ΑΤΕΙ ΠΕΙΡΑΙΑ – Ηλεκτρονικά Υπολογιστικά Συστήματα

Μηχανική Λογισμικού (Ε)

Άννα Ελένη Γεωργοπούλου (Α.Μ:39095 )

Περσεφόνη Ασφή (Α.Μ.:40200 )

Μαρία Μεϊμάρη (Α.Μ.:39196 )

# Ανάλυση Απαιήσεων

## 1.1 Σκοπός του κειμένου

Ο σκοπός αυτού του κειμένου είναι να περιγράψει τις λειτουργικές απαιτήσεις του λογισμικού που θα υλοποιηθεί για το online σύστημα κρατήσεων ενοικιαζόμενων.

## 1.2 Προορισμένο ακροατήριο και προτάσεις ανάγνωσης

Διευθυντής έργου, προγραμματιστές, τεχνική υποστήριξη, διαχειριστές εφαρμογής

(ιδιοκτήτης και υπάλληλος ενοικιαζόμενων δωματίων).

## 1.3 Στόχος και εύρος του προϊόντος

Στόχος της εφαρμογής είναι να δικτυώσει όλα τα καταλύματα του ξενοδοχείου έτσι ώστε να γίνει πιο εύκολη, πιο γρήγορη και πιο ασφαλές η διαδικασία που επιτελούν τόσο οι πελάτες για την κράτηση δωματίου και την εξόφληση αυτού όσο ο ιδιοκτήτης και οι υπάλληλοι του ξενοδοχείου ως προς την εξυπηρέτηση των πελατών και την διαχείριση των δωματίων. Επίσης θα έχει και ενημερωτικό ρόλο για το κάθε κατάλυμα-δωμάτιο ώστε να μπορεί ο διαχειριστής να παίρνει κάποια στατιστικά.

# Γενική Περιγραφή

## 2.1 Προέλευση και προοπτική του προϊόντος

Υπάρχει προοπτική ανάπτυξης ή αναβάθμισης του προϊόντος

## Γενική περιγραφή των λειτουργιών του προϊόντος

* Εισαγωγή στοιχείων πελάτη
* Αναζήτηση διαθεσιμότητας δωματίων ( μέγεθος και περιοχή) με βάση κριτηρίου το *“check in “* και το *” check out “*
* Εμφάνιση αποτελεσμάτων αναζήτησης
* Κράτηση δωματίου
* Συμπλήρωση φόρμας ( μέρες ενοικιάσεων)
* Κοστολόγηση
* Ηλεκτρονική πληρωμή
* Εκτύπωση στοιχείων
* Εμφάνιση στατιστικών (δωμάτια που ενοικιάζονται συχνότερα)
* Διαχείριση δωματίων
* Διαχείριση προσφορών
* Λίστα εξόδων
* Εκτύπωση απόδειξης
* Στατιστικά

## Κατηγορίες χρηστών

**Ιδιοκτήτης:** Το ξενοδοχείο διαθέτει έναν ιδιοκτήτη ο οποίος για να χρησιμοποιήσει στο σύστημα θα πρέπει να συνδεθεί με τον προσωπικό του κωδικό. Ο πράκτορας θα έχει δυνατότητα σε όλες τις λειτουργίες τις οποίες αναφερθήκαμε πιο πάνω.

**Υπάλληλος:** Έχει ένα προσωπικό κωδικό. Συνδεόμενος μπορεί να ελέγξει ποια δωμάτια είναι διαθέσιμα μέσω της μηχανής αναζήτησης, να κάνει κράτηση δωματίων και να κάνει οποιαδήποτε ενέργεια απαιτείται για την διαχείριση των δωματίων.

**Πελάτης:** Δημιουργεί έναν προσωπικό κωδικό συμπληρώνοντας την φόρμα δημιουργίας λογαριασμού προκειμένου να μπορέσει να εισαχθεί στο σύστημα και να κάνει οποιαδήποτε ενέργεια.

## Περιβάλλον λειτουργίας

* Το σύστημα είναι σχεδιασμένο για λειτουργικό σύστημα Windows XP.
* Οι ελάχιστες απαιτήσεις σε υλικό (hardware) είναι Pentium και μνήμη RAM 4GB και επεξεργαστή 2GHZ.
* Για τον Server λειτουργικό Linux, 2.8 GHz, 8GB RAM μνήμη.

## 

## Περιορισμοί στη σχεδίαση και την υλοποίηση

## ADSL 2Mbps για σύνδεση με τον κεντρικό server για τους πελάτες και LAN 1GBps για τον ιδιοκτήτη και τους υπαλλήλους.

* 1. **Τεκμηρίωση**
* Διαθέσιμο εγχειρίδιο σε μορφή .doc αλλά και online βοήθεια για τους χρήστες.

# Απαιτήσεις για τις εξωτερικές διεπαφές

## Διεπαφή χρήστη

Η διεπαφή με το χρήστη θα είναι όσο το δυνατό πιο εύκολη. Θα χρησιμοποιείται ένα Web Interface που θα ζητάει από τον χρήστη να εισάγει τον κωδικό. Μετά την σύνδεση θα εμφανίζονται διάφορα menu για την κάθε λειτουργία. Επιπλέον θα υπάρχουν και καρτέλες για τον ιδιοκτήτη και τους υπαλλήλους.

## Διεπαφές λογισμικού

Θα υπάρχει ένα σύστημα πληρωμών με Easy Bank.

# Λειτουργίες συστήματος

## Λειτουργία 1

**4.1.1 Περιγραφή και προτεραιότητα**

Ο εργαζόμενος θα μπορεί να εισέρχεται στο σύστημα με την πληκτρολόγηση του προσωπικού του κωδικού. Κωδικό θα έχουν μόνο ο ιδιοκτήτης, ο υπάλληλος και ο πελάτης. Θα είναι διαφορετικός για τον καθένα και κατά την είσοδο τους θα έχουν διαφορετικές λειτουργίες (ιδιοκτήτης - υπάλληλος - πελάτης). Η σύνδεση στο σύστημα έχει υψηλή προτεραιότητα γιατί πάντα θα πρέπει πρώτα να συνδεθούν.

* + 1. **Ακολουθίες ερεθίσματος / απόκρισης**
* **Εισαγωγή κωδικού**

**Ερέθισμα:** Το σύστημα θα ζητάει ένα κωδικό πρόσβασης και μετά την πληκτρολόγηση του θα περιμένει το πάτημα του enter

**Απόκριση:** Ο χρήστης θα πληκτρολογεί το κωδικό του και θα πατάει το enter

* **Εισαγωγή στοιχείων ή προϊόντος**

**Ερέθισμα:** Το σύστημα θα ζητάει τα στοιχεία πελάτη για εγγραφή και θα περιμένει το πάτημα του enter

**Απόκριση:** Ο υπάλληλος ή ο πελάτης θα πληκτρολογεί τα στοιχεία του και θα πατάει το enter.

* **Αναζήτηση**

**Ερέθισμα:** Το σύστημα θα ζητάει την λέξει κλειδί (μέγεθος δωματίου ή περιοχή) και θα περιμένει το πάτημα του enter.

**Απόκριση:** Ο υπάλληλος ή ο πελάτης θα γράφει τη λέξι κλειδί και θα πατάει το enter

* **Εκτύπωση**

**Ερέθισμα:** Το σύστημα περιμένει το πάτημα του εικονιδίου print για να εκτυπώσει την απόδειξη.

**Απόκριση:** Ο πελάτης πατάει το εικονίδιο που γράφει print

***4*.1.3** **Ανάλυση σε λειτουργικές απαιτήσεις**

**REQ-1:** Σύνδεση του ιδιοκτήτη ή του υπαλλήλου ή του πελάτη στο σύστημα.

Απαιτεί username και password. Σε περίπτωση εισαγωγής λανθασμένων στοιχείων ή μη συμπλήρωσης αυτών, εμφάνιση κατάλληλου μηνύματος για την κάθε περίπτωση.

**REQ-2:** Εισαγωγή στοιχείων πελάτη.

Σε περίπτωση που δεν συμπληρωθεί ένα από τα παρακάτω : ***Ονοματεπώνυμο, ημερομηνία γέννησης, διεύθυνση, τ.κ ,τηλέφωνο και αριθμός ταυτότητας***να εμφανίζεται κατάλληλο μήνυμα για το πεδίο που δεν έχει συμπληρωθεί και να μην επιτρέπετε να γίνει η εγγραφή.

**REQ-3:** Στατιστικά.

Για να εμφανιστούν στον ιδιοκτήτη τα στατιστικά θα πρέπει πρώτα να επιλέξει την χρονική περίοδο (μέρες ή μήνες), κατάλυμα και περιοχή και θα του εμφανίζονται με αύξουσα σειρά (αυτό με την μεγαλύτερη ζήτηση) τα δωμάτια.

Για να εμφανιστούν στον πελάτη τα στατιστικά θα πρέπει πρώτα να επιλέξει κατάλυμα και περιοχή και θα εμφανίζονται κατά αύξουσα σειρά τα δωμάτια που ενοικιάζονται συχνότερα.

**REQ-4:** Μετατροπή στις μέρες. Ο υπάλληλος θα μπορεί να αλλάξει τις μέρες κράτησης οποιαδήποτε στιγμή επιθυμεί.

**REQ-5:** Εκτύπωση απόδειξης. Η απόδειξη θα στέλνεται μέσω e-mail (ο οποίος θα το έχει συμπληρώσει κατά την καταχώρηση των στοιχείων του για την δημιουργία λογαριασμού) σε μορφή .pdf στον πελάτη αφού πληρώσει.

# Μη λειτουργικές απαιτήσεις

## Επιδόσεις

## Η συναλλαγή θα πρέπει να πραγματοποιείται σε χρόνο μικρότερο από 10 sec

1. Οι ασύγχρονοι χρήστες δεν θα πρέπει να ξεπερνούν τους 100.

## 5.2 Ασφάλεια πληροφορίας

Ασφάλεια θα παρέχεται κατά την εισαγωγή του προσωπικού κωδικού του χρήστη (ιδιοκτήτης – υπάλληλος - πελάτης) και κατά την πληρωμή του δωματίου απ’ τον πελάτη.

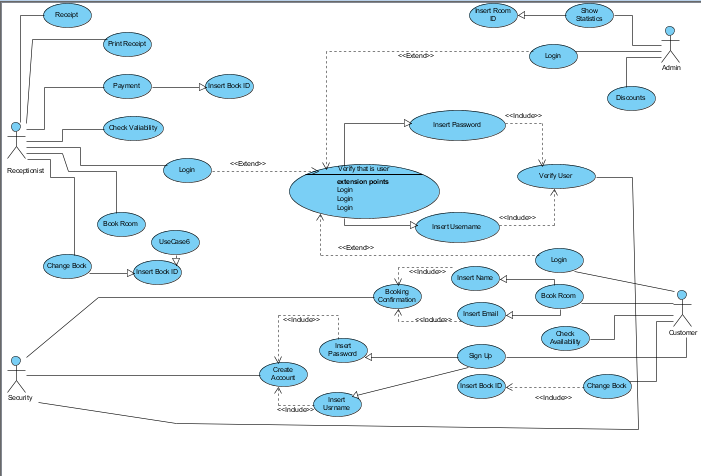
## 5.3 Επιχειρησιακοί κανόνες

Το σύστημα μπορεί να χρησιμοποιείται απ’ όλους (ιδιοκτήτης, υπάλληλοι, πελάτες).Ο καθένας από αυτούς θα μπορεί να το χρησιμοποιήσει σύμφωνα με τις δυνατότητες που έχουν αναφερθεί πιο πάνω.

**5.4 Φυσική Ασφάλεια**

Ο server θα φυλάσσεται στο γραφείο του ιδιοκτήτη των ενοικιαζόμενων δωματίων.

