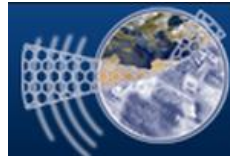




Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

**Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Μηχανικών Πληροφορικής**



Πτυχιακή Εργασία

**Portal διαχείρισης
αλυσίδας ξενοδοχείων**

**Μαρία Κουμή (ΑΜ: 1412)
Ιωάννης Αλεξανδράκης (ΑΜ:1165)**

Επιβλέπων Καθηγητής: Νικόλαος Παπαδάκης

**ΗΡΑΚΛΕΙΟ
2015**

Abstract

In modern times, the tourism industry is constantly evolving via the use of integrated information systems in order to provide customer service. The popularity of using the internet enhances the integration and the technologies developed are increasingly specialized to handle the volume of information and the public.

Particular emphasis has been given to internet search engines and booking services in tourism, to increase the profits of tourism enterprises. In this paper we study the technologies that such a booking system consists of, the expertise required, and the integrated design of such infrastructure.

Σύνοψη

Στην σύγχρονη εποχή, ο τουριστικός τομέας ενισχύεται διαρκώς μέσω της χρήσης ολοκληρωμένων πληροφοριακών συστημάτων για την εξυπηρέτηση των πελατών του. Η χρήση του διαδικτύου εξυπηρετεί άμεσα το σκοπό αυτό και οι τεχνολογίες που αναπτύσσονται είναι ολοένα και πιο εξειδικευμένες, για να χειριστούν τον όγκο πληροφορίας και κοινού.

Ιδιαίτερη έμφαση δίνεται στις διαδικτυακές μηχανές αναζήτησης και κράτησης τουριστικών υπηρεσιών, για την αύξηση των κερδών των τουριστικών επιχειρήσεων. Στην εργασία αυτή μελετώνται οι τεχνολογίες που συνθέτουν ένα τέτοιο ολοκληρωμένο σύστημα κρατήσεων, η τεχνογνωσία που απαιτείται, και η ολοκληρωμένη σχεδίαση μίας τέτοιας υποδομής.

Πίνακας Περιεχομένων

1	Εισαγωγή	- 1 -
1.1	Περίληψη	- 1 -
1.2	Κίνητρο για τη διεξαγωγή της εργασίας	- 1 -
1.3	Σκοπός και στόχοι της εργασίας	- 1 -
1.4	Δομή της εργασίας	- 2 -
2	Συστήματα Διαχείρισης Βάσεων Δεδομένων	- 3 -
2.1	Βάσεις Δεδομένων και Συστήματα Διαχείρισης	- 3 -
2.2	Πλεονεκτήματα χρήσης	- 4 -
3	Ανάλυση απαιτήσεων για DBMS	- 6 -
3.1	Βασικές έννοιες ανάλυσης συστήματος	- 6 -
3.1.1	Οντότητα	- 6 -
3.1.2	Ιδιότητα	- 6 -
3.1.3	Συσχέτιση	- 6 -
3.1.4	Οι διμελείς συσχετίσεις	- 7 -
3.1.5	Κλειδιά	- 7 -
3.2	Μοντέλο Οντοτήτων – Συσχετίσεων	- 7 -
3.3	Κανονικοποίηση	- 8 -
3.3.1	Πρώτη Κανονική Μορφή (1NF)	- 8 -
3.3.2	Δεύτερη Κανονική Μορφή (2NF)	- 8 -
3.3.3	Τρίτη Κανονική Μορφή (3NF)	- 9 -
3.3.4	Boyce-Codd κανονική μορφή (BCNF)	- 9 -
3.3.5	Ανώτερες Κανονικές Μορφές	- 9 -
3.4	Σχεδιασμός βάσης δεδομένων	- 9 -
4	Ταξιδιωτικά Συστήματα Κρατήσεων	- 11 -
4.1	Παγκόσμια συστήματα διανομής	- 11 -
4.1.1	Sabre	- 11 -
4.1.2	Worldspan	- 11 -
4.1.3	Galileo	- 12 -
4.1.4	Amadeus	- 12 -
4.1.5	Η χρήση των GDS στον ξενοδοχειακό τομέα	- 12 -
4.2	Διαδικτυακές μηχανές κρατήσεων	- 13 -
4.2.1	Content Inn by Amadeus	- 13 -
4.2.2	HIT-Hospitality	- 14 -
4.2.3	BOOKonlinenow by IMTEC	- 14 -
4.2.4	Travel Platform by Nelios.com	- 15 -
4.2.5	nsTravel, nsTIS by NetSemantics	- 15 -
4.2.6	SingularLogic Hotel	- 15 -
4.2.7	Sunsoft	- 16 -
4.2.8	Sysco	- 16 -
4.3	Διαδικασία online κράτησης	- 16 -
4.3.1	Βήμα 1 – Αναζήτηση βάσει προτιμήσεων	- 16 -
4.3.2	Βήμα 2 – Εμφάνιση διαθεσιμότητας	- 17 -
4.3.3	Βήμα 3 – Επιλογή χρήστη	- 17 -
4.3.4	Βήμα 4 – Καταχώρηση στοιχείων επισκεπτών	- 17 -
4.3.5	Βήμα 5 – Πληρωμή και ολοκλήρωση κράτησης	- 17 -
4.3.6	Βήμα 6 – Έκδοση ηλεκτρονικής κράτησης και επιβεβαίωση	- 18 -

5	Ανάλυση Εφαρμογής.....	- 19 -
5.1	Τεχνολογίες που χρησιμοποιήθηκαν	- 19 -
5.1.1	MySQL.....	- 19 -
5.1.2	PHP.....	- 19 -
5.1.3	HTML και CSS	- 20 -
5.1.4	JavaScript	- 20 -
6	Δομή εφαρμογής.....	- 21 -
6.1	Απαιτήσεις συστήματος	- 21 -
6.2	Επικοινωνία με τη Βάση Δεδομένων	- 22 -
6.3	Δομή βάσης δεδομένων.....	- 22 -
6.3.1	Οντότητες και Συσχετίσεις.....	- 22 -
6.3.2	Διάγραμμα Οντοτήτων – Συσχετίσεων Βάσης Δεδομένων	- 26 -
6.3.3	Σχεσιακό μοντέλο βάσης.....	- 28 -
6.4	Ανάλυση διαδικασίας κράτησης	- 31 -
6.4.1	Σελίδα αναζήτησης καταλυμάτων.....	- 31 -
6.4.2	Σελίδα αποτελεσμάτων διαθεσιμότητας.....	- 32 -
6.4.3	Σελίδα στοιχείων κράτησης.....	- 37 -
6.4.4	Σελίδα πληρωμής	- 37 -
6.4.5	Σελίδα επιβεβαίωσης και email.....	- 41 -
6.4.6	Σελίδα ακύρωσης κράτησης.....	- 41 -
7	Χρήστες συστήματος	- 43 -
7.1	Είδη χρηστών	- 43 -
7.2	Απλοί χρήστες	- 43 -
7.2.1	Εγγραφή νέου χρήστη	- 43 -
7.2.2	Είσοδος/Εξοδος χρήστη	- 45 -
7.2.3	Αυτόματη συμπλήρωση στοιχείων.....	- 46 -
7.3	Διαχειριστές.....	- 47 -
7.3.1	Δικαιώματα διαχειριστών.....	- 47 -
7.3.2	Διαχειριστικό πάνελ	- 47 -
7.3.3	Αναφορές συστήματος	- 55 -
8	Συμπεράσματα και Μελλοντικές επεκτάσεις	- 59 -
	Βιβλιογραφία	- 60 -
	ΠΑΡΑΡΤΗΜΑ Α.....	- 62 -
	ΠΑΡΑΡΤΗΜΑ Β.....	- 69 -

