η μέθοδος **“fillRelease”** από το bean. Αυτή η μέθοδος δεν δέχεται κάποιο όρισμα. Στέλνει ένα xml της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

Όπου **id** ο αριθμός του ξενοδοχείου του οποίου αναζητουμε τις πολιτικές.
Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **“fillRelease”, της web service “hoteldbFill”,** με σκοπό να παραλάβει ένα μήνυμα με τις πολιτικές που είναι καταχωρημένες στη βάση και συγκεκριμένα στον πίνακα **“releaseh”.**
Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<start>2010-02-19</start>
<end>2010-02-20</end>
<days>5</days>
</root>
```

Όπου **date_start** η ημερομηνία έναρξης και **date_end** η ημερομηνία λήξης και **days** οι ελάχιστες μέρες της πολιτικής. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης.

Κουμπί Add

Το κουμπί **“Add”** καλεί τη μέθοδο **“addHotelRelease”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“releaseDateStart”,** η οποία είναι συνδεδεμένη με την επιλεγμένη ημερομηνια από το ημερολόγιο **“Date Start”.**
- **“releaseDateEnd”,** η οποία είναι συνδεδεμένη με την επιλεγμένη ημερομηνια από το ημερολόγιο **“Date End”.**
- **“release”,** η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Days”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“addRelease”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<days>5</days>
<releaseST>2010-02-19</releaseST>
<releaseNT>2010-02-20</releaseND>
</root>
```

Όπου **id** ο αυξων αριθμος του ξενοδοχειου, **releaseST** η ημερομηνία έναρξης, **releaseND** η ημερομηνία λήξης και **release** το πλήθος των ελάχιστων ημερών της πολιτικής.
Μετά από μια επιτυχημένη προσθήκη θα καλεστεί η μέθοδος **“fillRelease”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Delete Hotel Release

Το κουμπί **“Delete Hotel Release”** καλεί τη μέθοδο **“deleteHotelRelease”** από το bean με σκοπό να διαγράψει τη πολιτικη από το ξενοδοχείο. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού (x). Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“fakeBy”** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη πολιτικη

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“deleteHotelRelease”, της web service “hoteldbDelete”,** αποστέλλοντας τη τιμή της μεταβλητης. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>4</id>
</root>
```

Όπου **id** ο αυξων αριθμος της πολιτικης.
Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **“fillRelease”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Panel: Insert Hotel's Early Booking



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής, διόρθωσης, προβολής και διαγραφής της πολιτικής πρόωρων κρατήσεων ένος ξενοδοχείου.

Αποτελείται από

- ✓ Το ημερολόγιο **"Date Start"** το οποίο είναι ένα **"rich:calendar"**
- ✓ Το ημερολόγιο **"Date End"** το οποίο είναι ένα **"rich:calendar"**
- ✓ Το ημερολόγιο **"Offer Start"** το οποίο είναι ένα **"rich:calendar"**
- ✓ Το ημερολόγιο **"Offer End"** το οποίο είναι ένα **"rich:calendar"**
- ✓ Το πλαίσιο επιλογής **"Percentage"** το οποίο είναι ένα **"h:selectBooleanCheckbox"**
- ✓ Το κουμπί **"Add/Edit Special Offer"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Τον **πίνακα** με τις πολιτικές ο οποίος είναι **"rich:dataTable"**
- ✓ Τον **scroller** ο οποίος είναι τύπου **"rich:dataScroller"**
- ✓ Τα **tooltip** επεξήγησης τα οποία είναι τύπου **"rich:tooltip"**

Επεξήγηση Ρόλων

Το ημερολογιο **"Date Start"** εμφανίζει και δίνει την επιλογή στο χρήστη να επιλέξει την ημερομηνία έναρξης της πολιτικης.

Το ημερολογιο **“Date End”** εμφανίζει και δίνει την επιλογή στο χρήστη να επιλέξει την ημερομηνία λήξης της πολιτικής.

Το ημερολογιο **“Offer Start”** εμφανίζει και δίνει την επιλογή στο χρήστη να επιλέξει την ημερομηνία έναρξης ισχύος της πολιτικής.

Το ημερολογιο **“Offer End”** εμφανίζει και δίνει την επιλογή στο χρήστη να επιλέξει την ημερομηνία λήξης ισχύος της πολιτικής.

Το κουμπί **“Add/Edit Special Offer”** είναι υπεύθυνο να παίρνει τις επιλεγμένες ημερομηνίες και την τιμή και να εισάγει τη πολιτική στο ξενοδοχείο.

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τις πολιτικές που έχουν εισαχθεί στο ξενοδοχειο. Οι τιμές που εμφανίζει είναι η ημερομηνία έναρξης η ημερομηνία λήξης, η ημερομηνία έναρξης ισχύος της πολιτικής, η ημερομηνία λήξης ισχύος της πολιτικής και η τιμή της προσφοράς.

Ο **scroller** είναι υπεύθυνος στο να αλλάζει ο χρήστης τις σελίδες με τις πολιτικές που φαίνονται μέσα στον πίνακα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Διαδικασία fillHotelEarlyBooking

Καλείται η μέθοδος **“fillHotelEarlyBooking”** από το bean. Αυτή η μέθοδος δεν δέχεται κάποιο όρισμα. Στέλνει ένα xml της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

Όπου **id** ο αριθμός του ξενοδοχείου του οποίου αναζητουμε τις πολιτικές.
Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **“fillHotelEarlyBooking”, της web service “hoteldbFill”,** με σκοπό να παραλάβει ένα μήνυμα με τις πολιτικές που είναι καταχωρημένες στη βάση και συγκεκριμένα στον πίνακα **“earlybooking”.**
Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<date_start>2010-02-19</date_start>
<date_end>2010-02-20</date_end>
<offerstart>2010-02-19</offerstart>
<offerend>2010-02-20</offerend>
<price>5</price>
<ebid>5</ebid>
</root>
```

Όπου **date_start** η ημερομηνία έναρξης και **date_end** η ημερομηνία λήξης, **offerstart** η ημερομηνία έναρξης ισχύος, **offerend** η ημερομηνία λήξης της ισχύος, **price** η τιμή της προσφοράς και **ebid** ο αύξων αριθμός της πολιτικής. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης.

Κουμπί Add/Edit Special Offer

Το κουμπί **“Add/Edit Special Offer”** καλεί τη μέθοδο **“addHotelEarlyBooking”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“hotelEarlyBookingDateStart”,** η οποία είναι συνδεδεμένη με την επιλεγμένη ημερομηνια από το ημερολόγιο **“Date Start”.**
- **“hotelEarlyBookingDateEnd”,** η οποία είναι συνδεδεμένη με την επιλεγμένη ημερομηνια από το ημερολόγιο **“Date End”.**
- **“hotelEarlyBookingOfferStart”,** η οποία είναι συνδεδεμένη με την επιλεγμένη ημερομηνια από το ημερολόγιο **“Offer Start”.**
- **“hotelEarlyBookingOfferEnd”,** η οποία είναι συνδεδεμένη με την επιλεγμένη ημερομηνια από το ημερολόγιο **“Offer End”.**
- **“percentageEarlyBooking”,** η οποία είναι συνδεδεμένη με το πλαίσιο επιλογής **“Percentage”.**
- **“hotelEarlyBookingPrice”,** η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Price”.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“addHotelEarlyBooking”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<date_start>2010-02-19</date_start>
<date_end>2010-02-20</date_end>
<offerstart>2010-02-19</offerstart>
<offerend>2010-02-20</offerend>
<price>54,45</price >
<id>2</id>
</root>
```

Όπου **id** ο αυξων αριθμος του ξενοδοχειου, **date_start** η ημερομηνία έναρξης, **date_end** η ημερομηνία λήξης, **offerstart** η ημερομηνία έναρξης ισχύος, **offerend** η ημερομηνία λήξης ισχύος και **price** η τιμή της προσφοράς.
Μετά από μια επιτυχημένη προσθήκη θα καλεστεί η μέθοδος **“fillHotelEarlyBooking”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Delete Hotel Early Booking

Το κουμπί **“Delete Hotel Early Booking (x)”** καλεί τη μέθοδο **“deleteHotelEarlyBooking”** από το bean με σκοπό να διαγράψει τη πολιτικη από το ξενοδοχείο. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού (x). Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“earlyBookingId”** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη πολιτικη

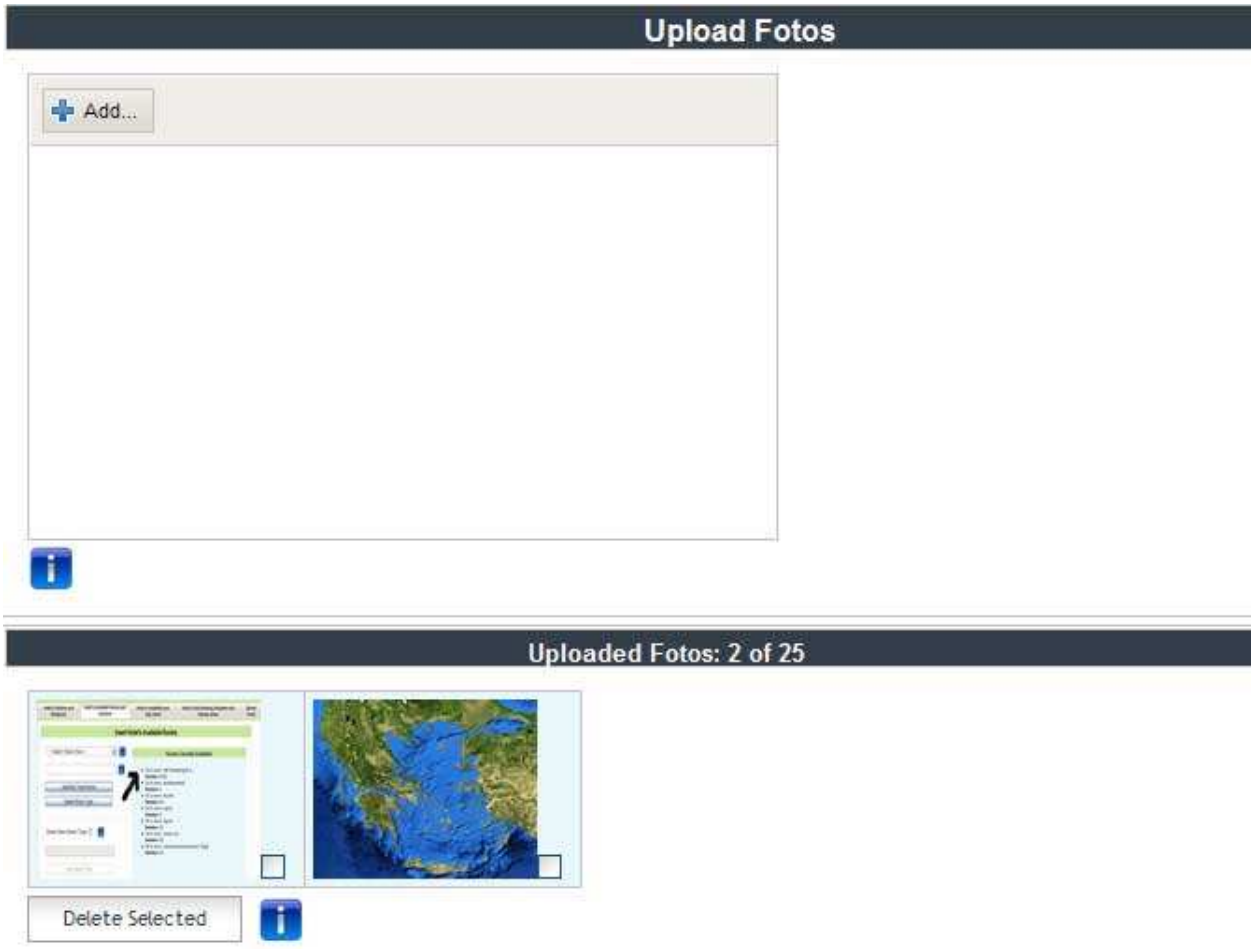
Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“deleteHotelEarlyBooking”, της web service “hoteldbDelete”,** αποστέλλοντας τη τιμή της μεταβλητης. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>4</id>
```

</root>

Όπου **id** ο αυξων αριθμος της πολιτικης.
Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **“fillHotelEarlyBooking”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Panel: Upload Hotel Fotos



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής, διόρθωσης, προβολής και διαγραφής των φωτογραφιών ένος ξενοδοχείου.

Αποτελείται από

- ✓ Τον uploader ο οποίος είναι ένας **“rich:fileUpload”**
- ✓ Το κουμπί **“Delete Selected”** το οποίο είναι ένα **“a4j:commandButton”**

- ✓ Τον **πίνακα** με τις φωτογραφίες ο οποίος είναι **“rich:dataGrid”**
- ✓ Τα **tooltip** επεξήγησης τα οποία είναι τύπου **“rich:tooltip”**

Επεξήγηση Ρόλων

Ο uploader είναι υπεύθυνος να παίρνει μια φωτογραφία και μετα τις απαραίτητες διαδικασίες να την αποθηκεύει στον server και στη βάση για το συγκεκριμένο ξενοδοχείο.

Το κουμπί **“Delete Selected”** είναι υπεύθυνο να παίρνει τις επιλεγμένες φωτογραφίες και να τις διαγράφει από τον server και από το ξενοδοχείο.

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τις φωτογραφίες που έχουν εισαχθεί για το συγκεκριμένο ξενοδοχείο.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Διαδικασία fillHotelFoto

Καλείται η μέθοδος **“fillHotelFoto”** από το bean. Αυτή η μέθοδος δέχεται έναν long σαν όρισμα ο οποίος αντιπροσωπεύει το id του ξενοδοχείου. Στέλνει ένα xml της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

Όπου **id** ο αριθμός του ξενοδοχείου του οποίου αναζητουμε τις φωτογραφίες.
Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **“fillHotelFoto”, της web service “hoteldbFill”,** με σκοπό να παραλάβει ένα μήνυμα με τις πολιτικές που είναι καταχωρημένες στη βάση και συγκεκριμένα στον πίνακα **“hotelfoto”.**
Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<foto>http://www.borker-lodge.com/hotelsData/HotelFotos/2/1.jpg</foto>
</root>
```

Όπου **foto** η διεύθυνση της φωτογραφίας. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης.

Διαδικασία Upload Fotos

Ο uploader καλεί τη μέθοδο **“listenerHotel”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο UploadEvent που στη περίπτωση μας είναι το event upload() του uploader. Είναι υπεύθυνη να διαβάσει το αρχείο που γίνεται upload και να ελέγξει στη συνέχεια αν υπάρχουν οι φακέλοι στο path Που πρεπει να κατασωρηθεί.

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“copyFile”,** αποστέλλοντας τη τιμή του προσωρινού αρχείου που γίνεται upload και το Path του αρχείου που πρέπει να δημιουργηθεί. Η μέθοδος αυτή παίρνει το προσωρινό αρχείο και μεταφράζοντας το σε πίνακα των 1024bytes καταφλερνει τη μεταφορά του.

Μετά από μια επιτυχημένη προσθήκη θα καλεστεί η μέθοδος **“fillHotelFoto”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Delete Selected

Το κουμπί **“Delete Selected”** καλεί τη μέθοδο **“deleteFotos”** από το bean με σκοπό να διαγράψει ένα πλήθος από επιλεγμενες φωτογραφίες του ξενοδοχείου. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη να διαβάσει τη λίστα από τις επιλεγμένες φωτογραφίες.
Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“deleteHotelFoto”, της web service “hoteldbDelete”,** αποστέλλοντας τη τιμή των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>4</id>
</root>
```

Όπου **id** ο αυξων αριθμος της πολιτικης.
Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **“fillHotelFoto”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Panel: Insert Hotel's Available Dates



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής, διόρθωσης, προβολής και διαγραφής της διαθεσιμότητας και τιμολόγησης των δωματίων ένος ξενοδοχείου.

Αποτελείται από

- ✓ Το μενου επιλογής **"Room Type"** το οποίο είναι ένα **"h:selectOneMenu"**
- ✓ Το μενου επιλογής **"Food"** το οποίο είναι ένα **"h:selectOneMenu"**
- ✓ Το μενου επιλογής **"Persons"** το οποίο είναι ένα **"h:selectOneMenu"**
- ✓ Το ημερολόγιο **"Date Start"** το οποίο είναι ένα **"rich:calendar"**
- ✓ Το ημερολόγιο **"Date End"** το οποίο είναι ένα **"rich:calendar"**
- ✓ Το κουμπί **"Add Availability"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το κουμπί **"Delete Availability"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Τον **πίνακα** με τις πολιτικές ο οποίος είναι **"rich:dataList"**
- ✓ Τον **scroller** ο οποίος είναι τύπου **"rich:dataScroller"**
- ✓ Τα **tooltip** επεξήγησης τα οποία είναι τύπου **"rich:tooltip"**

Επεξήγηση Ρόλων

Το μενού επιλογής **"Room Type"** εμφανίζει τη λίστα με τους διαθέσιμους τύπους δωματίων του ξενοδοχείου.

Το μενού επιλογής **"Food"** εμφανίζει τη λίστα με τους διαθέσιμου τύπους διατροφής για ένα δωμάτιο.

Το ημερολογιο **"Date Start"** εμφανίζει και δίνει την επιλογή στο χρήστη να επιλέξει την ημερομηνία έναρξης της διαθεσιμότητας.

Το ημερολογιο **“Date End”** εμφανίζει και δίνει την επιλογή στο χρήστη να επιλέξει την ημερομηνία λήξης της διαθεσιμότητας.

Το μενού επιλογής **“Persons”** εμφανίζει μια λίστα με τον διαθέσιμο αριθμό ανθρώπων στον οποίο θα απευθύνεται η διαθεσιμότητα.

Το κουμπί **“Add Availability”** είναι υπεύθυνο να παίρνει τις επιλογές του χρήστη και να εισάγει τη διαθεσιμότητα στο ξενοδοχείο.

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τις διαθεσιμότητες που έχουν εισαχθεί στο ξενοδοχειο.

Ο **scroller** είναι υπεύθυνος στο να αλλάζει ο χρήστης τις σελίδες με τις διαθεσιμότητες που φαίνονται μέσα στον πίνακα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Διαδικασία fillAvailable

Καλείται η μέθοδος **“fillAvailable”** από το bean. Αυτή η μέθοδος δεν δέχεται κάποιο όρισμα. Στέλνει ένα xml της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

Όπου **id** ο αριθμός του ξενοδοχείου του οποίου αναζητουμε τις διαθεσιμότητες.
Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **“fillAvailable”, της web service “hoteldbFill”,** με σκοπό να παραλάβει ένα μήνυμα με τις διαθεσιμότητες που είναι καταχωρημένες στη βάση και συγκεκριμένα στον πίνακα **“available”.**
Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<date_start>2010-02-19</date_start>
<date_end>2010-02-20</date_end>
<price>5</price>
<room>Double</room>
<food>BB</food>
<id>15</id>
<person>2</person>
<fake>1</fake>
</root>
```

Όπου **date_start** η ημερομηνία έναρξης και **date_end** η ημερομηνία λήξης, **room** ο τύπος του δωματιου, **food** ο τύπος της διατροφής, **price** η τιμή της διαθεσιμότητας, **Person** ο αριθμος των ατόμων. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης.

Κουμπί Add Available

Το κουμπί **“Add Available”** καλεί τη μέθοδο **“addAvailable”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“availableDateSrt”,** η οποία είναι συνδεδεμένη με την επιλεγμένη ημερομηνια από το ημερολόγιο **“Date Start”.**
- **“availableDateEnd”,** η οποία είναι συνδεδεμένη με την επιλεγμένη ημερομηνια από το ημερολόγιο **“Date End”.**
- **“availablePersons”,** η οποία είναι συνδεδεμένη με το μενου επιλογής **“Persons”.**
- **“availablePrice”,** η οποία είναι συνδεδεμένη με το πεδίο κειμένου **“Price”.**
- **“roomSizeAvailableSelected”,** η οποία είναι συνδεδεμένη με το μενου επιλογής **“Room Size”.**
- **“roomSizeAvailableSelected”,** η οποία είναι συνδεδεμένη με το μενου επιλογής **“Room Size”.**
- **“foodAvailableSelected”,** η οποία είναι συνδεδεμένη με το μενού επιλογής **“Food”.**
- **“hotelEditID”,** η οποία αντιπροσωπεύει το id του ξενοδοχείου**.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“addAvailable”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<date_start>2010-02-19</date_start>
<date_end>2010-02-20</date_end>
<food>BB</food>
<person>2</person>
<price>54,45</price >
<id>2</id>
<roomnumber>4</roomnumber>
<roomsize>Double</roomsize>
</root>
```

Όπου **id** ο αυξων αριθμος του ξενοδοχειου, **date_start** η ημερομηνία έναρξης, **date_end** η ημερομηνία λήξης, **food** ο τύπος διατροφής, **person** το πλήθος των ατόμων, **price** η τιμή της διαθεσιμότητας, **roomnumber** ο αριθμός των δωματίων που θα γίνουν διαθέσιμα.
Μετά από μια επιτυχημένη προσθήκη θα καλεστεί η μέθοδος **“fillAvailable”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Delete Availability

Το κουμπί **“Delete Availability”** καλεί τη μέθοδο **“deleteAvailable”** από το bean με σκοπό να διαγράψει τη πολιτικη από το ξενοδοχείο. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού (x). Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“availaleSelectedID”** τύπου String**,** η οποία είναι συνδεδεμένη με το πεδίο κειμένου και αντπροσωπεύει τον αύξων αριθμό της διαθεσιμότητας.

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“deleteAvailable”, της web service “hoteldbDelete”,** αποστέλλοντας τη τιμή της μεταβλητης. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>4</id>
</root>
```

Όπου **id** ο αυξων αριθμος της πολιτικης.
Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **“fillAvailable”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Panel: Insert Hotel’s Available Food



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής, διόρθωσης, προβολής και διαγραφής των διαθέσιμων τύπων διατροφής για ένα δεδομένο τύπο δωματίου και συγκεκριμένες ημερομηνίες που υποστηρίζει ένα ξενοδοχείο.

Αποτελείται από

- ✓ Το μενού επιλογής **“Food Type”** το οποίο είναι ένα **“h:selectOneMenu”**
- ✓ Το κουμπί **“Add/Edit Hotel Food”** το οποίο είναι ένα **“a4j:commandButton”**
- ✓ Το κουμπί **“Delete Hotel Food”** το οποίο είναι ένα **“a4j:commandButton”**
- ✓ Τον **πίνακα** με τα δωμάτια ο οποίος είναι **“rich:dataList”**
- ✓ Τα **tooltip** επεξήγησης τα οποία είναι τύπου **“rich:tooltip”**

Επεξήγηση Ρόλων

Το μενού επιλογής **“Food Type”** είναι υπεύθυνο να εμφανίζει τους τύπους διατροφής που έχουν εισαχθεί στο σύστημα.

Το κουμπί **“Add/Edit Hotel Food”** είναι υπεύθυνο να παίρνει τον επιλεγμένο τύπο διατροφής και την τιή και να τον εισάγει στο ξενοδοχείο στη βάση.

Το κουμπί **“Delete Hotel Food”** είναι υπεύθυνο να παίρνει τον επιλεγμένο τύπο διατροφής και να το διαγράφει από το ξενοδοχείο

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τις διατροφές που έχουν εισαχθεί στο ξενοδοχείο. Οι τιμές που εμφανίζει είναι ο το όνομα της διατροφής και η τιμή της.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Διαδικασία fillSelectedFood

Καλείται η μέθοδος **“fillSelectedFood”** από το bean. Αυτή η μέθοδος δεν παίρνει κάποιο όρισμα. Στέλνει ένα xml της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<roomsize>Double</roomsize>
<start>2010-02-19</start>
<end>2010-02-20 </end>
</root>
```

Όπου **id** ο αριθμός του ξενοδοχείου, **roomsize** ο τύπος δωματίου, **start** η ημερομηνία έναρξης της διαθεσιμότητας και **end** η ημερομηνια λήξης της διαθεσιμότητας.
Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **“fillSelectedFoodHotel”, της web service “hoteldbFill”,** με σκοπό να παραλάβει ένα μήνυμα με τις διαθέσιμες διατροφές του δωματίου για μια συγκεκριμένη περίοδο που είναι καταχωρημένες στη βάση και συγκεκριμένα στον πίνακα **“hotelfood”.**
Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<food>HalfBoard</food>
<price>15</price>
</root>
```

Όπου **food** το όνομα της διατροφής και **price** η τιμή της. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης.

Κουμπί Add/Edit Hotel Food

Το κουμπί **“Add/Edit Hotel Food”** καλεί τη μέθοδο **“addHotelFood”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“roomSizeFoodSelected”** τύπου String**,** η οποία είναι συνδεδεμένη με τον επιλεγμένο τύπο δωματίου**.**

- **“foodSelected”** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή από το μενού επιλογής **“Food Type”.**
- **“foodDateStart”** ημερομηνία συνδεδεμένη με την έναρξη της συγκεκριμένης διαθεσιμότητας**.**
- **“foodDateEnd”** ημερομηνία συνδεδεμένη με την λήξη της συγκεκριμένης διαθεσιμότητας**.**
- **“hotelEditID”** το Id του επιλεγμένου ξενοδοχείου**.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“addHotelFood”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<roomsize>Double</roomsize>
<start>2010-02-19</start>
<end>2010-02-20 </end>
<food>HalfBoard</food>
<price>15</price>
</root>
```

Όπου **roomsize** το όνομα του δωματίου, **id** ο αυξων αριθμος του ξενοδοχειου, **start** η ημερομηνία έναρξης της διαθεσιμότητας και **end** η ημερομηνια λήξης της διαθεσιμότητας, **food** το όνομα της διατροφής και **price** η τιμή της.
Μετά από μια επιτυχημένη προσθήκη θα καλεστεί η μέθοδος **“fillSelectedFood”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Delete Hotel Food

Το κουμπί **“Delete Hotel Food”** καλεί τη μέθοδο **“deleteHotelFood”** από το bean με σκοπό να διαγράψει τη παροχή από το δωματιο του ξενοδοχείου. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **“roomSizeFoodSelected”** τύπου String**,** η οποία είναι συνδεδεμένη με τον επιλεγμένο τύπο δωματίου**.**
- **“foodSelected”** τύπου String**,** η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή από το μενού επιλογής **“Food Type”.**
- **“foodDateStart”** ημερομηνία συνδεδεμένη με την έναρξη της συγκεκριμένης διαθεσιμότητας**.**
- **“foodDateEnd”** ημερομηνία συνδεδεμένη με την λήξη της συγκεκριμένης διαθεσιμότητας**.**
- **“hotelFoodPrice”** η τιμή της διατροφής.
- **“hotelEditID”** το Id του επιλεγμένου ξενοδοχείου

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“deleteHotelFood”, της web service “hoteldbDelete”,** αποστέλλοντας τη τιμή των μεταβλητων. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<roomsize>Double</roomsize>
<start>2010-02-19</start>
<end>2010-02-20 </end>
<food>HalfBoard</food>
<price>15</price>
</root>
```

Όπου **roomsize** το όνομα του δωματίου, **id** ο αυξων αριθμος του ξενοδοχειου, **start** η ημερομηνία έναρξης της διαθεσιμότητας και **end** η ημερομηνια λήξης της διαθεσιμότητας, **food** το όνομα της διατροφής και **price** η τιμή της.
Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **"fillSelectedFood",** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Panel: Insert Hotel's Restrictions



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής, διόρθωσης, προβολής και διαγραφής των περιορισμών των διαθεσιμοτητων ένος ξενοδοχείου.

Αποτελείται από

- ✓ Το πλαίσιο κειμένου **"Price"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το πλαίσιο κειμένου **"Days Threshold"** το οποίο είναι ένα **"h:inputText"**

- ✓ Το πλαίσιο επιλογής **“Percentage”** το οποίο είναι ένα **“h:selectBooleanCheckbox”**
- ✓ Το κουμπί **“Add/Edit Hotel Restriction”** το οποίο είναι ένα **“a4j:commandButton”**
- ✓ Τον **πίνακα** με τις πολιτικές ο οποίος είναι **“rich:dataList”**
- ✓ Τα **tooltip** επεξήγησης τα οποία είναι τύπου **“rich:tooltip”**

Επεξήγηση Ρόλων

Το πλαίσιο κειμένου **“Price”** δέχεται την τιμή του περιορισμού.

Το πλαίσιο κειμένου **“Days Threshold”** δέχεται τις ελάχιστες μέρες πριν ενεργοποιιηθεί ο περιορισμός.

Το πλαίσιο επιλογής **“Percentage”** υποδεικνύει αν η τιμή είναι ποσοστό η καθαρός αριθμός

Το κουμπί **“Add/Edit Hotel Restriction”** είναι υπεύθυνο να παίρνει τις επιλογές του χρήστη και να εισάγει τον περιορισμό στο ξενοδοχείο.

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τους περιορισμούς που έχουν εισαχθεί στο ξενοδοχειο.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Διαδικασία fillHotelRestriction

Καλείται η μέθοδος **“fillHotelRestriction”** από το bean. Αυτή η μέθοδος δεν δέχεται κάποιο όρισμα. Στέλνει ένα xml της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

Όπου **id** ο αριθμός της διαθεσιμότητας της οποίας αναζητουμε τους περιορισμούς.
Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **“fillHotelRestriction”, της web service “hoteldbFill”,** με σκοπό να παραλάβει ένα μήνυμα με τις πολιτικές που είναι καταχωρημένες στη βάση και συγκεκριμένα στον πίνακα **“restriction”.**
Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<date_start>2010-02-19</date_start>
<date_end>2010-02-20</date_end>
<roomname>Double</roomname>
<price>20</price>
<days>5</days>
<id>2</id>
</root>
```

Όπου **date_start** η ημερομηνία έναρξης και **date_end** η ημερομηνία λήξης, **roomname** το όνομα τοτ δωματίου, **days** οι ελάχιστες μέρες πριν ενεργοποιηθει ο περιορισμός, **price** η τιμή του περιορισμού και **id** ο αύξων αριθμός του περιορισμού. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις τιμές που λάβαμε από το μήνυμα απάντησης.

Κουμπί Add/Edit Hotel Restriction

Το κουμπί **"Add/Edit Hotel Restriction"** καλεί τη μέθοδο **"addHotelRestriction"** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **"foodDateStart",** η ημερομηνια έναρξης ισχύος του περιορισμού
- **"foodDateEnd",** η ημερομηνία λήξης ισχύος του περιορισμού**.**
- **"avvID",** το id της διαθεσιμότητας**.**
- **"hotelRestrictionPrice",** η οποία είναι συνδεδεμένη με το πεδίο κειμένου **"Price".**
- **"hotelRestrictionDays",** η οποία είναι συνδεδεμένη με το πεδίο κειμένου **"Days Threshold".**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **"addHotelRestriction", της web service "hoteldbAdd",** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<date_start>2010-02-19</date_start>
<date_end>2010-02-20</date_end>
<roomsize>Double</roomsize>
<days>5</days>
<price>54,45</price >
<id>2</id>
</root>
```

Όπου **id** ο αυξων αριθμος του ξενοδοχειου, **date_start** η ημερομηνία έναρξης, **date_end** η ημερομηνία λήξης, **roomsize** ο τύπος διαμονης του περιορισμου, **days** οι ελάχιστες μέρες μέχρι να ενεργοποιηθεί ο περιορισμός και **price** η τιμή του περιορισμού.
Μετά από μια επιτυχημένη προσθήκη θα καλεστεί η μέθοδος **"fillHotelRestriction",** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Κουμπί Delete Hotel Restriction

Το κουμπί **"Delete Hotel Restriction (x)"** καλεί τη μέθοδο **"deleteHotelRestriction"** από το bean με σκοπό να διαγράψει τη πολιτικη από το ξενοδοχείο. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού (x). Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

- **"restrictionsID"** η οποία είναι συνδεδεμένη με τον επιλεγμένο περιορισμο

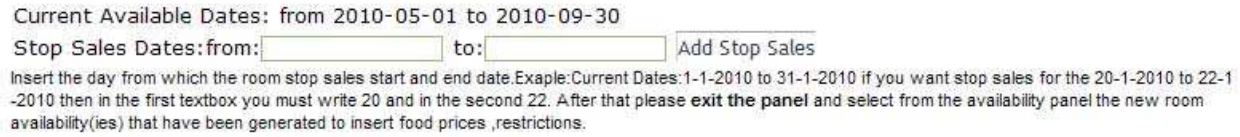
Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **"deleteHotelRestriction", της web service "hoteldbDelete",** αποστέλλοντας τη τιμή της μεταβλητης. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>4</id>
</root>
```

Όπου **id** ο αυξων αριθμος του περιορισμού.

Μετά από μια επιτυχημένη διόρθωση θα καλεστεί η μέθοδος **“fillHotelRestriction”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Panel: Insert Hotel’s Emergency Stop Sales



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής stop sales σε μια διαθεσιμότητα ένος ξενοδοχείου.

Αποτελείται από

- ✓ Το πλαίσιο κειμένου **“From”** το οποίο είναι ένα **“h:inputText”**
- ✓ Το πλαίσιο κειμένου **“To”** το οποίο είναι ένα **“h:inputText”**
- ✓ Το κουμπί **“Add Stop Sales”** το οποίο είναι ένα **“a4j:commandButton”**

Επεξήγηση Ρόλων

Το πλαίσιο κειμένου **“From”** δέχεται την ημέρα έναρξης του stop sale.

Το πλαίσιο κειμένου **“To”** δέχεται την ημέρα λήξης του stop sale

Το κουμπί **“Add Stop Sales”** είναι υπεύθυνο να παίρνει τις ημέρες έναρξης και λήξης και να εισάγει το stop sale στη διαθεσιμότητα του ξενοδοχείου.