**Διάργαμμα Περιπτώσεων Χρήσης**



# Κατάλογος Περιπτώσεων Χρήσης

|  |  |
| --- | --- |
| Εμπλεκόμενοι Ρόλοι | Περιπτώσεις Χρήσης |
| Υπάλληλος | UC-1.Σύνδεση  UC-2.Αναζήτηση διαθέσημου δωματίου  UC-3.Κράτηση δωματίου  UC-4. Επεξεργασία Κράτησης  UC-5.Εξόφληση λογαριασμού  UC-6. Δημιουργία απόδειξης  UC-7.Εκτύπωση απόδειξης |
| Διαχειριστής | UC-8.Σύνδεση  UC-9.Στατιστικά  UC-10.Διαχείριση προσφορών |
| Πελάτης | UC-11. Εγγραφή Πελάτη  UC-12.Σύνδεση Πελάτη  UC-13.Έλεγχος (αναζήτηση) διαθεσιμότητας δωματίου  UC-14.Κράτηση δωματίου  UC-15.Ακύρωση ή τροποποίηση κράτησης δωματίου |
|  |  |

**Πρότυπο Περιπτώσεων Χρήσης**

|  |  |  |  |
| --- | --- | --- | --- |
| Κωδικός Περίπτωσης: | UC-1 | | |
| Ονομασία: | Σύνδεση Υπαλλήλου | | |
| Δημιουργήθηκε από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη | Τελευταία ενημέρωση από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη |
| Ημερομηνία Συγγραφής: | 28/11/2013 | Ημερομηνία τελευταίας ενημέρωσης: | 10/01/2014 |

|  |  |
| --- | --- |
| Εμπλεκόμενοι Ρόλοι: | Υπάλληλος |
| Περιγραφή: | Ο υπάλληλος μπορεί να εισάγει τον κωδικό του |
| Γεγονός Εκκίνησης: | Ο υπάλληλος πληκτρολογεί τον σωστό κωδικό πρόσβασης |
| Προϋποθέσεις: | Ο κωδικός που θα πληκτρολογηθεί να είναι ο σωστός |
| Τελική Κατάσταση: | Σύνδεση |
| Φυσιολογική Ροή: | 1. Ο υπάλληλος πληκτρολογεί τον κωδικό πρόσβασης 2. Πατάει enter 3. Αν ο κωδικός είναι σωστός εμφανίζεται το menu με όλες τις επιλογές |
| Εναλλακτική Ροή: | - |
| Εξαιρέσεις: | Ο κωδικός πρόσβασης είναι λανθασμένος |
| Ενσωματώνει: | - |
| Προτεραιότητα: | 1 |
| Συχνότητα χρήσης: | - |
| Business Rules: | - |
| Ειδικές απαιτήσεις: | - |
| Υποθέσεις: | - |
| Σημειώσεις και ζητήματα: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Κωδικός Περίπτωσης: | UC-2 | | |
| Ονομασία: | Αναζήτηση διαθέσημου δωματίου | | |
| Δημιουργήθηκε από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη | Τελευταία ενημέρωση από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη |
| Ημερομηνία Συγγραφής: | 28/11/2013 | Ημερομηνία τελευταίας ενημέρωσης: | 10/01/2014 |

|  |  |
| --- | --- |
| Εμπλεκόμενοι Ρόλοι: | Υπάλληλος |
| Περιγραφή: | Ο υπάλληλος μπορεί να αναζητήσει ποιά δωμάτια είναι διαθέσιμα. |
| Γεγονός Εκκίνησης: | Ο υπάλληλος επιλέγει την ημερομηνία άφιξης και αναχώρησης, τον αριθμό των ατόμων ανά δωμάτιο και τον αριθμό των δωματίων |
| Προϋποθέσεις: | Ο υπάλληλος πρέπει να συμπληρώσει όλα τα πεδία |
| Τελική Κατάσταση: | Κράτηση δωματίου |
| Φυσιολογική Ροή: | 1. UC-1 2. Επιλογή ημερομηνίας άφιξης και αναχώρησης 3. Επιλογή αριθμού ατόμων/δωμάτιο 4. Επιλογή αριθμού δωματίων 5. Πάτημα **Αναζήτηση** 6. Εισάγει αναγνωριστικό κωδικό για κάποιο δωμάτιο στη φόρμα, ώστε το σύστημα να εμφανίσει τις πληροφορίες για το συγκεκριμένο δωμάτιο |
| Εναλλακτική Ροή: | Επιστροφή στο κεντρικό menu |
| Εξαιρέσεις: | Αν ένα από τα πεδία δεν συμπληρωθούν και πατηθεί **‘Αναζήτηση’** θα εμφανίζεται κατάλληλο μήνυμα που να λέει ότι δεν μπορεί να πραγματοποιηθεί αναζήτηση χωρίς να συμπληρωθούν όλα τα πεδία. Ο υπάλληλος θα πρέπει να πατήσει το **ok** που εμφανίζεται μαζί με το μήνυμα για να επιστρέψει στη φόρμα με τα στοιχεία. |
| Ενσωματώνει: | UC-1 |
| Προτεραιότητα: | 2 |
| Συχνότητα χρήσης: |  |
| Business Rules: |  |
| Ειδικές απαιτήσεις: |  |
| Υποθέσεις: |  |
| Σημειώσεις και ζητήματα: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Κωδικός Περίπτωσης: | UC-3 | | |
| Ονομασία: | Κράτηση δωματίου | | |
| Δημιουργήθηκε από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη | Τελευταία ενημέρωση από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη |
| Ημερομηνία Συγγραφής: | 28/11/2013 | Ημερομηνία τελευταίας ενημέρωσης: | 10/01/2014 |

|  |  |
| --- | --- |
| Εμπλεκόμενοι Ρόλοι: | Υπάλληλος |
| Περιγραφή: | Ο υπάλληλος μπορεί να κάνει κράτηση σ’ ένα δωμάτιο |
| Γεγονός Εκκίνησης: | Ο υπάλληλος επιλέγει κράτηση |
| Προϋποθέσεις: | Το δωμάτιο να είναι διαθέσιμο |
| Τελική Κατάσταση: | Κράτηση δωματίου |
| Φυσιολογική Ροή: | 1. UC-1  2. UC-2  3.Πάτημα **‘Συνέχεια’**  4.Εμφανίζεται φόρμα συμπλήρωσης των στοιχείων του πελάτη  5.Συμπλήρωση των στοιχείων του πελάτη  6.Πάτημα **‘Κράτηση’** |
| Εναλλακτική Ροή: | Επιστροφή στο κεντρικό menu |
| Εξαιρέσεις: |  |
| Ενσωματώνει: | UC-1, UC-2 |
| Προτεραιότητα: | 3 |
| Συχνότητα χρήσης: |  |
| Business Rules: | - |
| Ειδικές απαιτήσεις: | - |
| Υποθέσεις: | - |
| Σημειώσεις και ζητήματα: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Κωδικός Περίπτωσης: | UC-4 | | |
| Ονομασία: | Επεξεργασία Κράτησης | | |
| Δημιουργήθηκε από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη | Τελευταία ενημέρωση από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη |
| Ημερομηνία Συγγραφής: | 28/11/2013 | Ημερομηνία τελευταίας ενημέρωσης: | 10/01/2014 |

|  |  |
| --- | --- |
| Εμπλεκόμενοι Ρόλοι: | Υπάλληλος |
| Περιγραφή: | Ο υπάλληλος μπορεί να κάνει τροποποίηση ή διαγραφή κράτησης |
| Γεγονός Εκκίνησης: | Ο υπάλληλος πατάει ‘***Τροποποίηση ή Ακύρωση κράτησης’*** |
| Προϋποθέσεις: | Να έχει πραγματοποιηθεί κράτηση |
| Τελική Κατάσταση: | Τροποποίηση ή διαγραφη κράτησης |
| Φυσιολογική Ροή: | 1. UC-1 2. Πάτημα ‘***Τροποποίηση ή Ακύρωση κράτησης’*** 3. Εισαγωγή αναγνωριστικού κωδικού της κράτησης 4. Πάτημα **‘Επεξεργασία κράτησης’** 5. Εμφάνηση της φόρμας για την επεξεργασία της κράτησης 6. Επεξεργασία της κράτησης 7. Επιστροφή στο κεντρικό μενού |
| Εναλλακτική Ροή: | Επιστροφή στο κεντρικό μενού |
| Εξαιρέσεις: | Να μην υπάρχει κράτηση |
| Ενσωματώνει: | UC-1 |
| Προτεραιότητα: | 2 |
| Συχνότητα χρήσης: |  |
| Business Rules: | - |
| Ειδικές απαιτήσεις: | - |
| Υποθέσεις: | - |
| Σημειώσεις και ζητήματα: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Κωδικός Περίπτωσης: | UC-5 | | |
| Ονομασία: | Εξόφληση δωματίου | | |
| Δημιουργήθηκε από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη | Τελευταία ενημέρωση από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη |
| Ημερομηνία Συγγραφής: | 28/11/2013 | Ημερομηνία τελευταίας ενημέρωσης: | 10/01/2014 |

|  |  |
| --- | --- |
| Εμπλεκόμενοι Ρόλοι: | Υπάλληλος |
| Περιγραφή: | Ο υπάλληλος μπορεί να πραγματοποιήσει τη διαδικασία της εξόφλησης των δωματίων |
| Γεγονός Εκκίνησης: | Ο υπάλληλος πατάει **‘Εξόφληση Δωματίου’** |
| Προϋποθέσεις: | Να υπάρχει κράτηση |
| Τελική Κατάσταση: | Εξόφληση δωματίου |
| Φυσιολογική Ροή: | 1. UC-1 2. Πάτημα ‘**εξόφληση δωματίου’** 3. Εισαγωγή αναγνωρηστικού κωδικού κράτησης 4. Πάτημα Enter 5. Πάτημα δημιουργία απόδειξης 6. Επιστροφή στο κεντρικό menu |
| Εναλλακτική Ροή: | Επιστροφή στο κεντρικό menu |
| Εξαιρέσεις: | Να μην υπάρχει ο αναγνωρηστικός κωδικός κράτησης |
| Ενσωματώνει: | UC1,UC-6 |
| Προτεραιότητα: | 2 |
| Συχνότητα χρήσης: | - |
| Business Rules: | - |
| Ειδικές απαιτήσεις: | - |
| Υποθέσεις: | - |
| Σημειώσεις και ζητήματα: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Κωδικός Περίπτωσης: | UC-6 | | |
| Ονομασία: | Δημιουργία απόδειξης | | |
| Δημιουργήθηκε από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη | Τελευταία ενημέρωση από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη |
| Ημερομηνία Συγγραφής: | 28/11/2013 | Ημερομηνία τελευταίας ενημέρωσης: | 10/01/2014 |

|  |  |
| --- | --- |
| Εμπλεκόμενοι Ρόλοι: | Υπάλληλος |
| Περιγραφή: | Ο πράκτορας μπορεί να δημιουργήσει απόδειξη |
| Γεγονός Εκκίνησης: | Ο πράκτορας **‘Δημιουργία Απόδειξης’** |
| Προϋποθέσεις: | Να υπάρχει κράτηση |
| Τελική Κατάσταση: | Δημιουργία απόδειξης |
| Φυσιολογική Ροή: | 1.UC-1  2.UC-5  3. Πάτημα δημιουργία απόδειξης  4.Εκτύπωση απόδειξης  5. Επιλογή επιστροφής στο αρχικό menu  6.Επιστροφή στο κεντρικό menu |
| Εναλλακτική Ροή: | - |
| Εξαιρέσεις: |  |
| Ενσωματώνει: | UC-1, UC-5, UC-7 |
| Προτεραιότητα: | 3 |
| Συχνότητα χρήσης: | - |
| Business Rules: | - |
| Ειδικές απαιτήσεις: | - |
| Υποθέσεις: | - |
| Σημειώσεις και ζητήματα: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Κωδικός Περίπτωσης: | UC-7 | | |
| Ονομασία: | Εκτύπωση απόδειξης | | |
| Δημιουργήθηκε από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη | Τελευταία ενημέρωση από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη |
| Ημερομηνία Συγγραφής: | 28/11/2013 | Ημερομηνία τελευταίας ενημέρωσης: | 10/01/2014 |