Πίνακας Εικόνων

Εικόνα 1 – Σχεδίαση οντότητας	- 6 -
Εικόνα 2 – Σχεδίαση ιδιοτήτων οντότητας	- 6 -
Εικόνα 3 – Δομή αρχείων web server	- 21 -
Εικόνα 4 – Δομή αρχείων back-end διαχείρισης.....	- 21 -
Εικόνα 5 – Σχεδίαση οντότητας USER.....	- 23 -
Εικόνα 6 – Σχεδίαση οντότητας CREDIT CARD.....	- 23 -
Εικόνα 7 – Σχεδίαση οντότητας SITE.....	- 23 -
Εικόνα 8 – Σχεδίαση οντότητας COUNTRY.....	- 24 -
Εικόνα 9 – Σχεδίαση οντότητας CITY.....	- 24 -
Εικόνα 10 – Σχεδίαση οντότητας HOTEL.....	- 24 -
Εικόνα 11 – Σχεδίαση οντότητας SEASON.....	- 24 -
Εικόνα 12 – Σχεδίαση οντότητας ROOM.....	- 25 -
Εικόνα 13 – Σχεδίαση οντότητας CAPACITY.....	- 25 -
Εικόνα 14 – Σχεδίαση οντότητας RATE.....	- 25 -
Εικόνα 15 – Σχεδίαση οντότητας BOOKING.....	- 26 -
Εικόνα 16 – Καθολικό διάγραμμα οντοτήτων – συσχετίσεων του συστήματος.....	- 27 -
Εικόνα 17 – Σχεσιακό μοντέλο χρηστών	- 28 -
Εικόνα 18 – Σχεσιακό μοντέλο γλωσσών site.....	- 28 -
Εικόνα 19 – Σχεσιακό μοντέλο γεωγραφικών τοποθεσιών.....	- 29 -
Εικόνα 20 – Απεικόνιση συσχέτισης χωρών και χρηστών	- 29 -
Εικόνα 21 – Σχεσιακό μοντέλο ξενοδοχείου.....	- 30 -
Εικόνα 22 – Σχεσιακό μοντέλο δωματίων ξενοδοχείου.....	- 30 -
Εικόνα 23 – Απεικόνιση συσχέτισης ξενοδοχείων και δωματίων	- 31 -
Εικόνα 24 – Σχεσιακό μοντέλο κρατήσεων	- 31 -
Εικόνα 25 – Παράδειγμα συμπληρωμένης φόρμας αναζήτησης	- 32 -
Εικόνα 26 – Σελίδα αποτελεσμάτων – ξενοδοχείο	- 35 -
Εικόνα 27 – Σελίδα αποτελεσμάτων - δωμάτια	- 36 -
Εικόνα 28 – Φόρμα συμπλήρωσης στοιχείων κράτησης	- 37 -
Εικόνα 29 – Σελίδα πληρωμής με πιστ. κάρτα.....	- 38 -
Εικόνα 30 – Κλειδιά επικοινωνίας Stripe.....	- 38 -
Εικόνα 31 – Ανάλυση χρεώσεων μέσω Stripe	- 39 -
Εικόνα 32 – Φόρμα πληρωμής Stripe	- 40 -
Εικόνα 33 – Φόρμα ακυρώσεων	- 41 -
Εικόνα 34 – Σημείο εγγραφής χρήστη στη κεντρική σελίδα	- 43 -
Εικόνα 35 – Φόρμα εγγραφής νέου χρήστη.....	- 44 -
Εικόνα 36 – Φόρμα εγγραφής με συμπληρωμένα πεδία.....	- 44 -
Εικόνα 37 – Πίνακας βάσης καταχωρημένων χρηστών.....	- 44 -
Εικόνα 38 – Μήνυμα επιβεβαίωσης εγγραφής χρήστη.....	- 45 -
Εικόνα 39 – Μήνυμα σφάλματος εισόδου χρήστη στο σύστημα.....	- 45 -
Εικόνα 40 – Φόρμα επικοινωνίας με αυτομάτως συμπληρωμένα στοιχεία	- 46 -
Εικόνα 41 – Φόρμα εισόδου διαχειριστών.....	- 47 -
Εικόνα 42 – Σφάλμα εισόδου διαχειριστή	- 48 -
Εικόνα 43 – Διαχειριστικό πάνελ.....	- 48 -
Εικόνα 44 – Σύνδεσμος εισαγωγής ξενοδοχείου.....	- 48 -
Εικόνα 45 – Σελίδα προσθήκης στοιχείων ξενοδοχείου	- 49 -
Εικόνα 46 – Σελίδα διαχείρισης τιμών και σεζόν	- 52 -
Εικόνα 47 – Σύνδεσμος τροποποίησης ξενοδοχείου/τιμών	- 53 -
Εικόνα 48 – Σύνδεσμος διαγραφής ξενοδοχείου	- 53 -
Εικόνα 49 – Σύνδεσμοι διαχείρισης χρηστών.....	- 54 -
Εικόνα 50 – Φόρμα δημιουργίας χρήστη στο διαχειριστικό.....	- 54 -
Εικόνα 51 – Τροποποίηση χρήστη στο διαχειριστικό.....	- 54 -
Εικόνα 52 – Περιοχή στατιστικών αναφορών στο διαχειριστικό πάνελ.....	- 55 -
Εικόνα 53 – Σελίδα ανάλυσης οφειλών ξενοδοχείου.....	- 55 -
Εικόνα 54 – Δυνατότητες εκτύπωσης και αποθήκευσης γραφημάτων	- 56 -

Εικόνα 55- Σελίδα αναφορών πληρότητας ξενοδοχείου	- 57 -
Εικόνα 56 – Μενού προβολής κρατήσεων συστήματος	- 57 -
Εικόνα 57 – Κατάσταση κρατήσεων στο σύστημα.....	- 58 -

Πίνακας Σχημάτων

Σχήμα 1 – Κώδικας σύνδεσης στη βάση.....	- 22 -
Σχήμα 2 – Μεταβλητές σύνδεσης στη βάση	- 22 -

1 Εισαγωγή

1.1 Περίληψη

Στην εργασία αυτή θα γίνει μία μελέτη και ανάλυση των τεχνολογιών που συνθέτουν ένα σύστημα διαχείρισης ξενοδοχειακών κρατήσεων. Αρχικά, θα παρουσιαστούν οι τρόποι διαχείρισης σύνθετης πληροφορίας μέσω εξελιγμένων συστημάτων διαχείρισης βάσεων δεδομένων και τι είναι τα συστήματα αυτά. Εν συνεχεία, θα αποσαφηνιστούν κάποιοι βασικοί όροι ανάλυσης και σχεδιασμού ενός συστήματος κρατήσεων και η δημιουργία των σχετικών μοντέλων βάσεων δεδομένων τους. Στο επόμενο βήμα θα παρουσιαστούν τα πιο σύγχρονα συστήματα κρατήσεων και διαχείρισης ξενοδοχείων από μεγάλες εταιρίες πληροφορικής και τεχνολογίας, και πως το καθένα υλοποιεί το δικό του μηχανισμό κράτησης. Κατόπιν της ανάλυσης αυτής, θα προχωρήσουμε στην καταγραφή των απαιτήσεων του αντίστοιχου συστήματός μας για διαδικτυακές κρατήσεις, το σχεδιασμό και την δημιουργία του συστήματος της βάσης δεδομένων και την υλοποίηση των κατάλληλων διεπαφών επικοινωνίας με το χρήστη για την κράτηση δωματίων σε ένα ξενοδοχειακό κατάλυμα.

1.2 Κίνητρο για τη διεξαγωγή της εργασίας

Στην σύγχρονη εποχή, όλες οι τουριστικές επιχειρήσεις στοχεύουν στην καλύτερη δυνατή ποιότητα υπηρεσιών και την προβολή τους με σύγχρονα μέσα. Ο τουριστικός τομέας είναι αδύνατον να λειτουργήσει πλέον χωρίς την απαραίτητη πληροφοριακή υποδομή, τόσο εσωτερικά στα τμήματα των επιμέρους τουριστικών καταλυμάτων, όσο και εξωτερικά για την ηλεκτρονική προβολή και διανομή των τουριστικών υπηρεσιών. Η χρήση των νέων τεχνολογιών πληροφορικής, οφείλεται σε πολύ μεγάλο βαθμό στον μεγάλο όγκο πληροφορίας του τομέα και στην πολυσύνθετη δομή του. Ο τουρισμός περιλαμβάνει τόσο επιχειρήσεις με διαφορετικές κατηγορίες και χωρητικότητες, όσο και μεγαλύτερους ξενοδοχειακούς ομίλους. Παράλληλα, τα καταλύματα βρίσκονται σε κάθε γεωγραφική περιοχή του πλανήτη, ενώ παράλληλα ο πελάτης / επισκέπτης βρίσκεται σε μεγάλη απόσταση από αυτά, δυσχεραίνοντας τον στην επισκόπησή τους.

Χρησιμοποιώντας τις πιο σύγχρονες μεθόδους και συστήματα τεχνολογίας της πληροφορικής, βελτιώνεται η απόδοση και παραγωγικότητα των επιχειρήσεων αυτών. Παράλληλα, η χρήση πληροφοριακών συστημάτων στον τουρισμό ενισχύει την αναβάθμιση των παρεχόμενων υπηρεσιών και την ανάπτυξη νέων επιχειρηματικών δραστηριοτήτων και ανταγωνιστικού πλεονάσματος του χώρου. Η χρήση του διαδικτύου ως μέσο διάχυσης πληροφορίας εξυπηρετεί αντίστοιχα στην προβολή των υπηρεσιών τουρισμού, ανεξαρτήτως γεωγραφικής τοποθεσίας και στην άμεση διάδραση και εξυπηρέτηση του ενδιαφερόμενου κοινού.