Κουμπί Add Stop Sales Available

Το κουμπί **“Add Stop Sales”** καλεί τη μέθοδο **“addStopSalesAvailable”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

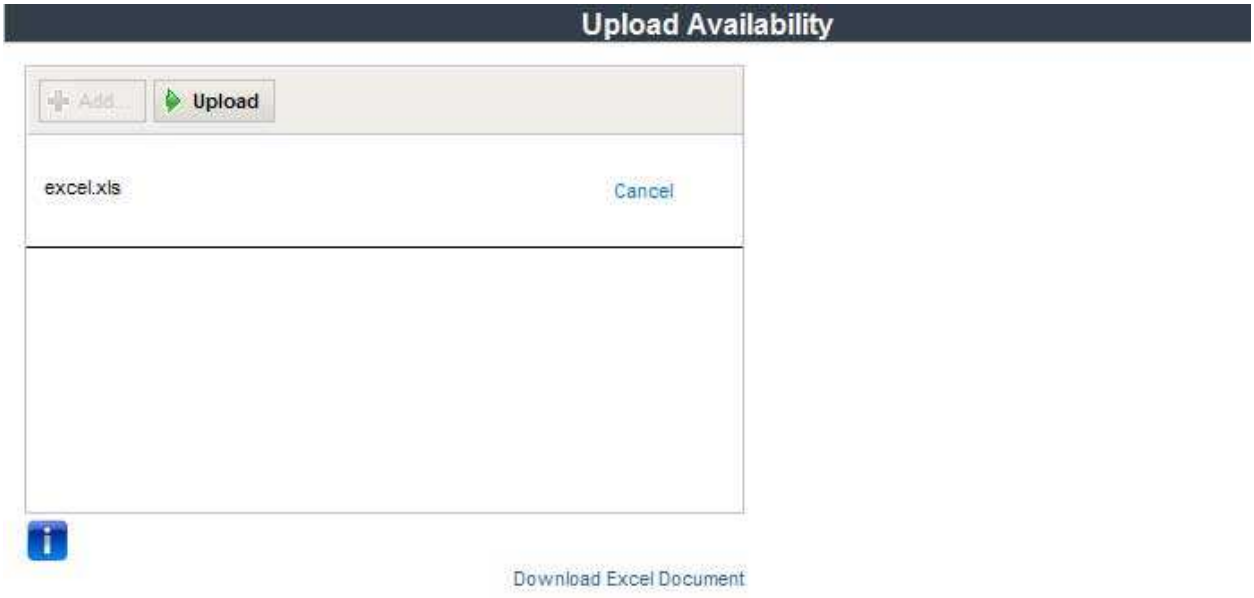
- **“stopslsDateSt”,** η ημερομηνια έναρξης του stopsale
- **“stopslsDateNd”,** η ημερομηνία λήξης του stopsale**.**
- **“avvID”,** το id της διαθεσιμότητας**.**

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“addRoomAvailable”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<st>2010-02-19</st>
<nd>2010-02-20</nd>
<id>2</id>
</root>
```

Όπου **id** ο αυξων αριθμος της διαθεσιμότητας, **st** η ημερομηνία έναρξης, **nd** η ημερομηνία λήξης του stop sale.
Μετά από μια επιτυχημένη προσθήκη θα καλεστεί η μέθοδος **“fillAvailable”,** με τη διαδικασια που αναλύσαμε παραπάνω, με σκοπό να ανανεωθούν τα περιεχόμενα του πίνακα και να εμφανιστούν οι αλλαγές που κάναμε.

Panel: Upload Availability



Λειτουργικότητα
Εμφανίζει τη διεπαφή της εισαγωγής διαθεσιμοτήτων από έτοιμα αρχεία excel

Αποτελείται από
- ✓ Τον uploader ο οποίος είναι ένας **“rich:fileUpload”**
- ✓ Τον σύνδεσμο **“Download Excel Document”**
- ✓ Τα **tooltip** επεξήγησης τα οποία είναι τύπου **“rich:tooltip”**

Επεξήγηση Ρόλων
Ο uploader είναι υπεύθυνος να παίρνει ένα αρχείο excel και μετα τις απαραιτητες διαδικασίες και ελέγχους να περνάει τη διαθεσιμότητα στο σύστημα.

Το Link **“Download Excel Document”** δίνει τη δυνατότητα στο χρηστη να κατεβάζει ένα αρχείο excel πρωτότυπο της μόρφής που δέχεται το σύστημα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Διαδικασία Upload Excel Availability

Ο uploader καλεί τη μέθοδο **“listenerExcelAvailability”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο UploadEvent που στη περίπτωση μας είναι το event upload() του uploader. Είναι υπεύθυνη να διαβάσει το αρχείο που γίνεται upload και να ελέγξει στη συνέχεια αν υπάρχουν οι φακέλοι στο path Που πρεπει να καταχωρηθεί.

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“copyFile”,** αποστέλλοντας τη τιμή του προσωρινού αρχείου που γίνεται upload και το Path του αρχείου που πρέπει να δημιουργηθεί. Η μέθοδος αυτή παίρνει το προσωρινό αρχείο και μεταφράζοντας το σε πίνακα των 1024bytes καταφερνει τη μεταφορά του.

Διαδικασία addExcelAvailability

Η διαδικασία **“addExcelAvailability”** καλείται όταν τελειώσει το Upload του αρχείου. Συγκεκριμένα καλείται η μέθοδος **“addExcelAvailability”** από το bean με σκοπό να προσθέσει τη διαθεσιμότητα στο ξενοδοχείο μετα τους απαραιτητους ελέγχους. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event uploadComplete() του uploader.

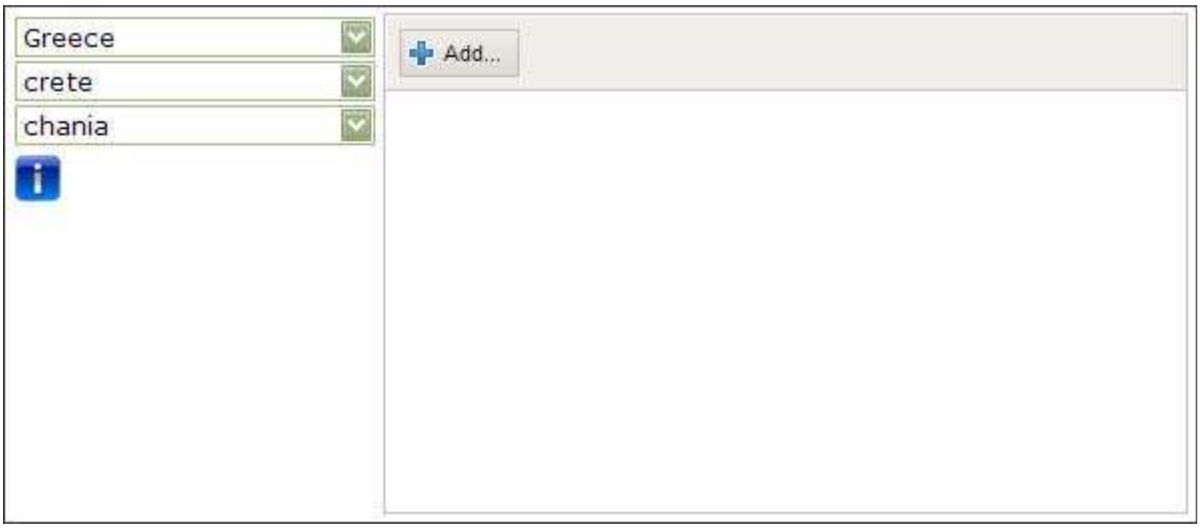
Μετα που θα διαβάσει το αρχείο θα επικοινωνήσει με τη μέθοδο **“addExcelAvailable”, της web service “hoteldbAdd”,** αποστέλλοντας τη τιμή των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<person>4</person>
<food>BB</food>
<price>24</price>
<date_start>2010-02-19</date_start>
<date_end>2010-02-20</date_end>
<id>4</id>
</root>
```

Όπου **id** ο αυξων αριθμος της διαθεσιμότητας, **person** ο αριθμός των ατόμων, **food** το όνομα της διατροφής, **date_start** η ημερομηνία έναρξης της διαθεσιμότητας, **date_end** η ημερομηνία λήξης της διαθεσιμότητας και **price** η τιμή της διαθεσιμότητας.

1.1.2 Upload General Fotos

Panel: Upload General Fotos



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής και διόρθωσης, των φωτογραφιών των περιοχών που υποστηρίζει το σύστημα.

Αποτελείται από

- ✓ Τον uploader ο οποίος είναι ένας **“rich:fileUpload”**
- ✓ Το μενού επιλογής **“Country”** το οποίο είναι ένα **“h:selectOneMenu”**
- ✓ Το μενού επιλογής **“Region”** το οποίο είναι ένα **“h:selectOneMenu”**
- ✓ Το μενού επιλογής **“City”** το οποίο είναι ένα **“h:selectOneMenu”**
- ✓ Το **tooltip** επεξήγησης το οποίο είναι τύπου **“rich:tooltip”**

Επεξήγηση Ρόλων

Ο uploader είναι υπεύθυνος να παίρνει μια φωτογραφία και μετα τις απαραίτητες διαδικασίες να την αποθηκεύει στον server και στη βάση για τη συγκεκριμένη περιοχή.

Τα μενού επιλογής είναι υπεύθυνα να εμφανίζουν τις περιοχες του συστήματος για να διαλέξει ο χρήστης. Οι περιοχές που εμφανίζονται είναι εκείνες που επιτρέπεται να παρέμβει ο χρήστης. Ο περιορισμός αυτός ορίζεται από τις επιλογές του διαχειρηστή και έχουμε μιλήσει σε προηγούμενη ενότητα για τη λειτουργία του.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Μενού Επιλογής Country

Το μενού επιλογής **“Country”** καλεί τη μέθοδο **“fillCbxRegionFoto”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event selectedIndexChanged() του μενού επιλογής. Είναι υπεύθυνη στο να διαβάσει τη μεταβλητή τύπου String με όνομα **“countryFotoHotel”**, η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή του μενού επιλογής, και να επικοινωνήσει με τη μέθοδο **“fillRegion”, της web service “hoteldbFill”,** αποστέλλοντας της την τιμή της μεταβλητής. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<root>
<country_name>Greece</country_name>
</root>
```

Όπου **country_name** το όνομα της χώρας της οποίας, έχει επιλέξει ο χρήστης να πάρει, τις επαρχίες. Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>22</id>
<region_name>Attica</region_name>
<flag>1</flag>
<id>25</id>
<region_name>Crete</region_name>
<flag>1</flag>
```

Το πεδίο **region_name** το οποίο αντιπροσωπεύει το όνομα της επαρχίας. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του μενού επιλογής **“Region”** με τις τιμές που λάβαμε από το μήνυμα απάντησης

Μενού Επιλογής Region

Το μενού επιλογής **“Region”** καλεί τη μέθοδο **“fillCbxCityFoto”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο ActionEvent που στη περίπτωση μας είναι το event selectedIndexChanged() του μενού επιλογής. Είναι υπεύθυνη στο να διαβάσει τις μεταβλητές:

1. String με όνομα **“countryFotoHotel”**, η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή του μενού επιλογής **country**
2. String με όνομα **“regionFotoHotel”**, η οποία είναι συνδεδεμένη με την επιλεγμένη τιμή του μενού επιλογής **region**

και να επικοινωνήσει με τη μέθοδο **“fillCity”, της web service “hoteldbFill”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<region_name>Crete</region_name>
</root>
```

Όπου **country_name** το όνομα της χώρας και **region_name** το όνομα της επαρχίας, της οποίας έχει επιλέξει ο χρήστης να πάρει τις πόλεις. Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<city_name>Chania</city_name>
</root>
```

Το πεδίο **city_name** το οποίο αντιπροσωπεύει το όνομα της πόλης και. Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του Μενου επιλογής **“City”** με τις τιμές που λάβαμε από το μήνυμα απάντησης

Διαδικασία Upload Fotos

Ο uploader καλεί τη μέθοδο **“listener”** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο UploadEvent που στη περίπτωση μας είναι το event upload() του uploader. Είναι υπεύθυνη να διαβάσει το αρχείο που γίνεται upload και να ελέγξει στη συνέχεια αν υπάρχουν οι φακέλοι στο path Που πρεπει να κατασωρηθεί.

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **“copyFile”,** αποστέλλοντας τη τιμή του προσωρινού αρχείου που γίνεται upload και το Path του αρχείου που πρέπει να δημιουργηθεί. Η μέθοδος αυτή παίρνει το προσωρινό αρχείο και μεταφράζοντας το σε πίνακα των 1024bytes καταφλερνει τη μεταφορά του.

Μετά από μια επιτυχημένη προσθήκη θα καλεστεί η μέθοδος **“addFotoCountry”, “addFotoRegion” ή “addFotoCity”,** αναλόγως σε ποιο επίπεδο έκανε προσθήκη ο χρήστης και θα ενημερώθεί η βάση για τη προσθήκη.

Διαδικασία addFotoCountry

Καλείται η μέθοδος **“addFotoCountry”** και παίρνει σαν όρισματα, μια μεταβλητή string **country** το όνομα της χώρας που προσθέτουμε τη φωτογραφία και μια μεταβλητή string **foto** για το url της φωτογραφίας. Στη συνέχεια επικοινωνήσει με τη μέθοδο **“addFotoCountry”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<foto>http://www.broker-lodge.com/hotelsData/Countries/Greece/Greece.jpg</foto>
</root>
```

Όπου **country_name** το όνομα της χώρας και **foto** το url της φωτογραφίας.

Διαδικασία addFotoRegion

Καλείται η μέθοδος **“addFotoRegion”** και παίρνει σαν όρισματα, μια μεταβλητή string **country** το όνομα της χώρας, μια μεταβλητή string **region** το όνομα της επαρχίας που προσθέτουμε τη φωτογραφία και μια μεταβλητή string **foto** για το url της φωτογραφίας. Στη συνέχεια επικοινωνήσει με τη μέθοδο **“addFotoRegion”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<region_name>Crete</region_name>
<foto>http://www.broker-lodge.com/hotelsData/Countries/Greece/Crete/Crete.jpg</foto>
</root>
```

Όπου **country_name** το όνομα της χώρας, **region_name** το όνομα της επαρχίας και **foto** το url της φωτογραφίας.

Διαδικασία addFotoCity

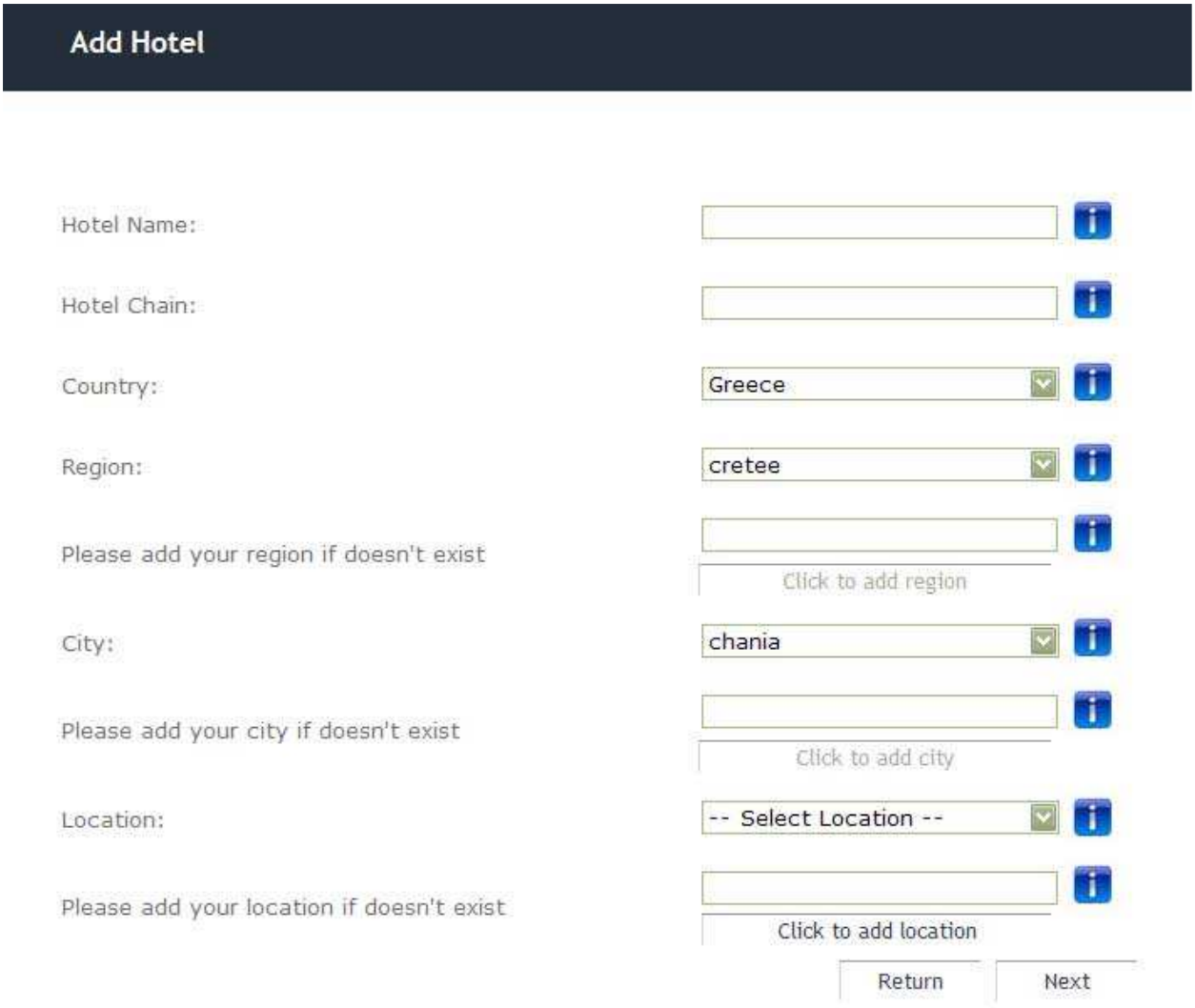
Καλείται η μέθοδος **“addFotoCity”** και παίρνει σαν όρισματα, μια μεταβλητή string **country** το όνομα της χώρας, μια μεταβλητή string **region** το όνομα της επαρχίας, μια μεταβλητή string **city** το όνομα της πόλης που προσθέτουμε τη φωτογραφία και μια μεταβλητή string **foto** για το url της φωτογραφίας. Στη συνέχεια επικοινωνήσει με τη μέθοδο **“addFotoCity”, της web service “hoteldbAdd”,** αποστέλλοντας τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<region_name>Crete</region_name>
<city_name>Chania</city_name>
<foto>http://www.broker-
lodge.com/hotelsData/Countries/Greece/Crete/Chania/Chania.jpg</foto>
</root>
```

Όπου **country_name** το όνομα της χώρας, **region_name** το όνομα της επαρχίας, **city_name** το όνομα της πόλης και **foto** το url της φωτογραφίας.

1.1.3 Add New Hotel

Procedure: Add Hotel Step 1



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής ενός καινούργιου ξενοδοχείου στη βάση. Το πρώτο βήμα περιλαμβάνει το όνομα του ξενοδοχείου και τις βασικές λειτουργίες για τη τοποθεσία του.

Αποτελείται από

- ✓ Το πεδίο κειμένου **"Hotel Name"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το πεδίο κειμένου **"Hotel Chain"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το μενού επιλογής **"Country"** το οποίο είναι ένα **"h:selectOneMenu"**
- ✓ Το μενού επιλογής **"Region"** το οποίο είναι ένα **"h:selectOneMenu"**
- ✓ Το πεδίο κειμένου **"Add Region"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Add Region"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το μενού επιλογής **"City"** το οποίο είναι ένα **"h:selectOneMenu"**
- ✓ Το πεδίο κειμένου **"Add City"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Add City"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το μενού επιλογής **"Location"** το οποίο είναι ένα **"h:selectOneMenu"**

- ✓ Το πεδίο κειμένου **“Add Location”** το οποίο είναι ένα **“h:inputText”**
- ✓ Το κουμπί **“Add Location”** το οποίο είναι ένα **“a4j:commandButton”**
- ✓ Τα **tooltip** επεξήγησης τα οποία είναι τύπου **“rich:tooltip”**
- ✓ Το κουμπί **“Return”** το οποίο είναι ένα **“h:commandButton”**
- ✓ Το κουμπί **“Next”** το οποίο είναι ένα **“a4j:commandButton”**

Επεξήγηση Ρόλων

Τα μενού επιλογής είναι υπεύθυνα να εμφανίζουν τις περιοχές του συστήματος για να διαλέξει ο χρήστης. Οι περιοχές που εμφανίζονται είναι εκείνες που επιτρέπεται να παρέμβει ο χρήστης. Ο περιορισμός αυτός ορίζεται από τις επιλογές του διαχειρηστή και έχουμε μιλήσει σε προηγούμενη ενότητα για τη λειτουργία του. Επίσης ο τρόπος λειτουργίας και διασύνδεσης των μενού επιλογής είναι ο ίδιος με αυτόν που έχουμε αναλύσει σε προηγούμενη ενότητα.

Το κουμπί **add Region** είναι υπεύθυνο να πάρει το κείμενο που εισήγαγε ο χρήστης στο πεδίο κειμένου **Region** και μετα τους απαραίτητους ελέγχους να προσθέσει την επαρχία κάτω από την επιλεγμένη χώρα.

Το κουμπί **add City** είναι υπεύθυνο να πάρει το κείμενο που εισήγαγε ο χρήστης στο πεδίο κειμένου **City** και μετα τους απαραίτητους ελέγχους να προσθέσει την πόλη κάτω από την επιλεγμένη χώρα και επαρχία.

Το κουμπί **add Location** είναι υπεύθυνο να πάρει το κείμενο που εισήγαγε ο χρήστης στο πεδίο κειμένου **Location** και μετα τους απαραίτητους ελέγχους να προσθέσει την τοποθεσία κάτω από την επιλεγμένη χώρα, επαρχία και πόλη.

Οι διαδικασίες πρόσθεσης επαρχίας, πόλης και τοποθεσίας είναι ίδιες με αυτές που έχουμε αναλύσει σε προηγούενη ενότητα.

Το κουμπί **Return** είναι υπεύθυνο να μηδενίζει τις μεταβλητές και να επιστρέφει τον χρήστη στη προηγούμενη σελίδα.

Το κουμπί **Next** είναι υπέυθυνο να προχωράει τον χρήστη στο επόμενο βήμα αφού πρώτα γίνουν οι απαραίτητοι έλεγχοι και διαπιστωθεί ότι δεν υπάρχει σφάλμα στη σελίδα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Procedure: Add Hotel Step 2



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής ενός καινούργιου ξενοδοχείου στη βάση. Το δεύτερο βήμα περιλαμβάνει πληροφορίες για τον τύπο και τη ποιότητα του ξενοδοχείου καθώς και πληροφορίες επικοινωνίας.

Αποτελείται από

- ✓ Το πεδίο κειμένου **"Address"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το μενού επιλογής **"Staying Type"** το οποίο είναι ένα **"h:selectOneMenu"**
- ✓ Το πεδίο κειμένου **"Stars"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το πεδίο κειμένου **"Total Rooms"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το πεδίο κειμένου **"Email"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το πεδίο κειμένου **"Phone"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το πεδίο κειμένου **"Fax"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το μενού επιλογής **"Check In"** το οποίο είναι ένα **"h:selectOneMenu"**
- ✓ Το μενού επιλογής **"Check Out"** το οποίο είναι ένα **"h:selectOneMenu"**
- ✓ Τα **tooltip** επεξήγησης τα οποία είναι τύπου **"rich:tooltip"**
- ✓ Το κουμπί **"Return"** το οποίο είναι ένα **"h:commandButton"**
- ✓ Το κουμπί **"Next"** το οποίο είναι ένα **"a4j:commandButton"**

Επεξήγηση Ρόλων

Το μενού επιλογής **“Staying Type”** είναι υπεύθυνο να εμφανίζει τους διαθέσιμους τύπους διαμονής για να επιλέξει ο χρήστης.

Το μενού επιλογής **“Stars”** είναι υπεύθυνο να εμφανίζει τους διαθέσιμους αριθμούς αστεριών για τη βαθμολογία του ξενοδοχείου.

Το μενού επιλογής **“Check In”** είναι υπεύθυνο να εμφανίζει τις διαθέσιμες ώρες για να επιλέξει ο χρήστης το check in του ξενοδοχείου.

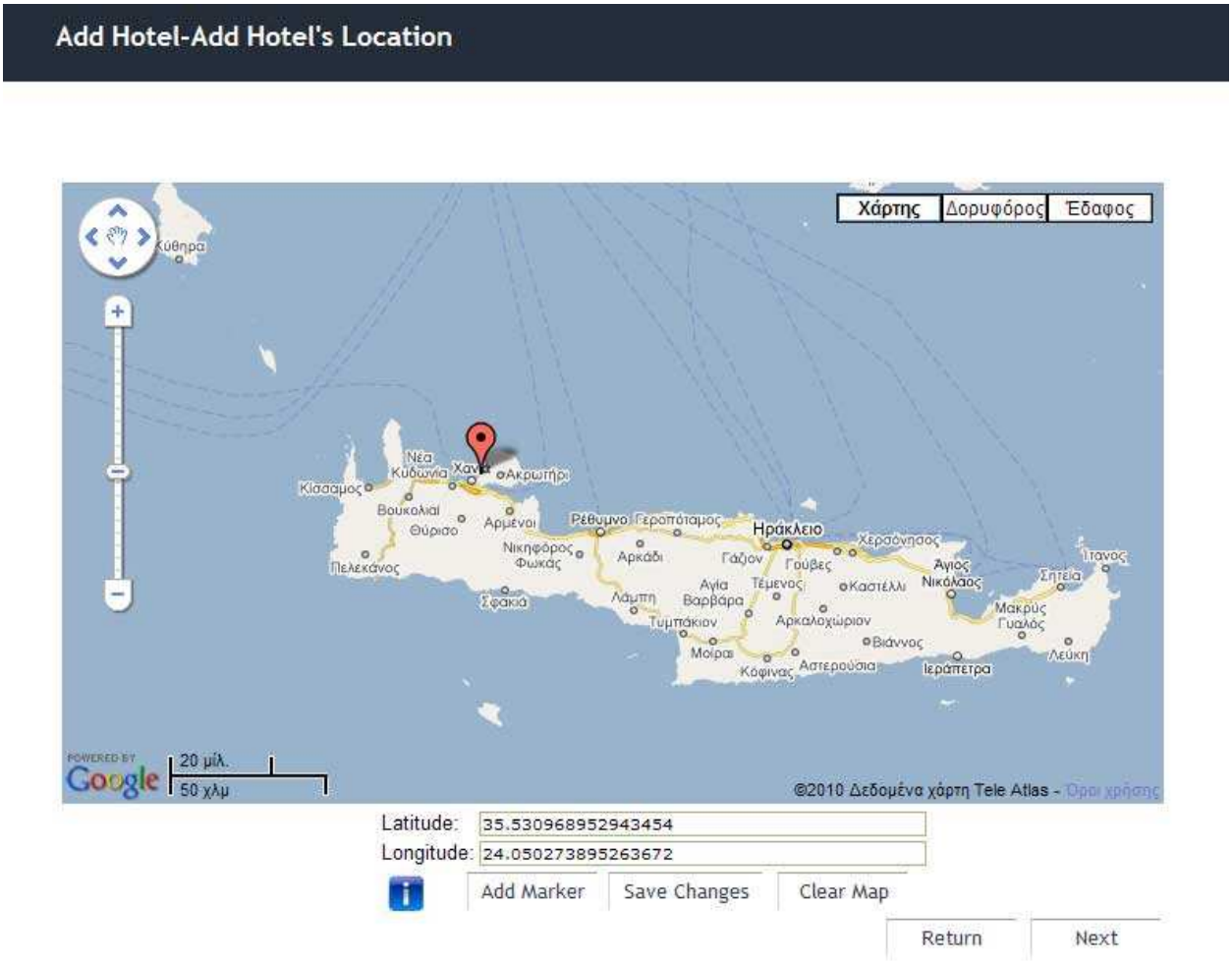
Το μενού επιλογής **“Check Out”** είναι υπεύθυνο να εμφανίζει τις διαθέσιμες ώρες για να επιλέξει ο χρήστης το check out του ξενοδοχείου.

Το κουμπί **Return** είναι υπεύθυνο να επιστρέφει τον χρήστη στο προηγούμενο βήμα.

Το κουμπί **Next** είναι υπεύθυνο να προχωράει τον χρήστη στο επόμενο βήμα αφού πρώτα γίνουν οι απαραίτητοι έλεγχοι και διαπιστωθεί ότι δεν υπάρχει σφάλμα στη σελίδα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Procedure: Add Hotel Step 3



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής ενός καινούργιου ξενοδοχείου στη βάση. Το τρίτο βήμα περιλαμβάνει την εισαγωγή της συγκεκριμένης τοποθεσίας του ξενοδοχείου με συντεταγμένες με τη βοήθεια των Google Maps.

Αποτελείται από

- ✓ Τον χάρτη Google Maps
- ✓ Το πεδίο κειμένου **"Latitude"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το πεδίο κειμένου **"Longitude"** το οποίο είναι ένα **"h:inputText"**
- ✓ Το κουμπί **"Add Marker"** το οποίο είναι ένα **"input"**
- ✓ Το κουμπί **"Save Changes"** το οποίο είναι ένα **"a4j:commandButton"**
- ✓ Το κουμπί **"Clear Map"** το οποίο είναι ένα **"input"**
- ✓ Τα **tooltip** επεξήγησης τα οποία είναι τύπου **"rich:tooltip"**
- ✓ Το κουμπί **"Return"** το οποίο είναι ένα **"h:commandButton"**
- ✓ Το κουμπί **"Next"** το οποίο είναι ένα **"a4j:commandButton"**

Επεξήγηση Ρόλων

Το κουμπί **"Add Marker"** είναι υπεύθυνο να προσθέτει μια πινέζα στο χάρτη την οποία ο χρήστης μπορεί να μετακινεί έως οτου βρει την τοποθεσία του ξενοδοχείου. Μια πινέζα μπορει να προσθεθεί κάθε φορά

Το κουμπί **"Clear Map"** είναι υπεύθυνο να αφαιρέσει την πινέζα από τον χάρτη καθαρίζοντας τον.

Το πλαίσιο κειμένου **"Latitude"** είναι υπεύθυνο να εμφανίζει κάθε φορά το γεωγραφικό μήκος της τοποθεσίας που βρίσκεται η πινέζα.

Το πλαίσιο κειμένου **"Longitude"** είναι υπεύθυνο να εμφανίζει κάθε φορά το γεωγραφικό πλάτος της τοποθεσίας που βρίσκεται η πινέζα.

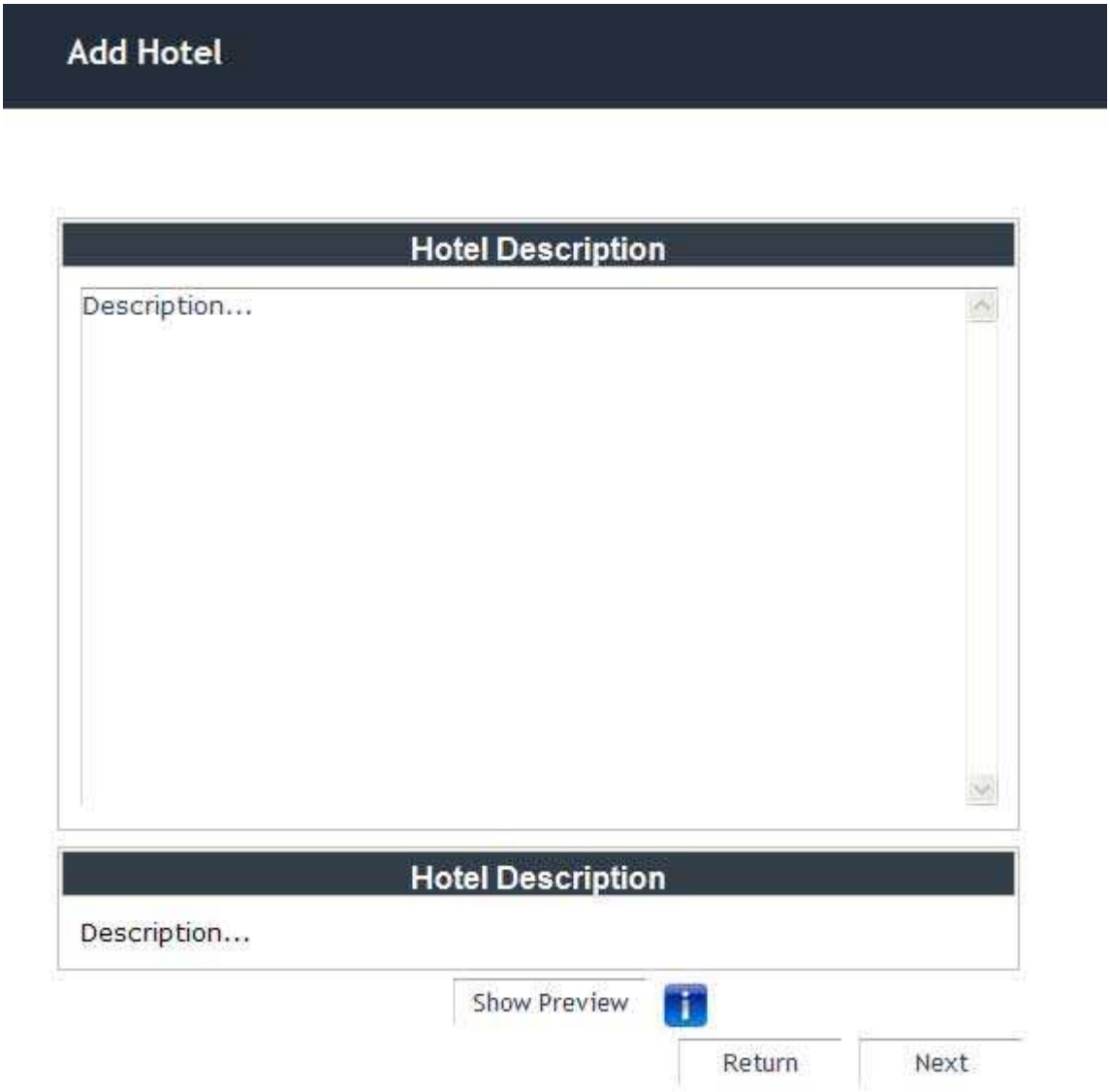
Το κουμπί **"Save Changes"** είναι υπεύθυνο να παίρνει τις τιμές από τα πλαίσια κειμένου και να τις προσθέτει στις μεταβλητές του bean. Συγκεκριμένα το latitude το δίνει στην **lang** και το longitude στη **lng.**

Το κουμπί **Return** είναι υπεύθυνο να επιστρέφει τον χρήστη στο προηγούμενο βήμα.

Το κουμπί **Next** είναι υπέυθυνο να προχωράει τον χρήστη στο επόμενο βήμα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Procedure: Add Hotel Step 4



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής ενός καινούργιου ξενοδοχείου στη βάση. Το τέταρτο βήμα περιλαμβάνει την εισαγωγή της περιγραφής του ξενοδοχείου.

Αποτελείται από

- ✓ Το πεδίο κειμένου **"Description"** το οποίο είναι ένα **"h:inputTextArea"**
- ✓ Το πεδίο κειμένου **"Description Preview"** το οποίο είναι ένα **"h:outputText"**
- ✓ Το κουμπί **"Show Preview"** το οποίο είναι ένα **"h:commandButton"**
- ✓ Το **tooltip** επεξήγησης το οποίο είναι τύπου **"rich:tooltip"**
- ✓ Το κουμπί **"Return"** το οποίο είναι ένα **"h:commandButton"**
- ✓ Το κουμπί **"Next"** το οποίο είναι ένα **"a4j:commandButton"**

Επεξήγηση Ρόλων

Το πλαίσιο κειμένου **“Description”** είναι υπεύθυνο να δέχεται τον κείμενο της περιγραφής του ξενοδοχείου.

Το πλαίσιο κειμένου **“Descrption Preview”** είναι υπεύθυνο να εμφανίζει την περιγραφή του ξενοδοχείου που έχει εισάγει ο χρήστης έως εκείνη τη στιγμή ενημερώνοντας τον για την ακριβή μορφή που θα έχει στη σελίδα απεικόνισης.

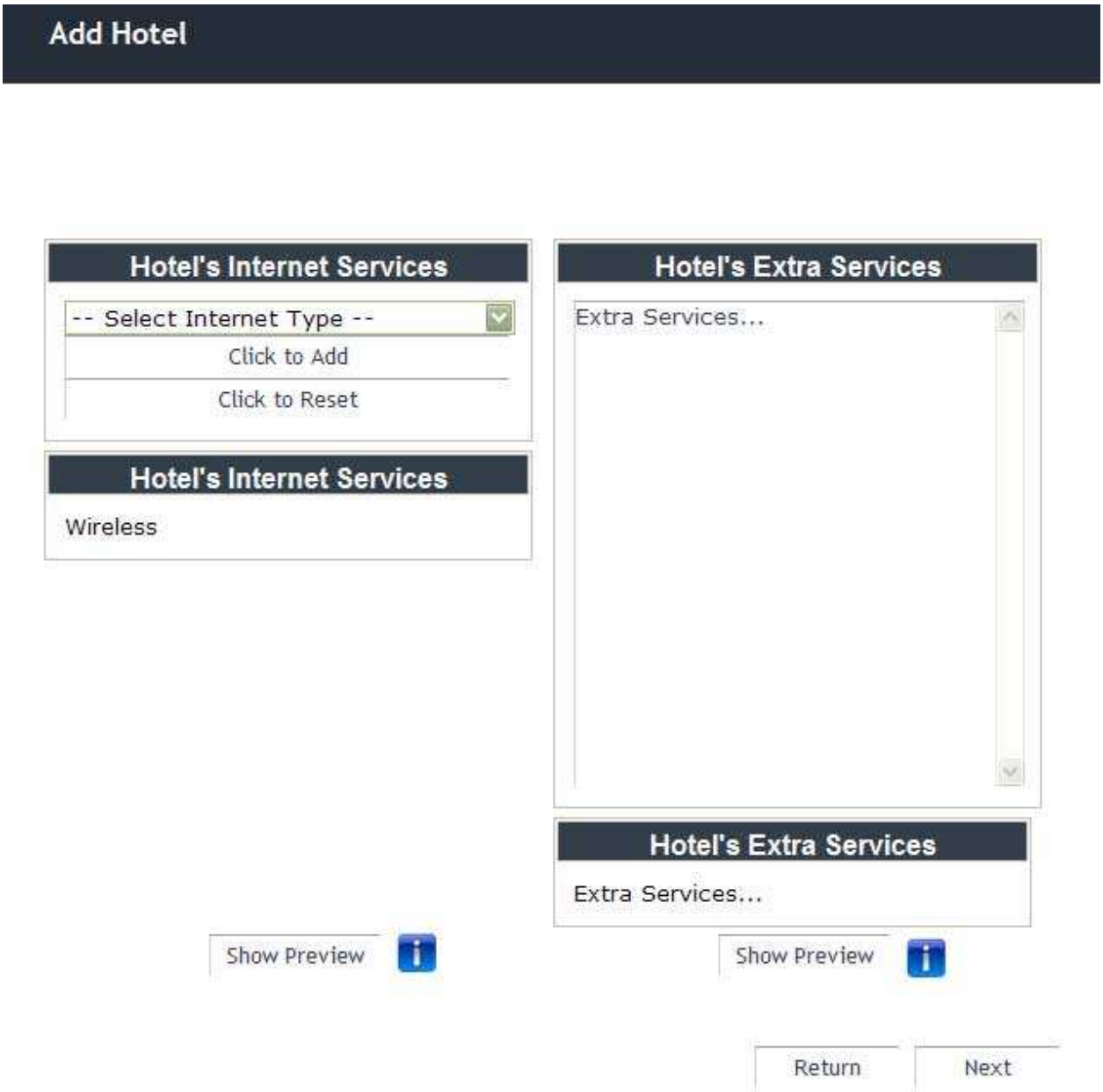
Το κουμπί **“Show Preview”** είναι υπεύθυνο να ανανεώνει τα περιεχόμενα του πλαίσιου απεικόνησης της περιγραφής για να ενημερωθεί με τις τελευταίες αλαγές.

Το κουμπί **Return** είναι υπεύθυνο να επιστρέφει τον χρήστη στο προηγούμενο βήμα.

Το κουμπί **Next** είναι υπέυθυνο να προχωράει τον χρήστη στο επόμενο βήμα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Procedure: Add Hotel Step 5



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής ενός καινούργιου ξενοδοχείου στη βάση. Το πέμπτο βήμα περιλαμβάνει την εισαγωγή των τύπων Internet που παρέχει το ξενοδοχείο καθώς και τις επιπρόσθετες υπηρεσίες.

Αποτελείται από

- ✓ Το μενού επιλογής **"Internet Type"** το οποίο είναι ένα **"h:selectOneMenu"**
- ✓ Το κουμπί **"Add"** το οποίο είναι ένα **"h:commandButton"**
- ✓ Το κουμπί **"Reset"** το οποίο είναι ένα **"h:commandButton"**

- ✓ Το πεδίο κειμένου **“Internet Preview”** το οποίο είναι ένα **“h:outputText”**
- ✓ Το πεδίο κειμένου **“Extra Services”** το οποίο είναι ένα **“h:inputTextArea”**
- ✓ Το πεδίο κειμένου **“ Extra Services Preview”** το οποίο είναι ένα **“h:outputText”**
- ✓ Τα κουμπιά **“Show Preview”** τα οποία είναι **“h:commandButton”**
- ✓ Το **tooltip** επεξήγησης το οποίο είναι τύπου **“rich:tooltip”**
- ✓ Το κουμπί **“Return”** το οποίο είναι ένα **“h:commandButton”**
- ✓ Το κουμπί **“Next”** το οποίο είναι ένα **“h:commandButton”**

Επεξήγηση Ρόλων

Το μενού επιλογής **Internet Type** είναι υπεύθυνο να εμφανίζει τους διαθέσιμους τύπους internet τους συστήματος για να τους προσθέσει ο χρήστης στο ξενοδοχείο.

Το κουμπί **Add** παίρνει τον επιλεγμένο τύπο internet και τον προσθέτει στο ξενοδοχείο.

Το κουμπί **Reset** καθαρίζει τη λίστα με τους εισαχθέντες τύπους internet στο ξενοδοχείο.

Το πλαίσιο κειμένου **internet Preview** είναι υπεύθυνο να εμφανίζει τη λίστα με τους εισαχθέντες τύπους internet στο ξενοδοχείο.

Το πλαίσιο κειμένου **“Extra Services”** είναι υπεύθυνο να δέχεται τον κείμενο των επιπρόσθετων υπηρεσιών που παρέχει το ξενοδοχείο.

Το πλαίσιο κειμένου **“ Extra Services Preview”** είναι υπεύθυνο να εμφανίζει το κείμενο των πρόσθετων υπηρεσών του ξενοδοχείου που έχει εισάγει ο χρήστης έως εκείνη τη στιγμή ενημερώνοντας τον για την ακριβή μορφή που θα έχει στη σελίδα απεικόνισης.

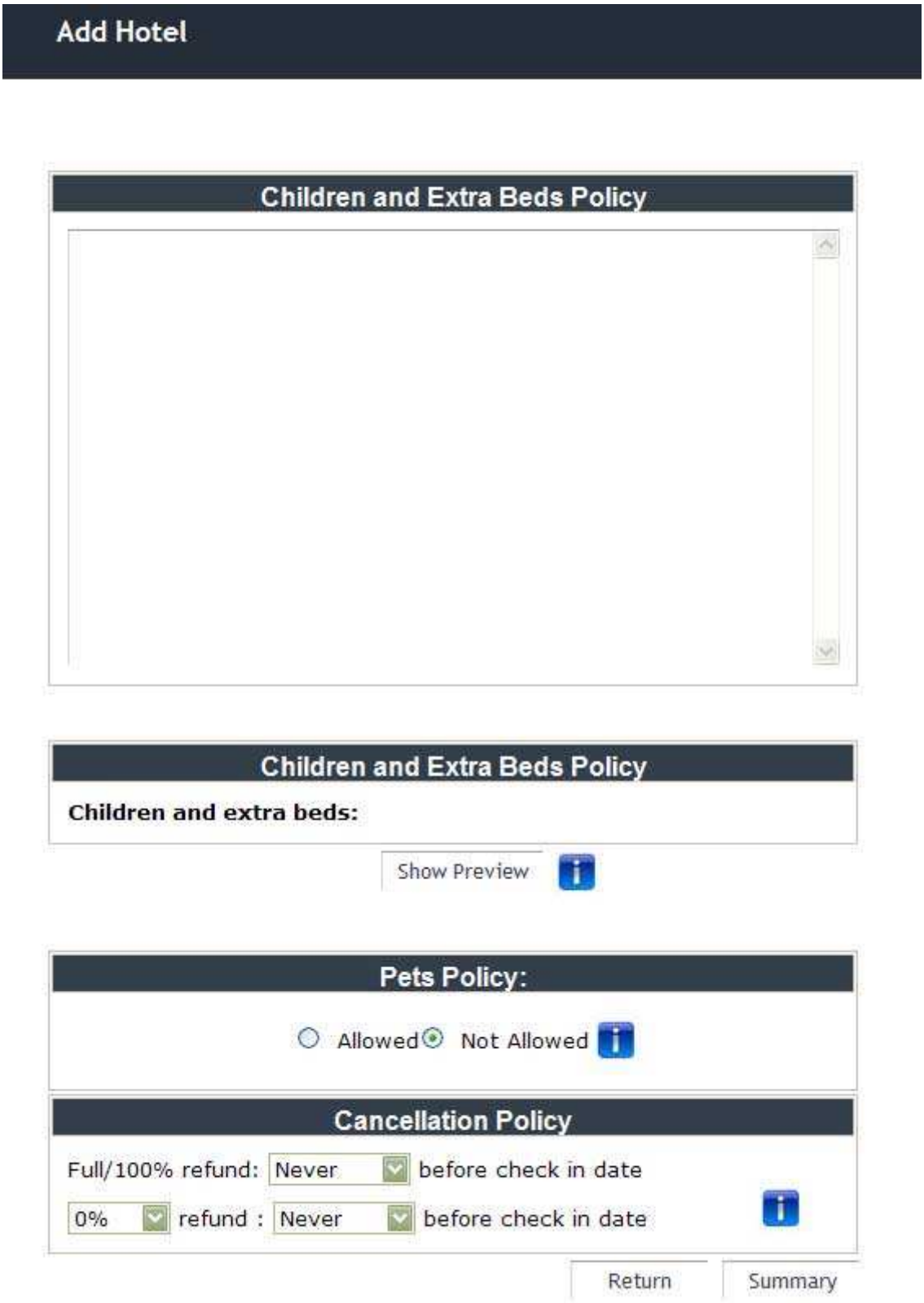
Τα κουμπιά **“Show Preview”** είναι υπεύθυνα να ανανεώνουν τα περιεχόμενα των πλαισίων απεικόνησης του internet και των υπηρεσιών για να ενημερωθεί με τις τελευταίες αλαγές.

Το κουμπί **Return** είναι υπεύθυνο να επιστρέφει τον χρήστη στο προηγούμενο βήμα.

Το κουμπί **Next** είναι υπέυθυνο να προχωράει τον χρήστη στο επόμενο βήμα.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Procedure: Add Hotel Step 6



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής ενός καινούργιου ξενοδοχείου στη βάση. Το έκτο βήμα περιλαμβάνει την εισαγωγή της πολιτικής παιδιών και επιπρόσθετων κρεβατιών καθώς και τις πολιτικές ακυρώσεων και κατοικιδίων.

Αποτελείται από

- ✓ Το πεδίο κειμένου **"Children & Bed Policy"** το οποίο είναι ένα **"h:inputTextArea"**
- ✓ Το πεδίο κειμένου **" Children & Bed Policy Preview"** το οποίο είναι ένα **"h:outputText"**
- ✓ Το κουμπί **"Show Preview"** το οποίο είναι **"h:commandButton"**
- ✓ Τα κουμπία επιλογής **"Pets Policy"** τα οποία περιλαμβάνονται σε ένα **"h:selectOneRadio"**
- ✓ Τα μενού επιλογής **"Cancellation Policy"** τα οποία είναι **"h:selectOneMenu"**
- ✓ Τα **tooltip** επεξήγησης τα οποία είναι τύπου **"rich:tooltip"**
- ✓ Το κουμπί **"Return"** το οποίο είναι ένα **"h:commandButton"**
- ✓ Το κουμπί **"Summary"** το οποίο είναι ένα **"a4j:commandButton"**

Επεξήγηση Ρόλων

Το πλαίσιο κειμένου **"Children & Bed Policy"** είναι υπεύθυνο να δέχεται το κείμενο της πολιτικής παιδιών και επιπροσθέτων κρεβατιών του ξενοδοχείου.

Το πλαίσιο κειμένου **"Children & Bed Policy Preview"** είναι υπεύθυνο να εμφανίζει το κείμενο της πολιτικής παιδιών και επιπροσθέτων κρεβατιών του ξενοδοχείου που έχει εισάγει ο χρήστης έως εκείνη τη στιγμή ενημερώνοντας τον για την ακριβή μορφή που θα έχει στη σελίδα απεικόνισης.

Το κουμπί **"Show Preview"** είναι υπεύθυνο να ανανεώσει το περιεχόμενο του πλαισίου απεικόνησης της πολιτικής για να ενημερωθεί ο χρήστης με τις τελευταίες αλαγές.

Τα κουμπία επιλογής **"Pets Policy"** δίνουν τη δυνατότητα στο χρήστη να επιλέξει αν θα επιτρέπονται τα κατοικίδια στο ξενοδοχείο ή όχι.

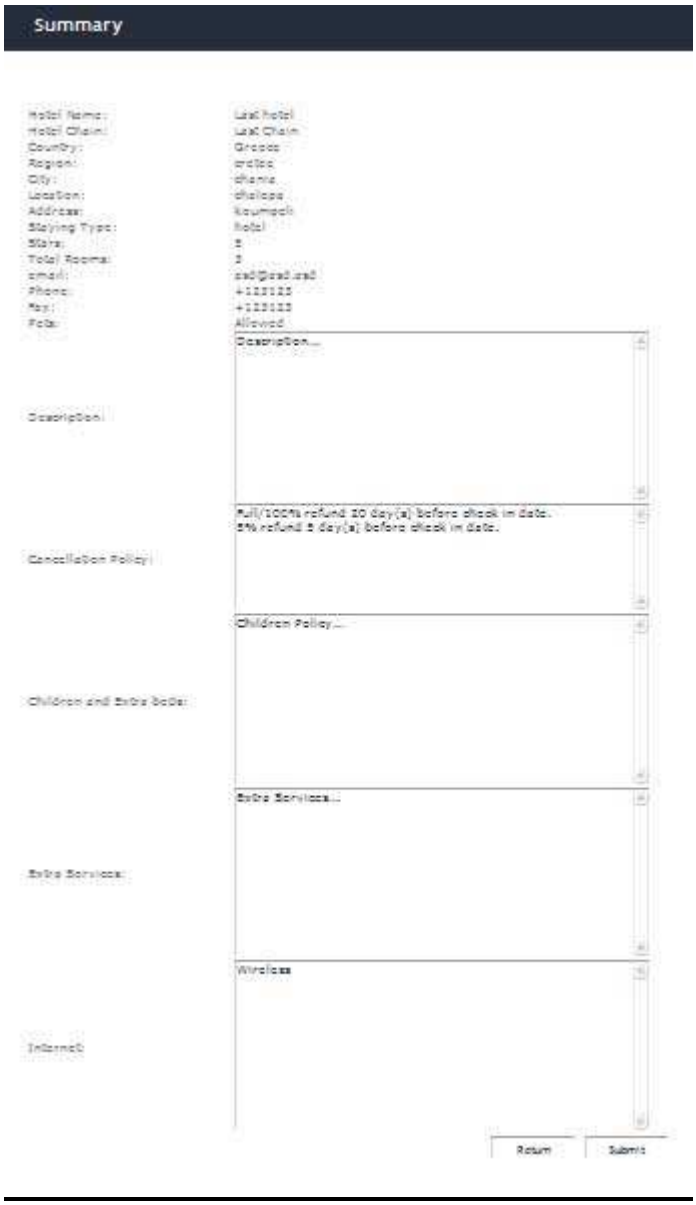
Τα μενού επιλογής **"Cancellation Policy"** δίνουν τη δυνατότητα στο χρήστη να εισάγει την πολιτική ακυρώσεων κρατήσεων. Το πρώτο μενού αφορα τον ελάχιστο αριθμό των ημερών που έως τότε θα επιστραφεί το 100% του ποσού της κράτηση. Τα δεύτερο μενού αφορά το ποσοστό της επιστροφής χρημάτων και το τρίτο μενού τις ελάχιστες μέρες που έως τότε θα εφαρμοστεί το ποσοστο της επιστροφής. Θεωρειται ότι μετα το πέρας των ημερων του τρίτου μενού δεν εφαρμόζεται επιστροφή χρημάτων.

Το κουμπί **Return** είναι υπεύθυνο να επιστρέφει τον χρήστη στο προηγούμενο βήμα.

Το κουμπί **Summary** είναι υπεύθυνο να προχωράει τον χρήστη στο επόμενο βήμα.

Τα **tooltip** είναι υπεύθυνα, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζουν ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Procedure: Add Hotel Step 7



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής ενός καινούργιου ξενοδοχείου στη βάση. Το έβδομο βήμα περιλαμβάνει την σύνοψη των πληροφοριών που εισήχθησαν στα προηγούμενα βήματα για επαλήθευση.

Αποτελείται από

- ✓ Το πεδίο κειμένου **“Hotel Name”** το οποίο είναι **“h:outputText”**
- ✓ Το πεδίο κειμένου **“Hotel Chain”** το οποίο είναι **“h:outputText”**
- ✓ Το πεδίο κειμένου **“Country”** το οποίο είναι **“h:outputText”**
- ✓ Το πεδίο κειμένου **“Region”** το οποίο είναι **“h:outputText”**
- ✓ Το πεδίο κειμένου **“City”** το οποίο είναι **“h:outputText”**

- ✓ Το πεδίο κειμένου **“Location”** το οποίο είναι **“h:outputText”**
- ✓ Το πεδίο κειμένου **“Address”** το οποίο είναι **“h:outputText”**
- ✓ Το πεδίο κειμένου **“Staying Type”** το οποίο είναι **“h:outputText”**
- ✓ Το πεδίο κειμένου **“Stars”** το οποίο είναι **“h:outputText”**
- ✓ Το πεδίο κειμένου **“Total Rooms”** το οποίο είναι **“h:outputText”**
- ✓ Το πεδίο κειμένου **“Email”** το οποίο είναι **“h:outputText”**
- ✓ Το πεδίο κειμένου **“Phone”** το οποίο είναι **“h:outputText”**
- ✓ Το πεδίο κειμένου **“Fax”** το οποίο είναι **“h:outputText”**
- ✓ Το πεδίο κειμένου **“Pets”** το οποίο είναι **“h:outputText”**
- ✓ Το πεδίο κειμένου **“Description”** το οποίο είναι ένα **“h:inputTextArea”**
- ✓ Το πεδίο κειμένου **“Cancelation”** το οποίο είναι ένα **“h:inputTextArea”**
- ✓ Το πεδίο κειμένου **“Children & Beds”** το οποίο είναι ένα **“h:inputTextArea”**
- ✓ Το πεδίο κειμένου **“Extra Services”** το οποίο είναι ένα **“h:inputTextArea”**
- ✓ Το πεδίο κειμένου **“Internet”** το οποίο είναι ένα **“h:inputTextArea”**
- ✓ Το κουμπί **“Return”** το οποίο είναι ένα **“a4j:commandButton”**
- ✓ Το κουμπί **“Summary”** το οποίο είναι ένα **“h:commandButton”**

Επεξήγηση Ρόλων

Όλα τα πλαίσια κειμένου είναι υπεύθυνα να απεικονίσουν την τιμή που είχε εισάγει ο χρήστης στα προηγούμενα βήματα.

Το κουμπί **Return** είναι υπεύθυνο να επιστρέφει τον χρήστη στο προηγούμενο βήμα.

Το κουμπί **Summary** είναι υπεύθυνο να προχωράει τον χρήστη στο επόμενο βήμα.

Διαδικασία AddHotel

Το κουμπί **Summary** καλει τη μέθοδο **addHotel** από το bean η οποία δέχεται σαν όρισμα ένα ActionEvent που στη περίπτωση μας είναι το event click() του κουμπιού. Είναι υπαύθυνη να διαβάσει τις παρακάτω μεταβλητές:

- **hotelName,** το όνομα του ξενοδοχείου
- **hotelChain,** το όνομα της αλυσίδας του ξενοδοχείου
- **hotelEmail,** το email του ξενοδοχείου
- **hotelAddress,** η διεύθυνση του ξενοδοχείου
- **hotelStars,** ο αριθμός των αστεριών του ξενοδοχείου
- **hotelPhone,** το τηλέφωνο του ξενοδοχείου
- **hotelFax,** το fax του ξενοδοχείου
- **hotelStayingType,** ο τύπος διαμονής του ξενοδοχείου
- **hotelCountry,** η χώρα του ξενοδοχείου
- **hotelRegion,** η επαρχία που βρίσκεται του ξενοδοχείο
- **hotelCity,** η πολη του ξενοδοχείου
- **hotelLocation,** η τοποθεσία του ξενοδοχείου
- **userId,** ο χρήστης που εισήγαγε το ξενοδοχείο
- **hotelDescription,** η περιγραφή του ξενοδοχείου
- **hotelTotalRooms,** ο αριμός του συνόλου των δωματίων του ξενοδοχείου
- **hotelCharge,** η πολιτική ακυρώσεων του ξενοδοχείου
- **hotelAddBed,** η πολιτική των επιπρόσθετων κρεβατιών του ξενοδοχείου
- **hotelAddPet,** η πολιτική των κατοικιδίων
- **hotelAddService,** οι επιπρόσθετες υπηρεσίες που παρέχει το ξενοδοχείο
- **hotelAddInternet,** οι τύποι internet Που παρέχει το ξενοδοχείο

- **hotelAddCheckin,** η ώρα που δέχεται τα checkin το ξενοδοχείο
- **hotelAddCheckout,** η ώρα που δέχεται τα checkout το ξενοδοχείο
- **lang,** ο δεκαδικός αριθμός του γεωγραφικού μήκους του ξενοδοχείου
- **lng,** ο δεκαδικός αριθμός του γεωγραφικού πλάτους του ξενοδοχείου

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **addHotel** της **webservice hoteldbAdd** και θα της στείλει τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα xml έγγραφο κωδικοποιημένο σε Base64 format της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>33</id>
<name>New Hotel</name>
<stayingtype>Hotel</stayingtype>
<stars>5</stars>
<email>newemail@email.gr</email>
<phone>+2101234567</phone>
<country_name>Greece</country_name>
<region_name>Attica</region_name>
<city_name>Athens City</city_name>
<location>Centre</location>
<hoteladdress>Centre</hoteladdress>
<description>Hotel Description</description>
<totalrooms>33</totalrooms>
<hotelchain>New Hotel Chain</hotelchain>
<fax>+2101234568</fax>
<chargingpolicy>20/20/1</chargingpolicy>
<bedchild>Bed Policy</bedchild>
<pets>Pet Policy</pets>
<services>Services</services>
<internet>Wireless</internet>
<checkin>12:00</checkin>
<checkout>10:00</checkout>
<latt>33,345345</latt>
<lon>23,234234</lon>
</root>
```