|  |  |
| --- | --- |
| Εμπλεκόμενοι Ρόλοι: | Υπάλληλος |
| Περιγραφή: | Ο πράκτορας μπορεί να δημιουργήσει απόδειξη |
| Γεγονός Εκκίνησης: | Ο πράκτορας **‘ Εκτύπωση Απόδειξης’** |
| Προϋποθέσεις: | Να υπάρχει κράτηση |
| Τελική Κατάσταση: | Εκτύπωση απόδειξης |
| Φυσιολογική Ροή: |  |
| Εναλλακτική Ροή: | - |
| Εξαιρέσεις: |  |
| Ενσωματώνει: | UC-1, UC-5, UC-6 |
| Προτεραιότητα: | 4 |
| Συχνότητα χρήσης: | - |
| Business Rules: | - |
| Ειδικές απαιτήσεις: | - |
| Υποθέσεις: | - |
| Σημειώσεις και ζητήματα: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Κωδικός Περίπτωσης: | UC-8 | | |
| Ονομασία: | Σύνδεση Διαχειριστή | | |
| Δημιουργήθηκε από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη | Τελευταία ενημέρωση από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη |
| Ημερομηνία Συγγραφής: | 28/11/2013 | Ημερομηνία τελευταίας ενημέρωσης: | 10/01/2014 |

|  |  |
| --- | --- |
| Εμπλεκόμενοι Ρόλοι: | Διαχειριστής |
| Περιγραφή: | Ο διαχειριστής μπορεί να εισάγει τον κωδικό του |
| Γεγονός Εκκίνησης: | Ο διαχειριστής πληκτρολογεί τον σωστό κωδικό πρόσβασης |
| Προϋποθέσεις: | Ο κωδικός που θα πληκτρολογηθεί να είναι ο σωστός |
| Τελική Κατάσταση: | Σύνδεση |
| Φυσιολογική Ροή: | Ο διαχειριστής πληκτρολογεί τον κωδικό πρόσβασης  Πατάει enter  Αν ο κωδικός είναι σωστός εμφανίζεται το menu με όλες τις επιλογές |
| Εναλλακτική Ροή: | - |
| Εξαιρέσεις: | Ο κωδικός πρόσβασης είναι λανθασμένος |
| Ενσωματώνει: | - |
| Προτεραιότητα: | 1 |
| Συχνότητα χρήσης: | - |
| Business Rules: | - |
| Ειδικές απαιτήσεις: | - |
| Υποθέσεις: | - |
| Σημειώσεις και ζητήματα: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Κωδικός Περίπτωσης: | UC-9 | | |
| Ονομασία: | Στατιστικά | | |
| Δημιουργήθηκε από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη | Τελευταία ενημέρωση από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη |
| Ημερομηνία Συγγραφής: | 28/11/2013 | Ημερομηνία τελευταίας ενημέρωσης: | 10/01/2014 |

|  |  |
| --- | --- |
| Εμπλεκόμενοι Ρόλοι: | Διαχειριστής |
| Περιγραφή: | Ο διαχειριστής εμφανίζει τα στατιστικά του συστήματος |
| Γεγονός Εκκίνησης: | Επιλογή εμφάνισης στατιστικών |
| Προϋποθέσεις: | Επιλογή στατιστικών |
| Τελική Κατάσταση: | Εμφάνιση στατιστικών |
| Φυσιολογική Ροή: | 1. UC-8 2. Επιλογή στατιστικών 3. Επιλογή δωματίου 4. Εμφάνιση στατιστικών 5. Επιλογή επιστροφή στο κεντρικό menu 6. Επιστροφή στο κεντρικό menu |
| Εναλλακτική Ροή: | - |
| Εξαιρέσεις: |  |
| Ενσωματώνει: | UC-8 |
| Προτεραιότητα: | 2 |
| Συχνότητα χρήσης: | 1 φορά την εβδομάδα |
| Business Rules: | - |
| Ειδικές απαιτήσεις: | - |
| Υποθέσεις: | - |
| Σημειώσεις και ζητήματα: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Κωδικός Περίπτωσης: | UC-10 | | |
| Ονομασία: | Διαχείριση προσφορών | | |
| Δημιουργήθηκε από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη | Τελευταία ενημέρωση από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη |
| Ημερομηνία Συγγραφής: | 28/11/2013 | Ημερομηνία τελευταίας ενημέρωσης: | 10/01/2014 |

|  |  |
| --- | --- |
| Εμπλεκόμενοι Ρόλοι: | Διαχειριστής |
| Περιγραφή: | Ο διαχειριστής διαχειρίζεται τις προσφορές των δωματίων |
| Γεγονός Εκκίνησης: | Επιλογή **‘Προσφορές’** |
| Προϋποθέσεις: | Επιλογή δωματίων |
| Τελική Κατάσταση: | Εμφάνιση προσφορών |
| Φυσιολογική Ροή: | 1. UC-8 2. Πάτημα **‘Προσφορές’** 3. Εμφάνηση φόρμας επεξεργασίας προσφορών 4. Επεξεργασία της φόρμας 5. Πάτημα Enter 6. Επιλογή **‘επιστροφή στο κεντρικό menu’** 7. Επιστροφή στο κεντρικό menu |
| Εναλλακτική Ροή: | 1. Επιστροφή στο κεντρικό menu |
| Εξαιρέσεις: |  |
| Ενσωματώνει: | UC-8 |
| Προτεραιότητα: | 2 |
| Συχνότητα χρήσης: | 2 φορές την εβδομάδα |
| Business Rules: | - |
| Ειδικές απαιτήσεις: | - |
| Υποθέσεις: | - |
| Σημειώσεις και ζητήματα: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Κωδικός Περίπτωσης: | UC-11 | | |
| Ονομασία: | Εγγραφή Πέλάτη | | |
| Δημιουργήθηκε από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη | Τελευταία ενημέρωση από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη |
| Ημερομηνία Συγγραφής: | 28/11/2013 | Ημερομηνία τελευταίας ενημέρωσης: | 10/01/2014 |

|  |  |
| --- | --- |
| Εμπλεκόμενοι Ρόλοι: | Πελάτης |
| Περιγραφή: | Ο πελάτης μπορεί να εισάγει τα στοιχεία του από το πληκτρολόγιο. |
| Γεγονός Εκκίνησης: | Ο πελάτης επιλέγει εγγραφή |
| Προϋποθέσεις: | Ο πελάτης πρέπει να συμπληρώσει όλα τα πεδία |
| Τελική Κατάσταση: | Εγγραφή πελάτη |
| Φυσιολογική Ροή: | 1. Πατά **‘Εγγραφή’** 2. Εμφάνηση της φόρμας εγγραφής 3. Συμπλήρωση της φόρμας εγγραφής 4. Πάτημα **‘Υποβολή’** |
| Εναλλακτική Ροή: |  |
| Εξαιρέσεις: | Αν ένα από τα στοιχεία του πελάτη δεν συμπληρωθούν και πατηθεί το **‘Υποβολή’** θα εμφανίζεται κατάλληλο μήνυμα που να λέει ότι δεν μπορεί να ολοκληρωθεί η εγγραφή χωρίς να συμπληρωθούν όλα τα πεδία. Ο πράκτορας θα πρέπει να πατήσει το ok που εμφανίζεται μαζί με το μήνυμα για να επιστρέψει στη φόρμα με τα στοιχεία. |
| Ενσωματώνει: |  |
| Προτεραιότητα: | 1 |
| Συχνότητα χρήσης: |  |
| Business Rules: |  |
| Ειδικές απαιτήσεις: |  |
| Υποθέσεις: |  |
| Σημειώσεις και ζητήματα: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Κωδικός Περίπτωσης: | UC-12 | | |
| Ονομασία: | ΣύνδεσηΠελάτη | | |
| Δημιουργήθηκε από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη | Τελευταία ενημέρωση από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη |
| Ημερομηνία Συγγραφής: | 28/11/2013 | Ημερομηνία τελευταίας ενημέρωσης: | 10/01/2014 |

|  |  |
| --- | --- |
| Εμπλεκόμενοι Ρόλοι: | Πελάτης |
| Περιγραφή: | Ο πελάτης μπορεί να εισάγει τον κωδικό του |
| Γεγονός Εκκίνησης: | Ο πελάτης πληκτρολογεί τον σωστό κωδικό πρόσβασης και το username |
| Προϋποθέσεις: | Ο κωδικός και το username που θα πληκτρολογηθούν να είναι οσωστά |
| Τελική Κατάσταση: | Σύνδεση |
| Φυσιολογική Ροή: | Ο πελάτης πληκτρολογεί τον κωδικό πρόσβασης και το username  Πατάει enter  Αν ο κωδικός και το username είναι σωστά εμφανίζεται το menu με όλες τις επιλογές |
| Εναλλακτική Ροή: | - |
| Εξαιρέσεις: | Ο κωδικός πρόσβασης ή/και το username είναι λανθασμένα |
| Ενσωματώνει: | - |
| Προτεραιότητα: | 2 |
| Συχνότητα χρήσης: | - |
| Business Rules: | - |
| Ειδικές απαιτήσεις: | - |
| Υποθέσεις: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Κωδικός Περίπτωσης: | UC-13 | | |
| Ονομασία: | Έλεγχος (αναζήτηση) διαθεσιμότητας δωματίου | | |
| Δημιουργήθηκε από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη | Τελευταία ενημέρωση από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη |
| Ημερομηνία Συγγραφής: | 28/11/2013 | Ημερομηνία τελευταίας ενημέρωσης: | 10/01/2014 |

|  |  |
| --- | --- |
| Εμπλεκόμενοι Ρόλοι: | Πελάτης |
| Περιγραφή: | Ο πελάτης μπορεί να αναζητήσει ποιά δωμάτια είναι διαθέσιμα. |
| Γεγονός Εκκίνησης: | Ο πελάτης επιλέγει την ημερομηνία άφιξης και αναχώρησης, τον αριθμό των ατόμων ανά δωμάτιο και τον αριθμό των δωματίων |
| Προϋποθέσεις: | Ο πελάτης πρέπει να συμπληρώσει όλα τα πεδία και να έχει πραγματοποιήση εγγραφή |
| Τελική Κατάσταση: | Κράτηση δωματίου |
| Φυσιολογική Ροή: | 1. UC-12 2. Επιλογή ημερομηνίας άφιξης και αναχώρησης 3. Επιλογή αριθμού ατόμων/δωμάτιο 4. Επιλογή αριθμού δωματίων 5. Πάτημα Αναζήτηση |
| Εναλλακτική Ροή: | Επιστροφή στο κεντρικό menu |
| Εξαιρέσεις: | Αν ένα από τα πεδία δεν συμπληρωθούν και πατηθεί ‘Αναζήτηση’ θα εμφανίζεται κατάλληλο μήνυμα που να λέει ότι δεν μπορεί να πραγματοποιηθεί αναζήτηση χωρίς να συμπληρωθούν όλα τα πεδία. Ο υπάλληλος θα πρέπει να πατήσει το ok που εμφανίζεται μαζί με το μήνυμα για να επιστρέψει στη φόρμα με τα στοιχεία. |
| Ενσωματώνει: | UC-12 |
| Προτεραιότητα: | 3 |
| Συχνότητα χρήσης: |  |
| Business Rules: |  |
| Ειδικές απαιτήσεις: |  |
| Υποθέσεις: |  |
| Σημειώσεις και ζητήματα: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Κωδικός Περίπτωσης: | UC-14 | | |
| Ονομασία: | Κράτηση δωματίου | | |
| Δημιουργήθηκε από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη | Τελευταία ενημέρωση από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη |
| Ημερομηνία Συγγραφής: | 28/11/2013 | Ημερομηνία τελευταίας ενημέρωσης: | 10/01/2014 |