Τα τελευταία χρόνια παρουσιάζεται ολοένα και μεγαλύτερη άνθιση στις διαδικτυακές ολοκληρωμένες μηχανές αναζήτησης και κράτησης τουριστικών υπηρεσιών, από αεροπορικά εισιτήρια μέχρι και δωμάτια ξενοδοχείων, τα οποία είναι ικανά να εξυπηρετήσουν μεγάλο όγκο χρηστών παράλληλα, χειρίζονται ένα αρκετά μεγάλο όγκο πληροφορίας και διενεργούν υψηλές υπολογιστικές πράξεις σε μικρούς χρόνους. Ο τρόπος σχεδίασης, δημιουργίας και απρόσκοπτης λειτουργίας ενός τέτοιου συστήματος αποτελεί το βασικό κίνητρο της συγκεκριμένης εργασίας.

1.3 Σκοπός και στόχοι της εργασίας

Η παρούσα εργασία έχει ως σκοπό την ανάπτυξη μίας σύγχρονης εφαρμογής κρατήσεων ξενοδοχειακών δωματίων μέσω του διαδικτύου. Παράλληλα, τα δεδομένα του συστήματος διατίθενται μέσω ενός κανονικοποιημένου συστήματος βάσης δεδομένων MySQL, προκειμένου να εξασφαλίζεται η ακεραιότητα των δεδομένων. Θα αναλυθεί η ροή εργασίας για να αναπτυχθεί ένα τέτοιο σύστημα, καθώς επίσης και οι απαιτήσεις λειτουργικότητάς του.

1.4 Δομή της εργασίας

Στο πρώτο Κεφάλαιο παρουσιάζεται περιληπτικά το περιεχόμενο, το κίνητρο και οι στόχοι της εργασίας. Το δεύτερο Κεφάλαιο κάνει αναφορά στο τρόπο διαχείρισης πληροφορίας με την τεχνολογία βάσεων δεδομένων και τα πλεονεκτήματα χρήσης τους. Στο τρίτο Κεφάλαιο γίνεται μία επισκόπηση στον τρόπο ανάλυσης συστημάτων διαχείρισης βάσεων δεδομένων και στο σχεδιασμό αυτών. Το τέταρτο Κεφάλαιο αναλύει τα συστήματα ταξιδιωτικών κρατήσεων και τους μηχανισμούς διανομής τους, παρουσιάζει τα γνωστότερα συστήματα διαχείρισης και online κρατήσεων ξενοδοχειακών κλινών, καθώς και τον τρόπο με τον οποίο πραγματοποιείται μία διαδικτυακή κράτηση. Το πέμπτο Κεφάλαιο παρουσιάζει τις τεχνολογίες με τις οποίες υλοποιήθηκε το παρόν σύστημα κρατήσεων. Στο Κεφάλαιο 6, αναλύεται ο τρόπος υλοποίησης της βάσης δεδομένων και της εφαρμογής που υλοποιήθηκε. Το Κεφάλαιο 7 επεξηγεί τη χρήση του συστήματος από τους χρήστες και τους διαχειριστές της εφαρμογής. Το όγδοο Κεφάλαιο ολοκληρώνει την εργασία, παρουσιάζοντας τα συμπεράσματα που προκύπτουν από την συγκεκριμένη εργασία, καθώς και όποιες μελλοντικές επεκτάσεις του συστήματος.

2 Συστήματα Διαχείρισης Βάσεων Δεδομένων

2.1 Βάσεις Δεδομένων και Συστήματα Διαχείρισης

Μια βάση δεδομένων είναι κάτι παραπάνω από μια απλή συλλογή αποθηκευμένων στοιχείων. Μια **Βάση Δεδομένων (DataBase)** είναι ένας οργανωμένος τρόπος αποθήκευσης πληροφοριών και πρόσβασης στα δεδομένα αυτά με ποικίλους τρόπους, χρησιμοποιώντας ειδικά προγράμματα. Ένας άλλος ορισμός είναι ότι μια βάση δεδομένων είναι ένα ολοκληρωμένο σύστημα που αποτελείται από τα δεδομένα (data) και το κατάλληλο λογισμικό (software), τα οποία χρησιμοποιώντας το υλικό (hardware) βοηθούν στην ενημέρωση και πληροφόρηση των χρηστών (users) [1].

Οι στόχοι μίας βάσης δεδομένων είναι οι εξής:

- Ο περιορισμός της πολλαπλής αποθήκευσης των ίδιων στοιχείων (**redundancy**).
- Ο καταμερισμός (**sharing**) των ίδιων στοιχείων σ' όλους τους χρήστες.
- Η ομοιομορφία (**uniformity**) στον χειρισμό και την αναπαράσταση των δεδομένων.
- Η επιβολή κανόνων ασφαλείας (**security**).
- Η διατήρηση της ακεραιότητας (**integrity**) και της αξιοπιστίας (**reliability**) των δεδομένων.
- Η ανεξαρτησία των δεδομένων (**data independence**) και των προγραμμάτων από τον φυσικό τρόπο αποθήκευσης των δεδομένων.

Το **Σύστημα Διαχείρισης Βάσεων Δεδομένων (DBMS, DataBase Management System)** είναι ένα σύνολο από προγράμματα που επιτρέπουν τον χειρισμό των δεδομένων μιας ή περισσοτέρων βάσεων δεδομένων που ανήκουν στο ίδιο σύστημα. Το DBMS περιέχει κάποια εργαλεία γενικής χρήσης για την δημιουργία και το χειρισμό των δεδομένων, εφόσον τα δεδομένα που υπάρχουν στις βάσεις δεδομένων είναι :

- **Ολοκληρωμένα (Integrated)**, δηλαδή τα δεδομένα πρέπει να είναι αποθηκευμένα σε ομοιόμορφα οργανωμένα σύνολα αρχείων, χωρίς την επανάληψη ή τον πλεονασμό (redundancy) των ίδιων στοιχείων.
- **Καταμεριζόμενα (Shared)**, δηλαδή να μπορούν περισσότεροι του ενός χρήστες να βλέπουν και να μοιράζονται τα ίδια δεδομένα την ίδια χρονική στιγμή.

Στα νεότερα DBMS, όπως την Oracle και την Informix, μπορεί να υπάρξει άμεση άντληση πληροφορίας, απουσία ειδικευμένων προγραμματιστών. Τα δεδομένα ενός DBMS μπορούν να χρησιμοποιηθούν σε κάθε μορφής ερώτημα (query) για την λήψη της κατάλληλης πληροφορίας.

Οι χρήστες των εφαρμογών αντλούν τα στοιχεία που τους ενδιαφέρουν από τη βάση δεδομένων χωρίς να είναι σε θέση να γνωρίζουν με ποιο τρόπο είναι οργανωμένα τα δεδομένα σ' αυτήν. Το DBMS έχει τον ρόλο του μεσάζοντα ανάμεσα στον χρήστη και τη βάση δεδομένων και μόνο μέσω του DBMS μπορεί ο χρήστης να αντλήσει πληροφορίες από τη βάση δεδομένων. Ένα DBMS μπορεί να είναι εγκατεστημένο σ' έναν μόνο υπολογιστή ή και σ' ένα δίκτυο υπολογιστών και μπορεί να χρησιμοποιείται από έναν ή περισσότερους χρήστες.