Στη συνέχεια θα προσθεθεί το ξενοδοχείο στη βάση και η μέθοδος θα επιστρέψει ένα SOAP μήνυμα με ένα xml εγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

Οπου Id ο αύξων αριθμός που πήρε το ξενοδοχείο όταν προσθέθηκε στη βάση.

Procedure: Add Hotel Step 8



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγής ενός καινούργιου ξενοδοχείου στη βάση. Το όγδοο και τελευταίο βήμα περιλαμβάνει την εισαγωγή της κεντρικής φωτογραφίας του ξενοδοχείου.

Αποτελείται από

- ✓ Τον uploader ο οποίος είναι ένας **"rich:fileUpload"**
- ✓ Το κουμπί **"Finish"** το οποίο είναι ένα **"h:commandButton"**
- ✓ Το **tooltip** επεξήγησης το οποίο είναι τύπου **"rich:tooltip"**

Επεξήγηση Ρόλων

Ο uploader είναι υπεύθυνος να παίρνει μια φωτογραφία και μετα τις απαραίτητες διαδικασίες να την αποθηκεύει στον server και στη βάση για το συγκεκριμένο ξενοδοχείο.

Το κουμπί **"Finish"** είναι υπεύθυνο να ολοκληρώνει τη διαδικασία της εισαγωγής ξενοδοχείου και να επιστρέφει τον χρήστη στο κεντρικό μενού επιλογής.

Το **tooltip** είναι υπεύθυνο, με το που τοποθετήσει ο χρήστης τον κέρσορα πάνω, να εμφανίζει ένα πλαίσιο με πληροφορίες για τη λειτουργικότητα της διεπαφής.

Διαδικασία Upload Fotos

Ο uploader καλεί τη μέθοδο **"listenerAddHotel"** από το bean. Αυτή η μέθοδος παίρνει σαν όρισμα ένα αντικείμενο UploadEvent που στη περίπτωση μας είναι το event upload() του uploader. Είναι υπεύθυνη να διαβάσει το αρχείο που γίνεται upload και να ελέγξει στη συνέχεια αν υπάρχουν οι φακέλοι στο path Που πρεπει να καταχωρηθεί.

Στη συνέχεια θα επικοινωνήσει με τη μέθοδο **"copyFile",** αποστέλλοντας τη τιμή του προσωρινού αρχείου που γίνεται upload και το Path του αρχείου που πρέπει να δημιουργηθεί. Η

μέθοδος αυτή παίρνει το προσωρινό αρχείο και μεταφράζοντας το σε πίνακα των 1024bytes καταφλερνει τη μεταφορά του.

Μετά από μια επιτυχημένη προσθήκη θα καλεστεί η μέθοδος **“addFotoAddHotel”,** πέρνοντας σαν όρισμα το id του ξενοδοχείου και το url της φωτογραφίας με σκοπό να επικοινωνήσει με τη μέθοδο **addFotoAddHotel** της **webservice hoteldbAdd** και να της δώσει τις τιμές των μεταβλητών. Η επικοινωνία περιλαμβάνει ένα xml έγγραφο κωδικοποιημένο σε Base64 format της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<foto>://www.broker-lodge.com/hotelsData/HotelFotos/2/mainhoteloutputXBfoto.jpg</foto>
</root>
```

1.1.4 Edit Hotel

Procedure: Edit Hotel

Η διαδικασία αυτή περιλαμβάνει την αλλαγή / διόρθωση των πληροφοριών ενός ξενοδοχείου από τον χρήστη. Τα βήματα και η μεθοδολογία είναι ίδια με τη **Procedure: Add Hotel** οπότε δεν θα την αναλύσουμε περαιτέρω,

1.1.5 Hotelier Statistics

Panel: Statistics

Η διαδικασία αυτή περιλαμβάνει την αναζήτηση, προβολή, ανάλυση και πληρωμή των ποσών από τις κρατήσεις που έχουν γίνει στο σύστημα. Τα η μεθοδολογία είναι ίδια με τη διαδικασία **Statistics** του admin οπότε δεν θα την αναλύσουμε περαιτέρω,

1.3 Customer

Panel: Top Destinations



Λειτουργικότητα

Εμφανίζει τη διεπαφή προβολής των προτεινόμενων ξενοδοχείων.

Αποτελείται από

- ✓ Τον **πίνακα** με τα ξενοδοχεία ο οποίος είναι **"rich:dataTable"**

Ο πίνακας αποτελείται από:

- ✓ Την εικόνα του ξενοδοχείου η οποια είναι **“h:graphicImage”**
- ✓ Το link με το όνομα του ξενοδοχείου το οποίο είναι **“h:commandLink”**
- ✓ Την εικόνα με τα αστέρια του ξενοδοχείου η οποία είναι **“h:graphicImage”**
- ✓ Το link με την άμεση κράτηση δωματίου το οποίο είναι **“h:commandLink”**

Επεξήγηση Ρόλων

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τα προεινόμενα ξενοδοχεία.

Η **εικόνα** είναι υπεύθυνη να δείχνει τη κεντρική εικόνα του ξενοδοχείου.

Το **link** είναι υπεύθυνο να δείχνει το όνομα του ξενοδοχείου και να οδήγει το χρήστη στη κεντρικη σελίδα του ξενοδοχείου

Η **εικόνα** με τ’αστεριαείναι υπευθυνη να δείχνει την κατάλληλη προκαθορισμένη εικόνα ανάλογα τα’αστέρια του ξενοδοχείου.

Το **link** με την άμεση κράτηση είναι υπεύθυνο να οδηγεί τον χρήστη στη φόρμα άμεσης κράτησης δωματίου

Διαδικασία fillTopDestinations

Καλείται η μέθοδος **“fillTopDestination”** από το bean. Αυτή η μέθοδος δεν δέχεται ορίσματα. Και δεν στέλνει κάποιο ωφέλιμο μήνυμα. Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **“fillTopDestination”, της web service “hoteldbFill”,** με σκοπό να παραλάβει ένα μήνυμα με τα προτεινόμενα ξενοδοχεία που είναι καταχωρημένα στη βάση και συγκεκριμένα στον πίνακα **“topdestinations”.**

Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

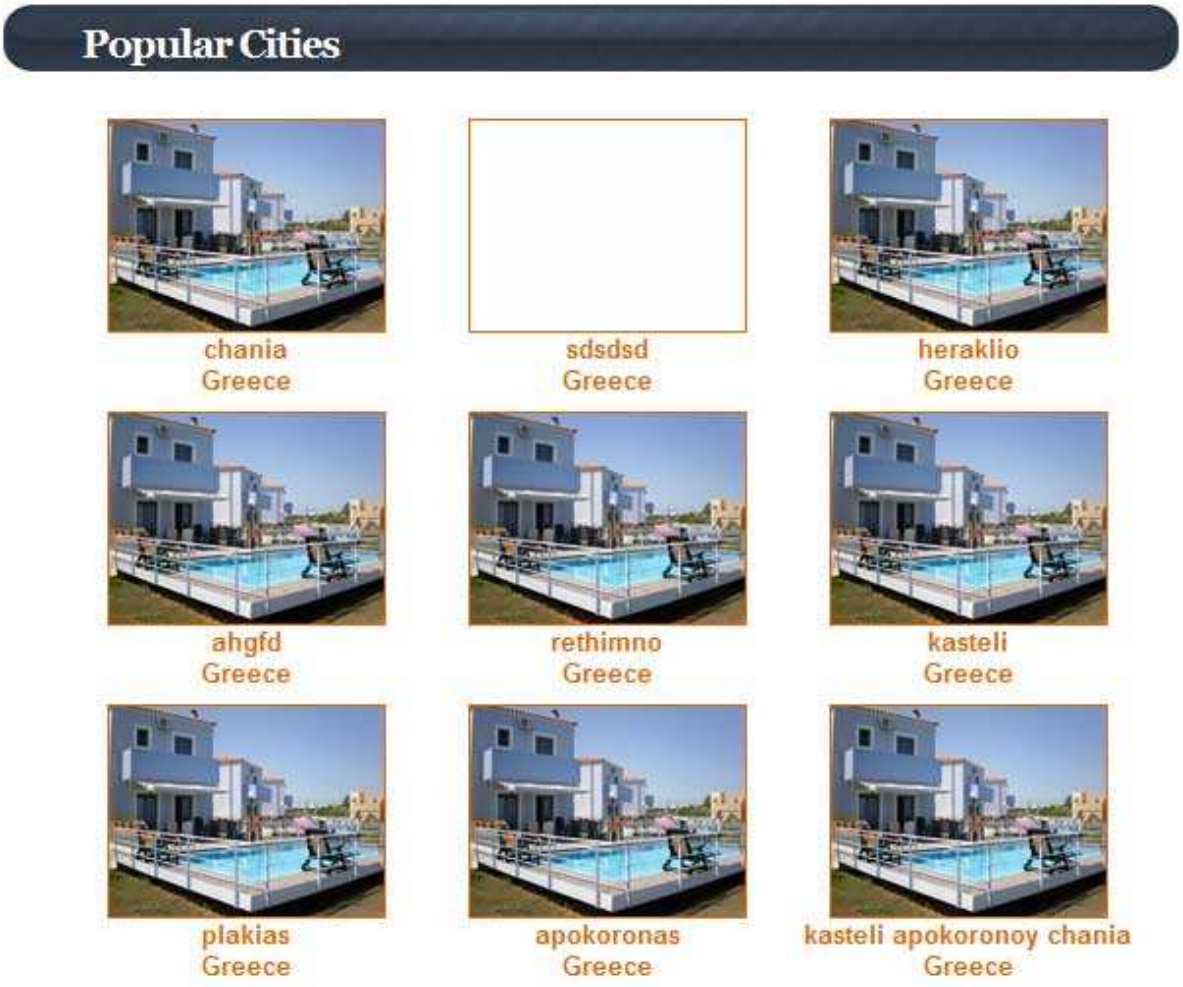
```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<name>Airport</name>
<id>23</id>
<city_name></city_name>
<country_name></country_name>
<stars></stars>
<foto></foto>
<suggestion></suggestion>
<sugname></sugname>
<city_name1></city_name1>
<country_name1></country_name1>
<countryID1></countryID1>
<foto1></foto1>
<id1></id1>
<relativepath></relativepath>
<serverpath></relativepath>
</root>
```

- **name –** Το ονομα του ξενοδοχειου
- **id -** Το id του ξενοδοχείου
- **city_name –** Το ονομα της πόλης του ξενοδοχείου
- **coyntry_name –** Το ονομα της χωρας του ξενοδοχείου
- **stars –** Τα αστερια του ξενοδοχείου

- **foto –** Το url της φωτογραφιας του ξενοδοχείου
- **relativepath –** Το relative path των φωτογραφιων
- **serverpath –** Το server path των φωτογραφιων

Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τα προτεινόμενα ξενοδοχεία που λάβαμε από το μήνυμα απάντησης.

Panel: Popular Cities



Λειτουργικότητα

Εμφανίζει τη διεπαφή προβολής των δημοφειλέστερων πόλεων του συστήματος.

Αποτελείται από

- ✓ Τον **πίνακα** με τα ξενοδοχεία ο οποίος είναι **“rich:dataTable”**

Ο πίνακας αποτελείται από:

- ✓ Την εικόνα της πόλης η οποια είναι **“h:graphicImage”**
- ✓ Το link με το όνομα της πόλης το οποίο είναι **“h:commandLink”**
- ✓ Το link με το όνομα της χώρας το οποίο είναι **“h:commandLink”**

Επεξήγηση Ρόλων

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τις δημοφειλέστερες πόλεις του συστηματος.

Η **εικόνα** είναι υπεύθυνη να δείχνει την φωτογραφία της πόλης.

Το **link** είναι υπεύθυνο να δείχνει το όνομα της πόλης και να οδήγει το χρήστη στη σελίδα αναζήτησης για ξενοδοχεία στη συγκεκριμένη πολη

Το **link** είναι υπεύθυνο να δείχνει το όνομα της χώρας και να οδήγει το χρήστη στη σελίδα αναζήτησης για ξενοδοχεία στη συγκεκριμένη χώρα

Διαδικασία fillTopCity

Καλείται η μέθοδος **“fillTopCity”** από το bean. Αυτή η μέθοδος δεν δέχεται ορίσματα. Και δεν στέλνει κάποιο ωφέλιμο μήνυμα. Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **“fillTopCity”, της web service “hoteldbFill”,** με σκοπό να παραλάβει ένα μήνυμα με τις δημοφειλέστερες πόλεις που είναι καταχωρημένες στη βάση**.**

Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>22</id>
<city_name></city_name>
<country_name></country_name>
<foto></foto>
<countryID></countryID>
</root>
```

- **id -** Το id της πόλης
- **city_name –** Το ονομα της πόλης
- **coyntry_name –** Το ονομα της χωρας
- **foto –** Το url της φωτογραφιας της πόλης
- **countryID –** Το id της χώρας

Στη συνέχεια θα ανανεωθούν τα περιεχόμενα του πίνακα με τις δημοφειλέστερες πόλεις που λάβαμε από το μήνυμα απάντησης.

Panel: Contact



Λειτουργικότητα

Εμφανίζει τη διεπαφή της αποστολής μηνύματος επικοινωνίας από τον πελάτη.

Αποτελείται από

- ✓ Το πεδίο εισαγωγης κειμενου **First Name** το οποίο είναι **“h:inputText”**
- ✓ Το πεδίο εισαγωγης κειμενου **Last Name** το οποίο είναι **“h:inputText”**
- ✓ Το πεδίο εισαγωγης κειμενου **Email** το οποίο είναι **“h:inputText”**
- ✓ Το πεδίο εισαγωγης κειμενου **Message** το οποίο είναι **“h:inputTextArea”**
- ✓ Το κουμπί **Send** το οποίο είναι **“a4j:commandButton”**
- ✓ Το κουμπί **Return** το οποίο είναι **“h:commandButton”**

Επεξήγηση Ρόλων

Το κουμπι **Send** είναι υπεύθυνο να παίρνει τα δεδομένα που εισήγαγε ο χρήστης στα πεδία κειμένου και να στέλνει το email επικοινωνιας

Το κουμπι **Return** είναι υπεύθυνο να οδηγει τον χρήστη στην κεντρικη σελιδα του site

Κουμπί Send

Καλείται η μέθοδος **“sendContactMail”** από το bean. Αυτή η μέθοδος δεν δέχεται ορίσματα. Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **“sendContactMail”, της web service “hoteldbOther”,** με σκοπό να παραλάβει ένα μήνυμα με τις δημοφειλέστερες πόλεις που είναι καταχωρημένες στη βάση**.** Στέλνει ένα xml της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<fname>FirstName</fname>
<lname>LastName</lname>
```

<email>someEmail</email>
<contactTopic>Topic</contactTopic>
<contactMessage>Message</contactMessage>
</root>

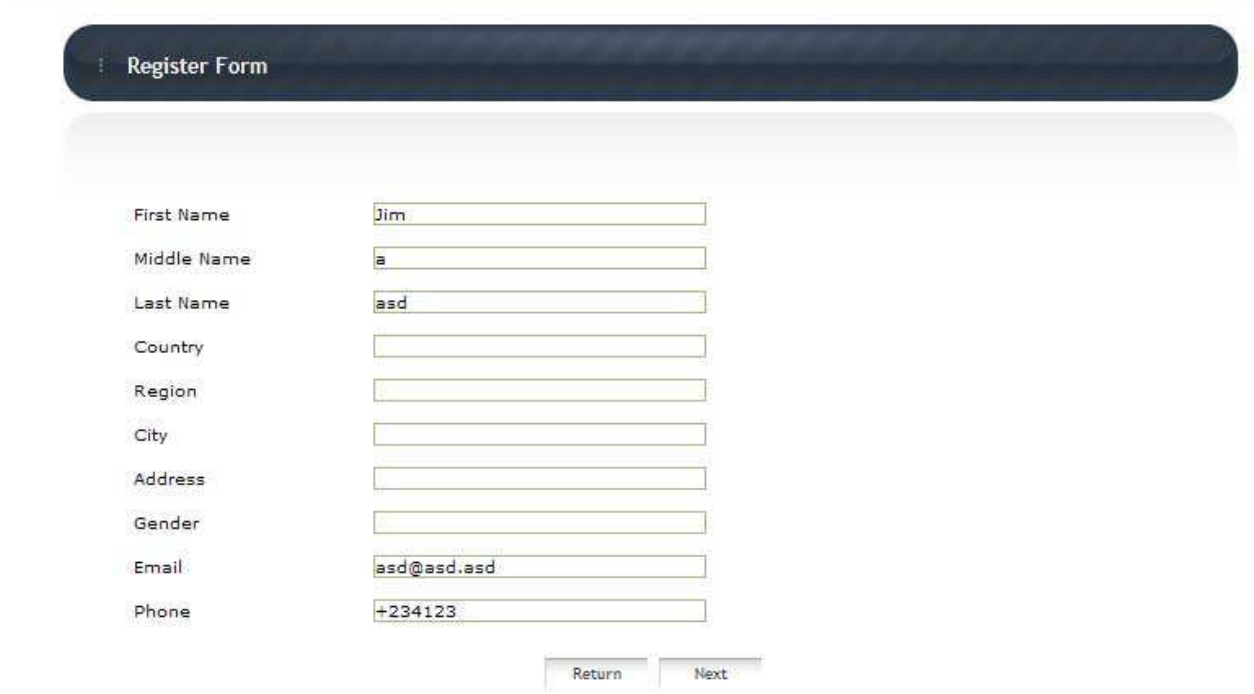
- **fname –** Το όνομα του χρήστη
- **lname -** Το επίθετο του χρήστη
- **email –** Το email του χρήστη
- **contactTopic –** Το θέμα του μηνυματος επικοινωνιας
- **contactMessage –** Το μήνυμα επικοινωνιας

Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

<?xml version="1.0" encoding="UTF-8"?>
<root>
<result>0</result>
</root>

- **result -** Ο κωδικός του απότελέσματος της αποστολής

Panel: Register Step1



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εγγραφής ενός χρήστη στο σύστημα.

Αποτελείται από

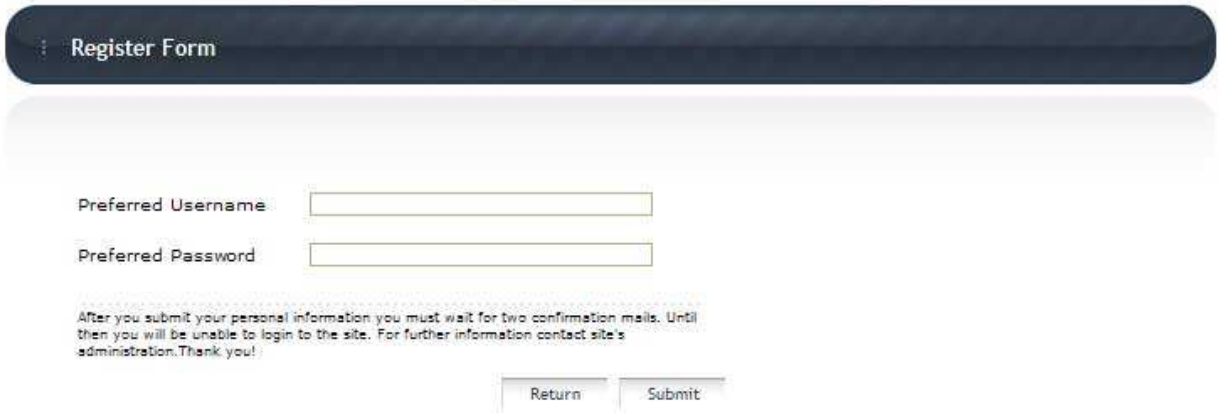
- ✓ Το πεδίο εισαγωγης κειμενου **First Name** το οποίο είναι **“h:inputText”**
- ✓ Το πεδίο εισαγωγης κειμενου **Middle Name** το οποίο είναι **“h:inputText”**
- ✓ Το πεδίο εισαγωγης κειμενου **Last Name** το οποίο είναι **“h:inputText”**
- ✓ Το πεδίο εισαγωγης κειμενου **Country** το οποίο είναι **“h:inputText”**
- ✓ Το πεδίο εισαγωγης κειμενου **Region** το οποίο είναι **“h:inputText”**
- ✓ Το πεδίο εισαγωγης κειμενου **City** το οποίο είναι **“h:inputText”**
- ✓ Το πεδίο εισαγωγης κειμενου **Address** το οποίο είναι **“h:inputText”**
- ✓ Το πεδίο εισαγωγης κειμενου **Gender** το οποίο είναι **“h:inputText”**
- ✓ Το πεδίο εισαγωγης κειμενου **Email** το οποίο είναι **“h:inputText”**
- ✓
- ✓ Το πεδίο εισαγωγης κειμενου **Phone** το οποίο είναι **“h:inputText”**
- ✓ Το κουμπί **Next** το οποίο είναι **“h:commandButton”**
- ✓ Το κουμπί **Return** το οποίο είναι **“h:commandButton”**

Επεξήγηση Ρόλων

Το κουμπι **Next** είναι υπεύθυνο να παίρνει τα δεδομένα που εισήγαγε ο χρήστης στα πεδία κειμένου και να προχωράει στο επόμενο βήμα της καταχώρησης.

Το κουμπι **Return** είναι υπεύθυνο να οδηγεί τον χρήστη στην κεντρικη σελιδα του site

Panel: Register Step2



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εγγραφής ενός χρήστη στο σύστημα.

Αποτελείται από

- ✓ Το πεδίο εισαγωγης κειμενου **Preferred UserName** το οποίο είναι **“h:inputText”**
- ✓ Το πεδίο εισαγωγης κειμενου **Preferred Password** το οποίο είναι **“h:inputText”**
- ✓ Το κουμπί **Submit** το οποίο είναι **“h:commandButton”**
- ✓ Το κουμπί **Return** το οποίο είναι **“h:commandButton”**

Επεξήγηση Ρόλων

Το κουμπι **Submit** είναι υπεύθυνο να παίρνει τα δεδομένα που εισήγαγε ο χρήστης στα πεδία κειμένου και να καταχώρει τον νέο χρήστη στο συστημα.

Το κουμπι **Return** είναι υπεύθυνο να οδηγει τον χρήστη στο πρώτο βήμα της διαδικασιας

Κουμπί Submit

Καλείται η μέθοδος **"addNewUser"** από το bean. Αυτή η μέθοδος δέχεται ορίσμα ένα ActionEvent που στη περιπτωση μας είναι το event click() του κουμπιου.
Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **"acceptUserAdminMail", της web service "hoteldbOther",** με σκοπό να ενημερώσει τον διαχειρηστη για την νεα αιτηση**.** Στέλνει ένα xml της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<fname>FirstName</fname>
<mname>M</mname>
<lname>LastName</lname>
<email>someEmail</email>
<country>Greece</country>
<region>Region</region>
<city>Chania</city>
<phone>+3403434</phone>
<address>Address</address>
<position>manager</position>
<department>sales</department>
<gender>male</gender>
<username>username</username>
<password>pass</password>
<md5username>asdadq32e3wd21e3wds</md5username>
<md5>asd23e23ddr23r23wdwr23er233</md5>
</root>
```

- **fname –** Το όνομα του χρήστη
- **mname –** Το μεσαιο ονομα του χρήστη
- **lname -** Το επίθετο του χρήστη
- **email –** Το email του χρήστη
- **country –** Η χωρα του χρηστη
- **region –** Η επαρχια του χρήστη
- **city –** Η πολη του χρηστη
- **phone –** Το τηλεφωνο του χρηστη
- **address –** Η διευθυνση του χρηστη
- **position –** Η θεση που καταλαμβανει στην εταιρια του
- **department –** Το τμημα της εταιριας στο οποιο υπηρετει
- **gender –** Το φυλο του χρηστη
- **username –** Το Username του χρηστη
- **password –** Ο κωδικος του χρηστη
- **md5Username –** Το hash με το username του χρηστη

- **md5 –** Το hash με το username και τον κωδικο του χρηστη

Στη συνέχεια ο χρήστης θα οδηγηθει στην αρχικη σελιδα.