|  |  |
| --- | --- |
| Εμπλεκόμενοι Ρόλοι: | Πελάτης |
| Περιγραφή: | Ο πελάτης μπορεί να κάνει κράτηση σ’ ένα δωμάτιο |
| Γεγονός Εκκίνησης: | Ο πελάτης επιλέγει κράτηση |
| Προϋποθέσεις: | Το δωμάτιο να είναι διαθέσιμο |
| Τελική Κατάσταση: | Κράτηση δωματίου |
| Φυσιολογική Ροή: | 1. UC-12  2. UC-13  3.Πάτημα **‘Συνέχεια’**  4.Εμφανίζεται φόρμα συμπλήρωσης των στοιχείων του πελάτη  5.Συμπλήρωση των στοιχείων του πελάτη  6.Πάτημα **‘Κράτηση’** |
| Εναλλακτική Ροή: | Επιστροφή στο κεντρικό menu |
| Εξαιρέσεις: |  |
| Ενσωματώνει: | UC-12, UC-13 |
| Προτεραιότητα: | 4 |
| Συχνότητα χρήσης: |  |
| Business Rules: | - |
| Ειδικές απαιτήσεις: | - |
| Υποθέσεις: | - |
| Σημειώσεις και ζητήματα: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Κωδικός Περίπτωσης: | UC-15 | | |
| Ονομασία: | Επεξεργασία Κράτησης | | |
| Δημιουργήθηκε από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη | Τελευταία ενημέρωση από: | Άννα Ελένη Γεωργοπούλου  Περσεφόνη Ασφή  Μαρία Μεϊμάρη |
| Ημερομηνία Συγγραφής: | 28/11/2013 | Ημερομηνία τελευταίας ενημέρωσης: | 10/01/2014 |

|  |  |
| --- | --- |
| Εμπλεκόμενοι Ρόλοι: | Πελάτης |
| Περιγραφή: | Ο πελάτης μπορεί να κάνει τροποποίηση ή διαγραφή κράτησης |
| Γεγονός Εκκίνησης: | Ο πελάτης πατάει ‘***Τροποποίηση ή Ακύρωση κράτησης’*** |
| Προϋποθέσεις: | Να έχει πραγματοποιηθεί κράτηση |
| Τελική Κατάσταση: | Τροποποίηση ή διαγραφη κράτησης |
| Φυσιολογική Ροή: | 1.UC-12  2.Πάτημα ‘***Τροποποίηση ή Ακύρωση κράτησης’***  3.Εισαγωγή αναγνωριστικούκωδικού της κράτησης  4.Πάτημα **‘Επεξεργασία κράτησης’**  5.Εμφάνηση της φόρμας για την επεξεργασία της κράτησης  6.Επεξεργασία της κράτησης  7.Πάτημα Enter  8.Επιστροφή στο κεντρικό μενού |
| Εναλλακτική Ροή: | Επιστροφή στο κεντρικό μενού |
| Εξαιρέσεις: | Να μην υπάρχει κράτηση |
| Ενσωματώνει: | UC-12 |
| Προτεραιότητα: | 5 |
| Συχνότητα χρήσης: |  |
| Business Rules: | - |
| Ειδικές απαιτήσεις: | - |
| Υποθέσεις: | - |
| Σημειώσεις και ζητήματα: | - |

**Εισαγωγή**

Ο τουριστικός τομέας εμφανίζει τα τελευταία χρόνιαεντυπωσιακό ρυθμό ανάπτυξης. Η πολυμορφικότητα των τουριστικών προϊόντων και υπηρεσιών σε συνδυασμό με τις δυνατότητες της κοινωνίας της πληροφορίας είχε σαν αποτέλεσμα την ανάπτυξη online διαδικτυακών τόπων για την αποτελεσματική πρόσβαση του καταναλωτή στο τουριστικό προϊόν. Ο στόχος είναι ο μελλοντικός τουρίστας να είναι σε θέση, με τη βοήθεια ενός φιλικού γραφικού περιβάλλοντος να μπορεί εύκολα, γρήγορα και οικονομικά να καθορίσει μόνος του το περιεχόμενο των διακοπών του έως και την τελευταία λεπτομέρεια.

Η χρήση των online διαδικτυακών τόπων και των συστημάτων κρατήσεων δίνει τη δυνατότητα στο χρήστη να έχει άμεση ενημέρωση σε ότι αφορά:

1. Πληροφορίες για πτήσεις, κρατήσεις θέσεων, τιμές εισητηρίων.

2. Πραγματοποίηση κρατήσεων σε ξενοδοχεία, τουριστικά πακέτα, ενοικιάσεις αυτοκινήτων.

3. Διαχείριση χρηματικών ποσών σε συνάλλαγμα και αγορές εισιτηρίων για διάφορες εκδηλώσεις.

4. Δυνατότητα πληροφόρησης των μετεωρολογικών συνθηκών

Βασικά κριτήρια για την επιλογή ενός συστήματος Online κράτησης αποτελεί η αξιοπιστία του πληροφοριακού υλικού, δηλαδή η πραγματική εικόνα των

παρεχόμενων τουριστικών υπηρεσιών καθώς και η έγκυρη πληροφόρηση σε ότι αφορά τις προσφερόμενες τιμές.

Για τις τουριστικές επιχειρήσεις τα συστήματα κρατήσεων:

α) αποτελούν το κύριο μέσο προώθησης και διακίνησης των τουριστικών προϊόντων,διαμορφώνοντας σε πολλές περιπτώσεις τη σχέση προσφοράς-ζήτησης,

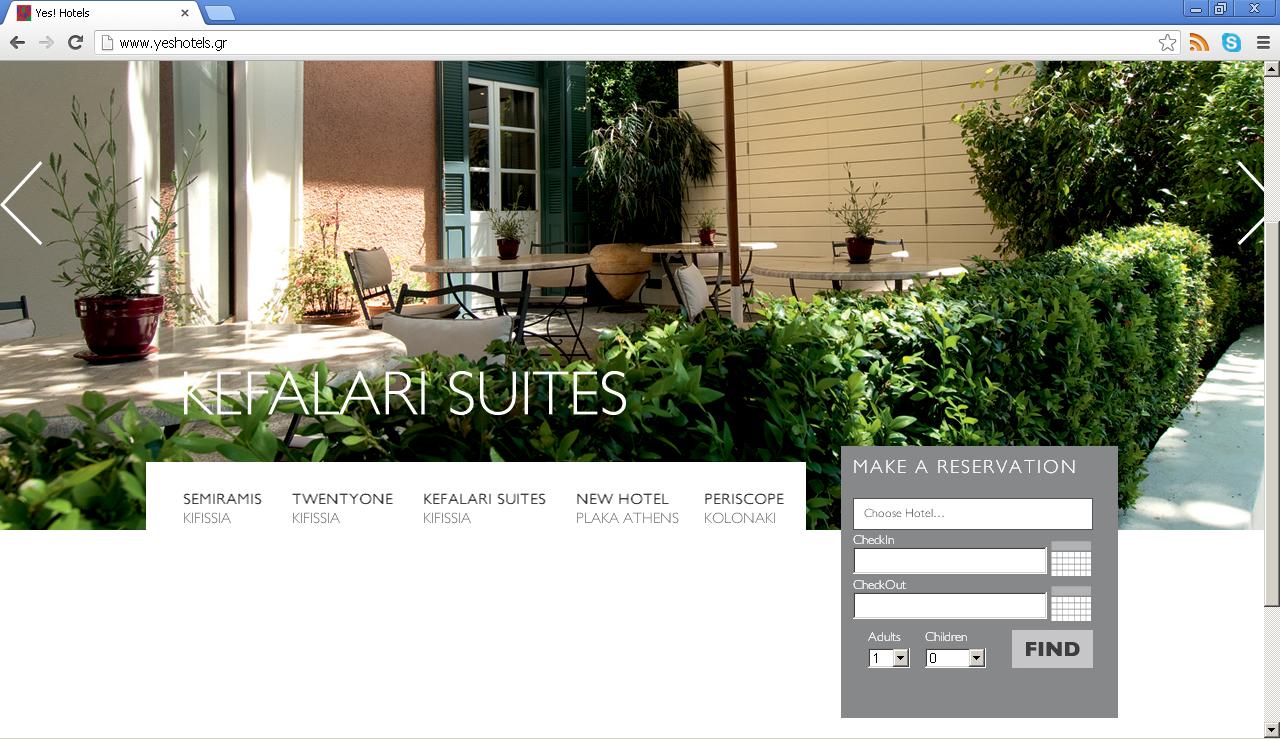
β) διευρύνουν την αγορά στην οποία στοχεύει η επιχείρηση,

γ) οδηγούν σε ανάπτυξη με γνώμονατον ορθολογικόσχεδιασμό.

Οι κρατήσεις του ξενοδοχείου μπορούν να προέρχονται από οποιοδήποτε σημείο του κόσμου. Οποιοσδήποτε έχει σύνδεση στο διαδίκτυο μπορεί να γίνει υποψήφιος πελάτης του ξενοδοχείου.Καταχωρώντας τα στοιχεία του στη φόρμα αναζήτησης ο πελάτης βλέπει τα αποτελέσματα,επιλέγει αυτό που θέλει και πραγματοποιεί μία κράτηση. Το site εμφανίζει διάφορες ανέσειςπου διαθέτει το εκάστοτε ξενοδοχείο καθώς και χάρτες του siteπου διευκολύνει τον πελάτη να δει που ακριβώς βρίσκεται το ξενοδοχείο καθώς και άλλαενδιαφέροντα από τουριστικής άποψης σημεία κοντά σε αυτό.Η διαθεσιμότητα των δωματίων του ξενοδοχείου είναι συνεχώς on-line, ώστε οι πελάτεςνα εξυπηρετούνται αποτελεσματικά και γρήγορα, χωρίς περαιτέρω συνεννοήσεις (τηλέφωνα,fax, e-mail) και χρονοβόρες διαδικασίες που μέχρι σήμερα οδηγούσαν σε ακυρώσειςκρατήσεων. Το σύστημα εμφανίζει αναλυτικά τις χρεώσεις για κάθε ημέρα και το σύνολο της κράτησης για την καλύτερη ενημέρωση των πελατών. Παρόλα αυτά, το προσωπικό είναιπάντοτε διαθέσιμο να απαντήσει και σε πιο γενικές ερωτήσεις, να κάνει προτάσεις και ναβοηθήσει σημαντικά στο σχεδιασμό του ταξιδιού.

**Παράδειγμα online συστήματος κρατήσεων ενοικιαζόμενων δωματίων**

Για την καλύτερη κατανόηση του αντικειμένου, ας πάρουμε ως παράδειγμα τον ιστότοπο <http://www.yeshotels.gr/>, και ας αναλύσουμε τις βασικές λειτουργίες του. Στο κάτω δεξιά μέρος της αρχικής σελίδας με τον τίτλο «Αναζήτηση», ο πελάτης μπορεί να δει τα διαθέσιμα δωμάτια σε κάποιο από τα ξενοδοχεία του συγκροτήματος. Ο ιστότοπος μπορεί επίσης να ενημερώνει τον πελάτη για την κράτησή του μέσω email.