Ένα **Σύστημα Βάσης Δεδομένων (ΣΒΔ) ή DBS (Data Base System)** αποτελείται από το υλικό, το λογισμικό, τη βάση δεδομένων και τους χρήστες. Είναι δηλαδή ένα σύστημα αποθήκευσης και αξιοποίησης των δεδομένων με τη βοήθεια ηλεκτρονικού υπολογιστή. Αναλυτικά:

- Το **υλικό (hardware)** αποτελείται από τους ηλεκτρονικούς υπολογιστές, τα περιφερειακά, τους σκληρούς δίσκους, κ.ά., όπου είναι αποθηκευμένα τα αρχεία της βάσης δεδομένων αλλά και τα προγράμματα που χρησιμοποιούνται για την επεξεργασία τους.
- Το **λογισμικό (software)** είναι τα προγράμματα που χρησιμοποιούνται για την επεξεργασία των δεδομένων (στοιχείων) της βάσης δεδομένων.
- Η **βάση δεδομένων (database)** αποτελείται από το σύνολο των αρχείων όπου είναι αποθηκευμένα τα δεδομένα του συστήματος. Τα στοιχεία αυτά μπορεί να βρίσκονται αποθηκευμένα σ' έναν φυσικό υπολογιστή αλλά και σε περισσότερους. Όμως, στον χρήστη δίνεται διαφανώς η εντύπωση ότι βρίσκονται συγκεντρωμένα στον ίδιο υπολογιστή. Ο κάθε χρήστης έχει διαφορετικά δικαιώματα και βλέπει διαφορετικό κομμάτι της βάσης δεδομένων, ανάλογα με τον σκοπό για τον οποίο συνδέεται.
- Οι **χρήστες (users)** μιας βάσης δεδομένων κατηγοριοποιούνται ως εξής:

- **Τελικοί χρήστες (end users)**. Χρησιμοποιούν κάποια εφαρμογή για τη λήψη πληροφορίας από μια βάση δεδομένων, έχουν τις λιγότερες δυνατότητες επέμβασης στα στοιχεία της βάσης δεδομένων, χρησιμοποιούν ειδικούς κωδικούς πρόσβασης και το σύστημα τους επιτρέπει ανάλογα πρόσβαση σε συγκεκριμένο κομμάτι της βάσης δεδομένων.
- **Προγραμματιστές εφαρμογών (application programmers)**. Αναπτύσσουν τις εφαρμογές του DBS σε κάποια από τις εφαρμόσιμες γλώσσες προγραμματισμού.
- **Διαχειριστής δεδομένων (data administrator – DA)**. Έχει τη διοικητική αρμοδιότητα και ευθύνη για την οργάνωση της βάσης δεδομένων και την απόδοση δικαιωμάτων πρόσβασης στους χρήστες.
- **Διαχειριστής βάσης δεδομένων (database administrator – DBA)**. Λαμβάνει οδηγίες από τον διαχειριστή δεδομένων και είναι αυτός που διαθέτει τις τεχνικές γνώσεις και αρμοδιότητες για τη σωστή και αποδοτική λειτουργία του DBMS. Οι αρμοδιότητές του είναι οι εξής : (α) Η απόφαση για το είδος των πληροφοριών που πρέπει να αποθηκευτούν. (β) Η απόφαση για τον τρόπο αποθήκευσης και πρόσβασης στις πληροφορίες αυτές. (γ) Η συνεργασία με τους τελικούς χρήστες. (δ) Η απόφαση για τον τρόπο εξασφάλισης των πληροφοριών. (ε) Η απόφαση για το κάθε πότε θα γίνονται αντίγραφα ασφαλείας (backup) των αρχείων. (στ) Η παρακολούθηση της σωστής λειτουργίας της βάσης δεδομένων και η απαιτούμενη προσαρμογή της.

2.2 Πλεονεκτήματα χρήσης

Τα πλεονεκτήματα ενός συστήματος βάσης δεδομένων, σε σύγκριση με τις παραδοσιακές μεθόδους παρακολούθησης, είναι αρκετά [2]. Ξεκινώντας από μικρή βάση δεδομένων, προκύπτουν τα εξής:

- **Οικονομία χώρου**, καταργώντας τους παραδοσιακά ογκώδεις φακέλους με έγγραφα.
- **Ταχύτητα** (το σύστημα μπορεί να ανακαλεί και να αλλάζει τα δεδομένα γρηγορότερα από τον άνθρωπο)
- **Λιγότερος κόπος**
- **Άμεση πληροφόρηση** (Ακριβείς και ενημερωμένες πληροφορίες διαθέσιμες κάθε στιγμή).

Επιπλέον, σε ένα περιβάλλον πολλών χρηστών, το σύστημα βάσης δεδομένων παρέχει στην επιχείρηση κεντρικό έλεγχο των δεδομένων της ώστε να προκύπτουν τα εξής πλεονεκτήματα:

Ο πλεονασμός μειώνεται στο ελάχιστο. Στα συμβατικά συστήματα (εκείνα που δεν είναι συστήματα βάσεων δεδομένων), η κάθε εφαρμογή έχει τα δικά της αρχεία. Αυτό το γεγονός οδηγεί πολύ συχνά σε υψηλό βαθμό πλεονασμού (επανάληψης) για τα αποθηκευμένα δεδομένα, με αποτέλεσμα τη σπατάλη αποθηκευτικού χώρου. Θα πρέπει εδώ να ξεκαθαρίσουμε ότι αυτό δε σημαίνει πως είναι πάντα δυνατό να εξαλειφθούν όλοι οι πλεονασμοί, ούτε πως είναι πάντα επιθυμητό. Μερικές φορές υπάρχουν σοβαροί επιχειρηματικοί ή τεχνικοί λόγοι που επιβάλλουν να τηρούνται ξεχωριστά αντίγραφα των ίδιων αποθηκευμένων δεδομένων.

Η ασυνέπεια μπορεί να αποφευχθεί ως ένα βαθμό. Το DBMS θα μπορεί να εγγυηθεί ότι η βάση δεδομένων δε θα είναι ποτέ ασυνεπής στα μάτια του χρηστή, εξασφαλίζοντας ότι κάθε αλλαγή που θα γίνεται σε οποιαδήποτε από δυο όμοιες καταχωρίσεις θα γίνεται αυτόματα και στην άλλη. Αυτή η διαδικασία είναι γνωστή με το όνομα *διάδοση ενημερώσεων*.

Χρήση κοινόχρηστων δεδομένων. Ο μερισμός δε σημαίνει μόνο ότι οι υπάρχουσες εφαρμογές μπορούν να μοιράζονται τα δεδομένα της βάσης δεδομένων αλλά και ότι είναι δυνατή η ανάπτυξη νέων εφαρμογών που θα μπορούν να χρησιμοποιούν τα ίδια αποθηκευμένα δεδομένα.

Επιβολή προτύπων. Η τυποποίηση της αναπαράστασης των δεδομένων διευκολύνει ιδιαίτερα την ανταλλαγή δεδομένων. Τα πρότυπα ονομασίας και τεκμηρίωσης των δεδομένων είναι επίσης πολύ επιθυμητά για να διευκολύνεται ο μερισμός και η καλύτερη κατανόηση των δεδομένων.

Εφαρμογή περιορισμών ασφαλείας. Έχοντας πλήρη δικαιοδοσία πάνω στη βάση δεδομένων, ο DBA (α) μπορεί να εξασφαλίσει ότι η πρόσβαση στη βάση δεδομένων θα μπορεί να γίνεται μόνο μέσω των κατάλληλων καναλιών και, κατά συνέπεια, (β) μπορεί να ορίσει κανόνες ασφαλείας με βάση τους οποίους θα γίνεται έλεγχος κάθε φορά που θα υπάρχει απόπειρα προσπέλασης εμπιστευτικών δεδομένων. Βέβαια τα συστήματα βάσης δεδομένων απαιτούν την ύπαρξη ενός καλού συστήματος ασφαλείας.

Διατήρηση ακεραιότητας. Το πρόβλημα της ακεραιότητας έγκειται στην εξασφάλιση της ακρίβειας των δεδομένων στη βάση. Η ασυμφωνία μεταξύ δυο καταχωρίσεων που υποτίθεται ότι αντιπροσωπεύουν το ίδιο «γεγονός» είναι ένα παράδειγμα έλλειψης ακεραιότητας. Φυσικά, αυτό το συγκεκριμένο πρόβλημα μπορεί να παρουσιαστεί μόνο αν υπάρχει πλεονασμός στα αποθηκευμένα δεδομένα. Ακόμη και αν δεν υπάρχει πλεονασμός όμως, πάλι υπάρχει περίπτωση να περιέχει η βάση δεδομένων λανθασμένες πληροφορίες.

Εξισορρόπηση αντικρουόμενων απαιτήσεων. Γνωρίζοντας τις συνολικές απαιτήσεις της επιχείρησης, σε αντιδιαστολή με τις απαιτήσεις των μεμονωμένων χρηστών, ο DBA (πάντα με τις οδηγίες του υπεύθυνου διαχείρισης δεδομένων) μπορεί να δομήσει το σύστημα με τέτοιο τρόπο ώστε να παρέχει γενικές υπηρεσίες που να είναι «βέλτιστες για την επιχείρηση». Για παράδειγμα μπορεί να επιλεγεί μια αναπαράσταση των αποθηκευμένων δεδομένων που να παρέχει γρήγορη πρόσβαση στις σημαντικές εφαρμογές ίσως σε βάρος της απόστασης άλλων εφαρμογών.