Panel: Search



Λειτουργικότητα

Εμφανίζει τη διεπαφή της αναζήτησης ενός ξενοδοχείου ή περιοχης.

Αποτελείται από

- ✓ Το πεδίο εισαγωγης κειμενου το οποίο είναι **“h:inputText”**
- ✓ Το ημερολογιο **checkin** το οποίο είναι **“rich:calendar”**
- ✓ Το ημερολογιο **checkout** το οποίο είναι **“rich:calendar”**
- ✓ Τα κουμπια μοναδικης επιλογης **SearchBy** τα οποια είναι
- ✓ Το κουμπί **Search** το οποίο είναι **“h:commandLink”**
- ✓ Τα link επιλογης νομισματος τα οποια είναι **“a4j:commandLink”**
- ✓ Το κουτι επιλογης **No Dates Yet** το οποιο είναι **“h:selectBooleanCheckBox”**

Επεξήγηση Ρόλων

Το κουμπι **Search** είναι υπεύθυνο να παίρνει τις επιλογές του χρήστη και να πραγματοποιει την αναζήτηση.

Τα ημερολόγια δίνουν την επιλογη στο χρήστη να επιλέξει τις ημερομηνιες εναρξης και λήξης της διαμονης του.

Το κουτι επιλογής επιτρέπει στο χρήστη να μην συμπεριλάβει ημερομηνίες στην αναζήτηση του

Τα κουμπια μοναδικης επιλογης επιτρέπουν στο χρήστη να επιλέξει αναμεσα σε αναζητηση περιοχής ή ξενοδοχείου

Τα link αλλαγής νομίσματος επιτρέπουν στο χρήστη να εναλλασει το νόμισμα που θα βλεπει τις τιμές.

Κουμπί Search
Καλείται η μέθοδος **“searchDestinationAction”** από το bean. Αυτή η μέθοδος δεν δέχεται ορίσμα.
Είναι υπεύθυνη να διαβάσει τις επιλογές του χρήστη και ανάλογια για το ειδος της αναζήτησης θα καλέσει την αντιστοιχη μεθοδο.
Στη περιπτωση της αναζητησης ξενοδοχείου:
Θα επικοινωνήσει με τη μέθοδο **“searchHotel”, της web service “hoteldbOther”,** με σκοπό να πραγματοποιηθει η αναζητηση**.** Στέλνει ένα xml της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<by>Search</by>
</root>
```

- **by –** Το κείμενο της αναζητησης

Το μήνυμα που λαμβάνει είναι της μορφης:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country>Country</country>
<foto>Url</foto>
<city>City</city>
<hotelid>2</hotelid>
<name>Hotel Name</name>
<rate>23,23</rate>
<stars>5</stars>
</root>
```

- **country –** Η χωρα του ξενοδοχειου
- **foto –** Η φωτογραφια του ξενοδοχειου
- **city –** Η πολη του ξενοδοχειου
- **hotelid –** Το Id του ξενοδοχειου
- **name –** Το ονομα του ξενοδοχειου
- **rate –** Η φθηνοτερη τιμη για μια μερα ενός διαθεσιμου δωματιου του ξενοδοχειου
- **stars –** Τα αστερια του ξενοδοχειου

Στη συνεχεια ο χρήστης θα οδηγηθει στη σελία αποτελεσμάτων η οποία θα γεμίσει με τα αποτελεσματα των ξενοδοχείων.

Στη περιπτωση αναζήτησης περιοχής:

Θα επικοινωνήσει με τη μέθοδο **“search” ή “searchNoDates”, της web service “hoteldbOther”,** αναλόγως την επιλογή ημερομηνιών του χρήστη, με σκοπό να πραγματοποιηθει η αναζητηση για περιοχη**.** Στέλνει ένα xml της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<by>Search</by>
<sdate>2010-02-20</sdate>
</root>
```

- **by –** Το κείμενο της αναζητησης
- **sdate –** Η ημερομηνια εναρξης από τις επιλογες του χρηστη

Το μήνυμα που λαμβάνει είναι της μορφης:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<countries-id ></countries-id >
<countries></countries>
<countries-foto></countries-foto>
<regions-countryid></regions-countryid>
<regions-id></regions-id>
<regions-country></regions-country>
<egions-region></egions-region>
<egions-foto></egions-foto>
<regions-num></regions-num>
<countries-num></countries-num>
<cities-num></cities-num>
<loc-num></loc-num>
<cities-countryid></cities-countryid>
<cities-id></cities-id>
<cities-country></cities-country>
<cities-region></cities-region>
<cities-city></cities-city>
<cities-foto></cities-foto>
<loc-id></loc-id>
<loc-countryid></loc-countryid>
<loc-cityid></loc-cityid>
<loc-country></loc-country>
<loc-region></loc-region>
<loc-city></loc-city>
<loc-location></loc-location>
<loc-foto></loc-foto>
</root>
```

- **countries-id –** Το id της χώρας
- **countries –** Το όνομα της χώρας
- **countries-foto –** Η φωτογραφία της χώρας
- **regions-countryid –** Το id της χώρας της επαρχιας
- **regions-id –** Το id της επαρχιας
- **regions-country –** Το ονομα της χώρας της επαρχιας

- **egions-region –** Το ονομα της επαρχιας
- **egions-foto –** Η φωτογραφια της επαρχιας
- **regions-num –** Ο αριθμός των ξενοδοχείων της επαρχιας
- **countries-num –** Ο αριθμός των ξενοδοχείων της χωρας
- **cities-num –** Ο αριθμός των ξενοδοχείων της πόλης
- **cities-countryid –** Το Id της χωρας της πολης
- **cities-id –** Το Id της πόλης
- **cities-country –** Το ονομα της χωρας της πολης
- **cities-region –** Το ονομα της επαρχιας της πολης
- **cities-city –** Το ονομα της πολης
- **cities-foto -** Η φωτογραφια της πολης
- **loc-id –** Το id της τοποθεσιας
- **loc-countryid –** Το id της χωρας της τοποθεσιας
- **loc-cityid –** Το id της πολης της τοποθεσιας
- **loc-country –** Το ονομα της χωρας της τοποθεσιας
- **loc-region –** Το ονομα της επαρχιας της τοποθεσιας
- **loc-city –** Το ονομα της πολης της τοποθεσιας
- **loc-location –** Το ονομα της τοποθεσιας
- **loc-foto –** Η φωτογραφια της τοποθεσιας

Στη συνεχεια ο χρήστης θα οδηγηθει στη σελίδα αποτελεσμάτων των περιοχών η οποία θα γεμίσει με τα αποτελεσματα των χωρών, επαρχιών, πόλεων και τοποθεσιών που βρέθηκαν.

Panel: Search Results: Destination



Λειτουργικότητα

Εμφανίζει τη διεπαφή της ανάρτησης των αποτελεσμάτων μιας αναζήτησης για τοποθεσία.

Αποτελείται από

- ✓ Τον **πίνακα** με τα ξενοδοχεία ο οποίος είναι **"rich:dataTable"**

Ο πίνακας αποτελείται από:

- ✓ Την εικόνα της χώρας/επαρχίας/πόλης/τοποθεσίας η οποια είναι **"h:graphicImage"**
- ✓ Το link με το όνομα της χώρας/επαρχίας/πόλης/τοποθεσίας το οποίο είναι **"h:commandLink"**
- ✓ Το link με τον αριθμό των ξενοδοχείων κάτω από τον προορισμο το οποίο είναι **"h:commandLink"**

Επεξήγηση Ρόλων

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τα αποτελεσματα της διαδικασιας.

Η **εικόνα** είναι υπεύθυνη να δείχνει την φωτογραφία της χώρας /επαρχίας/πόλης/τοποθεσίας.

Το **link** είναι υπεύθυνο να δείχνει το όνομα της χώρας /επαρχίας/πόλης/τοποθεσίας και να οδήγει το χρήστη στη σελίδα αναζήτησης για ξενοδοχεία στο συγκεκριμένο προορισμο

Το **link** είναι υπεύθυνο να δείχνει τον αριθμό των ξενοδοχείων που βρίσκονται κατω από έναν προορισμο και να οδηγεί τον χρήστη στην εμφάνιση των ξενοδοχείων

Διαδικασία Αναζήτησης σε συγκεκριμένο προορισμό

Καλείται η μέθοδος **"afterMiddleSearch"** από το bean. Αυτή η μέθοδος δέχεται ορίσμα ένα ActionEvent. Είναι υπεύθυνη στο να επικοινωνήσει με τη μέθοδο **"searchHotelCountry" ή "searchHotelCountryNoDates", της web service "hoteldbOther",** με σκοπό να παραλάβει ένα μήνυμα με τα αποτελέσματα της αναζήτησης**.**

Στέλνει ένα xml της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<by>Search</by>
<start></start>
<end></end>
</root>
```

- **by –** Το κείμενο της αναζητησης
- **start –** Η ημερομηνία εναρξης
- **end –** Η ημερομηνια λήξης

Η απάντηση που λαμβάνει από την μέθοδο της webservice είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<avid>22</avid>
<city>Chania</city>
<country>Greece</country>
<foto>fotoUrl</foto>
<hotelId>2</hotelId>
<name>HotelName</name>
<rate>8</rate>
<stars>5</stars>
<price>45,45</price>
<tapa>10</tapa>
<desc>Description</desc>
</root>
```

- **avid -** Το id της διαθεσιμοτητας
- **city –** Το ονομα της πόλης
- **coyntry –** Το ονομα της χωρας
- **foto –** Το url της φωτογραφιας του ξενοδοχείου
- **hotelid –** Το id του ξενοδοχειου
- **name –** Το ονομα του ξενοδοχείου
- **rate –** Η βαθμολογια του ξενοδοχειου
- **stars –** Τα αστερια του ξενοδοχειου
- **price –** Η τιμη του φθηνοτερου δωματιου
- **tapa –** Το ποσοστο κερδους στη τιμη
- **desc –** Η περιγραφη του ξενοδοχειου

Στη συνέχεια ο χρηστης θα οδηγηθει στην επομενη σελιδα οπου θα εμφανιστουν τα αποτελεσματα αναζητησης που λάβαμε από το μήνυμα απάντησης.
Η διαδικασια για τις επαρχιες, πολεις και τοποθεσίες είναι ακριβως η ίδια.

Panel: Search Results: Hotels



Λειτουργικότητα

Εμφανίζει τη διεπαφή της ανάρτησης των αποτελεσμάτων μιας αναζήτησης για ξενοδοχεία.

Αποτελείται από

- ✓ Τον **πίνακα** με τα ξενοδοχεία ο οποίος είναι **“rich:dataTable”**

Ο πίνακας αποτελείται από:

- ✓ Την εικόνα του ξενοδοχείου η οποια είναι **“h:graphicImage”**
- ✓ Το link με το όνομα του ξενοδοχειου το οποίο είναι **“h:commandLink”**
- ✓ Το πεδιο κειμενου με τη βαθμολογια των επισκεπτων το οποιο είναι **“h:outputText”**
- ✓ Το πεδιο κειμενου με τη πολη και τη χωρα του ξενοδοχειου το οποιο είναι **“h:outputText”**
- ✓ Το πεδιο κειμενου με την φθηνοτερη τιμη του ξενοδοχειου το οποιο είναι **“h:outputText”**
- ✓ Το πεδιο κειμενου με τη περιγραφη του ξενοδοχειου το οποιο είναι **“h:outputText”**
- ✓ Το κουμπί γρηγορης πραγματοποιησης κράτησης το οποίο είναι **“h:commandLink”**
- ✓ Την εικόνα των αστεριων του ξενοδοχειου η οποια είναι **“h:graphicImage”**

Επεξήγηση Ρόλων

Ο **πίνακας** είναι υπεύθυνος στο να εμφανίζει τη λίστα με τα αποτελεσματα αναζητησης των ξενοδοχειων.

Η **εικόνα** είναι υπεύθυνη να δείχνει την φωτογραφία του ξενοδοχειου.

Το **link** με το ονομα είναι υπεύθυνο να δείχνει το όνομα του ξενοδοχειου και να οδήγει το χρήστη στη κεντρικη σελιδα του ξενοδοχειου

Το κουμπι **booknow** είναι υπευθυνο να οδηγει τον χρηστη στη πραγματοποιηση κράτησης.

Panel: Hotel Page

leto hotel in greece chalepa ★★★★★ Type:katagwgi

Hotel Chain:letoch chain, Rating:Not available yet , Hotel's location :mayrogianni,sparta city,sparta, Greece



leto hotel in greece chalepa has 333 rooms.

Overview

afd adas dsadsa dsadsadsaa dsadsasaddasads dasdsasaddsasaddsadsa dasdassdadsadsad dasdasdasasd sadsasdsa asddsad. dasdsasadd.asd sadsdadssad dsadsasdsasasda sadsadsaddsaads asdsadd dsaasasdas sadsadasdsdasdasadddsa dsadssdaassdadadsaasddsaasdsadsdaasdssaddsasdadsasdasdadsaasa dsadsadsa

Click to show more photos

Available Rooms

27-07-2010 28-07-2010 Change Dates

Room		Room Price		Number	Reserved	Price for 1 day(s)
Royal	Details	110.0 €	3	Click here	0	0.0 €
signle	Details	366.3 €	1	Click here	0	0.0 €
double	Details	478.5 €	2	Click here	0	0.0 €
Suita	Details	1100.0 €	2	Click here	0	0.0 €

Total Price: 0.0 €

Proceed with booking

Hotel Policies

Children and extra beds: +453665654sss+45 65654+453665654sfd
sffdsfdsfsdffdsfdshgyagsdffadfg
sgdegdsfgdsdgdfgdfsdggdfgdfdgfdgfsdgdgfdgfdfsgdgfdgfgdfgfdgfdgdfg
gfsdgfsdsgdfsdfsdfsdfsdfsdsf
fdsdsfdfsdfsfdsdsfdfsdfsfdsfdsfdsdfsdfsdfsfdsfdsfdsfdsfdsdfsfdds
fdssdfsfdfdsdfsdffdsfdsfdsdsffdsdsffdsdfsdfsdfsdffdssfdfdsfsdfdsfsdsfd
fdsfdsdsffsdfdsfsdfdsfdssdffsdsfddfsfsdfsdsfdfdsfsdfdsfdsfdsdfsdfsdfs
fdsfdsdfsdfsfdsfsdfdsfdsdfsdsffdsfdsfdsfdsfdsfdsfdsdfsdfsdsffdsdsffdsf
fdsfdsfdsdfsdfsdfsfdsfdsfdsfdsfdsfdsfdsfdsfdsfdsfdsfdsfdsfdsfdsfdsfdsfs
fsdfdsfdsdfsdfsdsffdsdfsfdsdfsdfsfdsfdsfdsfdsdfsfdsfdsdsfdfsfdsdfsfdsd
ddddddddddddddddddddddddddddd

Canceling policy: If cancelled you will be charged the full price of the reservation.
Pet policy: Allowed
Check in from: 01:00
Check out until: 22:00

Hotel Facilities

24-reception, assad'sd

Hotel Extra Services

asdsdadssdadsasdasadsadsdadsadsaa sdasdasdadsadsasdasdasdaasdaaaaaa sad
Ticket Service.dsfdfsfdsf

dsfdsfdsdfsffdsfdsfds

fsdfdsfdsfdsfdsfdsfdsdfsfdsfds

adsdasdas
dasdsasaddsaasddsaasdads

Hotel Distances

Airport	33.0 Km
asd'sd	44.0 Km

Hotel Internet Services

sfsafaffafdsfsfdffdssfdsfddsfs dfdsffds ddsffdssdfsdfsfdsdfdsfdsfdsfdsffs dffd dssdfsdfsfdsdfdsfdfsfdsdsfsdf fdsdfsds fdssdfdsfdfsdfsdfdsfdsfdsff dsdsfdfsds sdfdssddsfsdffds dsffd sdffdssdf dsfdfdsf fsadsadsadasddsadssadadsadasdasdas sadsasdadsasdasdsaddsaadsdasassad dsasdaasasdsda

Λειτουργικότητα

Εμφανίζει τη διεπαφή της σελιδας του ξενοδοχείου με ολες τις πληροφοριες, τις πολιτικες, τα δωματια, τις διαθεσιμοτητες και τις φωτογραφιες.

Αποτελείται από

- ✓ Την εικόνα του ξενοδοχείου η οποια είναι **"h:graphicImage"**
- ✓ Πεδιο κειμενου το οποιο είναι **"h:outputText"** για τις παρακατω τιμές:
- ✓ Το ονομα του ξενοδοχειου, τον τυπο του ξενοδοχειου, την αλυσιδα του ξενοδοχειου, τη βαθμολογια από τους χρηστες, την τοποθεσια του ξενοδοχειου, το συνολικο αριθμο των δωματιων.
- ✓ Την εικόνα των αστεριων του ξενοδοχειου η οποια είναι **"h:graphicImage"**
- ✓ Το πλαισιο πολλαπλης προβολης πληροφοριων το οποιο είναι **"rich:panelBar"** το οποιο εμφανιζει τη περιγραφη του ξενοδοχειου και τις φωτογραφιες του.
- ✓ Τον πίνακα με τις πληροφοριες διαθεσιμοτητας δωματιων ο οποίος είναι **"rich:dataTable"**
- ✓ Πεδια κειμένου **"h:outputText"** για τις πολιτικες των παιδιων και εξτρα κρεβατιων, ακυρώσεων κρατησεων, κατοικιδιων, checkin και check out
- ✓ Πεδια κειμενου **"h:outputText"** για τα facilities του ξενοδοχειου, τις εξτρα υπηρεσιες, τις από στασεις και τις υπηρεσιες internet.

Επεξήγηση Ρόλων

Τα **πλαίσια κειμένου** είναι υπευθυνα να απεικονιζουν τις πληροφοριες και τη θεματολογια των μεταβλητων που είναι συνδεδεμενα

Το **πλαισιο πολλαπλης προβολης** είναι υπεύθυνο να εμφανίζει στη μια καρτέλα τη περιγραφη του ξενοδοχειου και στη δευτερη καρτελα τη λίστα με τις φωτογραφιες του ξενοδοχειου.

Η **εικόνα** είναι υπεύθυνη να δείχνει την φωτογραφία του ξενοδοχειου.

Ο **πινακας** είναι υπευθυνος να δείχνει τις πληροφοριες διαθεσιμοτητας των δωματιων καθως και τις τιμές τους με τις διατροφες. Δινει την επιλογη στο χρηστη να επιλέξει τον αριθμο των δωματιων καθως και να αλλαξει τον προεπιλεγμενο τυπο διατροφης. Διαθετει κουμπι με το οποιο ο χρηστης όταν επιλεξει τα δωματια που θελει με ολες τις επιλογες μπορει να προχωρησει στη πραγματοποιηση της κράτησης.

Διαδικασια fillHotelPage

Καλείται η μέθοδος **fillHotelPage** από το bean η οποια δέχεται ένα ActionEvent σαν ορισμα. Ειναι υπευθυνη να επικοινωνησει με τη **fillHotelPage** της webservice **hoteldbFill** και να της στείλει το xml εγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id></id>
<start></start>
<end></end>
<exchange></exchange>
<exchangeicon></exchangeicon>
</root>
```

- **id –** Το id του ξενοδοχείου

- **start –** Η ημερομηνια εναρξης αναζητησης του χρηστη
- **end –** Η ημερομηνια λήξης αναζητησης του χρηστη
- **exchange –** Το νομισμα που εχει επιλεξει ο χρηστης
- **exchangeIcon –** Το εικονιδιο του νομισματος

Το μήνυμα που θα παραλάβει από τη μέθοδο θα είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<name>Hotel Name</name>
<chain>Chain</chain>
<country>Greece</country>
<city>Chania</city>
<rating>8</rating>
<star>5</star>
<totalrooms>123</totalrooms>
<description>Description</description>
<bedchild>Policy</bedchild>
<pets>Allowed</pets>
<room>room</room>
<tapa>10</tapa>
<diafora>diafora</diafora>
<price>34</price>
<hid>3</hid>
<avid>33</avid>
<food>Halfboard</food>
<persons>2</persons>
<release>policy</release>
<fake>fake</fake>
<hotelnear>nearhotels</hotelnear>
<hotelnearid>nearids</hotelnearid>
<hotelnearstars>nearstars</hotelnearstars>
<checkin>10:00</checkin>
<SuperTapa>20</SuperTapa>
<checkout>12:00</checkout>
<internet>wireless</internet>
<services>services</services>
<chargingpolicy>policy</chargingpolicy>
<foto>fotoUrl</foto>
<genfoto>mainfotourl</genfoto>
<stayingtype>hotel</stayingtype>
<address>address</address>
<lon>23.454545</lon>
<lat>34.232312</lat>
<facility>facility</facility>
<location>location</location>
<dist>4</dist>
<station>airport</station>
<roomFac>facilities</roomFac>
<roomFood>foods</roomFood>
```

<extradiscription>extras</extradiscription>
<extratitle>titles</extratitle>
<extraprice>prices</extraprice>
</root>

- **name –** Το ονομα του ξενοδοχειου
- **chain –** Η αλυσιδα του ξενοδοχειου
- **country –** Η χωρα του ξενοδοχειου
- **city –** Η πολη του ξενοδοχειου
- **rating –** Η βαθμολογια του ξενοδοχειου
- **star –** Τα αστερια του ξενοδοχειου
- **totalrooms –** Ο συνολικος αριθμος των δωματιων του ξενοδοχειου
- **description –** Η περιγραφη του ξενοδοχειου
- **bedchild –** Η πολιτικη των παραπανησιων κρεβατιων και παιδιων
- **pets –** Η πολιτικη των κατοικιδιων
- **room -**
- **tapa –** Το ποσοστο του κερδους
- **diafora –** Η διαφορα του κερδους από το κοστος
- **price –** Η τιμη της διαθεσιμοτητας
- **hid –** Το id του ξενοδοχειου
- **avid –** Το id της διαθεσιμοτητας
- **food –** Ο τυπος διατροφης της διαθεσιμοτητας
- **persons –** Ο αριθμος των ατομων της διαθεσιμοτητας
- **release –** Η release policy του ξενοδοχειου
- **hotelnear –** Τα ξενοδοχεια που βρισκονται κοντα στο επιλεγμενο
- **hotelnearid –** Τα id των κοντινων ξενοδοχειων
- **hotelnearstars –** Τα αστερια των κοντινων ξενοδοχειων
- **checkin –** Το checkin του ξενοδοχειου
- **superTapa –** Το ποσοστο κερδους από το ξενοδοχειο
- **checkout –** Το checkout του ξενοδοχειου
- **internet –** Οι τυποι internet που παρεχει το ξενοδοχειο
- **services –** Οι υπηρεσιες του ξενοδοχειου
- **chargingpolicy –** Η πολιτικη ακυρωσεων
- **foto –** Οι φωτογραφιες του ξενοδοχειου
- **genfoto –** Η κεντρικη φωτογραφια του ξενοδοχειου
- **stayingtype –** Ο τυπος διαμονης του ξενοδοχειου
- **address –** Η διευθυνση του ξενοδοχειου
- **lon –** Το γεωγραφικο πλατος της τοποθεσιας του ξενοδοχειου
- **lat –** Το γεωγραφικο μηκος της τοποθεσιας του ξενοδοχειου
- **facility –** Τα facilities του ξενοδοχειου
- **location –** Η τοποθεσια του ξενοδοχειου
- **dist –** Οι αποστασεις από τα σημεια ενδιαφεροντος
- **station –** Οι σταθμοι ενδιαφεροντος
- **roomFac –** Τα facilities των δωματιων
- **roomFood –** Οι τυποι διατροφης των δωματιων
- **extradiscription –** Η περιγραφη των εξτρα υπηρεσιων
- **extratitle –** Ο τιτλος των εξτρα υπηρεσιων

- **extraprice –** Η τιμη των εξτρα υπηρεσιων

Στη συνέχεια θα φορτωθουν τα στοιχεία της σελίδας με τις πληροφοριες που λαβαμε από το μηνυμα απαντησης.

Panel: Booking



Λειτουργικότητα

Εμφανίζει τη διεπαφή της πραγματοποιησης κράτησης και της εισαγωγης προσωπικων στοιχειων.

Αποτελείται από

- ✓ Το πεδιο εισαγωγης κειμενου **firstname** το οποιο είναι **“h:inputText”**
- ✓ Το πεδιο εισαγωγης κειμενου **lastname** το οποιο είναι **“ h:inputText ”**
- ✓ Το πεδιο εισαγωγης κειμενου **middlename** το οποιο είναι **“ h:inputText ”**
- ✓ Το πεδιο εισαγωγης κειμενου **email** το οποιο είναι **“ h:inputText ”**
- ✓ Το πεδιο εισαγωγης κειμενου **confirm email** το οποιο είναι **“ h:inputText ”**
- ✓ Το πεδιο εισαγωγης κειμενου **phone** το οποιο είναι **“ h:inputText ”**
- ✓ Το πεδιο εισαγωγης κειμενου **mobile** το οποιο είναι **“ h:inputText ”**
- ✓ Το πεδιο εισαγωγης κειμενου **flight number** το οποιο είναι **“ h:inputText ”**

- ✓ Το πεδιο εισαγωγης κειμενου **card number** το οποιο είναι **“ h:inputText ”**
- ✓ Το πεδιο εισαγωγης κειμενου **CVC** το οποιο είναι **“ h:inputText ”**
- ✓ Το μενου επιλογης **guest number** το οποιο είναι **“h:selectOneMenu”**
- ✓ Το μενου επιλογης **children number** το οποιο είναι **“h:selectOneMenu”**
- ✓ Το μενου επιλογης **card type** το οποιο είναι **“h:selectOneMenu”**
- ✓ Τα μενου επιλογης **expire date** τα οποια είναι **“h:selectOneMenu”**
- ✓ Τα πεδια κειμένου με τις βασικες επιλογες του χρηστη για τη κρατηση τα οποια είναι **“h:outputText”**
- ✓ Το tooltip επεξηγηση το οποιο είναι ένα **“rich:tooltip”**
- ✓ Το κουμπί ακυρωσης της διαδικασιας το οποίο είναι **“h:commandButton”**
- ✓ Το κουμπί κατοχυρωσης το οποίο είναι **“a4j:commandButton”**

Επεξήγηση Ρόλων

Τα πεδια εισαγωγης κειμένου επιτρέπουν στο χρήστη να εισαγει τα προσωπικα του στοιχεια τα οποια είναι απαραιτητα για την συνεχεια της διαδικασιας της κρατησης.

Τα πειδια κειμενου είναι υπευθυνα να προβαλουν τις επιλογες που εχει κανει ο χρηστης στο προηγουμενο βημα.

Το κουμπι **cancel** είναι υπευθυνο να ακυρωνει τη διαδικασια της κρατησης και να επιστρεφει τον χρηστη στην προηγουμενη σελιδα

Το κουμπι **Proceed** είναι υπευθυνο να οδηγει τον χρηστη στην ολοκληρωση της διαδικασιας

Κουπι Proceed

Καλειται η μεθοδος **sendVaucherMail** από το bean η οποια δεν δεχεται κάποιο όρισμα. Είναι υπευθυνη να διαβάσει ολες τι μεταβλητες με τις επιλογες του χρηστη για τη κρατηση και να τις δομησει σε ένα xml εγγραφο για να το στείλει στη μέθοδο **sendVaucherMail** της webservice **hoteldbOther.** Το xml είναι της μορφης:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<hid></hid>
<fname></fname>
<mname></mname>
<lname></lname>
<email></email>
<hotelname></hotelname>
<hoteladdress></hoteladdress>
<hotelcity></hotelcity>
<hotelcountry></hotelcountry>
<hotelavID></hotelavID>
<exp></exp>
<card></card>
<cvc></cvc>
<roomNum></roomNum>
<extra></extra>
<bookfoodReserve></bookfoodReserve>
<phone></phone>
<adProfit></adProfit>
<bookdatestart></bookdatestart>
```

<hotelcheckin></hotelcheckin>
<bookdateend></bookdateend>
<hotelcheckout></hotelcheckout>
<bookrooms></bookrooms>
<bookprice></bookprice>
<bookfood></bookfood>
<bookcanceldate></bookcanceldate>
<bookcancelprice></bookcancelprice>
<hotelcharge></hotelcharge>
<hotelbeds></hotelbeds>
<hotelinternet></hotelinternet>
<hotelpets></hotelpets>
<mobilephone></mobilephone>
<flightinfo></flightinfo>
<release></release>
<numberOfPersons></numberOfPersons>
<personsAges></personsAges>
</root>

- **hid –** Το id του ξενοδοχειου
- **fname -** Το ονομα του πελατη
- **mname** - Το μεσαιο ονομα του πελατη
- **lname** - Το επιθετο το πελατη
- **email** - Το email του πελατη
- **hotelname** - Το ονομα του ξενοδοχειου
- **hoteladdress** - Η διευθυνση του ξενοδοχειου
- **hotelcity** - Η πολη του ξενοδοχειου
- **hotelcountry** - Η χωρα του ξενοδοχειου
- **hotelavID** - Το id της διαθεσιμοτητας που επελεξε να κανει κρατησης ο πελατης
- **exp** - Η ημερομηνια ληξης της καρτας του πελατη
- **card** - Ο αριθμος της καρτας του πελατη
- **cvc** - Ο αριθμος ασφαλειας της καρτας
- **roomNum** - Το πληθος των δωματιων της κρατησης για κάθε τυπο
- **extra** - Οι εξτρα υπηρεσιες που επελεξε ο πελατης
- **bookfoodReserve** - Ο τυπος διατροφης που επελεξε ο πελατης
- **phone** - Ο αριθμος τηλεφωνου του πελατη
- **adProfit** - Το ποσοστο κερδους από τη κρατηση
- **bookdatestart** - Η ημερομηνια εναξης της κρατησης
- **hotelcheckin** - Η ωρα που δεχεται το ξενοδοχειο checkin
- **bookdateend** - Η ημερομηνια ληξης της κρατησης
- **hotelcheckout** - Η ωρα που δεχεται το ξενοδοχειο checkout
- **bookrooms** - Τα δωματια που κράτησε ο πελατης
- **bookprice** - Η τιμη της κρατησης
- **bookfood** - Η επιλεγμενη διατροφη
- **bookcanceldate** - Η ημερομηνια που μπορει να ακυρωθει η κρατηση
- **bookcancelprice** - Η τιμη που θα πληρωσει ο πελατης σε περιπτωση ακυρωσης
- **hotelcharge** - Η πολιτικη ακύρωσης
- **hotelbeds** - Η πολιτικη εξτρα κρεβατιων

- **hotelinternet** - Η πολιτικη internet
- **hotelpets** - Η πολιτικη κατοικιδιων
- **mobilephone** - Το κινητο του πελατη
- **flightinfo** - Οι πληροφοριες πτησης του πελατη
- **release** - Η release policy του ξενοδοχειου
- **numberOfPersons** - Ο αριθμος των ατομων της κρατησης
- **personsAges –** Οι ηλικιες των ατομων

Στη συνέχεια θα προχωρήσει η διαδικασία της κράτησης και ο πελάτης θα ενημερώθει για το αποτέλεσμα της διαδικασίας.

Panel: My Reservation



Λειτουργικότητα

Εμφανίζει τη διεπαφή της εισαγωγης των στοιχειων της κρατησης για την συνεχεια στη σελιδα της κράτησης και της εμφανισης των πληροφοριων.

Αποτελείται από

- ✓ Το πεδιο εισαγωγης κειμενου **customer ID** το οποιο είναι **“h:inputText”**
- ✓ Το πεδιο εισαγωγης κειμενου **Booking ID** το οποιο είναι **“ h:inputText ”**
- ✓ Το tooltip επεξηγηση το οποιο είναι ένα **“rich:tooltip”**
- ✓ Το κουμπί ακυρωσης της διαδικασιας το οποίο είναι **“h:commandLink”**
- ✓ Το κουμπί κατοχυρωσης το οποίο είναι **“a4j:commandButton”**

Επεξήγηση Ρόλων

Τα πεδια εισαγωγης κειμένου επιτρέπουν στο χρήστη να εισαγει τα στοιχεια της κρατησης. Το customerID και το bookingID Που εχει λάβει στο voucher του.

Το link **back** είναι υπευθυνο να ακυρωνει τη διαδικασια και να επιστρεφει τον χρηστη στην αρχικη σελιδα

Το κουμπι **Proceed** είναι υπευθυνο να ελεγχει τα στοιχεια που εβαλε ο χρηστης και να τον οδηγει στην σελιδα της κρατησης.

Κουπι Proceed

Καλειται η μεθοδος **amendment** από το bean η οποια δεν δεχεται κάποιο όρισμα. Είναι υπευθυνη να διαβάσει ολες τι μεταβλητες με τις επιλογες του χρηστη και να τις δομησει σε ένα xml εγγραφο για να το στείλει στη μέθοδο **amendment** της webservice **hoteldbOther.** Το xml είναι της μορφης:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<refID>refid</refID>
<bookID>bookid</bookID>
</root>
```