Η διαδικασία της κράτησης είναι απλή. Ο πελάτης επιλέγει την ημερομηνία άφιξης και αναχώρησης, τον αριθμό των ατόμων ανά δωμάτιο και τον αριθμό των δωματίων. Κάνει κλικ στο κουμπί που λέει "Αναζήτηση". Το σύστημα θα εμφανίσει τη διαθεσιμότητα των δωματίων τις ημερομηνίες που έχουν επιλεγεί προηγουμένως. Αφού εμφανιστούν οι διαθέσιμοι τύποι δωματίων και οι τιμές τους, ο πελάτης μπορεί να επιλέξει το δωμάτιο και την τιμή που επιθυμεί διαβάζοντας τις σχετικές πληροφορίες. Στη συνέχεια κάνει κλικ στο κουμπί "Κράτηση". Έχοντας επιλέξει το δωμάτιο που επιθυμεί, μεταφέρεται στην φόρμα ασφαλούς κράτησης. Εδώ πρέπει να διαβάσει τις πληροφορίες σχετικά με την κράτηση του και να συμπληρώσει τα κενά πεδία. Επισήμανση δίνεται στην σωστή εισαγωγή του ονόματος και της ηλεκτρονικής διεύθυνσης (email). Στη συνέχεια κάνει κλικ στο κουμπί που λέει "Κράτηση". Το σύστημα θα ενημερωθεί αυτόματα με την κράτηση και θα αποστείλει τον αριθμό της κράτησης, τον αριθμό επιβεβαίωσης του ξενοδοχείου και τα στοιχεία επικοινωνίας με το ξενοδοχείο άμεσα με email. Αν χρειάζεται επιπλέον επικοινωνία πριν την κράτηση το τμήμα εξυπηρέτησης πελατών θα επικοινωνήσει με email μέσα σε δύο (2) με τέσσερις (4) ώρες. Η αίτηση για κράτηση θα οδηγήσει σε μια νέα οθόνη όπου θα εμφανιστεί ο αριθμός κράτησης. Πρέπει να κρατηθεί αυτός ο αριθμός και να χρησιμοποιηθεί σε κάθε επικοινωνία με το ξενοδοχείο. Επιπλέον, όλες οι κρατήσεις επιβεβαιώνονται άμεσα (το αργότερο μέσα σε 30 λεπτά) με ένα email το οποίο θα αναφέρει τον αριθμό κράτησης. Επίσης, στο τέλος του email θα αναφέρεται ο πάροχος του ξενοδοχείου (provider) και ο αριθμός επιβεβαίωσης (provider code) τον οποίο ο πελάτης μπορεί να χρησιμοποιήσει μαζί με το όνομα του για το check in στο ξενοδοχείο.

Σε περίπτωση ακύρωσης μιας κράτησης ξενοδοχείου ενδέχεται ο πελάτης να δικαιούται μερική ή ολική επιστροφή του ποσού της κράτησης του. Το τελικό ποσό που θα επιστραφεί θα βασίζεται στην ακυρωτική πολιτική του ξενοδοχείου και στην αντίστοιχη ποινή που αυτή συνεπάγεται. Το ποσό που επιστρέφεται είναι το συνολικό ποσό της κράτησης μείον την ποινή της ακυρωτική πολιτικής. Να σημειωθεί ότι για οποιαδήποτε αλλαγή ή ακύρωση κράτησης υπάρχει χρέωση υπηρεσιών από τα συνεργαζόμενα site. Η επιστροφή των χρημάτων με βάση τα παραπάνω πραγματοποιείται σε χρονικό διάστημα μεταξύ 15 και 30 ημερών. Η πολιτική ακύρωσης του ξενοδοχείου εμφανίζεται στο τελευταίο βήμα της διαδικασίας κράτησης ξενοδοχείου πριν την εισαγωγή των στοιχείων της πιστωτικής κάρτας καθώς και στο email επιβεβαίωσης της κράτησης.

Η διαδικασία πληρωμής γίνεται με πιστωτική κάρτα και είναι απόλυτα ασφαλής. Την ευθύνη για τις συναλλαγές αναλαμβάνει η συμβεβλημένη τράπεζα, η οποία εξακριβώνει online την εγκυρότητα των στοιχείων της πιστωτικής κάρτας. Προκειμένου να διαφυλαχθεί η ασφάλεια των συναλλαγών χρησιμοποιείται ένας διακομιστής υψηλής ασφάλειας (SSL 128 bit) ο οποίος κρυπτογραφεί τον αριθμό της πιστωτικής κάρτας σε μη αναγνώσιμη μορφή. Τα στοιχεία της κάρτας γίνονται γνωστά μόνο στο χρηματοπιστωτικό ίδρυμα που διεκπεραιώνει τη συναλλαγή. Οι πληροφορίες της κάρτας δεν είναι αναγνώσιμες στο προσωπικό και δεν αποθηκεύονται στο σύστημα με οποιαδήποτε μορφή. Εάν ο πελάτης επιλέξει να πληρώσει με πιστωτική κάρτα, θα οδηγηθεί μέσω μιας ιστοσελίδας σε έναν ασφαλή δικτυακό τόπο, όπου θα πρέπει να δώσει τα στοιχεία της πιστωτικής του κάρτας και να επιβεβαιώσει το ποσό που θα χρεωθεί στην κάρτα του. Πληρωμή, εκτός της πιστωτικής κάρτας, μπορεί να γίνει μόνο με κατάθεση σε τραπεζικό λογαριασμό και έπειτα από τηλεφωνική επικοινωνία με το τμήμα εξυπηρέτησης πελατών.

**Υλοποίηση online συστήματος κρατήσεων ενοικιαζόμενων δωματίων**

Στο σύστημά μας υπάρχουν τρεις βασικοί χρήστες ο Πελάτης (Customer), ο Υπάλληλος της ρεσεψιόν (Receptionist) και ο Ιδιοκτήτης (Manager). Όπως φαίνεται στο παρακάτω διάγραμμα, οι δυνατές ενέργειες για τον Υπάλληλο της ρεσεψιόν είναι οι παρακάτω:

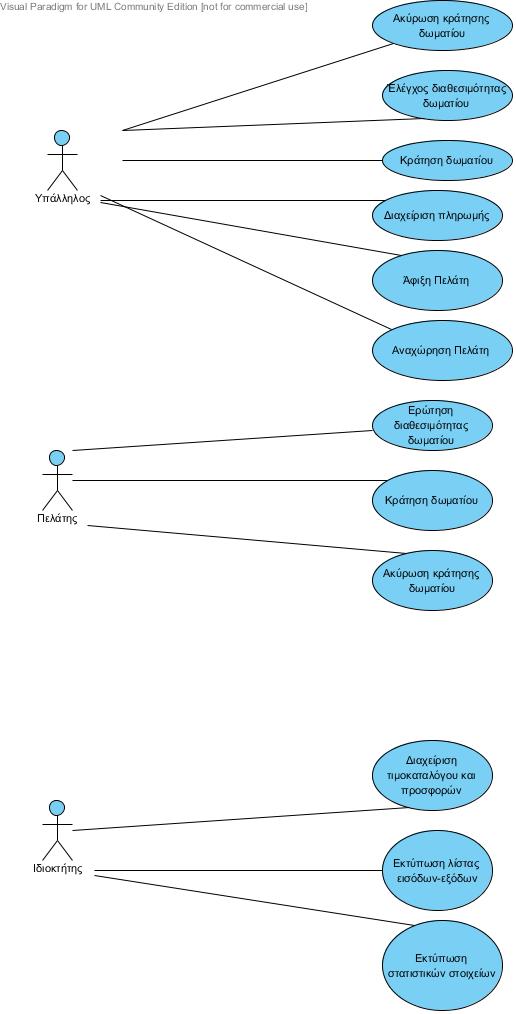
* Έλεγχος διαθεσιμότητας δωματίου
* Κράτηση δωματίου
* Ακύρωση κράτησης δωματίου
* Διαχείριση πληρωμής
* Άφιξη πελάτη
* Αναχώρηση πελάτη

Οι δυνατές ενέργειες για τον Πελάτη είναι οι παρακάτω:

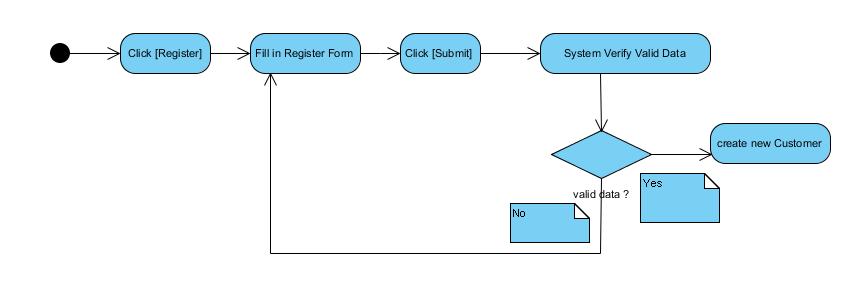
* Έλεγχος διαθεσιμότητας δωματίου
* Κράτηση δωματίου
* Ακύρωση κράτησης δωματίου

Οι δυνατές ενέργειες για τον Ιδιοκτήτη είναι οι παρακάτω:

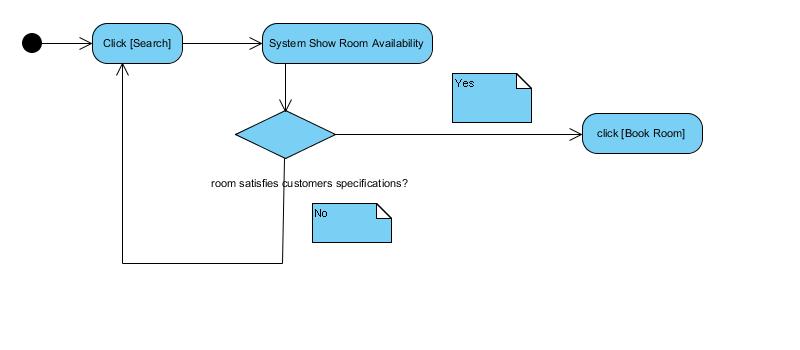
* Διαχείριση τιμοκαταλόγου και προσφορών
* Εκτύπωση λίστας εσόδων-εξόδων
* Εκτύπωση στατιστικών στοιχείων



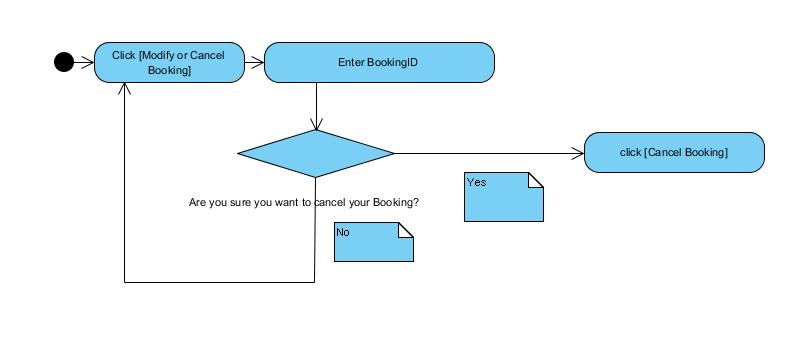
Στο ακόλουθο διάγραμμα ενεργειών παρουσιάζουμε τη διαδικασία της δημιουργίας ενός νέου λογαριασμού πελάτη στο σύστημα. Αφού ο πελάτης κάνει κλικ στο “Εγγραφή”, εισάγει τα στοιχεία του στις αντίστοιχες φόρμες και κάνει κλικ στο κουμπί που λέει "Υποβολή". Το σύστημα αφού ελέγξει ότι τα δεδομένα που έχει εισάγει ο χρήστης είναι έγκυρα προχωράει στη δημιουργία νέου λογαριασμού Πελάτη.



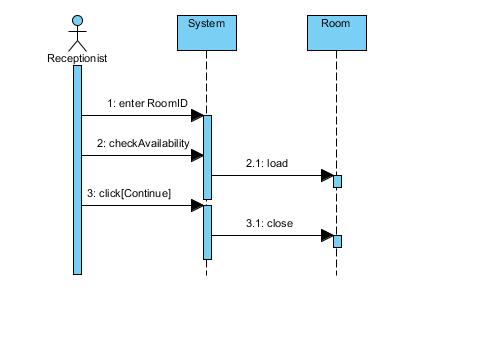
Στο ακόλουθο διάγραμμα ενεργειών παρουσιάζουμε τη διαδικασία της κράτησης δωματίου. Αφού ο πελάτης έχει επιλέξει την ημερομηνία άφιξης και αναχώρησης, τον αριθμό των ατόμων ανά δωμάτιο και τον αριθμό των δωματίων, και έχει κάνει κλικ στο κουμπί που λέει "Αναζήτηση", το σύστημα θα εμφανίσει τη διαθεσιμότητα των δωματίων για τις ημερομηνίες που έχουν επιλεγεί προηγουμένως. Αφού εμφανιστούν οι διαθέσιμοι τύποι δωματίων και οι τιμές τους, ο πελάτης μπορεί να επιλέξει το δωμάτιο με την τιμή που επιθυμεί κάνοντας κλικ στο κουμπί "Κράτηση Δωματίου". Έχοντας επιλέξει το δωμάτιο που επιθυμεί, μεταφέρεται στην φόρμα ασφαλούς κράτησης.