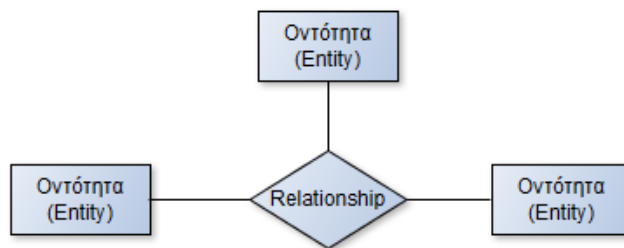
3 Ανάλυση απαιτήσεων για DBMS

3.1 Βασικές έννοιες ανάλυσης συστήματος

3.1.1 Οντότητα

Ο όρος **οντότητα (entity)** αντιστοιχεί σε κάθε αντικείμενο, έννοια, πρόσωπο ή κατάσταση με ανεξάρτητη ύπαρξη. Πρόκειται για οποιοδήποτε ανεξάρτητο αντικείμενο για το οποίο μπορεί να συγκεντρωθεί πληροφορία [1]. Η οντότητα είναι αντίστοιχη με την έννοια της εγγραφής στα αρχεία και τους πίνακες αλλά και με την έννοια του αντικειμένου στις σύγχρονες αντικειμενοστραφείς γλώσσες προγραμματισμού.

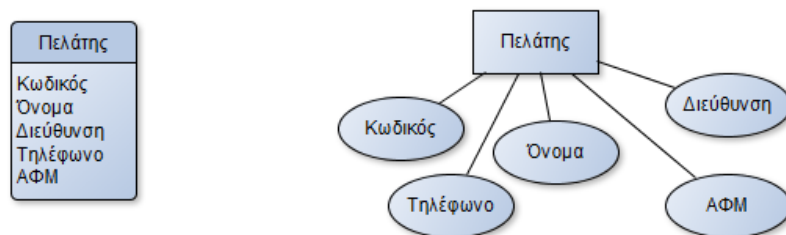
Μια βάση δεδομένων μπορεί να περιέχει πολλές διαφορετικές οντότητες, οι οποίες απεικονίζονται με ορθογώνια παραλληλόγραμμα και συσχετίζονται μεταξύ τους ανά δύο.



Εικόνα 1 – Σχεδίαση οντότητας

3.1.2 Ιδιότητα

Ο όρος **ιδιότητα (attribute)** ή πεδίο μιας οντότητας, αναφέρεται στα στοιχεία που περιγράφουν την οντότητα και την κάνουν να ξεχωρίζει από τα άλλα στοιχεία της ίδιας οντότητας [1]. Για παράδειγμα, η οντότητα «Πελάτης» μπορεί να έχει ως ιδιότητες τον *κωδικό πελάτη*, το *επώνυμο*, το *όνομα*, τη *διεύθυνση*, το *τηλέφωνο*, το *ΑΦΜ* κ.ά.. Οι ιδιότητες βοηθούν στον διαχωρισμό της μίας οντότητας από μία άλλη. Κάθε τύπος οντοτήτων περιγράφεται από ένα όνομα και από το σύνολο των ιδιοτήτων που περιέχει. Οι πληροφορίες αυτές αποτελούν το **σχήμα (schema)** της οντότητας. Οι ιδιότητες μίας οντότητας απεικονίζονται είτε σε μορφή πεδίων πίνακα εντός της οντότητας, είτε ως μεμονωμένες ελλείψεις, συνδεδεμένες με το ορθογώνιο παραλληλόγραμμο της οντότητας, όπως παρουσιάζεται στο παρακάτω σχήμα.



Εικόνα 2 – Σχεδίαση ιδιοτήτων οντότητας

3.1.3 Συσχέτιση

Μια **συσχέτιση (relationship)** συνδέει δύο ή και περισσότερες οντότητες μεταξύ τους και παριστάνεται με έναν ρόμβο μεταξύ των οντοτήτων στις οποίες αναφέρεται, συνδεδεμένος με αυτές [1]. Οι συσχετίσεις είναι απαραίτητες για την άντληση σχετικών πληροφοριών που αφορούν δύο ή και περισσότερες οντότητες, όπως για παράδειγμα ποιες παραγγελίες έγιναν από ποιον πελάτη (η οντότητα πελάτης ΠΑΡΑΓΓΕΛΝΕΙ μία ή περισσότερες παραγγελίες).

Όταν οι οντότητες που συμμετέχουν σε μια συσχέτιση είναι δύο, η συσχέτιση αποκαλείται **διμελής ή δυαδική**. Ο βαθμός μιας συσχέτισης είναι ίσος με το πλήθος των οντοτήτων που συμμετέχουν

σ' αυτήν. Μια συσχέτιση μπορεί και η ίδια να έχει ιδιότητες που να περιγράφουν ορισμένα χαρακτηριστικά της, όπως για παράδειγμα η συσχέτιση ΠΑΡΑΓΓΕΛΙΑ ανάμεσα στις οντότητες πελάτης και προϊόν μπορεί να περιέχει τις ιδιότητες Κωδικός Πελάτη, Κωδικός Προϊόντος, Κωδικός Παραγγελίας, Ποσότητα, κ.ά. Στην περίπτωση αυτή θα ήταν συνετό η δημιουργία μίας ακόμα οντότητας, ως Παραγγελία, η οποία θα περιέχει όλες τις παραπάνω ιδιότητες, και την μετονομασία της προηγούμενη συσχέτισης από ΠΑΡΑΓΓΕΛΙΑ σε ΣΥΝΑΜΜΑΓΗ, χωρίς ιδιότητες. Έτσι, η παραπάνω συσχέτιση θα μετατραπεί από διμελή σε **τριμελή**.

Ο σωστός σχεδιασμός μιας βάσης δεδομένων, απαιτεί την εκχώρηση ιδιοτήτων μόνο στις οντότητες και την δημιουργία συσχετίσεων ως λογικές συνδέσεις ανάμεσα στις οντότητες.

3.1.4 Οι διμελείς συσχετίσεις

Υπάρχουν τρία βασικά ήδη συνδέσεων στις διμελείς συσχετίσεις μεταξύ οντοτήτων [1], τα οποία διαμορφώνονται ως εξής :

- **Ένα-προς-ένα (1:1)**, όπου μια οντότητα συνδέεται **με μία και μόνο μία** άλλη οντότητα. Για παράδειγμα, η οντότητα νομός έχει μία μόνο πόλη σαν πρωτεύουσα, ενώ η οντότητα πρωτεύουσα αντιστοιχεί σ' έναν και μόνο έναν νομό. Στην περίπτωση των διμελών συσχετίσεων του τύπου ένα-προς-ένα, μπορούμε να ενώσουμε τα στοιχεία και των δύο ιδιοτήτων και να δημιουργήσουμε μια μοναδική οντότητα (πίνακα).
- **Ένα-προς-πολλά (1:M)**, όπου το στιγμιότυπο μιας οντότητας συνδέεται **με πολλά** στιγμιότυπα της άλλης οντότητας αλλά κάθε στιγμιότυπο της δεύτερης οντότητας συνδέεται **με ένα και μόνο** ένα στιγμιότυπο της πρώτης οντότητας. Για παράδειγμα, ένας πελάτης κάνει πολλές παραγγελίες, αλλά μια παραγγελία αντιστοιχεί σ' έναν και μόνο έναν πελάτη. Οι διμελείς συσχετίσεις του τύπου 1:N είναι οι πιο συχνά χρησιμοποιούμενες και οι πιο βολικές στη διαχείριση.
- **Πολλά-προς-πολλά (M:N)**, όπου σε κάθε στιγμιότυπο της μιας οντότητας αντιστοιχούν **πολλά** στιγμιότυπα της άλλης οντότητας και σε κάθε στιγμιότυπο της δεύτερης οντότητας αντιστοιχούν **πολλά** στιγμιότυπα της πρώτης οντότητας. Για παράδειγμα, ένας καθηγητής διδάσκει πολλούς μαθητές αλλά και ένας μαθητής διδάσκεται από πολλούς καθηγητές. Για την διαχείριση της συσχέτισης τύπου M:N, απαιτείται η δημιουργία ενός τρίτου πίνακα που θα περιέχει δύο μόνο ιδιότητες (πεδία), τα πεδία κλειδιά των δύο οντοτήτων, οπότε ο συνδυασμός τους θα είναι και το πεδίο κλειδί (σύνθετο κλειδί) του νέου πίνακα.

3.1.5 Κλειδιά

Ο όρος **κλειδί (key)** ή πιο σωστά **πρωτεύον κλειδί (primary key)** αναφέρεται σε μια ιδιότητα, ή σε ορισμένες περιπτώσεις ένα σύνολο ιδιοτήτων, η τιμή της οποίας είναι **μοναδική** σε ολόκληρη την οντότητα [1]. Πρακτικά, το πρωτεύον κλειδί έχει διαφορετική τιμή για κάθε στιγμιότυπο της οντότητας (ή για κάθε εγγραφή του πίνακα) και δεν μπορεί να έχει null τιμή. Ο συνδυασμός δύο ή και περισσότερων ιδιοτήτων για τη δημιουργία ενός πρωτεύοντος κλειδιού αποκαλείται **σύνθετο κλειδί**.