- **refID –** Το id της κρατησης
- **bookID -** Το id του πελατη

Στη συνέχεια εφόσον έχει εισάγει σωστά στοιχεία θα λάβει ένα μήνυμα απάντησης με τα στοιχεια της κράτησης. Το xml είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<fname>refid</fname>
<mname>bookid</mname>
<lname></lname>
<date_start></date_start>
<date_end></date_end>
<email></email>
<hotel></hotel>
<days></days>
<hotelprofit></hotelprofit>
<cardtype></cardtype>
<hotelfood></hotelfood>
<room></room>
<roomnum></roomnum>
<cancel></cancel>
</root>
```

- **fname –** Το ονομα του πελατη
- **mname -** Το μεσαιο ονομα του πελατη
- **lname –** Το επιθετο του πελατη
- **date_start –** Η ημερομηνια εναρξης της κρατησης
- **date_end –** Η ημερομηνια ληξης της κρατησης
- **email –** Το email του πελατη
- **hotel –** Το ονομα του ξενοδοχειου
- **days –** Οι συνολικες μερες της κρατησης
- **hotelprofit –** Το ποσοστο κερδους από τη κρατηση

- **cardtype –** Ο τυπος της πιστωτικης καρτας
- **hotelfood –** Ο τυπος διατροφης της κρατησης
- **room –** Οι τυποι δωματιων της κρατησης
- **roomnum –** Ο αριθμος των δωματιων από κάθε τυπο
- **cancel –** Η πολιτικη ακυρωσης της κρατησης

Στη συνέχεια ο χρηστης θα οδηγηθει στη σελιδα της κράτησης η οποια θα γεμίσει με τα στοιχεια που λάβαμε από το μηνυμα απαντησης.

Panel: My Reservation Page



Λειτουργικότητα

Εμφανίζει τη διεπαφή της σελιδας της κράτησης και της εμφανισης των πληροφοριων της. Επισης ο χρηστης εχει τη δυνατοτητα να αλλαξει τα στοιχεια της κρατησης του η να την ακυρωσει.

Αποτελείται από

- ✓ Τα πεδια κειμένου που εμφανιζουν τις πληροφοριες της κρατησης τα οποια είναι **“h:outputText”**
- ✓ Τον πινακα με τα δωμάτια της κρατησης ο οποιος είναι **“rich:dataTable”**
- ✓ Το tooltip επεξηγηση το οποιο είναι ένα **“rich:tooltip”**
- ✓ Το κουμπί ακυρωσης της κρατησης το οποίο είναι **“a4j:commanButton”**
- ✓ Το link αποσυνδεσης από τη σελιδα της κρατησης το οποίο είναι **“h:commandLink”**

Επεξήγηση Ρόλων

Τα πεδια κειμένου είναι υπευθυνα να δείχνουν τις πληροφοριες της κρατησης του πελατη.

Το link **back** είναι υπευθυνο να ακυρωνει τη διαδικασια και να επιστρεφει τον χρηστη στην αρχικη σελιδα

Το κουμπι **Cancel Reservation** είναι υπευθυνο να ακυρωνει την συγκεκριμενη κρατηση του πελατη.
Το κουμπι **Edit** είναι υπευθυνο να οδηγει τον χρηστη στο να αλλάξει τις πληροφοριες για το συγκεκριμενο δωματιο

Κουπι Cancel Reservation

Καλειται η μεθοδος **cancelReservation** από το bean η οποια δεν δεχεται κάποιο όρισμα. Είναι υπευθυνη να στειλει ένα xml μηνυμα με το id της κρατησης στη μέθοδο **cancelReservation** της webservice **hoteldbOther.** Το xml είναι της μορφης:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<merchREF>refid</merchREF>
</root>
```

- **merchREF –** Το id της κρατησης

Στη συνέχεια θα ακυρωθει η κράτηση και θα υπολογιστούν τα ακυρώτικα κόστη. Η απάντηση που θα λάβει από τη μέθοδο θα περιέχει ένα xml μηνυμα με το αποτελεσμα της διαδικασιας. Το xml είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<errorCode>0</errorCode>
<errorMessage>Message</errorMessage>
</root>
```

- **errorCode –** Ο κωδικος του αποτελεσματος της διαδικασιας
- **errorMessage -** Το μηνυμα απαντησης του αποτελεσματος

Στη συνέχεια ο χρηστης θα ενημερωθει για το αποτελεσμα της διαδικασιας από τα στοιχεια που λαβαμε από το μηνυμα απαντησης.

Κεφάλαιο 2 - Web Services

2.1 WebService: hoteldbAdd

2.1.1 Μέθοδος: addCountry

Είναι υπεύθυνη για την εισαγωγή μιας χώρας στη βάση δεδομένων και συγκεκριμένα στον πίνακα **country**. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
</root>
```

Όπου **country_name** το όνομα της χώρας που θέλει ο χρήστης να προστεθεί στη βάση.
Στη συνέχεια επικοινωνεί με τη stored procedure **"insertcountry"** της βάσης δεδομένων και της δίνει σαν όρισμα την τιμή του country_name πεδίου. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.2 Μέθοδος: editCountry

Είναι υπεύθυνη για την διόρθωση του ονόματος μιας χώρας στη βάση δεδομένων. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>NewValue</country_name>
<countryID>8</countryID>
</root>
```

Όπου **country_name** το όνομα της χώρας την οποία διορθώνουμε και **countryID** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων που αντιστοιχεί στην χώρα.
Στη συνέχεια επικοινωνεί με τη stored procedure **"editcountry"** της βάσης δεδομένων και της δίνει σαν όρισμα την τιμή των **country_name** και **countryID** πεδίων. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.3 Μέθοδος: addRegion

Είναι υπεύθυνη για την εισαγωγή μιας επαρχίας στη βάση δεδομένων και συγκεκριμένα στον πίνακα **region**. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<region_name>Crete</region_name>
</root>
```

Όπου **country_name** το όνομα της χώρας κάτω από την οποία θα προστεθεί η επαρχία και **region_name** το όνομα της επαρχίας προς προσθήκη.
Στη συνέχεια επικοινωνεί με τη stored procedure **“insertregion”** της βάσης δεδομένων και της δίνει σαν όρισμα την τιμή των **country_name** και **region_name** πεδίων. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.4 Μέθοδος: editRegion

Είναι υπεύθυνη για την διόρθωση του ονόματος μιας επαρχίας στη βάση δεδομένων. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<region_name>Creta</region_name>
<regionID>60</regionID>
</root>
```

Όπου **country_name** το όνομα της χώρας της οποίας την επαρχία διορθώνουμε, **region_name** το καινούργιο όνομα που θα πάρει η επαρχία μετά την διόρθωση και **regionID** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων που αντιστοιχεί στην επαρχία.
Στη συνέχεια επικοινωνεί με τη stored procedure **“editregion”** της βάσης δεδομένων και της δίνει σαν όρισμα την τιμή των **country_name, region_name** και **regionID** πεδίων. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.5 Μέθοδος: addCity

Είναι υπεύθυνη για την εισαγωγή μιας πόλης στη βάση δεδομένων και συγκεκριμένα στον πίνακα **city**. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
```

```
<country_name>Greece</country_name>
<region_name>Crete</region_name>
<city_name>Chania</city_name>
</root>
```

Όπου **country_name** το όνομα της χώρας, **region_name** το όνομα της επαρχίας κάτω από την οποία θα προστεθεί η πόλη και **city_name** το όνομα της πόλης προς προσθήκη.
Στη συνέχεια επικοινωνεί με τη stored procedure **“insertcity”** της βάσης δεδομένων και της δίνει σαν όρισμα την τιμή των **country_name, region_name** και **city_name** πεδίων. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.6 Μέθοδος: editCity

Είναι υπεύθυνη για την διόρθωση του ονόματος μιας πόλης στη βάση δεδομένων. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<city_name>Chania</city _name>
<cityID>66</cityID>
</root>
```

Όπου **city_name** το καινούργιο όνομα που θα πάρει η πόλη μετά την διόρθωση και **cityID** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων που αντιστοιχεί στην πόλη.
Στη συνέχεια επικοινωνεί με τη stored procedure **“editcity”** της βάσης δεδομένων και της δίνει σαν όρισμα την τιμή των **city_name** και **cityID** πεδίων. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.7 Μέθοδος: addLocation

Είναι υπεύθυνη για την εισαγωγή μιας τοποθεσίας στη βάση δεδομένων και συγκεκριμένα στον πίνακα **location**. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<region_name>Crete</region_name>
<city_name>Chania</city_name>
<location>Akrotiri</location>
</root>
```

Όπου **country_name** το όνομα της χώρας, **region_name** το όνομα της επαρχίας, **city_name** το όνομα της πόλη κάτω από την οποία θα προστεθεί η τοποθεσία και **location** το όνομα της τοποθεσίας προς προσθήκη.

Στη συνέχεια επικοινωνεί με τη stored procedure **“insertlocation”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές των πεδίων που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.8 Μέθοδος: editLocation

Είναι υπεύθυνη για την διόρθωση του ονόματος μιας τοποθεσίας στη βάση δεδομένων. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<location>Akrotiri</location>
<locationID>76</locationID>
</root>
```

Όπου **location** το καινούργιο όνομα που θα πάρει η τοποθεσία μετά την διόρθωση και **locationID** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων που αντιστοιχεί στην τοποθεσία.
Στη συνέχεια επικοινωνεί με τη stored procedure **“editlocation”** της βάσης δεδομένων και της δίνει σαν όρισμα την τιμή των **location** και **locationID** πεδίων. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.9 Μέθοδος: addCardType

Είναι υπεύθυνη για την εισαγωγή ενός τύπου πιστωτικής κάρτας στη βάση δεδομένων και συγκεκριμένα στον πίνακα **cardtype**. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<cardType>Visa</cardType>
</root>
```

Όπου **cardType** το όνομα του τύπου κάρτας που θέλει ο χρήστης να προστεθεί στη βάση. Στη συνέχεια επικοινωνεί με τη stored procedure **“insertcardtype”** της βάσης δεδομένων και της δίνει σαν όρισμα την τιμή του **cardType** πεδίου. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.10 Μέθοδος: editCardType

Είναι υπεύθυνη για την διόρθωση του ονόματος ενός τύπου πιστωτικής κάρτας στη βάση δεδομένων. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<cardType>Visa</cardType>
<cardTypeID>1</cardTypeID>
</root>
```

Όπου **cardType** το όνομα του τύπου τον οποίο διορθώνουμε και **cardTypeID** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων που αντιστοιχεί στον τύπο.
Στη συνέχεια επικοινωνεί με τη stored procedure **"editcardtype"** της βάσης δεδομένων και της δίνει σαν όρισμα την τιμή των **cardType** και **cardTypeID** πεδίων. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.11 Μέθοδος: addExchangeType

Είναι υπεύθυνη για την εισαγωγή ενός τύπου νομίσματος για μια συγκεκριμένη χώρα στη βάση δεδομένων και συγκεκριμένα στον πίνακα **exchangetype**. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<currency>Euro</currency>
</root>
```

Όπου **country_name** το όνομα της χώρας και **currency** το νόμισμα που θέλει ο χρήστης να προστεθεί στη βάση.
Στη συνέχεια επικοινωνεί με τη stored procedure **"insertexchange"** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές των πεδίων που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.12 Μέθοδος: editExchangeType

Είναι υπεύθυνη για την διόρθωση του ονόματος νομίσματος στη βάση δεδομένων. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<currency>Visa</currency>
<currencyID>1</currencyID>
</root>
```

Όπου **currency** το όνομα του νομίσματος το οποίο διορθώνουμε και **currnecyID** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων που αντιστοιχεί στο νόμισμα.
Στη συνέχεια επικοινωνεί με τη stored procedure **“editexchange”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές των πεδίων που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.13 Μέθοδος: addDistanceStation

Είναι υπεύθυνη για την εισαγωγή ενός σημείου ενδιαφέροντος στη βάση δεδομένων και συγκεκριμένα στον πίνακα **distancefrom**. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<station>Airport</station>
</root>
```

Όπου **station** το όνομα του σημείου ενδιαφέρνοτνος που θέλει ο χρήστης να προστεθεί στη βάση.
Στη συνέχεια επικοινωνεί με τη stored procedure **“insertdistancefrom”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές των πεδίων που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.14 Μέθοδος: editDistanceStation

Είναι υπεύθυνη για την διόρθωση του ονόματος ενός σημείου ενδιαφέρνοντος από τη βάση δεδομένων. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<station>Visa</station>
<stationID>1</stationID>
</root>
```

Όπου **station** το νεο όνομα του σημείου το οποίο διορθώνουμε και **currnecyID** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων που αντιστοιχεί στο σημείο ενδιαφέροντος.
Στη συνέχεια επικοινωνεί με τη stored procedure **“editdistancefrom”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές των πεδίων που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.15 Μέθοδος: addHotelNum

Είναι υπεύθυνη για την εισαγωγή του αριθμού των μέγιστων ξενοδοχείων που μπορει να βάλει ένας χρήστης στη βάση δεδομένων και συγκεκριμένα στον πίνακα **userhotel**. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>3</id>
<forward>5</forward>
</root>
```

Όπου **id** ο αύξων αριθμός που αντιστοιχεί στον χρήστη και **forward** ο αριθμός του περιορισμού των ξενοδοχείων.
Στη συνέχεια επικοινωνεί με τη βάση δεδομένων και αλλάζει τον αριθμό των ξενοδοχείων για τον συγκεκριμένο χρήστη εκτελόντας το αντίστοιχο ερώτημα. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.16 Μέθοδος: addMoney

Είναι υπεύθυνη για την εισαγωγή της προκαταβολής για έναν χρήστη στη βάση δεδομένων και συγκεκριμένα στον πίνακα **userhotel**. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>3</id>
<forward>20</forward>
</root>
```

Όπου **id** ο αύξων αριθμός που αντιστοιχεί στον χρήστη και **forward** ο αριθμός του ποσοστού.
Στη συνέχεια επικοινωνεί με τη βάση δεδομένων και αλλάζει τον αριθμό του ποσοστού για τον συγκεκριμένο χρήστη εκτελόντας το αντίστοιχο ερώτημα. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.17 Μέθοδος: addUserNameRestrCountry

Είναι υπεύθυνη για την εισαγωγή περιορισμού τοποθεσίας για έναν χρήστη στη βάση δεδομένων και συγκεκριμένα στον πίνακα **userhotel**. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country>Greece</country>
<md5username>7694f4a66316e53c8cdd9d9954bd611d</md5username>
</root>
```

Όπου **country** η χώρα που θα περιοριστεί ο χρήστης και **md5username** το συνθηματικο του χρήστη σε κωδικοποιημένη MD5 Hash μορφή.
Στη συνέχεια επικοινωνεί με τη stored procedure **insertrestrictioncountry2** και τις δίνει τις τιμές των πεδίων που έλαβε. Εφόσον η καταχώρηση ολοκληρωθεί χωρίς σφάλματα επικοινωνεί με τη stored procedure **selectadminmail** για να πάρει από τη βάση τα στοιχεία και τους κωδικούς του admin έτσι ώστε να στείλει mail στον χρήστη ότι μπορεί πλέον να εισάγει ξενοδοχείο. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.18 Μέθοδος: addUserNameRestrRegion

Είναι υπεύθυνη για την εισαγωγή περιορισμού τοποθεσίας για έναν χρήστη στη βάση δεδομένων και συγκεκριμένα στον πίνακα **userhotel**. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country>Greece</country>
<region>cretee</region>
<md5username>7694f4a66316e53c8cdd9d9954bd611d</md5username>
</root>
```

Όπου **country** η χώρα, **region** η επαρχία που θα περιοριστεί ο χρήστης και **md5username** το συνθηματικο του χρήστη σε κωδικοποιημένη MD5 Hash μορφή.
Στη συνέχεια επικοινωνεί με τις stored procedure **insertrestrictionregion1, insertrestrictionregion2, insertrestrictionregion3** και καταχωρεί τον περιορισμό για τον χρήστη. Εφόσον η καταχώρηση ολοκληρωθεί χωρίς σφάλματα επικοινωνεί με τη stored procedure **selectadminmail** για να πάρει από τη βάση τα στοιχεία και τους κωδικούς του admin έτσι ώστε να στείλει mail στον χρήστη ότι μπορεί πλέον να εισάγει ξενοδοχείο. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.19 Μέθοδος: addUserNameRestrCity

Είναι υπεύθυνη για την εισαγωγή περιορισμού τοποθεσίας για έναν χρήστη στη βάση δεδομένων και συγκεκριμένα στον πίνακα **userhotel**. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country>Greece</country>
<city>chania</city>
<region>cretee</region>
<md5username>7694f4a66316e53c8cdd9d9954bd611d</md5username>
```

</root>

Όπου **country** η χώρα, **region** η επαρχία, **city** η πόλη που θα περιοριστεί ο χρήστης και **md5username** το συνθηματικο του χρήστη σε κωδικοποιημένη MD5 Hash μορφή.
Στη συνέχεια επικοινωνεί με τις stored procedure **insertrestrictioncity1, insertrestrictioncity2, insertrestrictioncity3** και καταχωρεί τον περιορισμό για τον χρήστη. Εφόσον η καταχώρηση ολοκληρωθεί χωρίς σφάλματα επικοινωνεί με τη stored procedure **selectadminmail** για να πάρει από τη βάση τα στοιχεία και τους κωδικούς του admin έτσι ώστε να στείλει mail στον χρήστη ότι μπορεί πλέον να εισάγει ξενοδοχείο. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.20 Μέθοδος: addStayingType

Είναι υπεύθυνη για την εισαγωγή ενός τύπου ξενοδοχείου στη βάση δεδομένων και συγκεκριμένα στον πίνακα **stayingtype**. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<stayingtype>Hotel</stayingtype>
</root>
```

Όπου **stayingtype** το όνομα του τύπου ξενοδοχείου που θέλει ο χρήστης να προστεθεί στη βάση.
Στη συνέχεια επικοινωνεί με τη stored procedure **“insertstayingtype”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές των πεδίων που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.21 Μέθοδος: editStayingType

Είναι υπεύθυνη για την διόρθωση του ονόματος ενός τύπου ξενοδοχείου από τη βάση δεδομένων. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<stayingtype>Hotel</stayingtype>
<stayingtypeID>1</stayingtypeID>
</root>
```

Όπου **stayingtype** το νεο όνομα του τύπου το οποίο διορθώνουμε και **stayingtypeID** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων που αντιστοιχεί στον τύπο.
Στη συνέχεια επικοινωνεί με τη stored procedure **“editstayingtype”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές των πεδίων που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.22 Μέθοδος: addRoomSize

Είναι υπεύθυνη για την εισαγωγή ενός τύπου δωματίου στη βάση δεδομένων και συγκεκριμένα στον πίνακα **roomsize**. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<room>Single</room>
</root>
```

Όπου **room** το όνομα του τύπου δωματίου που θέλει ο χρήστης να προστεθεί στη βάση.
Στη συνέχεια επικοινωνεί με τη stored procedure **“insertroomsize”** της βάσης δεδομένων και της δίνει σαν όρισμα την τιμή του πεδίου που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.23 Μέθοδος: editRoomSize

Είναι υπεύθυνη για την διόρθωση του ονόματος ενός τύπου δωματίου από τη βάση δεδομένων. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<room>Single</room>
<roomID>2</roomID>
</root>
```

Όπου **room** το νεο όνομα του τύπου το οποίο διορθώνουμε και **roomID** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων που αντιστοιχεί στον τύπο.
Στη συνέχεια επικοινωνεί με τη stored procedure **“editroomsize”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές των πεδίων που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.24 Μέθοδος: addInternetType

Είναι υπεύθυνη για την εισαγωγή ενός τύπου internet στη βάση δεδομένων και συγκεκριμένα στον πίνακα **internet**. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<internet>Wireless</internet>
</root>
```

Όπου **internet** το όνομα του τύπου που θέλει ο χρήστης να προστεθεί στη βάση.
Στη συνέχεια επικοινωνεί με τη stored procedure **“insertinternettype”** της βάσης δεδομένων και της δίνει σαν όρισμα την τιμή του πεδίου που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.25 Μέθοδος: editInternetType

Είναι υπεύθυνη για την διόρθωση του ονόματος ενός τύπου internet από τη βάση δεδομένων. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<internet>Wireless</internet>
<internetID>2</internetID>
</root>
```

Όπου **internet** το νεο όνομα του τύπου το οποίο διορθώνουμε και **internetID** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων που αντιστοιχεί στον τύπο.
Στη συνέχεια επικοινωνεί με τη stored procedure **“editinternettype”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές των πεδίων που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.26 Μέθοδος: addFacility

Είναι υπεύθυνη για την εισαγωγή ενός hotel facility στη βάση δεδομένων και συγκεκριμένα στον πίνακα **facilities**. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<facilty>Swimming Pool</facility>
</root>
```

Όπου **facility** το όνομα του facility που θέλει ο χρήστης να προστεθεί στη βάση.
Στη συνέχεια επικοινωνεί με τη stored procedure **“insertfacility”** της βάσης δεδομένων και της δίνει σαν όρισμα την τιμή του πεδίου που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.27 Μέθοδος: editFacility

Είναι υπεύθυνη για την διόρθωση του ονόματος ενός hotel facility από τη βάση δεδομένων. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<facility>Swimming Pool</facility>
<facilityID>34</facilityID>
</root>
```

Όπου **facility** το νεο όνομα του facility το οποίο διορθώνουμε και **facilityID** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων που αντιστοιχεί στο facility.
Στη συνέχεια επικοινωνεί με τη stored procedure **“editfacility”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές των πεδίων που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.28 Μέθοδος: addRoomFacility

Είναι υπεύθυνη για την εισαγωγή ενός room facility στη βάση δεδομένων και συγκεκριμένα στον πίνακα **roomfacilities**. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<RoomFacility>Mini Bar</RoomFacility>
</root>
```

Όπου **RoomFacility** το όνομα του facility που θέλει ο χρήστης να προστεθεί στη βάση.
Στη συνέχεια επικοινωνεί με τη stored procedure **“insertroomfacility”** της βάσης δεδομένων και της δίνει σαν όρισμα την τιμή του πεδίου που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.29 Μέθοδος: editRoomFacility

Είναι υπεύθυνη για την διόρθωση του ονόματος ενός room facility από τη βάση δεδομένων. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<RoomFacility>Mini Bar</RoomFacility>
<RoomFacilityID>43</RoomFacilityID>
</root>
```

Όπου **RoomFacility** το νεο όνομα του facility το οποίο διορθώνουμε και **RoomFacilityID** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων που αντιστοιχεί στο facility.

Στη συνέχεια επικοινωνεί με τη stored procedure **“editroomfacility”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές των πεδίων που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.30 Μέθοδος: addFood

Είναι υπεύθυνη για την εισαγωγή των τύπων φαγητού από τη προσωρινή λίστα στη βάση δεδομένων που θα είναι διαθέσιμα στο σύστημα και συγκεκριμένα στον πίνακα **food**. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>1</id>
<food>BnB</food>
<flag>1</flag>
<id>2</id>
<food>HB</food>
<flag>1</flag>
</root>
```

Όπου **id** ο αύξων αριθμός που θα πάρει ο τύπος, **food** το όνομα του τύπου και **flag** το αν θα είναι ενεργοποιημένος ή όχι.
Έπειτα καλεί ένα ερώτημα και διαγράφει από τον πίνακα food τις παλιές καταχωρήσεις και στη συνέχεια επικοινωνεί με τη stored procedure **“insertfood”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές των πεδίων που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.30 Μέθοδος: addTopDestinations

Είναι υπεύθυνη για την εισαγωγή των προτεινόμενων ξενοδοχείων από τη προσωρινή λίστα στη βάση δεδομένων που θα είναι διαθέσιμα στο σύστημα. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<id>4</id>
<id>7</id>
<id>8</id>
<id>12</id>
<id>16</id>
<id>19</id>
</root>
```

- **Id –** Ο αυξων αριθμός του ξενοδοχείου

Έπειτα επικοινωνεί με τη stored procedure **“insertTopDestinations”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές από το όρισμα που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.30 Μέθοδος: addAdminMail

Είναι υπεύθυνη για την εισαγωγή/αλλαγη των στοιχείων email του administrtor. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<mail>2</mail>
<user>4</user>
<pass>7</pass>
</root>
```

- **mail –** Η διευθυνση email του admin
- **user –** Το συνθηματικο πρόσβασης στο email
- **pass –** Το password πρόσβασης στο email

Έπειτα επικοινωνεί με τη stored procedure **“insertAdminMail”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές από το όρισμα που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.30 Μέθοδος: addHotelDistance

Είναι υπεύθυνη για την εισαγωγή/αλλαγη μιας απόσταστης από ένα σημείο ενδιαφέροντος. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<station>Airport</station>
<distance>45</distance>
</root>
```

- **id –** Ο αυξων αριθμος του ξενοδοχειου
- **station –** Το σημειο ενδιαφεροντος
- **distance –** Η αποσταση από το σημειο

Έπειτα επικοινωνεί με τη stored procedure **“insertHotelDistances”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές από το όρισμα που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.30 Μέθοδος: addHotelFacility

Είναι υπεύθυνη για την εισαγωγή/αλλαγη ενός facility σε ένα ξενοδοχειο. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<facility>Airport</facility>
</root>
```

- **id –** Ο αυξων αριθμος του ξενοδοχειου
- **facility –** Το ονομα του facility

Έπειτα επικοινωνεί με τη stored procedure **“insertHotelFacilities”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές από το όρισμα που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.30 Μέθοδος: addRoomNum

Είναι υπεύθυνη για την εισαγωγή/αλλαγη του αριθμού δωματίων σε ένα ξενοδοχειο. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<number>12</number>
<roomsize>Double</roomsize>
<smoke>0</smoke>
</root>
```

- **id –** Ο αυξων αριθμος του ξενοδοχειου
- **number –** Ο αριθμός των δωματιων
- **roomsize –** Ο τυπος του δωματιου που θα προσθεθούν
- **smoke –** Εάν προκειται για δωματιο καπνιζοντων

Έπειτα επικοινωνεί με τη stored procedure **“insertNumberOfRooms”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές από το όρισμα που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια

ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.30 Μέθοδος: addRoomFac

Είναι υπεύθυνη για την εισαγωγή/αλλαγη ενός facility σε ένα δωματιο ξενοδοχειου. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<roomfacility>12</roomfacility>
<roomsize>Double</roomsize>
</root>
```

- **id –** Ο αυξων αριθμος του ξενοδοχειου
- **roomfacility –** Το Ονομα του room facility
- **roomsize –** Ο τυπος του δωματιου που θα προσθεθει το faiclity

Έπειτα επικοινωνεί με τη stored procedure **“inserthotelroomfacilities”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές από το όρισμα που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.30 Μέθοδος: addStopSales

Είναι υπεύθυνη για την εισαγωγή/αλλαγη stopsales ενός ξενοδοχειου. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<date_start>2010-02-20</date_start>
<date_end>2010-02-23</date_end>
</root>
```

- **id –** Ο αυξων αριθμος του ξενοδοχειου
- **date_start –** Η ημερομηνια εναρξης του stop sale
- **date_start –** Η ημερομηνια ληξης του stop sale

Έπειτα επικοινωνεί με τη stored procedure **“ insertstopsales”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές από το όρισμα που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.30 Μέθοδος: addRelease

Είναι υπεύθυνη για την εισαγωγή/αλλαγη των release policies ενός ξενοδοχειου. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<release>20</release>
<releaseST>2010-02-23</releaseST>
<releaseNT>2010-02-23</releaseNT>
</root>
```

- **id –** Ο αυξων αριθμος του ξενοδοχειου
- **releaseST –** Η ημερομηνια εναρξης του policy
- **releaseND –** Η ημερομηνια ληξης του policy
- **release –** Ο αριθμός των ημερων του Policy

Έπειτα επικοινωνεί με τη stored procedure **“insertrelease”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές από το όρισμα που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.30 Μέθοδος: addHotelEarlyBooking

Είναι υπεύθυνη για την εισαγωγή/αλλαγη των early booking policies ενός ξενοδοχειου. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<price>30</price>
<date_start>2010-02-23</date_start>
<date_end>2010-02-23</date_end>
<offerstart>2010-02-27</offerstart>
<offerend>2010-02-29</offerend>
</root>
```

- **id –** Ο αυξων αριθμος του ξενοδοχειου
- **date_start –** Η ημερομηνια εναρξης του policy
- **date_end –** Η ημερομηνια ληξης του policy
- **offerstart –** Η ημερομηνια εναρξης ισχυος του policy
- **offerend –** Η ημερομηνια ληξης ισχυος του policy
- **price –** Η τιμη του policy

Έπειτα επικοινωνεί με τη stored procedure **“insertholelearlybooking”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές από το όρισμα που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.30 Μέθοδος: addAvailable

Είναι υπεύθυνη για την εισαγωγή/αλλαγη των διαθεσιμοτητων ενός ξενοδοχειου. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<price>30</price>
<date_start>2010-02-23</date_start>
<date_end>2010-02-25</date_end>
<roomsize>Double</roomsize>
<roomnumber>4</roomnumber>
<person>3</person>
<food>Halfboard</food>
</root>
```

- **id –** Ο αυξων αριθμος του ξενοδοχειου
- **date_start –** Η ημερομηνια εναρξης της διαθεσιμοτητας
- **date_end –** Η ημερομηνια ληξης της διαθεσιμοτητας
- **roomsize –** Ο τυπος δωματιου της διαθεσιμοτητας
- **roomnumber –** Ο αριθμος των δωματιων της διαθεσιμοτητας
- **price –** Η τιμη της διαθεσιμοτητας
- **person –** Τα ατομα της διαθεσιμοτητας
- **food –** Ο τυπος διατροφης της διαθεσιμοτητας

Έπειτα επικοινωνεί με τη stored procedure **“insertavailable”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές από το όρισμα που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.30 Μέθοδος: addHotelFood

Είναι υπεύθυνη για την εισαγωγή/αλλαγη των τυπων διατροφης που μπορουν να προσθεθουν με τις διαθεσιμοτητες ενός ξενοδοχειου. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
```

<id>2</id>
<price>30</price>
<start>2010-02-23</ start>
< end>2010-02-25</end>
<roomsize>Double</roomsize>
<food>Halfboard</food>
</root>

- **id –** Ο αυξων αριθμος του ξενοδοχειου
- **date_start –** Η ημερομηνια εναρξης της διαθεσιμοτητας
- **date_end –** Η ημερομηνια ληξης της διαθεσιμοτητας
- **roomsize –** Ο τυπος δωματιου της διαθεσιμοτητας
- **price –** Η τιμη της διαθεσιμοτητας
- **food –** Ο τυπος διατροφης της διαθεσιμοτητας

Έπειτα επικοινωνεί με τη stored procedure **“inserthotelfood”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές από το όρισμα που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.30 Μέθοδος: addHotelRestriction

Είναι υπεύθυνη για την εισαγωγή/αλλαγη των περιορισμων ενός ξενοδοχειου. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<price>30</price>
<date_start>2010-02-23</date_start>
<date_end>2010-02-25</date_end>
<room_size>Double</room_size>
<days>5</days>
</root>

- **id –** Ο αυξων αριθμος του ξενοδοχειου
- **date_start –** Η ημερομηνια εναρξης του περιορισμου
- **date_end –** Η ημερομηνια ληξης του περιορισμου
- **roomsize –** Ο τυπος δωματιου του περιορισμου
- **price –** Η τιμη του περιορισμου
- **days –** Ο αριθμος των ημερων του περιορισμου

Έπειτα εκτελεί ένα ερώτημα στη βάση δεδομένων και δίνει τις τιμές από το όρισμα που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.30 Μέθοδος: addRoomAvailable

Είναι υπεύθυνη για την εισαγωγή/αλλαγη των διαθεσιμων δωματιων ενός ξενοδοχειου. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<st>2010-02-23</st>
<nd>2010-02-25</nd>
</root>
```

- **id –** Ο αυξων αριθμος του ξενοδοχειου
- **st –** Η ημερομηνια εναρξης της διαθεσιμοτητας
- **nd –** Η ημερομηνια ληξης της διαθεσιμοτητας

Έπειτα εκτελει κάποια ερωτηματα στη βάση δεδομένων και δίνει τις τιμές από το όρισμα που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.30 Μέθοδος: addExcelAvailable

Είναι υπεύθυνη για την εισαγωγή/αλλαγη των διαθεσιμοτητων ενός ξενοδοχειου Μεσα από ένα δεδομενο αρχειο excel. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<price>30</price>
<date_start>2010-02-23</date_start>
<date_end>2010-02-25</date_end>
<person>3</person>
<food>Halfboard</food>
</root>
```

- **id –** Ο αυξων αριθμος του ξενοδοχειου
- **date_start –** Η ημερομηνια εναρξης της διαθεσιμοτητας
- **date_end –** Η ημερομηνια ληξης της διαθεσιμοτητας
- **price –** Η τιμη της διαθεσιμοτητας
- **person –** Τα ατομα της διαθεσιμοτητας
- **food –** Ο τυπος διατροφης της διαθεσιμοτητας

Έπειτα εκτελει κάποια ερωτηματα στη βάση δεδομένων και δίνει τις τιμές από το όρισμα που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.30 Μέθοδος: addFotoCountry

Είναι υπεύθυνη για την εισαγωγή/αλλαγη της φωτογραφίας μιας χώρας. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<foto>fotoUrl</foto>
</root>
```

- **country_name –** Το ονομα της χώρας
- **foto –** Το Url της φωτογραφιας

Έπειτα επικοινωνεί με τη stored procedure **“insertfotocountry”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές από το όρισμα που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.30 Μέθοδος: addFotoRegion

Είναι υπεύθυνη για την εισαγωγή/αλλαγη της φωτογραφίας μιας επαρχιας. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<region_name>Crete</region_name>
<foto>fotoUrl</foto>
</root>
```

- **country_name –** Το ονομα της χώρας
- **region_name –** Το ονομα της επαρχιας
- **foto –** Το Url της φωτογραφιας

Έπειτα επικοινωνεί με τη stored procedure **“insertfotoregion”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές από το όρισμα που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.30 Μέθοδος: addFotoCity

Είναι υπεύθυνη για την εισαγωγή/αλλαγη της φωτογραφίας μιας Πολης. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<region_name>Crete</region_name>
<city_name>Chania</city_name>
<foto>fotoUrl</foto>
</root>
```

- **country_name –** Το ονομα της χώρας
- **region_name –** Το ονομα της επαρχιας
- **city_name –** Το ονομα της πολης
- **foto –** Το Url της φωτογραφιας

Έπειτα επικοινωνεί με τη stored procedure **“insertfotocity”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές από το όρισμα που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.30 Μέθοδος: addFotoAddHotel

Είναι υπεύθυνη για την εισαγωγή/αλλαγη της κεντρικης φωτογραφίας ενός ξενοδοχείου. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<foto>fotoUrl</foto>
</root>
```

- **id –** Ο αυξων αριθμος του ξενοδοχείου
- **foto –** Το Url της φωτογραφιας

Έπειτα επικοινωνεί με τη stored procedure **“insertfotoaddhotel”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές από το όρισμα που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.1.30 Μέθοδος: addHotel

Είναι υπεύθυνη για την εισαγωγή ενός ξενοδοχείου στη βάση. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<name></name>
<stayingtype></stayingtype>
<stars></stars>
<email></email>
<phone></phone>
<country_name></country_name>
<region_name></region_name>
<city_name></city_name>
<location></location>
<hoteladdress></hoteladdress>
<description></description>
<totalrooms></totalrooms>
<hotelchain></hotelchain>
<fax></fax>
<chargingpolicy></chargingpolicy>
<bedchild></bedchild>
<pets></pets>
<services></services>
<internet></internet>
<checkin></checkin>
<checkout></checkout>
<latt></latt>
<lon></lon>
</root>
```

- **id –** χρήστης που εισήγαγε το ξενοδοχείο
- **name –** Το ονομα του ξενοδοχείου
- **stayingtype -** ο τύπος διαμονής του ξενοδοχείου
- **stars -** ο αριθμός των αστεριών του ξενοδοχείου
- **email -** το email του ξενοδοχείου
- **phone -** το τηλέφωνο του ξενοδοχείου
- **country_name -** η χώρα του ξενοδοχείου
- **region_name -** η επαρχία που βρίσκεται του ξενοδοχείο
- **city_name -** η πολη του ξενοδοχείου
- **location -** η τοποθεσία του ξενοδοχείου
- **hoteladdress -** η διεύθυνση του ξενοδοχείου
- **description -** η περιγραφή του ξενοδοχείου
- **totalrooms -** ο αριμός του συνόλου των δωματίων του ξενοδοχείου
- **hotelchain -** το όνομα της αλυσίδας του ξενοδοχείου
- **fax -** το fax του ξενοδοχείου
- **charginpolicy -** η πολιτική ακυρώσεων του ξενοδοχείου

- **bedchild** - η πολιτική των επιπρόσθετων κρεβατιών του ξενοδοχείου
- **pets** - η πολιτική των κατοικιδίων
- **services** - οι επιπρόσθετες υπηρεσίες που παρέχει το ξενοδοχείο
- **internet** - οι τύποι internet Που παρέχει το ξενοδοχείο
- **checkin** - η ώρα που δέχεται τα checkin το ξενοδοχείο
- **checkout** - η ώρα που δέχεται τα checkout το ξενοδοχείο
- **latt** - ο δεκαδικός αριθμός του γεωγραφικού μήκους του ξενοδοχείου
- **lon** - ο δεκαδικός αριθμός του γεωγραφικού πλάτους του ξενοδοχείου

Έπειτα επικοινωνεί με τη stored procedure **"inserthoteltable"** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές από το όρισμα που έλαβε. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.2 WebService: hoteldbFill

2.2.1 Μέθοδος: fillCountry

Είναι υπεύθυνη στο να ανακτά το πλήθος των χωρών που είναι καταχωρημένες στη βάση δεδομένων και συγκεκριμένα στον πίνακα **country**. Δεν δέχεται σαν όρισμα κάποιο ωφέλημο μήνυμα. Επικοινωνεί με τη stored procedure **selectcountryadmin** και από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης που είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>7</id>
<country_name>Germany</country_name>
<flag>1</flag>
<id>8</id>
<country_name>Greece</country_name>
<flag>1</flag>
<id>11</id>
<country_name>Italy</country_name>
<flag>1</flag>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε χώρα στον πίνακα "**country"**. Το πεδίο **country_name** το οποίο αντιπροσωπεύει το όνομα της χώρας και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν η συγκεκριμένη χώρα είναι ενεργοποιημένη ή όχι.

2.2.2 Μέθοδος: fillRegion

Είναι υπεύθυνη στο να ανακτά το πλήθος των επαρχιών μιας χώρας που είναι καταχωρημένες στη βάση δεδομένων και συγκεκριμένα στον πίνακα **region**.

Το όρισμα που δέχεται είναι ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
</root>
```

Όπου **country_name** το όνομα της χώρας της οποίας, έχει επιλέξει ο χρήστης να πάρει, τις επαρχίες. Επικοινωνεί με τη stored procedure **selectregionadmin** δίνοντας της σαν όρισμα τη τιμή του πεδίου. Από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης το οποίο περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>22</id>
<region_name>Attica</region_name>
<flag>1</flag>
<id>25</id>
<region_name>Crete</region_name>
<flag>1</flag>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε επαρχία στον πίνακα “**region”**. Το πεδίο **region_name** το οποίο αντιπροσωπεύει το όνομα της επαρχίας και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν η συγκεκριμένη επαρχία είναι ενεργοποιημένη ή όχι.