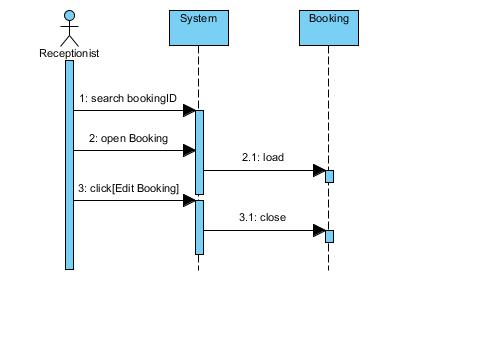
Στο ακόλουθο διάγραμμα ενεργειών παρουσιάζουμε τη διαδικασία της ακύρωσης δωματίου. Αφού ο πελάτης πάει στο κάτω μέρος της σελίδας και κάνει κλικ στο κουμπί ***“Τροποποίηση ή Ακύρωση κράτησης”*** μπορεί στη συνέχεια να δώσει τον αριθμό της κράτησης, και να κάνει κλικ στο κουμπί ***"Ακύρωση".*** Τότε το σύστημα θα εμφανίσει το ακόλουθο μήνυμα ***“Είσαστε βέβαιος ότι θέλετε να ακυρώσετε την κράτησή σας;»*** Ο πελάτης μπορεί να επιλέξει να ακυρώσει την κράτηση κάνοντας κλικ στο κουμπί "***Ακύρωση Κράτησης Δωματίου"***, οπότε και μεταφέρεται στην επόμενη φόρμα ακύρωσης κράτησης.



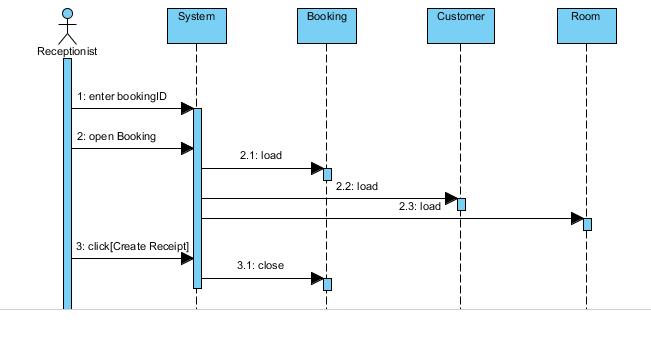
Στο ακόλουθο διάγραμμα ακολουθίας παρουσιάζουμε τη διαδικασία του ελέγχου διαθεσιμότητας δωματίου από τον υπάλληλο του ξενοδοχείου. Αφού ο υπάλληλος του ξενοδοχείου έχει επιλέξει την ημερομηνία άφιξης και αναχώρησης, τον αριθμό των ατόμων ανά δωμάτιο και τον αριθμό των δωματίων έπειτα από υπόδειξη του πελάτη, και έχει κάνει κλικ στο κουμπί που λέει ***"Αναζήτηση",*** το σύστημα θα εμφανίσει τα δωμάτια που είναι διαθέσιμα για τις ημερομηνίες που έχουν επιλεγεί προηγουμένως. Αφού εμφανιστούν οι διαθέσιμοι τύποι δωματίων, ο υπάλληλος του ξενοδοχείου μπορεί να εισάγει τον αναγνωριστικό κωδικό για κάποιο δωμάτιο στη φόρμα, ώστε το σύστημα να εμφανίσει τις πληροφορίες για το συγκεκριμένο δωμάτιο. Ύστερα, ο υπάλληλος του ξενοδοχείου, αφού ο πελάτης επιβεβαιώσει ότι επιθυμεί να μείνει στο συγκεκριμένο δωμάτιο, μπορεί να το επιλέξει κάνοντας κλικ στο κουμπί ***"Συνέχεια",*** ώστε να μεταβεί στη φόρμα κράτησης δωματίου.



Στο ακόλουθο διάγραμμα ακολουθίας παρουσιάζουμε τη διαδικασία της επεξεργασίας κράτησης δωματίου από τον υπάλληλο του ξενοδοχείου. Αφού ο υπάλληλος του ξενοδοχείου πάει κεντρικό μενού του λογισμικού και κάνει κλικ στο κουμπί ***“Τροποποίηση ή Ακύρωση κράτησης”*** μπορεί στη συνέχεια να εισάγει τον αναγνωριστικό κωδικό της κράτησης, και να κάνει κλικ στο κουμπί ***"Επεξεργασία Κράτησης”***, οπότε και μεταφέρεται στην επόμενη φόρμα επεξεργασίας των στοιχείων της κράτησης.

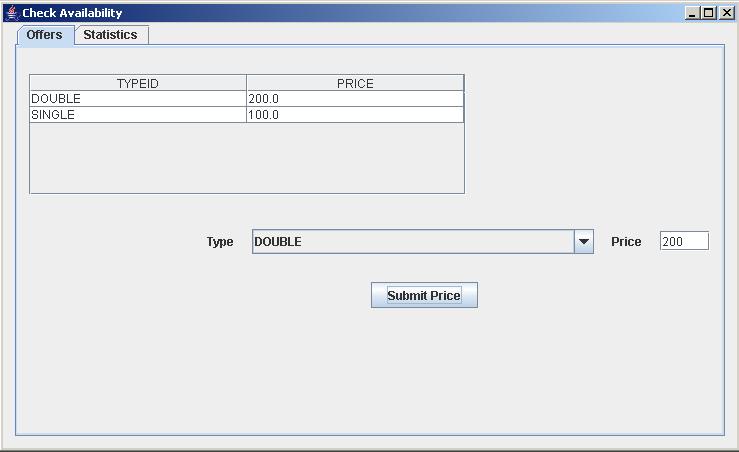


Στο ακόλουθο διάγραμμα ακολουθίας παρουσιάζουμε τη διαδικασία της εξόφλησης λογαριασμού από τον υπάλληλο του ξενοδοχείου. Αφού ο υπάλληλος του ξενοδοχείου πάει κεντρικό μενού του λογισμικού και κάνει κλικ στο κουμπί ***“Εξόφληση λογαριασμού”*** μπορεί στη συνέχεια να εισάγει τον αναγνωριστικό κωδικό της κράτησης και τον κωδικό του πελάτη, και να κάνει κλικ στο κουμπί ***" Δημιουργία Απόδειξης”***, οπότε και μεταφέρεται στην επόμενη φόρμα επεξεργασίας των στοιχείων της απόδειξης.

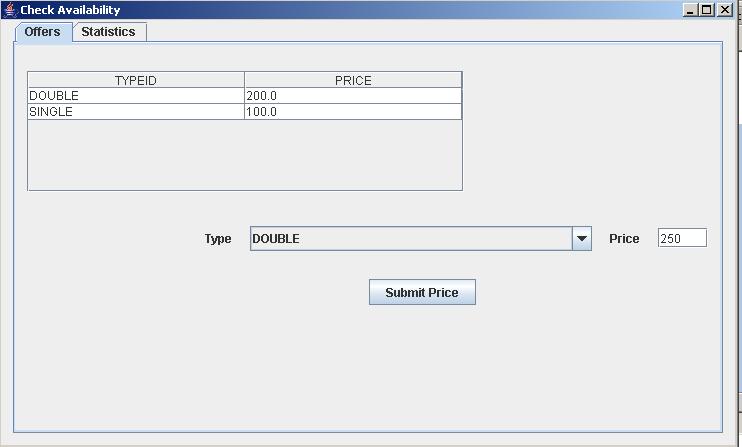


Στη συνέχεια παρουσιάζουμε την υλοποίηση της γραφικής διεπαφής για την εισαγωγή και επεξεργασία των δεδομένων της βάσης από τον Ιδιοκτήτη του ξενοδοχείου. Συγκεκριμένα, υλοποιήσαμε δύο καρτέλες **(Offers, Statistics)** για να δώσουμε τη δυνατότητα στον ιδιοκτήτη να θέτει τις τιμές των δωματίων **(Offers),** καθώς και να μπορεί να εκτυπώνει τα στοιχεία της βάσης για μία συγκεκριμένη χρονική περίοδο **(Statistics).** Η βάση δεδομένων αποτελείται από τέσσερις πίνακες τους **ROOMS, ROOMTYPE, BOOKING και CUSTOMER**.

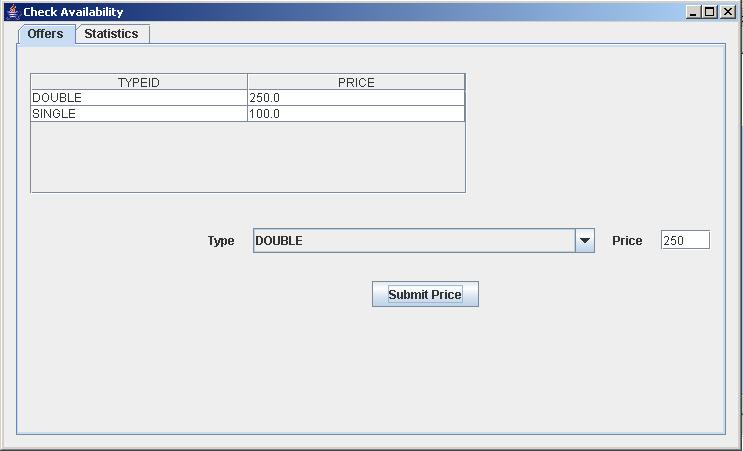
Στην παρακάτω εικόνα φαίνονται οι αρχικές τιμές (100 €, 200 €) των διαφορετικών τύπων δωματίων (SINGLE, DOUBLE).



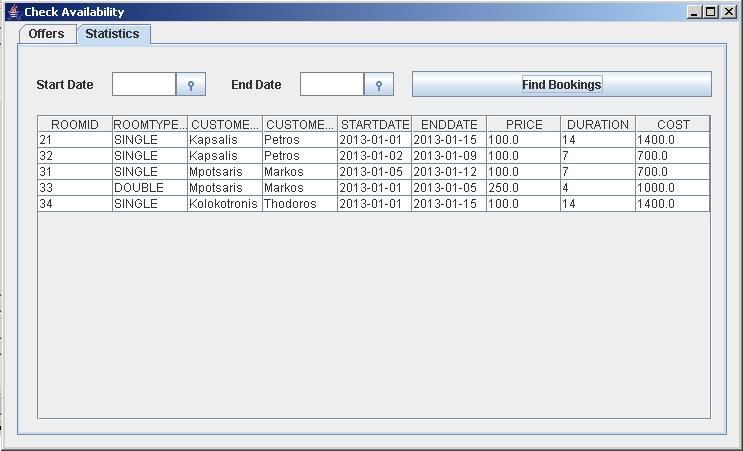
Στην παρακάτω εικόνα ο Ιδιοκτήτης του ξενοδοχείου εισάγει μία νέα τιμή (250€) για το δίκλινο δωμάτιο.