Ξένο κλειδί αποκαλείται μια ιδιότητα που είναι πρωτεύον κλειδί σε μια οντότητα αλλά που υπάρχει και σε μια άλλη οντότητα σαν απλή ιδιότητα. Τα ξένα κλειδιά είναι απαραίτητα για την δημιουργία των συσχετίσεων ανάμεσα στις οντότητες.

3.2 Μοντέλο Οντοτήτων – Συσχετίσεων

Το μοντέλο που έχει επικρατήσει σήμερα για την αναπαράσταση της δομής μιας βάσης δεδομένων είναι το **Μοντέλο Οντοτήτων-Συσχετίσεων (ΟΣ, Entity-Relationship Diagram, ER)** [1]. Οι βασικές έννοιες του μοντέλου αυτού είναι οι εξής :

- Οντότητες (Entities)
- Ιδιότητες ή Χαρακτηριστικά (Attributes)
- Συσχετίσεις (Relationships)

Για την αναπαράσταση ενός Μοντέλου Οντοτήτων – Συσχετίσεων χρησιμοποιούνται ειδικά διαγράμματα, όπου τα ορθογώνια συμβολίζουν τις οντότητες, οι ρόμβοι τις συσχετίσεις και οι ελλείψεις τις ιδιότητες, όπως αναλύθηκαν παραπάνω. Οι οντότητες συνδέονται μεταξύ τους μέσω των συσχετίσεών τους με γραμμές. Όλα τα παραπάνω αποτελούν τη λογική δομή μιας βάσης δεδομένων, μια εργασία που

είναι απαραίτητο να γίνει πριν από την καταχώριση και την επεξεργασία των στοιχείων (πληροφοριών) της βάσης δεδομένων. Το μοντέλο οντοτήτων-συσχετίσεων αποτελεί μια γενική περιγραφή των στοιχείων που θα απαρτίζουν μια βάση δεδομένων και απεικονίζει την εννοιολογική αντίληψη των δεδομένων, χωρίς τις ιδιαίτερες λεπτομέρειες υλοποίησης. Για τη διαμόρφωση του διαγράμματος οντοτήτων-συσχετίσεων, ακολουθούνται τα εξής βήματα :

- 1) Ορισμός των οντοτήτων (πινάκων) που θα ανήκουν στη βάση δεδομένων και θα δημιουργηθούν.
- 2) Ορισμός των ιδιοτήτων (πεδία) και τα πρωτεύοντα κλειδιά της κάθε οντότητας (πίνακα).
- 3) Καταγραφή και σχεδιασμός των συσχετίσεων ανάμεσα στις οντότητες.
- 4) Δημιουργία του διαγράμματος οντοτήτων-συσχετίσεων, όπου θα απεικονίζονται οι οντότητες, οι ιδιότητές τους και οι μεταξύ τους συσχετίσεις.

3.3 Κανονικοποίηση

Η μέθοδος της κανονικοποίησης βοηθάει στον λογικό σχεδιασμό μιας βάσης δεδομένων και είναι συμπληρωματική του μοντέλου οντοτήτων-συσχετίσεων. Με την κανονικοποίηση εξασφαλίζεται η λύση σε προβλήματα συνέπειας, πλεονασμού και εγκυρότητας των πληροφοριών της βάσης δεδομένων [3]. Η διαδικασία της κανονικοποίησης οδηγεί σε συνεχείς διασπάσεις των πινάκων σε πιο απλές και συμπαγείς μορφές, με κύριο στόχο την αποφυγή του πλεονασμού (επανάληψης) των δεδομένων, αφαιρώντας πεδία από τις αρχικές μεγάλες σχέσεις και τοποθετώντας τα σε νέες σχέσεις με τις ίδιες πληροφορίες. Οι βασικές μορφές της κανονικοποίησης είναι τρεις, η πρώτη (1η NF), η δεύτερη (2η NF) και η τρίτη (3η NF), και αναλύονται στις επόμενες ενότητες.

3.3.1 Πρώτη Κανονική Μορφή (1NF)

Η Πρώτη Κανονική Μορφή (1NF) θεωρείται γενικώς να είναι μέρος του ορισμού μιας σχέσης. Μία σχέση βρίσκεται σε πρώτη κανονική μορφή αν ικανοποιεί όλες τις ιδιότητες του ορισμού της σχέσης:

- 1) Κάθε όνομα γνωρίσματος (στήλης) είναι **μοναδικό**.
- 2) Όλες οι τιμές κάθε γνωρίσματος είναι **ίδιου τύπου**.
- 3) Κάθε τιμή γνωρίσματος είναι **ατομική** (μία τιμή και όχι πλειάδα τιμών).
- 4) Τα γνωρίσματα δεν έχουν διάταξη από τα αριστερά προς τα δεξιά..
- 5) Οι συστοιχίες (σειρές) δεν έχουν διάταξη από επάνω προς τα κάτω.
- 6) Δεν υπάρχουν δύο ίδιες σειρές (συστοιχίες) σε μία σχέση.

Εάν υπάρχει **καθορισμένο κλειδί** για τη σχέση τότε ικανοποιείται η απαίτηση της μοναδικότητας των συστοιχιών (σειρών). Με άλλα λόγια κάθε στήλη πρέπει να περιέχει μόνο μία τιμή σε κάθε της κελί. Μια γενικότερη συμβουλή για να καταλάβει κάποιος πότε ένας πίνακας δεν βρίσκεται σε πρώτη κανονική μορφή, θα πρέπει να προσέξει ώστε να μην υπάρχουν επαναλαμβανόμενα πεδία.

Τα βήματα που πρέπει να ακολουθηθούν ώστε να επέλθει ένας πίνακας με επαναλαμβανόμενα πεδία σε 1KM, είναι το σπάσιμο του αρχικού πίνακα σε δύο επί μέρους πίνακες. Ο ένας θα περιλαμβάνει τα βασικά στοιχεία του πίνακα και ο δεύτερος τα ονόματα των πεδίων που επαναλαμβάνονται μαζί με το πρωτεύον κλειδί του πρώτου πίνακα.

3.3.2 Δεύτερη Κανονική Μορφή (2NF)

Μία σχέση βρίσκεται σε Δεύτερη Κανονική Μορφή (2NF) εάν κάθε ένα από τα γνωρίσματά της που δεν είναι κλειδιά προσδιορίζονται από ολόκληρο το πρωτεύον κλειδί και όχι μόνο από ένα τμήμα του. Οι σχέσεις που έχουν μόνο ένα γνώρισμα σαν πρωτεύον κλειδί βρίσκονται αυτόματα και στη δεύτερη κανονική μορφή. Η λύση που ακολουθείται όπως και σε όλες τις περιπτώσεις που ένας πίνακας δεν είναι κανονικοποιημένος, είναι το σπάσιμο του πίνακα σε επιμέρους πίνακες.

- 1) Στον έναν πίνακα περιλαμβάνονται τα πεδία του κλειδιού και τα πεδία που εξαρτώνται και περιγράφουν όλο το σύνθετο κλειδί.
- 2) Στον άλλο πίνακα εισάγονται τα πεδία που εξαρτώνται και περιγράφουν μέρος από το σύνθετο κλειδί καθώς και το μέρος του σύνθετου κλειδιού.

3.3.3 Τρίτη Κανονική Μορφή (3NF)

Ένας πίνακας βρίσκεται σε Τρίτη Κανονική Μορφή (3NF) όταν είναι ήδη στην πρώτη 1NF και δεύτερη 2NF κανονική μορφή και κάθε στήλη που δεν είναι πρωτεύον κλειδί να είναι ανεξάρτητη από άλλη στήλη που δεν είναι επίσης πρωτεύον κλειδί.

3.3.4 Boyce-Codd κανονική μορφή (BCNF)

Η Boyce-Codd Κανονική Μορφή (BCNF) είναι μια πιο περιοριστική μορφή της 3NF, με την έννοια ότι κάθε σχέση στην BCNF είναι επίσης στην 3NF, αλλά δεν ισχύει πάντα το αντίστροφο. Η 3NF δεν αντιμετωπίζει τις παρακάτω περιπτώσεις:

- Μία σχέση να έχει δύο ή περισσότερα υποψήφια κλειδιά.
- Τα δύο υποψήφια κλειδιά να είναι σύνθετα.
- Να επικαλύπτονται (να έχουν τουλάχιστον ένα γνώρισμα κοινό)

Μία σχέση βρίσκεται σε κανονική μορφή BCNF εάν κάθε προσδιοριστικό της σχέσης είναι ένα υποψήφιο κλειδί. Δηλαδή οι μόνες συσχετίσεις που εμφανίζονται στο διάγραμμα οντοτήτων-συσχετίσεων είναι βέλη που ξεκινούν από υποψήφια κλειδιά. Στην πραγματικότητα, οι περισσότερες σχέσεις οι οποίες είναι στην 3NF είναι επίσης στην BCNF. Οποιαδήποτε σχέση που έχει μόνο δύο γνωρίσματα είναι σε BCNF.