2.2.3 Μέθοδος: fillCity

Είναι υπεύθυνη στο να ανακτά το πλήθος των πόλεων μιας επαρχίας που είναι καταχωρημένες στη βάση δεδομένων και συγκεκριμένα στον πίνακα **city**.

Το όρισμα που δέχεται είναι ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<region_name>Crete</region_name>
</root>
```

Όπου **country_name** το όνομα της χώρας και **region_name** το όνομα της επαρχίας της οποίας, έχει επιλέξει ο χρήστης να πάρει, τις επαρχίες. Επικοινωνεί με τη stored procedure **selectcityadmin** δίνοντας της σαν όρισμα τις τιμές των πεδίων που έλαβε. Από την απάντηση

που λαμβάνει χτίζει το μήνυμα απάντησης το οποίο περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>22</id>
<city_name>Chania</city_name>
<flag>1</flag>
<id>25</id>
<city_name>Rethymno</city_name>
<flag>1</flag>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε πόλη στον πίνακα "**city"**. Το πεδίο **city_name** το οποίο αντιπροσωπεύει το όνομα της πόλης και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν η συγκεκριμένη πόλη είναι ενεργοποιημένη ή όχι.

2.2.4 Μέθοδος: fillLocation

Είναι υπεύθυνη στο να ανακτά το πλήθος των τοποθεσιών μιας πόλης που είναι καταχωρημένες στη βάση δεδομένων και συγκεκριμένα στον πίνακα **location**.

Το όρισμα που δέχεται είναι ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<region_name>Crete</region_name>
<city_name>Chania</city_name>
</root>
```

Όπου **country_name** το όνομα της χώρας, **region_name** το όνομα της επαρχίας και **city_name** το όνομα τη πόλης της οποίας, έχει επιλέξει ο χρήστης να πάρει, τις τοποθεσίες. Επικοινωνεί με τη stored procedure **selectclocationadmin** δίνοντας της σαν όρισμα τις τιμές των πεδίων που έλαβε. Από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης το οποίο περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>22</id>
<location>Akrotiri</location>
<flag>1</flag>
<id>25</id>
<location>Chalepa</location>
<flag>1</flag>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε τοποθεσία στον πίνακα "**location"**. Το πεδίο **location** το οποίο αντιπροσωπεύει το όνομα της τοποθεσίας και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν η συγκεκριμένη τοποθεσία είναι ενεργοποιημένη ή όχι.

2.2.5 Μέθοδος: fillCardType

Είναι υπεύθυνη στο να ανακτά το πλήθος των τύπων καρτών που είναι καταχωρημένες στη βάση δεδομένων και συγκεκριμένα στον πίνακα **cardtype**. Δεν δέχεται σαν όρισμα κάποιο ωφέλημο μήνυμα. Επικοινωνεί με τη stored procedure **selectcardtypeadmin** και από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης που είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>1</id>
<cardtype>Visa</cardtype>
<flag>1</flag>
<id>2</id>
<cardtype>Mastercard</cardtype>
<flag>1</flag>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε τύπο κάρτας στον πίνακα "**cardtype"**. Το πεδίο **cardtype** το οποίο αντιπροσωπεύει το όνομα του τύπου και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν ο συγκεκριμένος τύπος είναι ενεργοποιημένος ή όχι.

2.2.6 Μέθοδος: fillExchangeType

Είναι υπεύθυνη στο να ανακτά το πλήθος των νομισμάτων-χωρών που είναι καταχωρημένες στη βάση δεδομένων και συγκεκριμένα στον πίνακα **exchangetype**. Δεν δέχεται σαν όρισμα κάποιο ωφέλημο μήνυμα. Επικοινωνεί με τη stored procedure **selectexchangetypedmin** και από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης που είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>1</id>
<country_name>Greece</country_name>
<currency>Euro</currency>
<flag>1</flag>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε νόμισμα-χώρα στον πίνακα "**exchangetype"**. Το πεδίο **currency** το οποίο αντιπροσωπεύει το όνομα του νομίσματος και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν το συγκεκριμένο νόμισμα είναι ενεργοποιημένο ή όχι.

2.2.7 Μέθοδος: fillDistanceFrom

Είναι υπεύθυνη στο να ανακτά το πλήθος των σημείων ενδιαφέροντος που είναι καταχωρημένα στη βάση δεδομένων και συγκεκριμένα στον πίνακα **distancefrom**. Δεν δέχεται σαν όρισμα κάποιο ωφέλημο μήνυμα. Επικοινωνεί με τη stored procedure **selectdistancefromadmin** και από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης που είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>1</id>
<station>Airport</station>
<flag>1</flag>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε σημείο ενδιαφέροντος στον πίνακα "**distancefrom"**. Το πεδίο **station** το οποίο αντιπροσωπεύει το όνομα του σημείου και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν το συγκεκριμένο σημείο είναι ενεργοποιημένο ή όχι.

2.2.8 Μέθοδος: fillUserHotel

Είναι υπεύθυνη στο να ανακτά το πλήθος των χρηστών που είναι καταχωρημένοι στη βάση δεδομένων και συγκεκριμένα στον πίνακα **userhotel**. Δεν δέχεται σαν όρισμα κάποιο ωφέλημο μήνυμα. Επικοινωνεί με τη stored procedure **selectuserhoteladmin** και από την απάντηση που λαμβάνει αναλύει τον περιορισμό σε περιοχή του χρήστη για να καλέσει τις procedures **selectuserhotelrecountry, selectuserhotelreregion** και **selectuserhotelrecity** για να πάρει τα ονόματα των περιοχών που είναι περιορισμένος ο χρήστης.

Στη συνέχεια χτίζει το μήνυμα απάντησης που είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>1</id>
<user_name>q</user_name>
<pass_word>q</pass_word>
<flag>1</flag>
<rest>Greece</rest>
<forward>10</forward>
<hotelNum>5</hotelNum>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε χρήστη στον πίνακα "**userhotel"**.
Το πεδίο **user_name** το οποίο αντιπροσωπεύει το συνθηματικό του χρήστη
Το πεδίο **pass_word** το οποίο αντιπροσωπεύει τον κωδικό του χρήστη
Το πεδίο **flag** στο οποίο φαίνεται εάν ο συγκεκριμένος χρήστης είναι ενεργοποιημένος ή όχι.
Το πεδίο **rest** στο οποίο φαίνεται ο περιορισμός του χρήστη.
Το πεδίο **forward** το οποίο αντιπροσωπεύει το ποσοστό της προκαταβολης του χρήστη.
Το πεδίο **hotelNum** στο οποίο περιλαμβάνεται ο αριθμός των ξενοδοχείων που μπορεί να βάλει ένας χρήστης.

2.2.9 Μέθοδος: fillRegionHotel

Είναι υπεύθυνη στο να ανακτά το πλήθος των ενεργοποιημένων επαρχιών μιας χώρας που είναι καταχωρημένες στη βάση δεδομένων και συγκεκριμένα στον πίνακα **region**.

Το όρισμα που δέχεται είναι ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
</root>
```

Όπου **country_name** το όνομα της χώρας της οποίας, έχει επιλέξει ο χρήστης να πάρει, τις επαρχίες. Επικοινωνεί με τη stored procedure **selectregion** δίνοντας της σαν όρισμα τη τιμή του πεδίου. Από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης το οποίο περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<region_name>Attica</region_name>
<region_name>Crete</region_name>
</root>
```

Όπου **region_name** το όνομα της επαρχίας.

2.2.10 Μέθοδος: fillCityHotel

Είναι υπεύθυνη στο να ανακτά το πλήθος των ενεργοποιημένων πόλεων μιας επαρχίας που είναι καταχωρημένες στη βάση δεδομένων και συγκεκριμένα στον πίνακα **city**.

Το όρισμα που δέχεται είναι ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
<region_name>Crete</region_name>
</root>
```

Όπου **country_name** το όνομα της χώρας και **region_name** το όνομα της επαρχίας της οποίας, έχει επιλέξει ο χρήστης να πάρει, τις επαρχίες. Επικοινωνεί με τη stored procedure **selectcity** δίνοντας της σαν όρισμα τις τιμές των πεδίων που έλαβε. Από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης το οποίο περιλαμβάνει ένα SOAP μήνυμα, κωδικοποιημένο σε μορφή Base64 που περιέχει το παρακάτω xml έγγραφο:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<root>
<city_name>Chania</city_name>
<city_name>Rethymno</city_name>
</root>
```

Όπου **city_name** το όνομα της πόλης.

2.2.11 Μέθοδος: fillStayingType

Είναι υπεύθυνη στο να ανακτά το πλήθος των τύπων ξενοδοχείου που είναι καταχωρημένοι στη βάση δεδομένων και συγκεκριμένα στον πίνακα **stayingtype**. Δεν δέχεται σαν όρισμα κάποιο ωφέλημο μήνυμα. Επικοινωνεί με τη stored procedure **selectstayingtypeadmin** και από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης που είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>1</id>
<stayingtype>Hotel</stayingtype>
<flag>1</flag>
</root>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε τύπο στον πίνακα "**stayingtype"**. Το πεδίο **stayingtype** το οποίο αντιπροσωπεύει το όνομα του τύπου και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν ο συγκεκριμένος τύπος είναι ενεργοποιημένος ή όχι.

2.2.12 Μέθοδος: fillRoomSize

Είναι υπεύθυνη στο να ανακτά το πλήθος των τύπων δωματίων που είναι καταχωρημένοι στη βάση δεδομένων και συγκεκριμένα στον πίνακα **roomsize**. Δεν δέχεται σαν όρισμα κάποιο ωφέλημο μήνυμα. Επικοινωνεί με τη stored procedure **selectroomsizeadmin** και από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης που είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>1</id>
<room>Single</room>
<flag>1</flag>
</root>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε τύπο στον πίνακα "**roomsize"**. Το πεδίο **room** το οποίο αντιπροσωπεύει το όνομα του τύπου και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν ο συγκεκριμένος τύπος είναι ενεργοποιημένος ή όχι.

2.2.13 Μέθοδος: fillInternetType

Είναι υπεύθυνη στο να ανακτά το πλήθος των τύπων internet που είναι καταχωρημένοι στη βάση δεδομένων και συγκεκριμένα στον πίνακα **internet**. Δεν δέχεται σαν όρισμα κάποιο ωφέλημο μήνυμα. Επικοινωνεί με τη stored procedure **selecthotelinternettypeadmin** και από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης που είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>1</id>
<internet>Wireless</internet>
<flag>1</flag>
</root>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε τύπο στον πίνακα "**internet"**. Το πεδίο **internet** το οποίο αντιπροσωπεύει το όνομα του τύπου και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν ο συγκεκριμένος τύπος είναι ενεργοποιημένος ή όχι.

2.2.14 Μέθοδος: fillFacilities

Είναι υπεύθυνη στο να ανακτά το πλήθος των hotel facilities που είναι καταχωρημένα στη βάση δεδομένων και συγκεκριμένα στον πίνακα **facilities**. Δεν δέχεται σαν όρισμα κάποιο ωφέλημο μήνυμα. Επικοινωνεί με τη stored procedure **selectfacilitiesadmin** και από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης που είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>1</id>
<facility_name>Swimming Pool</facility_name>
<flag>1</flag>
</root>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε facility στον πίνακα "**facilities"**. Το πεδίο **facility_name** το οποίο αντιπροσωπεύει το όνομα του facility και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν το συγκεκριμένο facility είναι ενεργοποιημένο ή όχι.

2.2.15 Μέθοδος: fillRoomFacilities

Είναι υπεύθυνη στο να ανακτά το πλήθος των room facilities που είναι καταχωρημένα στη βάση δεδομένων και συγκεκριμένα στον πίνακα **roomfacilities**. Δεν δέχεται σαν όρισμα κάποιο ωφέλημο μήνυμα. Επικοινωνεί με τη stored procedure **selectroomfacilitiesadmin** και από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης που είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>1</id>
<RoomFacility>Mini Bar</RoomFacility>
<flag>1</flag>
```

```
</root>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε facility στον πίνακα “**roomfacilities”**. Το πεδίο **RoomFacility** το οποίο αντιπροσωπεύει το όνομα του facility και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν το συγκεκριμένο facility είναι ενεργοποιημένο ή όχι.

2.2.16 Μέθοδος: fillFoodType

Είναι υπεύθυνη στο να ανακτά το πλήθος των τύπων φαγητού που είναι καταχωρημένοι στη βάση δεδομένων και συγκεκριμένα στον πίνακα **food**. Δεν δέχεται σαν όρισμα κάποιο ωφέλημο μήνυμα. Επικοινωνεί με τη stored procedure **selectfoodadmin** και από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης που είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>1</id>
<food>BnB</food>
<flag>1</flag>
</root>
```

Όπου **id** ο αύξων αριθμός από την αρίθμηση της βάσης δεδομένων και είναι μοναδικός για κάθε τύπο στον πίνακα “**food”**. Το πεδίο **food** το οποίο αντιπροσωπεύει το όνομα του τύπου φαγητού και τέλος το πεδίο **flag** στο οποίο φαίνεται εάν ο συγκεκριμένος τύπος είναι ενεργοποιημένος ή όχι.

2.2.17 Μέθοδος: fillHotelAdmin

Είναι υπεύθυνη να ανακτά το πλήθος των ξενοδοχείων που βρίσκονται στον πίνακα **hoteltable**. Δεν δέχεται σαν όρισμα κάποιο ωφέλημο μήνυμα. Επικοινωνεί με τη stored procedure **selectHotelAdmin** και από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης που είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>1</id>
<flag>1</flag>
<dateadd>2010-02-20</dateadd>
<name>Hotel Name</name>
<user_name>username</user_name>
<stars>5</stars>
<email>email@email.com</email>
<phone>+302101231234</phone>
<hoteladdress>Hotel Address</hoteladdress>
<totalrooms>33</totalrooms>
```

<hotelchain>Hotel Chain</hotelchain>
<fax>+302101231235</fax>
<stayingtype>Hotel</stayingtype>
<city_name>Athens City</city_name>
<location>Centre</location>
<region_name>Attica</region_name>
<country_name>Greece</country_name>
<checkout>10:00</checkout>
<checkin>12:00</checkin>
<rating>8</rating>
</root>

- **id -** ο αύξων αριθμός του ξενοδοχείου από την αρίθμηση της βάσης δεδομένων
- **flag -** η ενδειξη του αν το ξενοδοχείο είναι ενεργοποιημένο ή οχι.
- **dateadd –** η ημερομηνία που προσθέθηκε το ξενοδοχείο στη βάση
- **name –** το όνομα του ξενοδοχείου
- **user_name –** Το username του χρήστη που εισήγαγε το ξενοδοχείο
- **start –** Ο αριθμός των αστεριων του ξενοδοχείου
- **email –** Το email του ξενοδοχείου
- **phone –** Το τηλέφωνο του ξενοδοχείου
- **hoteladdress –** Η διεύθυνση του ξενοδοχείου
- **totalrooms –** Ο συνολικός αριθμός των δωματίων του ξενοδοχείου
- **hotelchain –** Το όνομα της αλυσίδας που ανήκει το ξενοδοχείου
- **fax –** Το fax του ξενοδοχείου
- **stayingtype –** Ο τύπος διαμονής του ξενοδοχείου
- **city_name –** Το όνομα της πολης του ξενοδοχείου
- **location –** Το όνομα της τοποθεσίας του ξενοδοχείου
- **region_name –** Το όνομα της επαρχίας του ξενοδοχείου
- **country_name –** Το όνομα της χώρας του ξενοδοχείου
- **checkout –** Η ώρα που δέχεται checkout το ξενοδοχείο
- **checkin –** Η ώρα που δέχεται checkin το ξενοδοχείο
- **rating –** Η βαθμόλογία που έχει προκύψει από τους επισκέπτες του ξενοδοχείου

2.2.18 Μέθοδος: fillHotelAdminCountry

Είναι υπεύθυνη να ανακτά το πλήθος των ξενοδοχείων του πίνακα **hoteltable** που πληρούν τις επιλογές αναζήτησης του χρήστη. Το όρισμα που δέχεται είναι της μορφής:

<?xml version="1.0" encoding="UTF-8"?>
<root>
<country_name>Greece</country_name>
</root>

- **country_name –** Το όνομα της χώρας, κάτω από την οποία ναζητούμε τα ξενοδοχεία

Επικοινωνεί με τη stored procedure **selectHotelAdminCountry** και από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης. Το μήνυμα απάντησης είναι ίδιο με το μήνυμα της **fillHotelAdmin.**

2.2.19 Μέθοδος: fillHotelAdminName

Είναι υπεύθυνη να ανακτά το πλήθος των ξενοδοχείων του πίνακα **hoteltable** που πληρούν τις επιλογές αναζήτησης του χρήστη. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<user_name>username</user_name>
</root>
```

- **user_name –** Το όνομα του χρήστη του οποίου αναζητουμε τα ξενοδοχεία

Επικοινωνεί με τη stored procedure **selectHotelAdminName** και από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης. Το μήνυμα απάντησης είναι ίδιο με το μήνυμα της **fillHotelAdmin.**

2.2.20 Μέθοδος: fillHotelAdminCountryName

Είναι υπεύθυνη να ανακτά το πλήθος των ξενοδοχείων του πίνακα **hoteltable** που πληρούν τις επιλογές αναζήτησης του χρήστη. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<user_name>username</user_name>
<country_name>Greece</country_name>
</root>
```

- **user_name –** Το όνομα του χρήστη του οποίου αναζητουμε τα ξενοδοχεία
- **country_name –** Το όνομα της χώρας, κάτω από την οποία ναζητούμε τα ξενοδοχεία

Επικοινωνεί με τη stored procedure **selectHotelAdminCountryName** και από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης. Το μήνυμα απάντησης είναι ίδιο με το μήνυμα της **fillHotelAdmin.**

2.2.20 Μέθοδος: fillHotelAdminDate

Είναι υπεύθυνη να ανακτά το πλήθος των ξενοδοχείων του πίνακα **hoteltable** που πληρούν τις επιλογές αναζήτησης του χρήστη. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
```

<dateadd>username</dateadd>
</root>

- **dateadd –** Η ημερομηνία που εισήχθει το ξενοδοχείο στη βάση

Επικοινωνεί με τη stored procedure **selectHotelAdminDate** και από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης. Το μήνυμα απάντησης είναι ίδιο με το μήνυμα της **fillHotelAdmin.**

2.2.20 Μέθοδος: fillHotelAdminCountryDate

Είναι υπεύθυνη να ανακτά το πλήθος των ξενοδοχείων του πίνακα **hoteltable** που πληρούν τις επιλογές αναζήτησης του χρήστη. Το όρισμα που δέχεται είναι της μορφής:

<?xml version="1.0" encoding="UTF-8"?>
<root>
<dateadd>username</dateadd>
<country_name>Greece</country_name>
</root>

- **dateadd –** Η ημερομηνία που εισήχθει το ξενοδοχείο στη βάση
- **country_name –** Το όνομα της χώρας, κάτω από την οποία ναζητούμε τα ξενοδοχεία

Επικοινωνεί με τη stored procedure **selectHotelAdminCountryDate** και από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης. Το μήνυμα απάντησης είναι ίδιο με το μήνυμα της **fillHotelAdmin.**

2.2.20 Μέθοδος: fillHotelAdminNameDate

Είναι υπεύθυνη να ανακτά το πλήθος των ξενοδοχείων του πίνακα **hoteltable** που πληρούν τις επιλογές αναζήτησης του χρήστη. Το όρισμα που δέχεται είναι της μορφής:

<?xml version="1.0" encoding="UTF-8"?>
<root>
<dateadd>username</dateadd>
<user_name>username</user_name>
</root>

- **dateadd –** Η ημερομηνία που εισήχθει το ξενοδοχείο στη βάση
- **user_name –** Το όνομα του χρήστη του οποίου αναζητουμε τα ξενοδοχεία

Επικοινωνεί με τη stored procedure **selectHotelAdminNameDate** και από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης. Το μήνυμα απάντησης είναι ίδιο με το μήνυμα της **fillHotelAdmin.**

2.2.20 Μέθοδος: fillHotelAdminCountryNameDate

Είναι υπεύθυνη να ανακτά το πλήθος των ξενοδοχείων του πίνακα **hoteltable** που πληρούν τις επιλογές αναζήτησης του χρήστη. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<dateadd>username</dateadd>
<user_name>username</user_name>
<country_name>Greece</country_name>
</root>
```

- **dateadd –** Η ημερομηνία που εισήχθει το ξενοδοχείο στη βάση
- **user_name –** Το όνομα του χρήστη του οποίου αναζητουμε τα ξενοδοχεία
- **country_name –** Το όνομα της χώρας, κάτω από την οποία ναζητούμε τα ξενοδοχεία

Επικοινωνεί με τη stored procedure **selectHotelAdminCountryNameDate** και από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης. Το μήνυμα απάντησης είναι ίδιο με το μήνυμα της **fillHotelAdmin.**

2.2.20 Μέθοδος: fillTopDestinationAdmin

Είναι υπεύθυνη να ανακτά το πλήθος των προτεινόμενων ξενοδοχείων του πίνακα **topdestinations**. Δεν δέχεται κάποιο ωφέλιμο μήνυμα. Επικοινωνεί με τη stored procedure **selectTopDestinationsAdmin** και από τα δεδομένα που επιστρέφει χτίζει το μήνυμα απάντησης που είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>username</id>
<user_name>username</user_name>
<name>Greece</name>
<order>Greece</order>
</root>
```

- **id –** Ο αύξων αριθμός του ξενοδοχείου
- **user_name –** Το όνομα του χρήστη ο οποίος έχει εισάγει το ξενοδοχείο
- **name –** Το όνομα του ξενοδοχείου
- **order –** Ο αριθμός της κατάταξης στη λίστα με τα προτεινόμενα ξενοδοχεία.

2.2.17 Μέθοδος: fillHotelStatsAdmin

Είναι υπεύθυνη να ανακτά το πλήθος των κρατήσεων που βρίσκονται στον πίνακα **booking** που πληρούν τις επιλογές αναζήτησης του χρήστη. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<filter>all</filter>
</root>
```

- **filter –** Το φίλτρο για τις πληρωμενες, μη πληρωμένες ή ολες τις κρατήσεις

Αναλόγως των επιλογών του χρήστη εκτελεί ένα ερώτημα στη βάση και από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης που είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<user>username</user>
<name>Hotel Name</name>
<date_start>2010-02-22</date_start>
<date_end>2010-02-24</date_end>
<days>2</days>
<hotelextra>Extra</hotelextra>
<hotelfood>Halfboard</hotelfood>
<roomNum>2</roomNum>
<hotelprofit>5</hotelprofit>
<adminprofit>25</adminprofit>
<completed>1</completed>
<id>14</id>
<country>Greece</country>
<forward>25</forward>
</root>
```

- **id -** ο αύξων αριθμός της κράτησης από την αρίθμηση της βάσης δεδομένων
- **date_start -** Η ημερομηνία του checkin της κράτησης.
- **date_end –** Η ημερομηνία του checkout της κράτησης
- **name –** το όνομα του ξενοδοχείου
- **user –** Το username του χρήστη που εισήγαγε το ξενοδοχείο
- **days –** Ο αριθμός των ημερών της κράτησης
- **hotelextra –** Το κείμενο από τις πρόσθετες παροχές του ξενοδοχείου που περιλαμβάνονται στη κράτηση
- **hotelfood –** Ο τύπος της διατροφής της κράτησης
- **adminprofit –** Το ποσό του κερδους του διαχειρηστή από την κράτηση
- **hotelprofit –** Το ποσό του κέρδους του πράκτορα από τη κράτηση
- **roomNum –** Ο αριθμός των δωματίων που έχει κλείσει ο πελάτης
- **completed –** Δείχνει ποια και αν έχουν καταβληθεί τα μερίσματα από το κέρδος της κράτησης
- **country –** Η χώρα του ξενοδοχείου της κράτησης.
- **forward –** Το ποσό της προκαταβολής της κράτησης.

2.2.17 Μέθοδος: fillHotelStatsAdminCountry

Είναι υπεύθυνη να ανακτά το πλήθος των κρατήσεων που βρίσκονται στον πίνακα **booking** που πληρούν τις επιλογές αναζήτησης του χρήστη. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<filter>all</filter>
<country>Greece</country>
</root>
```

- **filter –** Το φίλτρο για τις πληρωμενες, μη πληρωμένες ή ολες τις κρατήσεις
- **country –** Η χωρα που ανήκουν τα ξενοδοχεία των οποίων ψάχνουμε τις κρατήσεις

Αναλόγως των επιλογών του χρήστη εκτελεί ένα ερώτημα στη βάση και από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης το οποίο είναι ίδιο με αυτό της **fillHotelStatsAdmin.**

2.2.17 Μέθοδος: fillHotelStatsAdminName

Είναι υπεύθυνη να ανακτά το πλήθος των κρατήσεων που βρίσκονται στον πίνακα **booking** που πληρούν τις επιλογές αναζήτησης του χρήστη. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<filter>all</filter>
<user>username</user>
</root>
```

- **filter –** Το φίλτρο για τις πληρωμενες, μη πληρωμένες ή ολες τις κρατήσεις
- **user –** Το username του χρήστη που ανήκουν τα ξενοδοχεία των οποίων ψάχνουμε τις κρατήσεις

Αναλόγως των επιλογών του χρήστη εκτελεί ένα ερώτημα στη βάση και από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης το οποίο είναι ίδιο με αυτό της **fillHotelStatsAdmin.**

2.2.17 Μέθοδος: fillHotelStatsAdminCountryName

Είναι υπεύθυνη να ανακτά το πλήθος των κρατήσεων που βρίσκονται στον πίνακα **booking** που πληρούν τις επιλογές αναζήτησης του χρήστη. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<filter>all</filter>
<user>username</user>
<country>Greece</country>
```

</root>

- **filter –** Το φίλτρο για τις πληρωμενες, μη πληρωμένες ή ολες τις κρατήσεις
- **user –** Το username του χρήστη που ανήκουν τα ξενοδοχεία των οποίων ψάχνουμε τις κρατήσεις
- **country –** Η χωρα που ανήκουν τα ξενοδοχεία των οποίων ψάχνουμε τις κρατήσεις

Αναλόγως των επιλογών του χρήστη εκτελεί ένα ερώτημα στη βάση και από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης το οποίο είναι ίδιο με αυτό της **fillHotelStatsAdmin.**

2.2.17 Μέθοδος: fillHotelStatsAdminDate

Είναι υπεύθυνη να ανακτά το πλήθος των κρατήσεων που βρίσκονται στον πίνακα **booking** που πληρούν τις επιλογές αναζήτησης του χρήστη. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<filter>all</filter>
<dateSt>username</dateSt>
<dateNd>Greece</dateNd>
</root>
```

- **filter –** Το φίλτρο για τις πληρωμενες, μη πληρωμένες ή ολες τις κρατήσεις
- **dateSt –** Η ημερομηνία εναρξης της περιόδου αναζήτησης
- **dateNd –** Η ημερομηνια λήξης της περιόδου αναζήτησης

Αναλόγως των επιλογών του χρήστη εκτελεί ένα ερώτημα στη βάση και από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης το οποίο είναι ίδιο με αυτό της **fillHotelStatsAdmin.**

2.2.17 Μέθοδος: fillHotelStatsAdminCountryDate

Είναι υπεύθυνη να ανακτά το πλήθος των κρατήσεων που βρίσκονται στον πίνακα **booking** που πληρούν τις επιλογές αναζήτησης του χρήστη. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<filter>all</filter>
<dateSt>username</dateSt>
<dateNd>Greece</dateNd>
<country>Greece</country>
</root>
```

- **filter –** Το φίλτρο για τις πληρωμενες, μη πληρωμένες ή ολες τις κρατήσεις
- **dateSt –** Η ημερομηνία εναρξης της περιόδου αναζήτησης

- **dateNd –** Η ημερομηνια λήξης της περιόδου αναζήτησης
- **country –** Η χωρα που ανήκουν τα ξενοδοχεία των οποίων ψάχνουμε τις κρατήσεις

Αναλόγως των επιλογών του χρήστη εκτελεί ένα ερώτημα στη βάση και από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης το οποίο είναι ίδιο με αυτό της **fillHotelStatsAdmin.**

2.2.17 Μέθοδος: fillHotelStatsAdminNameDate

Είναι υπεύθυνη να ανακτά το πλήθος των κρατήσεων που βρίσκονται στον πίνακα **booking** που πληρούν τις επιλογές αναζήτησης του χρήστη. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<filter>all</filter>
<dateSt>username</dateSt>
<dateNd>Greece</dateNd>
<user>username</user>
</root>
```

- **filter –** Το φίλτρο για τις πληρωμενες, μη πληρωμένες ή ολες τις κρατήσεις
- **dateSt –** Η ημερομηνία εναρξης της περιόδου αναζήτησης
- **dateNd –** Η ημερομηνια λήξης της περιόδου αναζήτησης
- **user –** Το username του χρήστη που ανήκουν τα ξενοδοχεία των οποίων ψάχνουμε τις κρατήσεις

Αναλόγως των επιλογών του χρήστη εκτελεί ένα ερώτημα στη βάση και από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης το οποίο είναι ίδιο με αυτό της **fillHotelStatsAdmin.**

2.2.17 Μέθοδος: fillHotelStatsAdminCountryNameDate

Είναι υπεύθυνη να ανακτά το πλήθος των κρατήσεων που βρίσκονται στον πίνακα **booking** που πληρούν τις επιλογές αναζήτησης του χρήστη. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<filter>all</filter>
<dateSt>username</dateSt>
<dateNd>Greece</dateNd>
<user>username</user>
<country>Greece</country>
</root>
```

- **filter –** Το φίλτρο για τις πληρωμενες, μη πληρωμένες ή ολες τις κρατήσεις

- **dateSt –** Η ημερομηνία εναρξης της περιόδου αναζήτησης
- **dateNd –** Η ημερομηνια λήξης της περιόδου αναζήτησης
- **user –** Το username του χρήστη που ανήκουν τα ξενοδοχεία των οποίων ψάχνουμε τις κρατήσεις
- **country –** Η χωρα που ανήκουν τα ξενοδοχεία των οποίων ψάχνουμε τις κρατήσεις

Αναλόγως των επιλογών του χρήστη εκτελεί ένα ερώτημα στη βάση και από την απάντηση που λαμβάνει χτίζει το μήνυμα απάντησης το οποίο είναι ίδιο με αυτό της **fillHotelStatsAdmin.**

2.2.17 Μέθοδος: fillHotelDistanceFinal

Είναι υπεύθυνη να ανακτά τις αποστάσεις ενός ξενοδοχείου από τα σημεία ενδιαφέροντος. Τα δεδομένα βρίσκονται στον πίνακα **distances**. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

- **id –** Ο αυξων αριθμός του ξενοδοχείου

Επικοινωνεί με την **selectHotelDistances** δίνοντας της σαν όρισμα τη τιμή της μεταβλητής. Από τα αποτελέσματα της procedure χτίζει το μήνυμα απάντησης που έχει τη μορφή:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<station>Airport</station>
<distance>10</distance>
</root>
```

2.2.17 Μέθοδος: fillSelectedFacilitiesFinal

Είναι υπεύθυνη να ανακτά τα facilities ενός ξενοδοχείου. Τα δεδομένα βρίσκονται στον πίνακα **hotelfacilities**. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

- **id –** Ο αυξων αριθμός του ξενοδοχείου

Επικοινωνεί με την **selectHotelFacilities** δίνοντας της σαν όρισμα τη τιμή της μεταβλητής. Από τα αποτελέσματα της procedure χτίζει το μήνυμα απάντησης που έχει τη μορφή:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<facility_name>Bar</facility_name>
</root>
```

2.2.17 Μέθοδος: fillRoomNumber

Είναι υπεύθυνη να ανακτά τους τύπους δωματίων ενός ξενοδοχείου καθώς και το πλήθος τους. Τα δεδομένα βρίσκονται στον πίνακα **numberofrooms**. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

- **id –** Ο αυξων αριθμός του ξενοδοχείου

Εκτελεί ένα ερώτημα στη βάση δίνοντας της σαν όρισμα τη τιμή της μεταβλητής. Από τα αποτελέσματα που δέχεται χτίζει το μήνυμα απάντησης που έχει τη μορφή:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>4</id>
<number>34</number>
<flag>0</flag>
<smoke>0</smoke>
<room>4</room>
</root>
```

- **id –** Ο αυξων αριθμός του δωματίου
- **number –** Το πλήθος των δωματίων
- **flag –** Αν είναι ενεργοποιημένο ή όχι
- **smoke –** Αν πρόκειται για δωμάτιο καπνιστών
- **room –** Το όνομα του τύπου δωματίου

2.2.17 Μέθοδος: fillSelectedRoomFacilities

Είναι υπεύθυνη να ανακτά τα facilities του επιλεγμένου τύπου δωματίου ενός ξενοδοχείου. Τα δεδομένα βρίσκονται στον πίνακα **hotelroomfacilities**. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
```

<RoomFacilities>2</RoomFacilities>
</root>

- **id –** Ο αυξων αριθμός του ξενοδοχείου
- **RoomFacilities –** Ο επιλεγμένος τύπος δωματίου

Καλει την Porcedure **selectHotelRoomFacilities** δίνοντας της σαν όρισμα τις τιμές των μεταβλητών. Από τα αποτελέσματα που δέχεται χτίζει το μήνυμα απάντησης που έχει τη μορφή:

<?xml version="1.0" encoding="UTF-8"?>
<root>
<RoomFacility>Mini Bar</RoomFacility>
</root>

- **RoomFacility –** Ο τύπος facility που περιλαμβάνεται στο δωμάτιο του ξενοδοχείου.