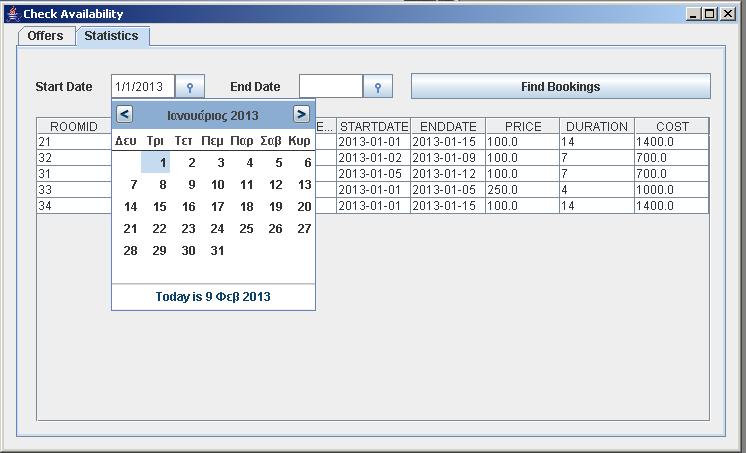
Στην παρακάτω εικόνα φαίνεται η ενημερωμένη βάση δεδομένων. Τώρα η τιμή των δίκλινων δωματίων έχει ενημερωθεί (250 €).



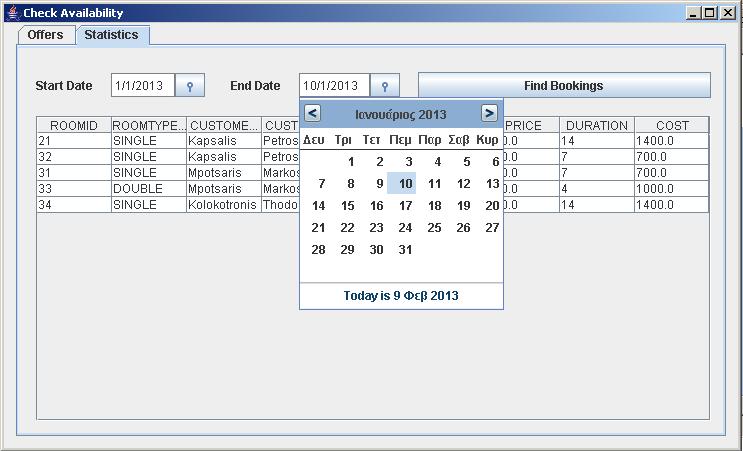
Στην αρχική εικόνα της καρτέλα Statistics φαίνονται όλες οι κρατήσεις για όλα τα δωμάτια.



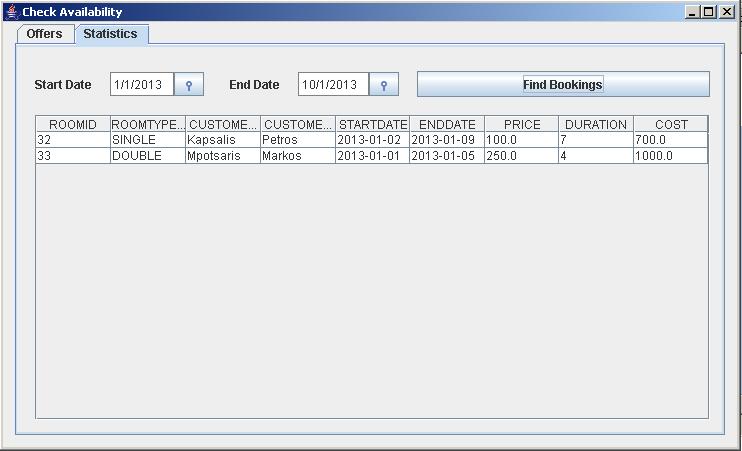
Στην παρακάτω εικόνα o ιδιοκτήτης του ξενοδοχείου επιλέγει την αρχική ημερομηνία του διαστήματος για το οποίο θέλει να ενημερωθεί με τις απαραίτητες πληροφορίες για τις κρατήσεις.



Στην παρακάτω εικόνα o ιδιοκτήτης του ξενοδοχείου επιλέγει την τελική ημερομηνία του διαστήματος για το οποίο θέλει να ενημερωθεί με τις απαραίτητες πληροφορίες για τις κρατήσεις.



Αφού o ιδιοκτήτης του ξενοδοχείου πατήσει το κουμπί Find Bookings, η καρτέλα Statistics ενημερώνεται με τις απαραίτητες πληροφορίες για τις κρατήσεις για το χρονικό διάστημα που επέλεξε προηγουμένως.



**Συμπεράσματα**

Στη σημερινή εποχή τα συστήματα κρατήσεων για τις τουριστικές επιχειρήσεις αποτελούν το κύριο μέσο προώθησης και διακίνησης των τουριστικών προϊόντων τους. Για το λόγο αυτό, όλο και περισσότερες τουριστικές επιχειρήσεις υιοθετούν τα συγκεκριμένα συστήματα διαχείρισης των δεδομένων τους. Στην αρχή της εργασίας μας ασχοληθήκαμε την περιγραφή ενός υπάρχοντος οnline συστήματος κρατήσεων ενοικιαζομένων δωματίων. Έπειτα περάσαμε στη συγκεκριμένη υλοποίηση του συστήματος ενημέρωσης των τιμών των ενοικιαζόμενων δωματίων, καθώς και στην προβολή των δεδομένων της βάσης για συγκεκριμένο χρονικό διάστημα. Τέλος, στο παράρτημα παρατίθεται ο κώδικας σε Java, με τον οποίο υλοποιήθηκαν όλα τα παραπάνω.

**User Acceptance Test (UAT)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1.** | **Test Case** |  |  |  |
| **Test #** | **Tasks** | **Expected Result** | **Notes** | **Pass /Fail** |
| **1.1** | Εισαγωγη Password + Username | Μπορει και γραφει Username+Password |  | Pass |
| **1.2** | Πατημα Clear | Σβησιμο γραμμενων TextField |  | Pass |
| **1.3** | Πατημα Εxit | Eξοδος απο την εφαρμογη |  | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **2.** | **Test Case** |  |  |  |
| **Test #** | **Tasks** | **Expected Result** | **Notes** | **Pass /Fail** |
| **2.1** | Εισαγωγη σωστου Password + Username | Γραφεται Username+Password |  | Pass |
| **2.2** | Πατημα Login | Μηνυμα Success | Βγαζει success | Pass |
| **2.3** | Πατημα Εxit | Eξοδος απο την εφαρμογη |  | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **3.** | **Test Case** |  |  |  |
| **Test #** | **Tasks** | **Expected Result** | **Notes** | **Pass /Fail** |
| **3.1** | Εισαγωγη λανθασμενου Password + Username | Γραφεται Username+Password |  | Pass |
| **3.2** | Πατημα Login | Μηνυμα λαθους | Βγαζει no success | Fail |
| **3.3** | Πατημα Εxit | Eξοδος απο την εφαρμογη |  | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **4.** | **Test Case** |  |  |  |
| **Test #** | **Tasks** | **Expected Result** | **Notes** | **Pass /Fail** |
| **4.1** | Εισαγωγη ονοματεπώνυμου + email | Γραφεται ονοματεπώνυμο + email |  | Pass |
| **4.2** | Πατημα Clear | Σβησιμο γραμμενων TextField |  | Pass |
| **4.3** | Πατημα Εxit | Eξοδος απο την εφαρμογη |  | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **5.** | **Test Case** |  |  |  |
| **Test #** | **Tasks** | **Expected Result** | **Notes** | **Pass /Fail** |
| **5.1** | Εισαγωγη σωστου ονοματεπώνυμου + email | Γραφεται ονοματεπώνυμο + email |  | Pass |
| **5.2** | Πατημα Κράτηση | Μηνυμα Success | Βγαζει success | Pass |
| **5.3** | Πατημα Εxit | Eξοδος απο την εφαρμογη |  | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **6.** | **Test Case** |  |  |  |
| **Test #** | **Tasks** | **Expected Result** | **Notes** | **Pass /Fail** |
| **6.1** | Εισαγωγη λανθασμενου ονοματεπώνυμου + email | Γραφεται ονοματεπώνυμο + email |  | Pass |
| **6.2** | Πατημα Κράτηση | Μηνυμα λαθους | Βγαζει no success | Fail |
| **6.3** | Πατημα Εxit | Eξοδος απο την εφαρμογη |  | Pass |

**Παράρτημα**

**Πηγαίος κώδικας σε Java**

% main.java

package javaapplication1;

import java.beans.Statement;

import java.sql.Connection;

import java.sql.Driver;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.logging.Level;

import java.util.logging.Logger;

public class Main {

public static void main(String[] args) {

Main main=new Main();

NewJFrame mainFrame = new NewJFrame();

mainFrame.setVisible(true);

}

}

% DatabaseHandler.java

package javaapplication1;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.ResultSetMetaData;

import java.sql.SQLException;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.Vector;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.table.DefaultTableModel;

public class DatabaseHandler {

public static void main(String[] args) {

DatabaseHandler db = new DatabaseHandler();

db.printRooms();

db.printRoomTypes();

}

public void insertIntoDatabase(int id, String type, boolean booked) {

Connection conn = null;

PreparedStatement stmt = null;

ResultSet rs = null;

try {

conn = getHSQLConnection();

String sql = "INSERT INTO ROOMS VALUES (?,?,?)";

stmt = conn.prepareStatement(sql);

stmt.setInt(1, id);

stmt.setString(2, type);

stmt.setInt(3, booked ? 1 : 0);

stmt.execute();

} catch (Exception ex) {

Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);

} finally {

try {

stmt.close();

conn.close();

} catch (SQLException ex) {

Logger.getLogger(DatabaseHandler.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

public void setPriceForRoomType(String roomTypeId, float price) {

Connection conn = null;

PreparedStatement stmt = null;

ResultSet rs = null;

try {

conn = getHSQLConnection();

String sql = "UPDATE ROOMTYPE SET PRICE=? WHERE TYPEID=?";

stmt = conn.prepareStatement(sql);

stmt.setFloat(1, price);

stmt.setString(2, roomTypeId);

stmt.execute();

printRoomTypes();

} catch (Exception ex) {

Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);

} finally {

try {

stmt.close();

conn.close();

} catch (SQLException ex) {

Logger.getLogger(DatabaseHandler.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

public void printRooms() {

Connection conn = null;

PreparedStatement stmt = null;

ResultSet rs = null;

try {

conn = getHSQLConnection();

String sql = "SELECT \* FROM ROOMS";

stmt = conn.prepareStatement(sql);

rs = stmt.executeQuery();

while (rs.next()) {

System.out.println("Room " + rs.getInt("RoomId") + " | " + rs.getString("RoomTypeId") + " | " + rs.getString("Booked"));

}

} catch (Exception ex) {

Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);

} finally {

try {

rs.close();

stmt.close();

conn.close();

} catch (SQLException ex) {

Logger.getLogger(DatabaseHandler.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

public void printRoomTypes() {

Connection conn = null;

PreparedStatement stmt = null;

ResultSet rs = null;

try {

conn = getHSQLConnection();

String sql = "SELECT \* FROM ROOMTYPE";

stmt = conn.prepareStatement(sql);

rs = stmt.executeQuery();

while (rs.next()) {

System.out.println(rs.getString("TypeId") + " | " + rs.getString("Price"));

}

} catch (Exception ex) {

Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);

} finally {

try {

rs.close();

stmt.close();

conn.close();

} catch (SQLException ex) {

Logger.getLogger(DatabaseHandler.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

private static Connection getHSQLConnection() throws Exception {

Class.forName("org.hsqldb.jdbcDriver");

String url = "jdbc:hsqldb:file:hotel.db";

Connection conn = DriverManager.getConnection(url, "sa", "");

conn.setAutoCommit(true);

return conn;

}

public static DefaultTableModel buildTableModel(ResultSet rs)

throws SQLException {

ResultSetMetaData metaData = rs.getMetaData();

// names of columns

Vector<String> columnNames = new Vector<String>();

int columnCount = metaData.getColumnCount();

for (int column = 1; column <= columnCount; column++) {

columnNames.add(metaData.getColumnName(column));

}

// data of the table

Vector<Vector<Object>> data = new Vector<Vector<Object>>();

while (rs.next()) {

Vector<Object> vector = new Vector<Object>();

for (int columnIndex = 1; columnIndex <= columnCount; columnIndex++) {

vector.add(rs.getObject(columnIndex));

}

data.add(vector);

}

return new DefaultTableModel(data, columnNames);

}

public DefaultTableModel getPriceTableModel() {

Connection conn = null;