Τα βήματα που ακολουθούνται ώστε να επιτευχθεί κανονικοποίηση ως προς BCNF είναι τα ακόλουθα:

- 1) Δημιουργία μίας λίστας με όλα τα γνωρίσματα.
- 2) Έλεγχος εάν κάθε προσδιοριστικό γνώρισμα μπορεί να είναι υποψήφιο κλειδί.
- 3) Για τα προσδιοριστικά γνωρίσματα που δεν είναι κλειδιά δημιουργείται μία σχέση για την κάθε συναρτησιακή εξάρτηση.
- 4) Γίνεται συσχέτιση των γνωρισμάτων που δεν είναι κλειδιά με την αρχική σχέση.

3.3.5 Ανώτερες Κανονικές Μορφές

3.3.5.1 Τέταρτη Κανονική Μορφή (4NF)

Μία σχέση είναι σε Τέταρτη Κανονική Μορφή (4NF) εάν είναι σε BCNF και δεν περιέχει εξαρτήσεις πολλαπλών τιμών (multivalued dependencies). Με άλλα λόγια μια σχέση είναι σε 4NF εάν και μόνο εάν οι εξαρτήσεις πολλαπλών τιμών που ικανοποιεί είναι στην πραγματικότητα συναρτησιακές εξαρτήσεις που ξεκινούν από υποψήφια κλειδιά.

3.3.5.2 Πέμπτη Κανονική Μορφή (5NF)

Μία σχέση είναι σε Πέμπτη Κανονική Μορφή (5NF) εάν και μόνο εάν οι μόνες εξαρτήσεις σύζευξης που ικανοποιεί είναι στην πραγματικότητα συναρτησιακές εξαρτήσεις που ξεκινούν από υποψήφια κλειδιά. Επίσης η πέμπτη κανονική μορφή απομακρύνει εξαρτήσεις σύζευξης.

3.4 Σχεδιασμός βάσης δεδομένων

Όπως είναι εύκολα κατανοητό, η βασικότερη εργασία κατά τον σχεδιασμό μιας σχεσιακής βάσης δεδομένων είναι ο ορισμός των πινάκων που θα χρησιμοποιηθούν καθώς και τα πεδία που θα περιέχει ο καθένας απ' αυτούς. Η διαδικασία αυτή αποκαλείται **κατασκευή του σχήματος (schema)** μιας βάσης δεδομένων [1]. Τα βασικά βήματα για τον σχεδιασμό και την ανάπτυξη μιας βάσης δεδομένων είναι:

- 1) Η ανάλυση των απαιτήσεων
- 2) Ο εννοιολογικός σχεδιασμός και η καταγραφή των οντοτήτων, ιδιοτήτων και των συσχετίσεών τους.
- 3) Η δημιουργία του διαγράμματος Οντοτήτων-Συσχετίσεων
- 4) Η μετατροπή του διαγράμματος Ο-Σ στο σχεσιακό σχήμα της βάσης δεδομένων.
- 5) Η υλοποίηση της βάσης δεδομένων, βάσει του καταγεγραμμένου σχεσιακού σχήματος.

Οι κανόνες που πρέπει να ακολουθηθούν ρητά κατά την υλοποίηση και το σχεδιασμό μιας σχεσιακής βάσης δεδομένων είναι οι εξής :

- Η κάθε **οντότητα** πρέπει να παριστάνεται ως ένας ξεχωριστός **πίνακας**.
- Η κάθε **στήλη** του πίνακα αντιστοιχεί σε μια **ιδιότητα** της οντότητας.

- Η κάθε **γραμμή** του πίνακα αντιστοιχεί σε ένα **στιγμιότυπο** της οντότητας.
- Η κάθε **γραμμή** πρέπει να είναι **μοναδική**, δηλ. αποκλείεται να υπάρχουν δύο ή και περισσότερες γραμμές που να περιέχουν τα ίδια ακριβώς στοιχεία.
- Η σειρά εμφάνισης των γραμμών δεν έχει καμία σημασία.
- Η κάθε **στήλη** έχει μια δική της **μοναδική ονομασία**.
- Οι **τιμές** που ανήκουν στην ίδια στήλη πρέπει να είναι του **ιδίου τύπου**, δηλ. ή όλες αριθμοί ή όλες αλφαριθμητικές κοκ.
- Το **πρωτεύον κλειδί (primary key)** μιας οντότητας, δεν πρέπει να είναι **ποτέ null**.
- Απαγορεύεται η επανάληψη της ίδιας τιμής στο πρωτεύον κλειδί.
- Το πρωτεύον κλειδί μιας οντότητας αποκαλείται ξένο κλειδί (foreign key) σε μια άλλη οντότητα, με την οποία υπάρχει συσχετισμός.
- Μπορεί να υπάρχουν πολλές γραμμές που να έχουν την ίδια τιμή στο ξένο κλειδί.

4 Ταξιδιωτικά Συστήματα Κρατήσεων

Η χρήση ταξιδιωτικών συστημάτων κρατήσεων στον τουριστικό τομέα αυξάνει ποσοτικά και ποιοτικά τις υπηρεσίες προς τον τελικό πελάτη, ενώ παράλληλα οι τουριστικές επιχειρήσεις μπορούν να εξάγουν ακριβείς μετρήσεις για τα προϊόντα και τις παρεχόμενες υπηρεσίες τους [4]. Ο τουριστικός τομέας θα μπορούσε να κατηγοριοποιηθεί ως εξής:

- Προμηθευτές του τουριστικού προϊόντος, π.χ. ξενοδοχεία, αεροπορικές εταιρίες, εταιρίες ενοικίασεως αυτοκινήτων κ.ά.
- Μεσάζοντες υπηρεσιών, π.χ. τουριστικά γραφεία, παγκόσμια συστήματα διανομής (GDS- Global Distribution Systems), κ.ά.
- Τελικοί καταναλωτές των υπηρεσιών, όπως οι τουρίστες, οι συνεδριακοί πελάτες κ.λπ.

4.1 Παγκόσμια συστήματα διανομής

Στις αρχές του 1960 υλοποιούνται τα πρώτα **μηχανογραφημένα συστήματα κρατήσεων (Computer Reservation Systems – CRS)**, ως διαχειριστικά συστήματα πληροφόρησης με τη χρήση βάσεων δεδομένων. Η μηχανογράφηση της διαδικασίας κρατήσεων συμβάλει σε αξιοσημείωτη μείωση του χρόνου εξυπηρέτησης των πελατών καθώς και την εξεύρεση εναλλακτικών λύσεων σε προβλήματα μετακινήσεων [4].

Στις αρχές της δεκαετίας του 1970 τα CRS εξελίσσονται σε συστήματα διανομής και ανταλλαγής πληροφοριών και κρατήσεων μεταξύ των αεροπορικών εταιριών και των τουριστικών πρακτόρων. Τα τουριστικά γραφεία έχουν πλέον την δυνατότητα να συνδεθούν με συγκεκριμένες αεροπορικές εταιρίες για να διαμορφώσουν μία πλήρη κράτηση σύμφωνα με τις απαιτήσεις του πελάτη. Στα μέσα της δεκαετίας του 1980, εξελιγμένα αμερικανικά συστήματα κρατήσεων CRS, επεκτείνονται στην ευρωπαϊκή αγορά, με αρκετά προβλήματα συνεργασίας με όλες τις ευρωπαϊκές αεροπορικές εταιρίες. Έτσι, γίνεται αναγκαία η δημιουργία ενός **Παγκόσμιου Συστήματος Διανομής (Global Distribution System – GDS)**, συνδέοντας όλα τα ευρωπαϊκά ταξιδιωτικά πρακτορεία με τα παγκόσμια ταξιδιωτικά γραφεία των αεροπορικών εταιριών, καθώς και με τους υπόλοιπους τουριστικούς κλάδους. Τα *Global Distribution Systems (GDS)* αποτελούν τον «μεσάζοντα» μεταξύ του ταξιδιωτικού πράκτορα και των επιχειρήσεων που παρέχουν τουριστικές υπηρεσίες (αεροπορικές εταιρίες, ξενοδοχειακές επιχειρήσεις, εταιρίες ενοικίασης αυτοκινήτων κ.λπ.). Αποτελούν το σημαντικότερο κανάλι διανομής των τουριστικών προϊόντων διεθνώς, αφού ελαχιστοποιούν το χρόνο εξυπηρέτησης και διασφαλίζουν την κράτηση σε ελάχιστο χρόνο.