2.2.17 Μέθοδος: fillStopSalesHotel

Είναι υπεύθυνη να ανακτά τα stopsales ενός ξενοδοχείου. Τα δεδομένα βρίσκονται στον πίνακα **stopsales**. Το όρισμα που δέχεται είναι της μορφής:

<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>

- **id –** Ο αυξων αριθμός του ξενοδοχείου

Καλει την procedure **selectStopSales** δίνοντας της σαν όρισμα τη τιμή της μεταβλητής. Από τα αποτελέσματα που δέχεται χτίζει το μήνυμα απάντησης που έχει τη μορφή:

<?xml version="1.0" encoding="UTF-8"?>
<root>
<date_start>2010-02-20</date_start>
<date_end>2010-02-25</date_end >
<id>5</id>
<flag>0</flag>
</root>

- **date_start –** Η ημερομηνία έναρξης του stopsale
- **date_end–** Η ημερομηνία λήξης του stopsale
- **id –** Ο αύξων αριθμός της καταγραφής στη βάση
- **flag –** Εάν το stopsale είναι ενεργοποιημένο ή οχι

2.2.17 Μέθοδος: fillRelease

Είναι υπεύθυνη να ανακτά τα release policies ενός ξενοδοχείου. Τα δεδομένα βρίσκονται στον πίνακα **releaseh**. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

- **id –** Ο αυξων αριθμός του ξενοδοχείου

Εκτελεί ένα ερώτημα στη βάση δίνοντας τη τιμή της μεταβλητής. Από τα αποτελέσματα που δέχεται χτίζει το μήνυμα απάντησης που έχει τη μορφή:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<start>2010-02-20</start>
<end>2010-02-25</end >
<id>5</id>
<days>5</days>
</root>
```

- **start –** Η ημερομηνία έναρξης του policy
- **end–** Η ημερομηνία λήξης του policy
- **id –** Ο αύξων αριθμός της καταγραφής στη βάση
- **days –** Ο αριθμός των ημερών για να ενεργοποιηθεί το Policy

2.2.17 Μέθοδος: fillHotelEarlyBooking

Είναι υπεύθυνη να ανακτά τα early booking policies ενός ξενοδοχείου. Τα δεδομένα βρίσκονται στον πίνακα **earlybooking**. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

- **id –** Ο αυξων αριθμός του ξενοδοχείου

Καλει τη procedure **selectHotelEarlyBooking** δίνοντας τη τιμή της μεταβλητής. Από τα αποτελέσματα που δέχεται χτίζει το μήνυμα απάντησης που έχει τη μορφή:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<date_start>2010-02-20</date_start>
<date_end>2010-02-25</date_end>
<ebid>5</ebid>
<price>45</price>
<offerstart>2010-02-26</offerstart>
<offerend>2010-02-28</offerend>
</root>
```

- **date_start –** Η ημερομηνία έναρξης του policy
- **date_end–** Η ημερομηνία λήξης του policy
- **ebid –** Ο αύξων αριθμός της καταγραφής στη βάση
- **price –** Η τιμή που θα ισχύσει όταν ενεργοποιηθεί το Policy
- **offerstart –** Η ημερομηνία ισχύος του Policy
- **offerend –** η ημερομηνια λήξης ισχύος του Policy

2.2.17 Μέθοδος: fillHotelFoto

Είναι υπεύθυνη να ανακτά τις φωτογραφίες ενός ξενοδοχείου. Τα δεδομένα βρίσκονται στον πίνακα **hotelfoto**. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

- **id –** Ο αυξων αριθμός του ξενοδοχείου

Καλει τη procedure **selectHotelFoto** δίνοντας τη τιμή της μεταβλητής. Από τα αποτελέσματα που δέχεται χτίζει το μήνυμα απάντησης που έχει τη μορφή:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<foto>http://www.borker-lodge.com/hotelsData/HotelFotos/2/1.jpg</foto>
</root>
```

- **foto –** Το Url της φωτογραφίας

2.2.17 Μέθοδος: fillAvailable

Είναι υπεύθυνη να ανακτά τις διαθεσιμότητες των δωματίων ενός ξενοδοχείου. Τα δεδομένα βρίσκονται στον πίνακα **available**. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

- **id –** Ο αυξων αριθμός του ξενοδοχείου

Καλει τη procedure **selectAvailableHotel** δίνοντας τη τιμή της μεταβλητής. Από τα αποτελέσματα που δέχεται χτίζει το μήνυμα απάντησης που έχει τη μορφή:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<date_start>2010-02-20</date_start>
<date_end>2010-02-25</date_end>
<id>5</id>
<price>45</price>
<room>Double</room>
<flag>0</flag>
<person>2010-02-28</person>
<food>HalfBoard</food>
</root>
```

- **date_start –** Η ημερομηνία έναρξης της διαθεσιμότητας
- **date_end–** Η ημερομηνία λήξης της διαθεσιμότητας
- **id –** Ο αύξων αριθμός της καταγραφής στη βάση
- **price –** Η τιμή της διαθεσιμότητας
- **room –** Ο τύπος δωματίου της διαθεσιμότητας
- **flag –** Εάν η διαθεσιμότητα είναι ενεργοποιημένη ή όχι
- **person –** Ο αριθμός των ατόμων που αφορα η διαθεσιμότητα
- **food –** Ο τύπος διατροφής που περιλαμβάνει η διαθεσιμότητα.

2.2.17 Μέθοδος: fillSelectedFoodHotel

Είναι υπεύθυνη να ανακτά τους διαθέσιμους τύπους διατροφής που μπορούν να προσθεθούν σε μια διαθεσιμότητα δωματίου ενός ξενοδοχείου. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<roomsize>Double</rooomsize>
```

<start>2010-02-20</start>
<end>2010-02-22</end>
</root>

- **id –** Ο αυξων αριθμός της διαθεσιμότητας
- **roomsize –** Ο τύπος δωματίου
- **start –** Η ημερομηνία έναρξης της διαθεσιμότητας
- **end –** Η ημερομηνία λήξης της διαθεσιμότητας

Καλει τη procedure **selectHotelFoodHotel** δίνοντας τις τιμές των μεταβλητών. Από τα αποτελέσματα που δέχεται χτίζει το μήνυμα απάντησης που έχει τη μορφή:

<?xml version="1.0" encoding="UTF-8"?>
<root>
<price>45</price>
<flag>0</flag>
<food>HaldBoard</food>
</root>

- **price –** Η τιμή του επιλεγμένου τύπου διατροφής που θα προσθεθεί στη διαθεσιμότητα
- **flag –** Εάν η καταγραφή είναι ενεργοποιημένη ή όχι
- **food –** Ο τύπος διατροφής που θα προσθεθεί στη διαθεσιμότητα αντικαθηστώντας τον προεπιλεγμενο.

2.2.17 Μέθοδος: fillHotelRestriction

Είναι υπεύθυνη να ανακτά τους περιορισμούς ενός ξενοδοχείου. Το όρισμα που δέχεται είναι της μορφής:

<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>

- **id –** Ο αυξων αριθμός του ξενοδοχείου

Εκτελεί ένα ερώτημα στη βάση δίνοντας τη τιμη της μεταβλητης. Από τα αποτελέσματα που δέχεται χτίζει το μήνυμα απάντησης που έχει τη μορφή:

<?xml version="1.0" encoding="UTF-8"?>
<root>
<date_start>2010-02-20</date_start>
<date_end>2010-02-25</date_end>
<id>5</id>

<price>45</price>
<roomName>Double</roomName>
<days>0</days>
</root>

- **date_start –** Η ημερομηνία έναρξης του περιορισμού
- **date_end–** Η ημερομηνία λήξης του περιορισμού
- **id –** Ο αύξων αριθμός της καταγραφής στη βάση
- **price –** Η τιμή του περιορισμού
- **roomName –** Ο τύπος δωματίου της διαθεσιμότητας
- **days –** Ο αριθμός των ημερών του περιορισμού

2.2.17 Μέθοδος: fillTopCity

Είναι υπεύθυνη να ανακτά τις δημοφιλείς πόλεις του συστήματος. Δεν δέχεται κάποιο ωφέλιμο μήνυμα στο όρισμα.

Θα επικοινωνήσει με τη stored procedure **selecttopcities**. Από τα αποτελέσματα που δέχεται χτίζει το μήνυμα απάντησης που έχει τη μορφή:

<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>22</id>
<city_name></city_name>
<country_name></country_name>
<foto></foto>
<countryID></countryID>
</root>

- **id -** Το id της πόλης
- **city_name –** Το ονομα της πόλης
- **coyntry_name –** Το ονομα της χωρας
- **foto –** Το url της φωτογραφιας της πόλης
- **countryID –** Το id της χώρας

2.2.17 Μέθοδος: fillTopDestination

Είναι υπεύθυνη να ανακτά τα προτεινομενα ξενοδοχεία του συστηματος. Δεν δέχεται κάποιο ωφέλιμο μήνυμα στο όρισμα.

Θα επικοινωνήσει με τη βάση καλόντας stored procedures και εκτελοντας ερωτηματα ανακτώντας τα προτεινόμενα ξενοδοχεία τα οποια εχουν διαθεσιμοτητες και τιμές. Από τα αποτελέσματα που δέχεται χτίζει το μήνυμα απάντησης που έχει τη μορφή:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<name>Airport</name>
<id>23</id>
<city_name></city_name>
<country_name></country_name>
<stars></stars>
<foto></foto>
<suggestion></suggestion>
<sugname></sugname>
<city_name1></city_name1>
<country_name1></country_name1>
<countryID1></countryID1>
<foto1></foto1>
<id1></id1>
<relativepath></relativepath>
<serverpath></relativepath>
</root>
```

- **name –** Το ονομα του ξενοδοχειου
- **id -** Το id του ξενοδοχείου
- **city_name –** Το ονομα της πόλης του ξενοδοχείου
- **coyntry_name –** Το ονομα της χωρας του ξενοδοχείου
- **stars –** Τα αστερια του ξενοδοχείου
- **foto –** Το url της φωτογραφιας του ξενοδοχείου
- **relativepath –** Το relative path των φωτογραφιων
- **serverpath –** Το server path των φωτογραφιων

2.2.17 Μέθοδος: fillHotelPage

Είναι υπεύθυνη να ανακτά ολες τις πληροφορίες ενός ξενοδοχείου όπως επισης και τη διαθεσιμοτητα των δωματιων με τις τιμες τους για μια συγκεκριμενη περιοδο ημερομηνιων.
Το xml που δέχεται είναι της μορφης:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id></id>
<start></start>
<end></end>
```

<exchange></exchange>
<exchangeicon></exchangeicon>
</root>

- **id –** Το id του ξενοδοχείου
- **start –** Η ημερομηνια εναρξης αναζητησης του χρηστη
- **end –** Η ημερομηνια λήξης αναζητησης του χρηστη
- **exchange –** Το νομισμα που εχει επιλεξει ο χρηστης
- **exchangeIcon –** Το εικονιδιο του νομισματος

Θα επικοινωνήσει με τη βάση καλόντας stored procedures και εκτελοντας ερωτηματα ανακτώντας τις πληροφοριες του ξενοδοχειου καθως και τις τιμες και τις διαθεσιμοτητες για τις επιλεγμενες ημερομηνιες. Από τα αποτελέσματα που δέχεται χτίζει το μήνυμα απάντησης που έχει τη μορφή:

<?xml version="1.0" encoding="UTF-8"?>
<root>
<name>Hotel Name</name>
<chain>Chain</chain>
<country>Greece</country>
<city>Chania</city>
<rating>8</rating>
<star>5</star>
<totalrooms>123</totalrooms>
<description>Description</description>
<bedchild>Policy</bedchild>
<pets>Allowed</pets>
<room>room</room>
<tapa>10</tapa>
<diafora>diafora</diafora>
<price>34</price>
<hid>3</hid>
<avid>33</avid>
<food>Halfboard</food>
<persons>2</persons>
<release>policy</release>
<fake>fake</fake>
<hotelnear>nearhotels</hotelnear>
<hotelnearid>nearids</hotelnearid>
<hotelnearstars>nearstars</hotelnearstars>
<checkin>10:00</checkin>
<SuperTapa>20</SuperTapa>
<checkout>12:00</checkout>
<internet>wireless</internet>
<services>services</services>
<chargingpolicy>policy</chargingpolicy>
<foto>fotoUrl</foto>
<genfoto>mainfotourl</genfoto>
<stayingtype>hotel</stayingtype>

<address>address</address>
<lon>23.454545</lon>
<lat>34.232312</lat>
<facility>facility</facility>
<location>location</location>
<dist>4</dist>
<station>airport</station>
<roomFac>facilities</roomFac>
<roomFood>foods</roomFood>
<extradiscription>extras</extradiscription>
<extratitle>titles</extratitle>
<extraprice>prices</extraprice>
</root>

- **name –** Το ονομα του ξενοδοχειου
- **chain –** Η αλυσιδα του ξενοδοχειου
- **country –** Η χωρα του ξενοδοχειου
- **city –** Η πολη του ξενοδοχειου
- **rating –** Η βαθμολογια του ξενοδοχειου
- **star –** Τα αστερια του ξενοδοχειου
- **totalrooms –** Ο συνολικος αριθμος των δωματιων του ξενοδοχειου
- **description –** Η περιγραφη του ξενοδοχειου
- **bedchild –** Η πολιτικη των παραπανησιων κρεβατιων και παιδιων
- **pets –** Η πολιτικη των κατοικιδιων
- **room -**
- **tapa –** Το ποσοστο του κερδους
- **diafora –** Η διαφορα του κερδους από το κοστος
- **price –** Η τιμη της διαθεσιμοτητας
- **hid –** Το id του ξενοδοχειου
- **avid –** Το id της διαθεσιμοτητας
- **food –** Ο τυπος διατροφης της διαθεσιμοτητας
- **persons –** Ο αριθμος των ατομων της διαθεσιμοτητας
- **release –** Η release policy του ξενοδοχειου
- **hotelnear –** Τα ξενοδοχεια που βρισκονται κοντα στο επιλεγμενο
- **hotelnearid –** Τα id των κοντινων ξενοδοχειων
- **hotelnearstars –** Τα αστερια των κοντινων ξενοδοχειων
- **checkin –** Το checkin του ξενοδοχειου
- **superTapa –** Το ποσοστο κερδους από το ξενοδοχειο
- **checkout –** Το checkout του ξενοδοχειου
- **internet –** Οι τυποι internet που παρεχει το ξενοδοχειο
- **services –** Οι υπηρεσιες του ξενοδοχειου
- **chargingpolicy –** Η πολιτικη ακυρωσεων
- **foto –** Οι φωτογραφιες του ξενοδοχειου
- **genfoto –** Η κεντρικη φωτογραφια του ξενοδοχειου
- **stayingtype –** Ο τυπος διαμονης του ξενοδοχειου
- **address –** Η διευθυνση του ξενοδοχειου
- **lon –** Το γεωγραφικο πλατος της τοποθεσιας του ξενοδοχειου
- **lat –** Το γεωγραφικο μηκος της τοποθεσιας του ξενοδοχειου

- **facility –** Τα facilities του ξενοδοχειου
- **location –** Η τοποθεσια του ξενοδοχειου
- **dist –** Οι αποστασεις από τα σημεια ενδιαφεροντος
- **station –** Οι σταθμοι ενδιαφεροντος
- **roomFac –** Τα facilities των δωματιων
- **roomFood –** Οι τυποι διατροφης των δωματιων
- **extradiscription –** Η περιγραφη των εξτρα υπηρεσιων
- **extratitle –** Ο τιτλος των εξτρα υπηρεσιων
- **extraprice –** Η τιμη των εξτρα υπηρεσιων

2.3 WebService: hoteldbDelete

2.3.1 Μέθοδος: deleteCountry

Είναι υπεύθυνη για την απενεργοποίηση/ενεργοποίηση μιας επαρχίας στη βάση δεδομένων. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<region_name>22</region_name>
</root>
```

Όπου **region_name** το id της επαρχίας την οποία θέλουμε να απενεργοποιήσουμε / ενεργοποιήσουμε.
Στη συνέχεια επικοινωνεί με τη stored procedure **“deleteregion”** της βάσης δεδομένων και της δίνει σαν όρισμα την τιμή του **region_name** πεδίου. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.3.2 Μέθοδος: deleteRegion

Είναι υπεύθυνη για την απενεργοποίηση/ενεργοποίηση μιας επαρχίας στη βάση δεδομένων. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
```

<region_name>22</region_name>
</root>

Όπου **region_name** το id της επαρχίας την οποία θέλουμε να απενεργοποιήσουμε / ενεργοποιήσουμε.
Στη συνέχεια επικοινωνεί με τη stored procedure **“deleteregion”** της βάσης δεδομένων και της δίνει σαν όρισμα την τιμή του **region_name** πεδίου. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.3.3 Μέθοδος: deleteCity

Είναι υπεύθυνη για την απενεργοποίηση/ενεργοποίηση μιας πόλης στη βάση δεδομένων. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>22</id>
</root>

Όπου **id** ο αριθμός της πόλης την οποία θέλουμε να απενεργοποιήσουμε / ενεργοποιήσουμε.
Στη συνέχεια επικοινωνεί με τη stored procedure **“deletecity”** της βάσης δεδομένων και της δίνει σαν όρισμα την τιμή του **id** πεδίου. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.3.4 Μέθοδος: deleteLocation

Είναι υπεύθυνη για την απενεργοποίηση/ενεργοποίηση μιας τοποθεσίας στη βάση δεδομένων. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>22</id>
</root>

Όπου **id** ο αριθμός της τοποθεσίας την οποία θέλουμε να απενεργοποιήσουμε / ενεργοποιήσουμε.
Στη συνέχεια επικοινωνεί με τη stored procedure **“deletelocation”** της βάσης δεδομένων και της δίνει σαν όρισμα την τιμή του **id** πεδίου. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.3.5 Μέθοδος: deleteCardType

Είναι υπεύθυνη για την απενεργοποίηση/ενεργοποίηση ενός τύπου κάρτας στη βάση δεδομένων. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<cardType>22</cardType>
</root>
```

Όπου **cardType** ο αριθμός του τύπου τον οποίο θέλουμε να απενεργοποιήσουμε / ενεργοποιήσουμε.
Στη συνέχεια επικοινωνεί με τη stored procedure **"deletecardtype"** της βάσης δεδομένων και της δίνει σαν όρισμα την τιμή του **cardType** πεδίου. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.3.6 Μέθοδος: deleteExchangeType

Είναι υπεύθυνη για την απενεργοποίηση/ενεργοποίηση ενός νομίσματος στη βάση δεδομένων. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<currency>22</currency>
</root>
```

Όπου **currency** ο αριθμός του νομίσματος το οποίο θέλουμε να απενεργοποιήσουμε / ενεργοποιήσουμε.
Στη συνέχεια επικοινωνεί με τη stored procedure **"deleteexchangetype"** της βάσης δεδομένων και της δίνει σαν όρισμα την τιμή του **currency** πεδίου. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.3.7 Μέθοδος: deleteDistanceFrom

Είναι υπεύθυνη για την απενεργοποίηση/ενεργοποίηση ενός σημείου ενδιαφέροντος στη βάση δεδομένων. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<station>22</station>
</root>
```

Όπου **station** ο αριθμός του σημείου το οποίο θέλουμε να απενεργοποιήσουμε / ενεργοποιήσουμε.
Στη συνέχεια επικοινωνεί με τη stored procedure **“deletedistancefrom”** της βάσης δεδομένων και της δίνει σαν όρισμα την τιμή του **station** πεδίου. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.3.8 Μέθοδος: deleteUserHotel

Είναι υπεύθυνη για την απενεργοποίηση/ενεργοποίηση ενός χρήστη στη βάση δεδομένων. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<User>2</User>
</root>
```

Όπου **User** ο αριθμός του χρήστη τον οποίο θέλουμε να απενεργοποιήσουμε / ενεργοποιήσουμε.
Στη συνέχεια επικοινωνεί με τη stored procedure **“deleteuserhotel”** της βάσης δεδομένων και της δίνει σαν όρισμα την τιμή του **User** πεδίου. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.3.9 Μέθοδος: deleteStayingType

Είναι υπεύθυνη για την απενεργοποίηση/ενεργοποίηση ενός τύπου ξενοδοχείου στη βάση δεδομένων. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<stayingtype>2</stayingtype>
</root>
```

Όπου **stayingtype** ο αριθμός του τύπου τον οποίο θέλουμε να απενεργοποιήσουμε / ενεργοποιήσουμε.
Στη συνέχεια επικοινωνεί με τη stored procedure **“deletestayingtype”** της βάσης δεδομένων και της δίνει σαν όρισμα την τιμή του **stayingtype** πεδίου. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.3.10 Μέθοδος: deleteRoomSize

Είναι υπεύθυνη για την απενεργοποίηση/ενεργοποίηση ενός τύπου δωματίου στη βάση δεδομένων. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

Όπου **id** ο αριθμός του τύπου τον οποίο θέλουμε να απενεργοποιήσουμε / ενεργοποιήσουμε.
Στη συνέχεια επικοινωνεί με τη stored procedure **“deleteroomsize”** της βάσης δεδομένων και της δίνει σαν όρισμα την τιμή του **id** πεδίου. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.3.11 Μέθοδος: deleteInternetType

Είναι υπεύθυνη για την απενεργοποίηση/ενεργοποίηση ενός τύπου internet στη βάση δεδομένων. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<internet>2</internet>
</root>
```

Όπου **internet** ο αριθμός του τύπου τον οποίο θέλουμε να απενεργοποιήσουμε / ενεργοποιήσουμε.
Στη συνέχεια επικοινωνεί με τη stored procedure **“deleteinternettype”** της βάσης δεδομένων και της δίνει σαν όρισμα την τιμή του **internet** πεδίου. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.3.12 Μέθοδος: deleteFacility

Είναι υπεύθυνη για την απενεργοποίηση/ενεργοποίηση ενός hotel facility στη βάση δεδομένων. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<facility_id>2</facility_id>
</root>
```

Όπου **facility_id** ο αριθμός του facility τον οποίο θέλουμε να απενεργοποιήσουμε / ενεργοποιήσουμε.
Στη συνέχεια επικοινωνεί με τη stored procedure **“deletefacilities”** της βάσης δεδομένων και της δίνει σαν όρισμα την τιμή του **facility_id** πεδίου. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.3.13 Μέθοδος: deleteRoomFac

Είναι υπεύθυνη για την απενεργοποίηση/ενεργοποίηση ενός room facility στη βάση δεδομένων. Παίρνει σαν όρισμα ένα String κωδικοποιημένο σε Base64 το οποίο εμπεριέχει ένα xml έγγραφο της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

Όπου **id** ο αριθμός του facility τον οποίο θέλουμε να απενεργοποιήσουμε / ενεργοποιήσουμε. Στη συνέχεια επικοινωνεί με τη stored procedure **"deleteroomfacilities"** της βάσης δεδομένων και της δίνει σαν όρισμα την τιμή του **id** πεδίου. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.3.13 Μέθοδος: deleteHotelDistance

Είναι υπεύθυνη για την διαγραφή μιας απόστασης για ένα ξενοδοχείο από τη βάση δεδομένων. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<station>Airport</station>
</root>
```

- **id –** Ο αυξων αριθμός του ξενοδοχείου
- **station –** Το επιλεγμένο σημείο ενδιαφέροντος προς διαγραφή

Στη συνέχεια επικοινωνεί με τη stored procedure **"deleteHotelDistances"** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές του ορίσματος. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.3.13 Μέθοδος: deleteHotelFacEdit

Είναι υπεύθυνη για την διαγραφή ενός facility ξενοδοχείου από τη βάση δεδομένων. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<facility>Bar</facility>
</root>
```

- **id –** Ο αυξων αριθμός του ξενοδοχείου
- **facility –** Το επιλεγμένο facility προς διαγραφή

Στη συνέχεια επικοινωνεί με τη stored procedure **“deleteHotelFacilities”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές του ορίσματος. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.3.13 Μέθοδος: deleteRoomNumber

Είναι υπεύθυνη για την ενεργοποίηση/απενεργοποίηση μιας καταγραφής για το πλήθος δωματίων ενος ξενοδοχείου από τη βάση δεδομένων. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<room>Double</room>
<smoke>0</smoke>
</root>
```

- **id –** Ο αυξων αριθμός του ξενοδοχείου
- **room –** Ο επιλεγμένος τύπος δωματίου
- **smoke –** Εάν προκειται για τα δωμάτια καπνιστών από τον επιλεγμένο τύπο

Στη συνέχεια επικοινωνεί με τη stored procedure **“deleteNumberOfRooms”** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές του ορίσματος. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.3.13 Μέθοδος: deleteRoomFacHotel

Είναι υπεύθυνη για την διαγραφή ενός facility από ένα δωματιο ενος ξενοδοχείου από τη βάση δεδομένων. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<roomfacility>Mini Bar</roomfacility>
<roomsize>Double</roomsize>
</root>
```

- **id –** Ο αυξων αριθμός του ξενοδοχείου

- **roomfacility –** Το facility προς διαγραφή
- **roomsize –** Ο τύπος δωματίου από τον οποίο θα διαγραφεί το facility

Στη συνέχεια επικοινωνεί με τη stored procedure **"deleteHotelRoomFacilities"** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές του ορίσματος. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

3.13 Μέθοδος: deleteStopSales

Είναι υπεύθυνη για την ενεργοποίηση/απενεργοποίηση μιας καταγραφής stopsale ενος ξενοδοχείου από τη βάση δεδομένων. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

- **id –** Ο αυξων αριθμός του stopsale

Στη συνέχεια επικοινωνεί με τη stored procedure **"deleteStopSalesFlag"** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές του ορίσματος. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.3.13 Μέθοδος: deleteHotelRelease

Είναι υπεύθυνη για την διαγραφή ενα release policy ενος ξενοδοχείου από τη βάση δεδομένων. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

- **id –** Ο αυξων αριθμός του policy

Στη συνέχεια εκτελεί ένα ερώτημα στη βάσης δεδομένων και διαγράφει τη καταγραφή με την τιμη του ορίσματος. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

3.13 Μέθοδος: deleteHotelEarlyBooking

Είναι υπεύθυνη για την απενεργοποίηση μιας προσφορας early booking ενος ξενοδοχείου από τη βάση δεδομένων. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

- **id –** Ο αυξων αριθμός του early booking

Στη συνέχεια επικοινωνεί με τη stored procedure **"deleteHotelEarlyBooking"** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές του ορίσματος. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.3.13 Μέθοδος: deleteHotelFoto

Είναι υπεύθυνη για την διαγραφή μιας φωτογραφίας ενος ξενοδοχείου από τη βάση δεδομένων. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<foto>fotoUrl</foto>
</root>
```

- **id –** Ο αυξων αριθμός του policy
- **foto –** Το url της φωτογραφίας

Στη συνέχεια επικοινωνεί με τη stored procedure **"deleteHotelFoto"** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές του ορίσματος. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.3.13 Μέθοδος: deleteAvailable

Είναι υπεύθυνη για την ενεργοποίηση/απενεργοποίηση μιας διαθεσιμότητας ενος ξενοδοχείου από τη βάση δεδομένων. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

- **id –** Ο αυξων αριθμός της διαθεσιμότητας

Στη συνέχεια επικοινωνεί με τη stored procedure **"deleteAvailable"** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές του ορίσματος. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.3.13 Μέθοδος: deleteHotelFood

Είναι υπεύθυνη για την ενεργοποίηση/απενεργοποίηση ενός τύπου διατροφης μεσα σε μια διαθεσιμότητα ενος ξενοδοχείου από τη βάση δεδομένων. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<roomsize>2</roomsize>
<food>2</food>
<start>2</start>
<end>2</end>
</root>
```

- **id –** Ο αυξων αριθμός του ξενοδοχείου
- **roomsize –** Ο τύπος δωματίου της διαθεσιμοτητας
- **food** - Ο τύπος διατροφής προς διαγραφή
- **start –** Η ημερομηνια εναρξης της διαθεσιμοτητας
- **end –** Η ημερομηνια λήξης της διαθεσιμοτητας

Στη συνέχεια επικοινωνεί με τη stored procedure **"deleteHotelFood"** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές του ορίσματος. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.3.13 Μέθοδος: deleteHotelRestriction

Είναι υπεύθυνη για την ενεργοποίηση/απενεργοποίηση ενός περιορισμού ενος ξενοδοχείου από τη βάση δεδομένων. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

- **id –** Ο αυξων αριθμός του περιορισμού

Στη συνέχεια επικοινωνεί με τη stored procedure **"deleteHotelRestriction"** της βάσης δεδομένων και της δίνει σαν όρισμα τις τιμές του ορίσματος. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.4 WebService: hoteldbOther

2.4.1 Μέθοδος: adminPayAll

Είναι υπεύθυνη να θέσει ως πληρωμένες ένα σύνολο από κρατήσεις. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<id>3</id>
<id>6</id>
</root>
```

- **id –** Ο αυξων αριθμός της κράτησης

Στη συνέχεια εκτελει ένα ερωτημα στη βάση δεδομένων και δίνει τις τιμές του ορίσματος. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.4.1 Μέθοδος: acceptUserAdminMail

Είναι υπεύθυνη να στείλει του διαχειρηστη ένα mail Με την αιτηση του νεου χρήστη για εισαγωγη στο σύστημα. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<fname>FirstName</fname>
<mname>M</mname>
<lname>LastName</lname>
<email>someEmail</email>
<country>Greece</country>
<region>Region</region>
<city>Chania</city>
<phone>+3403434</phone>
<address>Address</address>
<position>manager</position>
<department>sales</department>
<gender>male</gender>
<username>username</username>
<password>pass</password>
<md5username>asdadq32e3wd21e3wds</md5username>
<md5>asd23e23ddr23r23wdwr23er233</md5>
</root>
```

- **fname –** Το όνομα του χρήστη
- **mname –** Το μεσαιο ονομα του χρήστη
- **lname -** Το επίθετο του χρήστη
- **email –** Το email του χρήστη
- **country –** Η χωρα του χρηστη
- **region –** Η επαρχια του χρήστη
- **city –** Η πολη του χρηστη
- **phone –** Το τηλεφωνο του χρηστη
- **address –** Η διευθυνση του χρηστη
- **position –** Η θεση που καταλαμβανει στην εταιρια του
- **department –** Το τμημα της εταιριας στο οποιο υπηρετει
- **gender –** Το φυλο του χρηστη
- **username –** Το Username του χρηστη
- **password –** Ο κωδικος του χρηστη
- **md5Username –** Το hash με το username του χρηστη
- **md5 –** Το hash με το username και τον κωδικο του χρηστη

Στη συνέχεια καλει τη stored procedure **insertNewUserHotel** και δίνει τις βασικες τιμές του ορίσματος. Στη συνέχεια θα στείλει το mail με τα στοιχεια του πελατη και ένα Link για την

ενεργοποιηση του χρηστη. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.4.2 Μέθοδος: adminPay

Είναι υπεύθυνη να αλλάξει τη κατασταση μιας κράτησης ανάμεσα σε πληρωμένη, πληρωμένη η προκαταβολη και μη πληρωμενη. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
</root>
```

- **id –** Ο αυξων αριθμός της κράτησης

Στη συνέχεια εκτελει δυο ερωτηματα στη βάση δεδομένων και δίνει τις τιμές του ορίσματος. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.4.3 Μέθοδος: changePasswordAdmin

Είναι υπεύθυνη να αλλάξει τον κωδικό ενός χρήστη. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<id>2</id>
<password>pass</password>
<md5>md5Hash</md5>
</root>
```

- **id –** Ο αυξων αριθμός του χρήστη
- **password –** Ο νεος κωδικός του χρήστη
- **md5 –** Το hash του νεου κωδικού

Στη συνέχεια εκτελει τη stored procedure **“changepass”** και της δίνει τις τιμές του ορίσματος. Η μέθοδος αυτή δεν επιστρέφει κάποια ωφέλιμη τιμή, αντίθετα στη περίπτωση που παρουσιαστεί κάποιο σφάλμα κατά την εκτέλεση του προγράμματος θα εκτυπωθεί στο log του server το μήνυμα σφάλματος.

2.4.1 Μέθοδος: sendContactMail

Είναι υπεύθυνη να στείλει του διαχειρηστη ένα mail με το μήμυμα επικοινωνιας που συμπληρωσε ο πελατης από την αντιστοιχη φόρμα του site. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<fname>FirstName</fname>
<lname>LastName</lname>
<email>someEmail</email>
<contactTopic>Topic</contactTopic>
<contactMessage>Message</contactMessage>
</root>
```

- **fname –** Το όνομα του χρήστη
- **lname -** Το επίθετο του χρήστη
- **email –** Το email του χρήστη
- **contactTopic –** Το θέμα του μηνυματος επικοινωνιας
- **contactMessage –** Το μήνυμα επικοινωνιας

Στη συνέχεια θα στείλει το mail επικοινωνιας στον διαχειρηστη. Η μέθοδος επιστρέφει έναν κωδικο με το αποτελεσμα της διαδικασιας. Το xml απαντησης είναι της μορφης:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<result>0</result>
</root>
```

- **result -** Ο κωδικός του απότελέσματος της αποστολής

2.4.1 Μέθοδος: amendment

Είναι υπεύθυνη να στείλει να ανακτησει τα δεδομενα μιας κράτησης εφόσον τα στοιχεια που εισήγαγε ο χρήστης είναι σωστά. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
```

<refID>refid</refID>
<bookID>bookid</bookID>
</root>

- **refID –** Το id της κρατησης
- **bookID -** Το id του πελατη

Στη συνέχεια θα εκτελεσει ένα ερωτημα στη βάση και εφοσον τα στοιχεια είναι σωστα θα λάβει τις πληροφοριες της κράτησης τις οποίες θα εισάγει σε ένα μήνυμα και θα το στείλει σαν απάντηση. Το xml είναι της μορφής:

<?xml version="1.0" encoding="UTF-8"?>
<root>
<fname>refid</fname>
<mname>bookid</mname>
<lname></lname>
<date_start></date_start>
<date_end></date_end>
<email></email>
<hotel></hotel>
<days></days>
<hotelprofit></hotelprofit>
<cardtype></cardtype>
<hotelfood></hotelfood>
<room></room>
<roomnum></roomnum>
<cancel></cancel>
</root>

- **fname –** Το ονομα του πελατη
- **mname -** Το μεσαιο ονομα του πελατη
- **lname –** Το επιθετο του πελατη
- **date_start –** Η ημερομηνια εναρξης της κρατησης
- **date_end –** Η ημερομηνια ληξης της κρατησης
- **email –** Το email του πελατη
- **hotel –** Το ονομα του ξενοδοχειου
- **days –** Οι συνολικες μερες της κρατησης
- **hotelprofit –** Το ποσοστο κερδους από τη κρατηση
- **cardtype –** Ο τυπος της πιστωτικης καρτας
- **hotelfood –** Ο τυπος διατροφης της κρατησης
- **room –** Οι τυποι δωματιων της κρατησης
- **roomnum –** Ο αριθμος των δωματιων από κάθε τυπο
- **cancel –** Η πολιτικη ακυρωσης της κρατησης

2.4.1 Μέθοδος: cancelReservation

Είναι υπεύθυνη να ακυρώσει μια κράτηση και να υπολογίσει τα ακυρωτικά κόστη. Εφόσον προκύψουν ακυρωτικά θα επικοινωνήσει με τη τράπεζα και θα επιστρέψει το αντιστοιχο ποσο στο πελάτη. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<merchREF>refid</merchREF>
</root>
```

- **merchREF –** Το id της κρατησης

Στη συνέχεια θα εκτελεσει ένα ερωτημα στη βαση για να παρει τα στοιχεια της κρατησης. Θα καλέσει τη μεθοδο **resolveCancel** η οποία θα υπολογίσει αν υπαρχουν ακυρωτικα και αν βρει θα επικοινωνησει με τη τραπεζα να τα εισπραξει. Επειτα θα ακυρωθει η κράτηση και θα χτιστεί το μήνυμα απαντησης προς το πελατη με το αποτελεσμα της διαδικασιας. Το xml είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<errorCode>0</errorCode>
<errorMessage>Message</errorMessage>
</root>
```

- **errorCode –** Ο κωδικος του αποτελεσματος της διαδικασιας
- **errorMessage -** Το μηνυμα απαντησης του αποτελεσματος

2.4.1 Μέθοδος: sendVaucherMail

Είναι υπεύθυνη να πραγματοποιήσει μια κράτηση στο σύστημα επείτα από αιτηση του πελατη. Η διαδικασια περιλαμβάνει επικοινωνια με τη τράπεζα και χρέωση της πιστωτικης του πελατη, καταχώρηση της κρατησης στη βάση και αποστολλη ενημερωτικών email καθώς και το vaucher στον πελατη. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<hid></hid>
<fname></fname>
<mname></mname>
<lname></lname>
<email></email>
```

<hotelname></hotelname>
<hoteladdress></hoteladdress>
<hotelcity></hotelcity>
<hotelcountry></hotelcountry>
<hotelavID></hotelavID>
<exp></exp>
<card></card>
<cvc></cvc>
<roomNum></roomNum>
<extra></extra>
<bookfoodReserve></bookfoodReserve>
<phone></phone>
<adProfit></adProfit>
<bookdatestart></bookdatestart>
<hotelcheckin></hotelcheckin>
<bookdateend></bookdateend>
<hotelcheckout></hotelcheckout>
<bookrooms></bookrooms>
<bookprice></bookprice>
<bookfood></bookfood>
<bookcanceldate></bookcanceldate>
<bookcancelprice></bookcancelprice>
<hotelcharge></hotelcharge>
<hotelbeds></hotelbeds>
<hotelinternet></hotelinternet>
<hotelpets></hotelpets>
<mobilephone></mobilephone>
<flightinfo></flightinfo>
<release></release>
<numberOfPersons></numberOfPersons>
<personsAges></personsAges>
</root>

- **hid –** Το id του ξενοδοχειου
- **fname -** Το ονομα του πελατη
- **mname -** Το μεσαιο ονομα του πελατη
- **lname -** Το επιθετο το πελατη
- **email -** Το email του πελατη
- **hotelname -** Το ονομα του ξενοδοχειου
- **hoteladdress -** Η διευθυνση του ξενοδοχειου
- **hotelcity -** Η πολη του ξενοδοχειου
- **hotelcountry -** Η χωρα του ξενοδοχειου
- **hotelavID -** Το id της διαθεσιμοτητας που επελεξε να κανει κρατησης ο πελατης
- **exp -** Η ημερομηνια ληξης της καρτας του πελατη
- **card -** Ο αριθμος της καρτας του πελατη
- **cvc -** Ο αριθμος ασφαλειας της καρτας
- **roomNum -** Το πληθος των δωματιων της κρατησης για κάθε τυπο
- **extra -** Οι εξτρα υπηρεσιες που επελεξε ο πελατης
- **bookfoodReserve -** Ο τυπος διατροφης που επελεξε ο πελατης