PreparedStatement stmt = null;

ResultSet rs = null;

DefaultTableModel model = null;

try {

conn = getHSQLConnection();

String sql = "SELECT \* FROM ROOMTYPE";

stmt = conn.prepareStatement(sql);

rs = stmt.executeQuery();

model = buildTableModel(rs);

} catch (Exception ex) {

Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);

} finally {

try {

rs.close();

stmt.close();

conn.close();

} catch (SQLException ex) {

Logger.getLogger(DatabaseHandler.class.getName()).log(Level.SEVERE, null, ex);

}

}

return model;

}

public DefaultTableModel getBookingTableModel(Date startDate, Date endDate) {

Connection conn = null;

PreparedStatement stmt = null;

ResultSet rs = null;

DefaultTableModel model = null;

SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd");

try {

conn = getHSQLConnection();

String sql = "select roomId, roomTypeId, customerLastName, customerFirstName, "

+ "startDate, endDate, price,DATEDIFF ( 'day', startDate,endDate ) as duration, " +

" DATEDIFF ( 'day', startDate,endDate )\*price as cost" +

" from customer, rooms, roomtype, booking "

+ "where rooms.roomtypeid=roomtype.typeid and booking.roomid=rooms.roomid and "

+ " booking.customerid=customer.customerid " ;

if(startDate != null) {

sql += " and startDate >= '"+sdf.format(startDate)+"'";

}

if (endDate != null) {

sql+="and endDate <= '"+sdf.format(endDate)+"'";

}

stmt = conn.prepareStatement(sql);

rs = stmt.executeQuery();

model = buildTableModel(rs);

} catch (Exception ex) {

Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);

} finally {

try {

rs.close();

stmt.close();

conn.close();

} catch (SQLException ex) {

Logger.getLogger(DatabaseHandler.class.getName()).log(Level.SEVERE, null, ex);

}

}

return model;

}

}

% NewJFrame.java

package javaapplication1;

import java.text.SimpleDateFormat;

import java.util.Date;

import javax.swing.JButton;

public class NewJFrame extends javax.swing.JFrame {

public NewJFrame() {

initComponents();

}

private void initComponents() {

systemMessage = new javax.swing.JDialog();

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

jTabbedPane1 = new javax.swing.JTabbedPane();

offersPane = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

priceText = new javax.swing.JTextField();

roomTypeCombo = new javax.swing.JComboBox();

jLabel2 = new javax.swing.JLabel();

submitPriceButton = new javax.swing.JButton();

jScrollPane2 = new javax.swing.JScrollPane();

priceTable = new javax.swing.JTable(db.getPriceTableModel());

jPanel1 = new javax.swing.JPanel();

jScrollPane3 = new javax.swing.JScrollPane();

bookingTable = new javax.swing.JTable();

startDatePicker = new org.jdesktop.swingx.JXDatePicker();

endDatePicker = new org.jdesktop.swingx.JXDatePicker();

jButton1 = new javax.swing.JButton();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

systemMessage.setTitle("SystemMessage");

org.jdesktop.layout.GroupLayout systemMessageLayout = new org.jdesktop.layout.GroupLayout(systemMessage.getContentPane());

systemMessage.getContentPane().setLayout(systemMessageLayout);

systemMessageLayout.setHorizontalGroup(

systemMessageLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(0, 400, Short.MAX\_VALUE)

);

systemMessageLayout.setVerticalGroup(

systemMessageLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(0, 300, Short.MAX\_VALUE)

);

jTable1.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

{null, null, null, null},

{null, null, null, null},

{null, null, null, null},

{null, null, null, null}

},

new String [] {

"Title 1", "Title 2", "Title 3", "Title 4"

}

));

jScrollPane1.setViewportView(jTable1);

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Check Availability");

jLabel1.setText("Type");

priceText.setText("0.0");

priceText.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

priceTextActionPerformed(evt);

}

});

roomTypeCombo.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "SINGLE", "DOUBLE" }));

roomTypeCombo.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

roomTypeComboActionPerformed(evt);

}

});

roomTypeCombo.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

roomTypeComboMouseClicked(evt);

}

});

jLabel2.setText("Price");

submitPriceButton.setText("Submit Price");

submitPriceButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

submitPriceButtonActionPerformed(evt);

}

});

jScrollPane2.setViewportView(priceTable);

org.jdesktop.layout.GroupLayout offersPaneLayout = new org.jdesktop.layout.GroupLayout(offersPane);

offersPane.setLayout(offersPaneLayout);

offersPaneLayout.setHorizontalGroup(

offersPaneLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(offersPaneLayout.createSequentialGroup()

.add(offersPaneLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(offersPaneLayout.createSequentialGroup()

.add(354, 354, 354)

.add(submitPriceButton))

.add(org.jdesktop.layout.GroupLayout.TRAILING, offersPaneLayout.createSequentialGroup()

.add(190, 190, 190)

.add(jLabel1)

.add(18, 18, 18)

.add(roomTypeCombo, 0, 106, Short.MAX\_VALUE)

.add(18, 18, 18)

.add(jLabel2)

.add(18, 18, 18)

.add(priceText, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 50, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE))

.add(offersPaneLayout.createSequentialGroup()

.addContainerGap()

.add(jScrollPane2, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 437, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap())

);

offersPaneLayout.setVerticalGroup(

offersPaneLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(offersPaneLayout.createSequentialGroup()

.add(26, 26, 26)

.add(jScrollPane2, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 121, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.add(34, 34, 34)

.add(offersPaneLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)

.add(roomTypeCombo, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.add(jLabel1)

.add(priceText, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.add(jLabel2))

.add(28, 28, 28)

.add(submitPriceButton)

.addContainerGap(61, Short.MAX\_VALUE))

);

jTabbedPane1.addTab("Offers", offersPane);

bookingTable.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

{null, null, null, null},

{null, null, null, null},

{null, null, null, null},

{null, null, null, null}

},

new String [] {

"Title 1", "Title 2", "Title 3", "Title 4"

}

));

jScrollPane3.setViewportView(bookingTable);

jButton1.setText("Find Bookings");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jLabel3.setText("Start Date");

jLabel4.setText("End Date");

org.jdesktop.layout.GroupLayout jPanel1Layout = new org.jdesktop.layout.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(jPanel1Layout.createSequentialGroup()

.add(18, 18, 18)

.add(jPanel1Layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(org.jdesktop.layout.GroupLayout.TRAILING, jScrollPane3, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, 429, Short.MAX\_VALUE)

.add(jPanel1Layout.createSequentialGroup()

.add(jLabel3)

.addPreferredGap(org.jdesktop.layout.LayoutStyle.UNRELATED)

.add(startDatePicker, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.add(26, 26, 26)

.add(jLabel4)

.addPreferredGap(org.jdesktop.layout.LayoutStyle.UNRELATED)

.add(endDatePicker, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.add(18, 18, 18)

.add(jButton1, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, 105, Short.MAX\_VALUE)))

.addContainerGap())

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(jPanel1Layout.createSequentialGroup()

.add(24, 24, 24)

.add(jPanel1Layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)

.add(startDatePicker, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.add(endDatePicker, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.add(jLabel3)

.add(jLabel4)

.add(jButton1))

.add(18, 18, 18)

.add(jScrollPane3, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, 238, Short.MAX\_VALUE)

.addContainerGap())

);

jTabbedPane1.addTab("Statistics", jPanel1);

org.jdesktop.layout.GroupLayout layout = new org.jdesktop.layout.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(layout.createSequentialGroup()

.addContainerGap()

.add(jTabbedPane1, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, 462, Short.MAX\_VALUE)

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(layout.createSequentialGroup()

.add(jTabbedPane1, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, 340, Short.MAX\_VALUE)

.addContainerGap())

);

pack();

}// </editor-fold>

private void roomTypeComboActionPerformed(java.awt.event.ActionEvent evt) {

}

private void priceTextActionPerformed(java.awt.event.ActionEvent evt) {

}

private void roomTypeComboMouseClicked(java.awt.event.MouseEvent evt) {

}

private void submitPriceButtonActionPerformed(java.awt.event.ActionEvent evt) {

float price=Float.parseFloat(priceText.getText());

String roomType=(String)roomTypeCombo.getSelectedItem();

db.setPriceForRoomType(roomType,price);

priceTable.setModel(db.getPriceTableModel());

System.out.println("Set price for room type "+roomType+" to "+price);

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

Date startDate=startDatePicker.getDate();

Date endDate=endDatePicker.getDate();

bookingTable.setModel(db.getBookingTableModel(startDate,endDate));

}

private DatabaseHandler db=new DatabaseHandler();

private javax.swing.JTable bookingTable;

private org.jdesktop.swingx.JXDatePicker endDatePicker;

private javax.swing.JButton jButton1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JScrollPane jScrollPane3;

private javax.swing.JTabbedPane jTabbedPane1;

private javax.swing.JTable jTable1;

private javax.swing.JPanel offersPane;

private javax.swing.JTable priceTable;

private javax.swing.JTextField priceText;

private javax.swing.JComboBox roomTypeCombo;

private org.jdesktop.swingx.JXDatePicker startDatePicker;

private javax.swing.JButton submitPriceButton;

private javax.swing.JDialog systemMessage;

}

% hotel.db.script

CREATE SCHEMA PUBLIC AUTHORIZATION DBA

CREATE MEMORY TABLE ROOMS(ROOMID SMALLINT NOT NULL PRIMARY KEY,ROOMTYPEID VARCHAR(6),BOOKED BOOLEAN)

CREATE MEMORY TABLE ROOMTYPE(TYPEID VARCHAR(6) NOT NULL PRIMARY KEY,PRICE FLOAT)

CREATE MEMORY TABLE CUSTOMER(CUSTOMERID INTEGER NOT NULL PRIMARY KEY,CUSTOMERLASTNAME VARCHAR(20),CUSTOMERFIRSTNAME VARCHAR(20),CUSTOMERADDRESS VARCHAR(20))

CREATE MEMORY TABLE BOOKING(ROOMID INTEGER NOT NULL,CUSTOMERID INTEGER NOT NULL,STARTDATE DATE,ENDDATE DATE)

CREATE USER SA PASSWORD ""

GRANT DBA TO SA

SET WRITE\_DELAY 0 MILLIS

SET SCHEMA PUBLIC

INSERT INTO ROOMS VALUES(21,'SINGLE',FALSE)

INSERT INTO ROOMS VALUES(22,'SINGLE',FALSE)

INSERT INTO ROOMS VALUES(23,'DOUBLE',TRUE)

INSERT INTO ROOMS VALUES(24,'SINGLE',FALSE)

INSERT INTO ROOMS VALUES(31,'SINGLE',FALSE)

INSERT INTO ROOMS VALUES(32,'SINGLE',FALSE)

INSERT INTO ROOMS VALUES(33,'DOUBLE',TRUE)

INSERT INTO ROOMS VALUES(34,'SINGLE',FALSE)

INSERT INTO ROOMTYPE VALUES('DOUBLE',33.0E0)

INSERT INTO ROOMTYPE VALUES('SINGLE',3.0E0)

INSERT INTO CUSTOMER VALUES(1,'Kapsalis','Petros','petros@gmail.com')

INSERT INTO CUSTOMER VALUES(2,'Mpotsaris','Markos','markos@gmail.com')

INSERT INTO CUSTOMER VALUES(3,'Androutsos','Odysseus','odysseus@gmail.com')

INSERT INTO CUSTOMER VALUES(4,'Kolokotronis','Thodoros','thodoros@gmail.com')

INSERT INTO BOOKING VALUES(21,1,'2013-01-01','2013-01-15')

INSERT INTO BOOKING VALUES(31,2,'2013-01-05','2013-01-12')

INSERT INTO BOOKING VALUES(32,1,'2013-01-02','2013-01-09')

INSERT INTO BOOKING VALUES(33,2,'2013-01-01','2013-01-05')

INSERT INTO BOOKING VALUES(34,4,'2013-01-01','2013-01-15')