- **phone** - Ο αριθμος τηλεφωνου του πελατη
- **adProfit** - Το ποσοστο κερδους από τη κρατηση
- **bookdatestart** - Η ημερομηνια εναξης της κρατησης
- **hotelcheckin** - Η ωρα που δεχεται το ξενοδοχειο checkin
- **bookdateend** - Η ημερομηνια ληξης της κρατησης
- **hotelcheckout** - Η ωρα που δεχεται το ξενοδοχειο checkout
- **bookrooms** - Τα δωματια που κράτησε ο πελατης
- **bookprice** - Η τιμη της κρατησης
- **bookfood** - Η επιλεγμενη διατροφη
- **bookcanceldate** - Η ημερομηνια που μπορει να ακυρωθει η κρατηση
- **bookcancelprice** - Η τιμη που θα πληρωσει ο πελατης σε περιπτωση ακυρωσης
- **hotelcharge** - Η πολιτικη ακύρωσης
- **hotelbeds** - Η πολιτικη εξτρα κρεβατιων
- **hotelinternet** - Η πολιτικη internet
- **hotelpets** - Η πολιτικη κατοικιδιων
- **mobilephone** - Το κινητο του πελατη
- **flightinfo** - Οι πληροφοριες πτησης του πελατη
- **release** - Η release policy του ξενοδοχειου
- **numberOfPersons** - Ο αριθμος των ατομων της κρατησης
- **personsAges –** Οι ηλικιες των ατομων

Στη συνέχεια θα ελεχθουν και θα αναλυθουν τα στοιχεια του ορίσματος, θα υπολογίσθει το κόστος της κρατησης και θα γίνει η επικοινωνια με τη τράπεζα για τη χρέωση. Εφόσον η χρέωση είναι επιτυχης θα γίνει η καταχώρηση της κρατησης στη βάση και θα αποσταλουν τα ενημερωτικα email καθως και το vaucher στον πελατη. Τελος η μεθοδος θα επιστρέψει ένα μηνυμα και ένα κωδικο που αντιπροσωπευουν το αποτελεσμα της ολης διαδικασιας. Το xml είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<errorCode>0</errorCode>
<errorMessage>Message</errorMessage>
</root>
```

- **errorCode –** Ο κωδικος του αποτελεσματος της διαδικασιας
- **errorMessage** - Το μηνυμα απαντησης του αποτελεσματος

2.4.1 Μέθοδος: searchHotel

Είναι υπεύθυνη να πραγματοποιήσει μια αναζήτηση για ξενοδοχείο στη βάση με το όρισμα που έλαβε. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<by>Search</by>
</root>
```

- **by –** Το κείμενο της αναζητησης

Θα καλεστεί η stored procedure **searchhotel** και θα της δωθει το όρισμα που λάβαμε, εφόσον η αναζήτηση είναι επιτυχής θα δομηθεί ένα μήνυμα απάντησης με τα ξενοδοχεία που βρέθηκαν. Το xml είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<country>Country</country>
<foto>Url</foto>
<city>City</city>
<hotelid>2</hotelid>
<name>Hotel Name</name>
<rate>23,23</rate>
<stars>5</stars>
</root>
```

- **country –** Η χωρα του ξενοδοχειου
- **foto –** Η φωτογραφια του ξενοδοχειου
- **city –** Η πολη του ξενοδοχειου
- **hotelid –** Το Id του ξενοδοχειου
- **name –** Το ονομα του ξενοδοχειου
- **rate –** Η φθηνοτερη τιμη για μια μερα ενός διαθεσιμου δωματιου του ξενοδοχειου
- **stars –** Τα αστερια του ξενοδοχειου

2.4.1 Μέθοδος: searchNoDates

Είναι υπεύθυνη να πραγματοποιήσει μια αναζήτηση για περιοχη βάση ενός ορίσματος και μιας ενδεικτικής ημερομηνιας εφόσον ο πελάτης δεν επελεξε ημερομηνιες. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<by>Search</by>
<sdate>2010-02-20</sdate>
</root>
```

- **by –** Το κείμενο της αναζητησης
- **sdate –** Η ημερομηνια εναρξης από τις επιλογες του χρηστη

Θα εκτελεστουν κάποια ερωτηματα στη βάση και εφόσον η αναζήτηση είναι επιτυχής θα δομηθεί ένα μήνυμα απάντησης με τους προορισμους που βρέθηκαν. Το xml είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<countries-id ></countries-id >
<countries></countries>
<countries-foto></countries-foto>
<regions-countryid></regions-countryid>
<regions-id></regions-id>
<regions-country></regions-country>
<egions-region></egions-region>
<egions-foto></egions-foto>
<regions-num></regions-num>
<countries-num></countries-num>
<cities-num></cities-num>
<loc-num></loc-num>
<cities-countryid></cities-countryid>
<cities-id></cities-id>
<cities-country></cities-country>
<cities-region></cities-region>
<cities-city></cities-city>
<cities-foto></cities-foto>
<loc-id></loc-id>
<loc-countryid></loc-countryid>
<loc-cityid></loc-cityid>
<loc-country></loc-country>
<loc-region></loc-region>
<loc-city></loc-city>
<loc-location></loc-location>
<loc-foto></loc-foto>
</root>
```

- **countries-id –** Το id της χώρας
- **countries –** Το όνομα της χώρας
- **countries-foto –** Η φωτογραφία της χώρας
- **regions-countryid –** Το id της χώρας της επαρχιας
- **regions-id –** Το id της επαρχιας
- **regions-country –** Το ονομα της χώρας της επαρχιας
- **egions-region –** Το ονομα της επαρχιας
- **egions-foto –** Η φωτογραφια της επαρχιας
- **regions-num –** Ο αριθμός των ξενοδοχείων της επαρχιας

- **countries-num –** Ο αριθμός των ξενοδοχείων της χωρας
- **cities-num –** Ο αριθμός των ξενοδοχείων της πόλης
- **cities-countryid –** Το Id της χωρας της πολης
- **cities-id –** Το Id της πόλης
- **cities-country –** Το ονομα της χωρας της πολης
- **cities-region –** Το ονομα της επαρχιας της πολης
- **cities-city –** Το ονομα της πολης
- **cities-foto -** Η φωτογραφια της πολης
- **loc-id –** Το id της τοποθεσιας
- **loc-countryid –** Το id της χωρας της τοποθεσιας
- **loc-cityid –** Το id της πολης της τοποθεσιας
- **loc-country –** Το ονομα της χωρας της τοποθεσιας
- **loc-region –** Το ονομα της επαρχιας της τοποθεσιας
- **loc-city –** Το ονομα της πολης της τοποθεσιας
- **loc-location –** Το ονομα της τοποθεσιας
- **loc-foto –** Η φωτογραφια της τοποθεσιας

2.4.1 Μέθοδος: search

Είναι υπεύθυνη να πραγματοποιήσει μια αναζήτηση για περιοχη βάση ενός ορίσματος. Η διαδικασια είναι η ιδια με αυτην της searchNoDates οποτε δεν θα αναλυθεί περαιτερω.

2.4.1 Μέθοδος: searchHotelCountry

Είναι υπεύθυνη να πραγματοποιήσει μια αναζήτηση για ξενοδοχεια σε μια συγκεκριμένη χώρα μεσα σε συγκεκριμένες ημερομηνίες. Το όρισμα που δέχεται είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<by>Search</by>
<start></start>
<end></end>
</root>
```

- **by –** Το κείμενο της αναζητησης
- **start –** Η ημερομηνία εναρξης
- **end –** Η ημερομηνια λήξης

Θα εκτελεστουν κάποια ερωτηματα στη βάση και εφόσον η αναζήτηση είναι επιτυχής θα δομηθεί ένα μήνυμα απάντησης με τα ξενοδοχεια που βρέθηκαν. Το xml είναι της μορφής:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<avid>22</avid>
<city>Chania</city>
<country>Greece</country>
<foto>fotoUrl</foto>
<hotelId>2</hotelId>
<name>HotelName</name>
<rate>8</rate>
<stars>5</stars>
<price>45,45</price>
<tapa>10</tapa>
<desc>Description</desc>
</root>
```

- **avid -** Το id της διαθεσιμοτητας
- **city –** Το ονομα της πόλης
- **coyntry –** Το ονομα της χωρας
- **foto –** Το url της φωτογραφιας του ξενοδοχείου
- **hotelid –** Το id του ξενοδοχειου
- **name –** Το ονομα του ξενοδοχείου
- **rate –** Η βαθμολογια του ξενοδοχειου
- **stars –** Τα αστερια του ξενοδοχειου
- **price –** Η τιμη του φθηνοτερου δωματιου
- **tapa –** Το ποσοστο κερδους στη τιμη
- **desc –** Η περιγραφη του ξενοδοχειου

2.4.1 Μέθοδος: searchHotelCountryNoDates

Είναι υπεύθυνη να πραγματοποιήσει μια αναζήτηση για ξενοδοχεια σε μια συγκεκριμένη χώρα μεσα σε ενδεικτικες ημερομηνίες εφόσον ο χρηστης δεν επέλεξε συγκεκριμένες. Η διαδικασία είναι ιδια με αυτή της **searchHotelCountry** οποτε δεν θα αναλυθει περαιτερω.

2.4.1 Μέθοδος: searchHotelRegion

Είναι υπεύθυνη να πραγματοποιήσει μια αναζήτηση για ξενοδοχεια σε μια συγκεκριμένη επαρχια μεσα σε συγκεκριμενες ημερομηνιες. Η διαδικασία είναι ιδια με αυτή της **searchHotelCountry** οποτε δεν θα αναλυθει περαιτερω.

2.4.1 Μέθοδος: searchHotelRegionNoDates

Είναι υπεύθυνη να πραγματοποιήσει μια αναζήτηση για ξενοδοχεια σε μια συγκεκριμένη επαρχια μεσα σε ενδεικτικες ημερομηνίες εφόσον ο χρηστης δεν επέλεξε συγκεκριμένες. Η διαδικασία είναι ιδια με αυτή της **searchHotelCountryNoDates** οποτε δεν θα αναλυθει περαιτερω.

2.4.1 Μέθοδος: searchHotelCity

Είναι υπεύθυνη να πραγματοποιήσει μια αναζήτηση για ξενοδοχεια σε μια συγκεκριμένη πολη μεσα σε συγκεκριμενες ημερομηνιες. Η διαδικασία είναι ιδια με αυτή της **searchHotelCountry** οποτε δεν θα αναλυθει περαιτερω.

2.4.1 Μέθοδος: searchHotelCityNoDates

Είναι υπεύθυνη να πραγματοποιήσει μια αναζήτηση για ξενοδοχεια σε μια συγκεκριμένη πολη μεσα σε ενδεικτικες ημερομηνίες εφόσον ο χρηστης δεν επέλεξε συγκεκριμένες. Η διαδικασία είναι ιδια με αυτή της **searchHotelCountryNoDates** οποτε δεν θα αναλυθει περαιτερω.

2.4.1 Μέθοδος: searchHotelLocation

Είναι υπεύθυνη να πραγματοποιήσει μια αναζήτηση για ξενοδοχεια σε μια συγκεκριμένη τοποθεσια μεσα σε συγκεκριμενες ημερομηνιες. Η διαδικασία είναι ιδια με αυτή της **searchHotelCountry** οποτε δεν θα αναλυθει περαιτερω.

2.4.1 Μέθοδος: searchHotelLocationNoDates

Είναι υπεύθυνη να πραγματοποιήσει μια αναζήτηση για ξενοδοχεια σε μια συγκεκριμένη τοποθεσια μεσα σε ενδεικτικες ημερομηνίες εφόσον ο χρηστης δεν επέλεξε συγκεκριμένες. Η διαδικασία είναι ιδια με αυτή της **searchHotelCountryNoDates** οποτε δεν θα αναλυθει περαιτερω.

Κεφάλαιο 3 - Βάση Δεδομένων

3.1 Stored Procedures

3.1.1 insertCountry

Εισάγει μια χώρα στη βάση. Παίρνει σαν όρισμα ένα **VarChar(45)** που αντιπροσωπεύει το όνομα της χώρας προς εισαγωγή. Χρησιμοποιείται από τη μέθοδο **addCountry** της webservice **“hoteldbAdd”**. Δεν επιστρέφει κάποια τιμή στη μέθοδο που τη κάλεσε.

3.1.2 editCountry

Διορθώνει το όνομα μιας επαρχίας στη βάση. Παίρνει σαν όρισμα δύο μεταβλητές:

1. **VarChar(45)** όπου αντιπροσωπεύει το νέο όνομα της χώρας
2. **BigInt(20)** η οποία αντιπροσωπεύει τον αύξων αριθμό από την αρίθμηση της βάσης και υποδηλώνει τη θέση της χώρας στον πίνακα **country** όπου θα γίνει η αλλαγή.

Χρησιμοποιείται από τη μέθοδο **editCountry** της webservice **“hoteldbAdd”**. Δεν επιστρέφει κάποια τιμή στη μέθοδο που τη κάλεσε.

3.1.3 selectCountryAdmin

Επιστρέφει όλες τις χώρες από τη βάση δεδομένων που βρίσκονται στον πίνακα **country**. Χρησιμοποιείται από τη μέθοδο **fillCountry** της webservice **“hoteldbFill”.**

3.1.4 deleteCountry

Απενεργοποιέι ή Ενεργοποιέι μια χώρα στη βάση δεδομένων. Παίρνει σαν όρισμα μια μεταβλητή **BigInt(20)** η οποία αντιπροσωπεύει τη θέση της χώρας. Χρησιμοποιείται από τη μέθοδο **deleteCountry** της webservice **“hoteldbDelete”.**

3.1.5 insertRegion

Εισάγει μια επαρχία στη βάση. Παίρνει σαν όρισμα δύο μεταβλητές:

1. **VarChar(45)** όπου αντιπροσωπεύει το όνομα της χώρας κάτω από την οποία θα καταχωρηθεί η επαρχία
2. **VarChar(45)** η οποία αντιπροσωπεύει το όνομα της επαρχίας προς εισαγωγή.

Χρησιμοποιείται από τη μέθοδο addRegion της webservice “hoteldbAdd”. Δεν επιστρέφει κάποια τιμή στη μέθοδο που τη κάλεσε.

3.1.6 editRegion

Διορθώνει το όνομα μιας επαρχίας στη βάση. Παίρνει σαν όρισμα δύο μεταβλητές:

1. **VarChar(45)** όπου αντιπροσωπεύει το νέο όνομα της επαρχίας
2. **BigInt(20)** η οποία αντιπροσωπεύει τον αύξων αριθμό από την αρίθμηση της βάσης και υποδηλώνει τη θέση του πίνακα **region** όπου θα γίνει η αλλαγή.

Χρησιμοποιείται από τη μέθοδο **editRegion** της webservice **“hoteldbAdd”**. Δεν επιστρέφει κάποια τιμή στη μέθοδο που τη κάλεσε.

3.1.7 selectRegionAdmin

Επιστρέφει όλες τις επαρχίας μιας χώρας από τη βάση δεδομένων που βρίσκονται στον πίνακα **region**. Παίρνει σαν όρισμα μια μεταβλητή **VarChar(45)** η οποία αντιπροσωπεύει το όνομα της χώρας της οποίας θέλουμε τις επαρχίες. Χρησιμοποιείται από τη μέθοδο **fillRegion** της webservice **“hoteldbFill”.**

3.1.8 deleteRegion

Απενεργοποιέι ή Ενεργοποιέι μια επαρχία στη βάση δεδομένων. Παίρνει σαν όρισμα μια μεταβλητή **BigInt(20)** η οποία αντιπροσωπεύει τη θέση της επαρχίας. Χρησιμοποιείται από τη μέθοδο **deleteRegion** της webservice **“hoteldbDelete”.**

3.1.5 insertCity

Εισάγει μια πόλη στη βάση. Παίρνει σαν όρισμα τις εξής μεταβλητές:

1. **VarChar(45)** όπου αντιπροσωπεύει το όνομα της χώρας κάτω από την οποία θα καταχωρηθεί η πόλη
2. **VarChar(45)** η οποία αντιπροσωπεύει το όνομα της επαρχίας κάτω από την οποία θα καταχωρηθεί η πόλη
3. **VarChar(45)** η οποία αντιπροσωπεύει το όνομα της πόλης προς εισαγωγή.

Χρησιμοποιείται από τη μέθοδο addCity της webservice “hoteldbAdd”. Δεν επιστρέφει κάποια τιμή στη μέθοδο που τη κάλεσε.

3.1.6 editCity

Διορθώνει το όνομα μιας πόλης στη βάση. Παίρνει σαν όρισμα δύο μεταβλητές:

1. **VarChar(45)** όπου αντιπροσωπεύει το νέο όνομα της πόλης
2. **BigInt(20)** η οποία αντιπροσωπεύει τον αύξων αριθμό από την αρίθμηση της βάσης και υποδηλώνει τη θέση του πίνακα **city** όπου θα γίνει η αλλαγή.

Χρησιμοποιείται από τη μέθοδο **editCity** της webservice **“hoteldbAdd”**. Δεν επιστρέφει κάποια τιμή στη μέθοδο που τη κάλεσε.

3.1.7 selectCityAdmin

Επιστρέφει όλες τις πόλεις από τη βάση δεδομένων που βρίσκονται στον πίνακα **city**. Παίρνει σαν όρισμα δύο μεταβλητές:

1. **VarChar(45)** όπου αντιπροσωπεύει το όνομα της χώρας κάτω από την οποία αναζητούμε τις πόλεις
2. **VarChar(45)** το όνομα της επαρχίας στην οποία ανήκουν οι πόλεις που αναζητούμε.

. Χρησιμοποιείται από τη μέθοδο **fillCity** της webservice **“hoteldbFill”.**

3.1.8 deleteCity

Απενεργοποιεί ή Ενεργοποιεί μια πόλη στη βάση δεδομένων. Παίρνει σαν όρισμα μια μεταβλητή **BigInt(20)** η οποία αντιπροσωπεύει τη θέση της πόλης. Χρησιμοποιείται από τη μέθοδο **deleteCity** της webservice **“hoteldbDelete”.**

3.1.9 insertLocation

Εισάγει μια τοποθεσία στη βάση. Παίρνει σαν όρισμα τις εξής μεταβλητές:

1. **VarChar(45)** όπου αντιπροσωπεύει το όνομα της χώρας κάτω από την οποία θα καταχωρηθεί η τοποθεσία
2. **VarChar(45)** το όνομα της επαρχίας κάτω από την οποία θα καταχωρηθεί η τοποθεσία
3. **VarChar(45)** το όνομα της πόλης κάτω από την οποία θα καταχωρηθεί η τοποθεσία
4. **VarChar(45)** η οποία αντιπροσωπεύει το όνομα της τοποθεσίας προς εισαγωγή.

Χρησιμοποιείται από τη μέθοδο **addCity** της webservice **“hoteldbAdd”**. Δεν επιστρέφει κάποια τιμή στη μέθοδο που τη κάλεσε.

3.1.10 editLocation

Διορθώνει το όνομα μιας τοποθεσίας στη βάση. Παίρνει σαν όρισμα δύο μεταβλητές:

1. **VarChar(45)** όπου αντιπροσωπεύει το νέο όνομα της τοποθεσίας
2. **BigInt(20)** η οποία αντιπροσωπεύει τον αύξων αριθμό από την αρίθμηση της βάσης και υποδηλώνει τη θέση του πίνακα **location** όπου θα γίνει η αλλαγή.

Χρησιμοποιείται από τη μέθοδο **editLocation** της webservice **“hoteldbAdd”**. Δεν επιστρέφει κάποια τιμή στη μέθοδο που τη κάλεσε.

3.1.11 selectLocationAdmin

Επιστρέφει όλες τις τοποθεσίες μιας πόλης από τη βάση δεδομένων που βρίσκονται στον πίνακα **location**. Παίρνει σαν όρισμα τρείς μεταβλητές:

1. **VarChar(45)** όπου αντιπροσωπεύει το όνομα της χώρας
2. **VarChar(45)** το όνομα της επαρχίας
3. **VarChar(45)** το όνομα της πόλης στην οποία ανήκουν οι τοποθεσίες που αναζητούμε.

Χρησιμοποιείται από τη μέθοδο **fillLocation** της webservice **“hoteldbFill”.**

3.1.12 deleteLocation

Απενεργοποιέι ή Ενεργοποιέι μια τοποθεσία στη βάση δεδομένων. Παίρνει σαν όρισμα μια μεταβλητή **BigInt(20)** η οποία αντιπροσωπεύει τη θέση της τοποθεσία. Χρησιμοποιείται από τη μέθοδο **deleteLocation** της webservice **“hoteldbDelete”.**

3.1.13 insertCardType

Εισάγει ένα τύπο πιστωτικής κάρτας στη βάση. Παίρνει σαν όρισμα ένα **VarChar(45)** που αντιπροσωπεύει το όνομα του τύπου προς εισαγωγή. Χρησιμοποιείται από τη μέθοδο **addCardType** της webservice **“hoteldbAdd”**. Δεν επιστρέφει κάποια τιμή στη μέθοδο που τη κάλεσε.

3.1.14 editCardType

Διορθώνει το όνομα ενός τύπου κάρτας στη βάση. Παίρνει σαν όρισμα δύο μεταβλητές:

3. **VarChar(45)** όπου αντιπροσωπεύει το νέο όνομα του τύπου
4. **BigInt(20)** η οποία αντιπροσωπεύει τον αύξων αριθμό από την αρίθμηση της βάσης και υποδηλώνει τη θέση του τύπου στον πίνακα **cardtype** όπου θα γίνει η αλλαγή.

Χρησιμοποιείται από τη μέθοδο **editCardType** της webservice **“hoteldbAdd”**. Δεν επιστρέφει κάποια τιμή στη μέθοδο που τη κάλεσε.

3.1.15 deleteCardType

Απενεργοποιέι ή Ενεργοποιέι ένα τύπο πιστωτικής στη βάση δεδομένων. Παίρνει σαν όρισμα μια μεταβλητή **BigInt(20)** η οποία αντιπροσωπεύει τη θέση του τύπου. Χρησιμοποιείται από τη μέθοδο **deleteCardType** της webservice **“hoteldbDelete”.**

3.1.16 selectcardtypeadmin

Επιστρέφει όλες τις χώρες από τη βάση δεδομένων που βρίσκονται στον πίνακα **country**. Χρησιμοποιείται από τη μέθοδο **fillCountry** της webservice **“hoteldbFill”.**

3.1.17 insertexchange

Εισάγει ένα νόμισμα στη βάση. Παίρνει σαν όρισμα δύο μεταβλητές:

3. **VarChar(45)** όπου αντιπροσωπεύει το όνομα της χώρας στην οποία θα αντιστοιχιστεί το νόμισμα.
4. **VarChar(45)** η οποία αντιπροσωπεύει το όνομα του νομίσματος προς εισαγωγή.

Χρησιμοποιείται από τη μέθοδο addExchangeType της webservice “hoteldbAdd”. Δεν επιστρέφει κάποια τιμή στη μέθοδο που τη κάλεσε.

3.1.18 editexchange

Διορθώνει το όνομα ενός νομίσματος στη βάση. Παίρνει σαν όρισμα δύο μεταβλητές:

5. **VarChar(45)** όπου αντιπροσωπεύει το νέο όνομα του νομίσματος
6. **BigInt(20)** η οποία αντιπροσωπεύει τον αύξων αριθμό από την αρίθμηση της βάσης και υποδηλώνει τη θέση του νομίσματος στον πίνακα **exchangetype** όπου θα γίνει η αλλαγή.

Χρησιμοποιείται από τη μέθοδο **editExchangeType** της webservice **"hoteldbAdd"**. Δεν επιστρέφει κάποια τιμή στη μέθοδο που τη κάλεσε.

3.1.19 deleteexchangetype

Απενεργοποιέι ή Ενεργοποιέι ένα νόμισμα στη βάση δεδομένων. Παίρνει σαν όρισμα μια μεταβλητή **BigInt(20)** η οποία αντιπροσωπεύει τη θέση του νομίσματος. Χρησιμοποιείται από τη μέθοδο **deleteExchangeType** της webservice **"hoteldbDelete".**

3.1.20 selectexchangetypedmin

Επιστρέφει όλα τα νομίσματα από τη βάση δεδομένων που βρίσκονται στον πίνακα **exchangetype**. Χρησιμοποιείται από τη μέθοδο **fillExchangeType** της webservice **"hoteldbFill".**

3.1.21 insertdistancefrom

Εισάγει ένα σημείο ενδιαφέροντος στη βάση. Παίρνει σαν όρισμα ένα **VarChar(45)** που αντιπροσωπεύει το όνομα του σημείου προς εισαγωγή. Χρησιμοποιείται από τη μέθοδο **addDistanceFrom** της webservice **"hoteldbAdd"**. Δεν επιστρέφει κάποια τιμή στη μέθοδο που τη κάλεσε.

3.1.22 editdistancefrom

Διορθώνει το όνομα ενός σημείου ενδιαφέροντος στη βάση. Παίρνει σαν όρισμα δύο μεταβλητές:

1. **VarChar(45)** όπου αντιπροσωπεύει το νέο όνομα του σημείου
2. **BigInt(20)** η οποία αντιπροσωπεύει τον αύξων αριθμό από την αρίθμηση της βάσης και υποδηλώνει τη θέση του σημείου στον πίνακα **distancefrom** όπου θα γίνει η αλλαγή.

Χρησιμοποιείται από τη μέθοδο **editDistanceFrom** της webservice **"hoteldbAdd"**. Δεν επιστρέφει κάποια τιμή στη μέθοδο που τη κάλεσε.

3.1.23 selectdistancefromadmin

Επιστρέφει όλα τα σημεία ενδιαφέροντος από τη βάση δεδομένων που βρίσκονται στον πίνακα **distancefrom**. Χρησιμοποιείται από τη μέθοδο **fillDistanceFrom** της webservice **"hoteldbFill".**

3.1.24 deletedistancefrom

Απενεργοποιέι ή Ενεργοποιέι ένα σημείο ενδιαφέροντος στη βάση δεδομένων. Παίρνει σαν όρισμα μια μεταβλητή **BigInt(20)** η οποία αντιπροσωπεύει τη θέση του σημείου στη βάση. Χρησιμοποιείται από τη μέθοδο **deleteDistanceFrom** της webservice **"hoteldbDelete".**

3.1.25 insertrestrictioncountry2

Εισάγει τον περιορισμό ενός χρήστη σε μια ορισμένη χώρα. Παίρνει σαν όρισμα δύο μεταβλητές:

1. **VarChar(45)** όπου αντιπροσωπεύει το συνθηματικό του χρήστη σε κωδικοποιημένη MD5 Hash μορφή.
2. **VarChar(45)** οπου αντιπροσωπεύει το όνομα της χώρας που θα περιορίσουμε τον χρήστη.

Χρησιμοποιείται από τη μέθοδο **addUserNameRestrCountry** της webservice **"hoteldbAdd"**. Επιστρέφει τα δεδομένα του χρήστη που βρίσκονται στους πίνακες **userhotel** και **userhotelinfo**.

3.1.26 selectuserhoteladmin

Επιστρέφει όλους τους χρήστες από τη βάση δεδομένων που βρίσκονται στον πίνακα **userhotel**. Χρησιμοποιείται από τη μέθοδο **fillUserHotel** της webservice **"hoteldbFill".**

3.1.27 selectuserhotelrecountry

Επιστρέφει το όνομα μιας χώρας από τη βάση δεδομένων που βρίσκονται στον πίνακα **country**. Παίρνει σαν όρισμα τη μεταβλητή

1. **BigInt(45)** όπου αντιπροσωπεύει τη θέση της χώρας στο πίνακα **country**

Χρησιμοποιείται από τη μέθοδο **fillUserHotel** της webservice **"hoteldbFill"** είναι υπεύθυνη να επιστρέψει τη χώρα στην οποία έχουμε περιορίσει τον χρήστη.

3.1.28 selectuserhotelreregion

Επιστρέφει τα ονόματα της χώρας και της επαρχίας από τη βάση δεδομένων και σχετίζονται με τον περιορισμό του χρήστη. Παίρνει σαν όρισμα τη μεταβλητή

1. **BigInt(45)** όπου αντιπροσωπεύει τη θέση της επαρχίας στο πίνακα **region**

Χρησιμοποιείται από τη μέθοδο **fillUserHotel** της webservice **"hoteldbFill"**.

3.1.29 selectuserhotelrecity

Επιστρέφει τα ονόματα της χώρας, της επαρχίας και της πόλης από τη βάση δεδομένων τα οποία σχετίζονται με τον περιορισμό του χρήστη. Παίρνει σαν όρισμα τη μεταβλητή

1. **BigInt(45)** όπου αντιπροσωπεύει τη θέση της πόλης στο πίνακα **city**

Χρησιμοποιείται από τη μέθοδο **fillUserHotel** της webservice **"hoteldbFill"**.

3.1.30 deleteuserhotel

Απενεργοποιέι ή Ενεργοποιέι έναν χρήστη στη βάση δεδομένων. Παίρνει σαν όρισμα μια μεταβλητή **BigInt(20)** η οποία αντιπροσωπεύει τη θέση του χρήστη στη βάση. Χρησιμοποιείται από τη μέθοδο **deleteUserHotel** της webservice **"hoteldbDelete".**

3.1.31 selectadminmail

Επιστρέφει τα δεδομένα για το email του administrator και τους κωδικούς πρόσβασης στο webmail του έτσι ώστε να σταλούν τα mails.

3.1.32 insertrestrictionregion1

Επιστρέφει τον τρέχον περιορισμό για έναν χρήστη καθώς και τα id μιας δεδομένης χώρας και επαρχίας. Παίρνει σαν ορίσματα:

1. **VarChar(45)** όπου αντιπροσωπεύει το συθηματικό του χρήστη σε MD5 Hash Μορφή
2. **VarChar(45)** όπου αντιπροσωπεύει το όνομα της χώρας
3. **VarChar(45)** όπου αντιπροσωπεύει το όνομα της επαρχίας

Χρησιμοποιείται από τη μέθοδο **addUserNameRestrRegion** της webservice **"hoteldbAdd"**.

3.1.33 insertrestrictionregion2

Επιστρέφει το id της χώρας η οποία σχετίζεται με ένα δεδομένο id επαρχίας. Παίρνει σαν όρισμα:

1. **BigInt(20)** όπου αντιπροσωπεύει το id της επαρχίας

Χρησιμοποιείται από τη μέθοδο **addUserNameRestrRegion** της webservice **"hoteldbAdd"**.

3.1.34 insertrestrictionregion3

Εισάγει τον περιορισμό ενός χρήστη σε μια ορισμένη επαρχία. Παίρνει σαν όρισμα δύο μεταβλητές:

1. **VarChar(45)** όπου αντιπροσωπεύει το συνθηματικό του χρήστη σε κωδικοποιημένη MD5 Hash μορφή.
2. **VarChar(45)** οπου αντιπροσωπεύει τον περιορισμό σε μια επαρχία.

Χρησιμοποιείται από τη μέθοδο **addUserNameRestrRegion** της webservice **"hoteldbAdd"**.
Επιστρέφει τα δεδομένα του χρήστη που βρίσκονται στους πίνακες **userhotel** και **userhotelinfo**.

3.1.35 insertrestrictioncity1

Επιστρέφει τον τρέχον περιορισμό για έναν χρήστη καθώς και τα id μιας δεδομένης επαρχίας και πόλης. Παίρνει σαν ορίσματα:

1. **VarChar(45)** όπου αντιπροσωπεύει το συθηματικό του χρήστη σε MD5 Hash Μορφή
2. **VarChar(45)** όπου αντιπροσωπεύει το όνομα της χώρας
3. **VarChar(45)** όπου αντιπροσωπεύει το όνομα της επαρχίας
4. **VarChar(45)** όπου αντιπροσωπεύει το όνομα της πόλης

Χρησιμοποιείται από τη μέθοδο **addUserNameRestrCity** της webservice **“hoteldbAdd”**.

3.1.36 insertrestrictioncity2

Επιστρέφει το id της επαρχίας η οποία σχετίζεται με ένα δεδομένο id πόλης. Παίρνει σαν όρισμα:

1. **BigInt(20)** όπου αντιπροσωπεύει το id της πόλης

Χρησιμοποιείται από τη μέθοδο **addUserNameRestrCity** της webservice **“hoteldbAdd”**.

3.1.37 insertrestrictioncity3

Εισάγει τον περιορισμό ενός χρήστη σε μια ορισμένη πόλη. Παίρνει σαν όρισμα δύο μεταβλητές:

1. **VarChar(45)** όπου αντιπροσωπεύει το συνθηματικό του χρήστη σε κωδικοποιημένη MD5 Hash μορφή.
2. **VarChar(45)** οπου αντιπροσωπεύει τον περιορισμό σε μια πόλη.

Χρησιμοποιείται από τη μέθοδο **addUserNameRestrCity** της webservice **“hoteldbAdd”**.
Επιστρέφει τα δεδομένα του χρήστη που βρίσκονται στους πίνακες **userhotel** και **userhotelinfo**.

3.1.38 selectcity

Επιστρέφει τις ενεργοποιημένες πόλεις του συστήματος. Παίρνει σαν όρισμα:

- **VarChar(45) –** Το ονομα της χώρας πανω από τη πόλη
- **VarChar(45) -** Το ονομα της επαρχιας πανω από τη πολη

Χρησιμοιποιείται από τη μέθοδο **fillCityHotel** της webservice **“hoteldbFill”.**
Επιστρέφει τα περιεχόμενα του πινακα **city.**

3.1.39 insertstayingtype

Εισάγει ένα τύπο διαμονης στο συστημα. Δέχεται το όρισμα:

- **VarChar(45) –** Το ονομα του τυπου που θα προστεθει

Χρησιμοποιείται από τη μεθοδο **addStayingType.**

3.1.40 editstayingtype

3.1.41 deletestayingtype

Απενεργοποιέι ή Ενεργοποιέι έναν τύπο ξενοδοχείου στη βάση δεδομένων. Παίρνει σαν όρισμα μια μεταβλητή **BigInt(20)** η οποία αντιπροσωπεύει τη θέση του τύπου στη βάση. Χρησιμοποιείται από τη μέθοδο **deleteStayingType** της webservice **“hoteldbDelete”.**

3.1.42 insertroomsize

Εισάγει ένα τύπο δωματιου στο συστημα. Δέχεται το όρισμα:

- **VarChar(45) –** Το ονομα του τυπου που θα προστεθει

Χρησιμοποιείται από τη μεθοδο **addRoomSize.**

3.1.44 editroomsize

3.1.45 selectroomsizeadmin

3.1.46 deleteroomsize

Απενεργοποιέι ή Ενεργοποιέι έναν τύπο δωματίου στη βάση δεδομένων. Παίρνει σαν όρισμα μια μεταβλητή **BigInt(20)** η οποία αντιπροσωπεύει τη θέση του τύπου στη βάση. Χρησιμοποιείται από τη μέθοδο **deleteRoomSize** της webservice **“hoteldbDelete”.**

3.1.47 insertinternettype

Εισάγει ένα τύπο ιντερνετ στο συστημα. Δέχεται το όρισμα:

- **VarChar(45) –** Το ονομα του τυπου που θα προστεθει

Χρησιμοποιείται από τη μεθοδο **addInternetType**

3.1.48 editinternettype

3.1.49 insertfacility

Εισάγει ένα facility στο συστημα. Δέχεται το όρισμα:

- **VarChar(45) –** Το ονομα του facility που θα προστεθει

Χρησιμοποιείται από τη μεθοδο **addFacility**

3.1.50 editfacility

3.1.50 deletefacilities

Απενεργοποιέι ή Ενεργοποιέι έναν τύπο facility στη βάση δεδομένων. Παίρνει σαν όρισμα μια μεταβλητή **BigInt(20)** η οποία αντιπροσωπεύει τη θέση του facility στη βάση. Χρησιμοποιείται από τη μέθοδο **deleteFacility** της webservice **“hoteldbDelete”.**

3.1.51 selecthotelinternettypeadmin

3.1.52 deleteinternettype

Απενεργοποιέι ή Ενεργοποιέι έναν τύπο internet στη βάση δεδομένων. Παίρνει σαν όρισμα μια μεταβλητή **BigInt(20)** η οποία αντιπροσωπεύει τη θέση του τύπου στη βάση. Χρησιμοποιείται από τη μέθοδο **deleteInternetType** της webservice **“hoteldbDelete”.**

3.1.53 insertroomfacility

Εισάγει ένα room facility στο συστημα. Δέχεται το όρισμα:

- **VarChar(45) –** Το ονομα του room facility που θα προστεθει

Χρησιμοποιείται από τη μεθοδο **addRoomFacility**

3.1.54 editroomfacility

3.1.55 selectfacilitiesadmin

3.1.56 selectroomfacilitiesadmin

3.1.57 deleteroomfacilities

Απενεργοποιέι ή Ενεργοποιέι έναν τύπο room facility στη βάση δεδομένων. Παίρνει σαν όρισμα μια μεταβλητή **BigInt(20)** η οποία αντιπροσωπεύει τη θέση του τύπου στη βάση. Χρησιμοποιείται από τη μέθοδο **deleteRoomFac** της webservice **“hoteldbDelete”.**

3.1.58 insertfood

Εισάγει ένα τύπο διατροφής στο συστημα. Δέχεται το όρισμα:

- **VarChar(45) –** Το ονομα του τυπου που θα προστεθει

Χρησιμοποιείται από τη μεθοδο **addFood**

3.1.59 selectfoodadmin

.

3.1.58 inserttopdestinations

Εισάγει τα προτεινομενα ξενοδοχεία στο συστημα. Δέχεται το όρισμα:

- **VarChar(45) –** Το ονομα του τυπου που θα προστεθει

Χρησιμοποιείται από τη μεθοδο **addFood**