Το 1987 εμφανίζονται τα δυο πρώτα ευρωπαϊκά συστήματα διανομής, το **Amadeus** και το **Galileo** που συνδέονται με τα εταιρικά και τα ταξιδιωτικά πρακτορεία, διευκολύνοντας την επικοινωνία με τις αεροπορικές εταιρίες, η οποία πραγματοποιείται πλέον μέσω δικτύου. Οι νέες βάσεις δεδομένων που υλοποιούνται από τα συστήματα αυτά παρέχουν επαυξημένη πληροφόρηση στους ταξιδιώτες, ενώ ταυτόχρονα επιτρέπουν στους υπαλλήλους των υπηρεσιών αυτών να ικανοποιούν τις συνθήκες κράτησης του εκάστοτε πελάτη.

Σε παγκόσμιο επίπεδο, τα κυριότερα συστήματα GDS, είναι τα ακόλουθα:

4.1.1 Sabre [5]

Θεωρείται ένα από τα πιο ευέλικτα συστήματα της αγοράς, με πρόσβαση σε περισσότερες από 370 αεροπορικές εταιρίες και 59 εταιρίες ενοικίασης αυτοκινήτων και 31.000 ξενοδοχεία για online κρατήσεις, ενώ συνεργάζεται με τις ναυτιλιακές εταιρίες ΔΑΝΕ και ΕΡΜΕΣ. Διαθέτει online εφαρμογή, ως *Travelocity*, μέσω της οποίας μπορούν να γίνουν κρατήσεις στην καθορισμένη τιμή της αεροπορικής εταιρίας.

4.1.2 Worldspan [6]

Συnergάζεται με περισσότερες από 360 αεροπορικές εταιρίες, 40 εταιρίες ενοικίασης αυτοκινήτων, 16.000 τουριστικά γραφεία και περίπου 22.000 ξενοδοχεία. Επιπλέον παρέχει την δυνατότητα για πληροφορίες κρατήσεων ευρωπαϊκών σιδηροδρομικών εταιριών και ακτοπλοϊκές εταιρίες, ενώ παρέχει πληροφορίες σε εταιρίες που ασχολούνται με e-business κ.λπ..

4.1.3 Galileo [7]

Πρόκειται για ένα προηγμένο σύστημα κρατήσεων στο οποίο είναι συνδεδεμένες περισσότερες από 511 αεροπορικές εταιρίες, 39 εταιρίες ενοικίασης αυτοκινήτου, 46.046 ξενοδοχειακές μονάδες, 368 Tour Operators καθώς και όλες οι κύριες γραμμές κρουαζιέρων σε παγκόσμιο επίπεδο. Επίσης το συγκεκριμένο σύστημα κρατήσεων, παρέχει σε περισσότερους από 41.200 ταξιδιωτικούς πράκτορες τη δυνατότητα πρόσβασης σε πληροφορίες που αφορούν αεροπορικά αλλά και άλλα είδους εισιτήρια, τη δυνατότητα πραγματοποίησης πολλαπλών κρατήσεων καθώς και την έκδοση αεροπορικών εισιτηρίων σε 116 χώρες. Αποκλειστικός διανομέας της Galileo International στην Ελλάδα είναι η Galileo Hellas (θυγατρική εταιρία της Ολυμπιακής Αεροπορίας). Η δημιουργία του συγκεκριμένου συστήματος συμβάλει στην ανάπτυξη και μεγιστοποίηση της απόδοσης των τουριστικών επιχειρήσεων, καθώς επίσης και στην αναβάθμιση της ποιότητας παροχής του τουριστικού προϊόντος στην Ελλάδα.

4.1.4 Amadeus [8]

Το συγκεκριμένο GDS, εξυπηρετεί τις ανάγκες του marketing, της διανομής καθώς και των πωλήσεων των προϊόντων της τουριστικής βιομηχανίας. Παρέχει πληροφορίες για περισσότερες από 400 αεροπορικές εταιρίες, 63.000 ξενοδοχεία, δεκάδες εταιρίες ενοικίασης αυτοκινήτων και σιδηροδρομικούς οργανισμούς, ενώ το χρησιμοποιούν περισσότερα από τα 2/3 όλων των ταξιδιωτικών πρακτορείων στην Ευρώπη. Ο συνολικός αριθμός των σημείων πώλησης υπολογίζεται στις 290.000, σε περισσότερες από 210 αγορές παγκοσμίως. Επίσης, συνεργάζεται με το Worldspan και με πολλές ακτοπλοϊκές εταιρίες.

Στην ελληνική τουριστική αγορά, κυρίαρχη θέση στα συστήματα GDS κατέχουν το *Galileo*, το *Sabre*, το *Amadeus* και το *Worldspan*, μέσα από τα οποία ο πελάτης έχει τη δυνατότητα να πραγματοποιήσει την οποιαδήποτε κράτηση ανάλογα με την επιλογή του.

Οι επιλογή τους και η χρήση τους από τα τουριστικά πρακτορεία, εξαρτάται από τις παρακάτω ιδιότητες:

- 1) **Αξιοπιστία του πληροφοριακού υλικού.** Ένα τουριστικό πρακτορείο έχει την δυνατότητα άμεσης ενημέρωσης για τις όποιες αλλαγές προκύπτουν στις πτήσεις, τις κρατήσεις θέσεων, και την διαθεσιμότητα των κλινών σε κάθε συνεργαζόμενο ξενοδοχείο.
- 2) **Έγκυρη πληροφόρηση σε θέματα τιμών** του καταναλωτή. Κάθε τουριστικό πρακτορείο έχει άμεση πρόσβαση σε κάθε επίπεδο τιμολόγησης για κάθε επιμέρους υπηρεσία ενός τουριστικού πακέτου, για λόγους ανταγωνιστικότητας.
- 3) **Πληροφόρηση για την ικανοποίηση** του καταναλωτή. Ο βαθμός ικανοποίησης του πελάτη ποικίλλει ανάλογα με τις εκάστοτε ανάγκες του, και οριοθετείται μέσω τεχνικών marketing και την χρήση προγραμμάτων επιβράβευσης πελατών (loyalty schemes).
- 4) **Σύντομος χρόνος απάντησης.** Η ταχύτητα απάντησης του συστήματος ιδιαίτερα για τα μεγάλα τουριστικά πρακτορεία με αυξημένο όγκο ζήτησης είναι σημαντικός παράγοντας, αφού συνήθως η αναζήτηση διαθεσιμότητας, η κράτηση και η επιβεβαίωση των κρατήσεων, γίνεται σε αληθινούς χρόνους (real time –online).
- 5) **Συνεχής διεύρυνση των παρεχόμενων υπηρεσιών** και προϊόντων.
- 6) **Σύστημα τιμολόγησης** των προσφερόμενων υπηρεσιών, παρέχοντας τη δυνατότητα επιλογής πολλαπλών εναλλακτικών μεθόδων τιμολόγησης και χρέωσης του πελάτη.
- 7) **Ευκολία στη χρήση** και στο επίπεδο της **τεχνικής υποστήριξης**. Ο βαθμός ευκολίας στη χρήση του συστήματος σχετίζεται με τον κατάλληλο σχεδιασμό και την ευχρηστία της σελίδας αναζήτησης και πραγματοποίησης κρατήσεων. Όσον αφορά την τεχνική υποστήριξη, σε περίπτωση δυσλειτουργίας του συστήματος, θα πρέπει να υπάρχει άμεση και ταχεία λύση.

4.1.5 Η χρήση των GDS στον ξενοδοχειακό τομέα

Η χρήση των GDS σε σχέση με τις κρατήσεις δωματίων ξενοδοχειακών καταλυμάτων έχουν σαν στόχο την αποτελεσματική διάθεση των διαθέσιμων δωματίων, με απευθείας πώλησή τους σε νέες και υπάρχουσες αγορές, και την αύξηση της ετήσιας πληρότητάς τους [4]. Σημαντική κινητικότητα μέσω των συστημάτων κρατήσεων έχουν αναπτύξει οι μεγάλες αλυσίδες ξενοδοχείων, που διαθέτουν μέσω των συστημάτων αυτών τα δωμάτια των ξενοδοχείων τους. Επίσης, είναι δυνατό να προσφέρουν και άλλα

προϊόντα όπως τις αίθουσες εκδηλώσεων και συνεδρίων, και λοιπές εγκαταστάσεις, για τη σεζόν λειτουργίας τους. Οι ξενοδοχειακές μονάδες έχουν την δυνατότητα πλέον να χρησιμοποιούν ένα κεντρικό σύστημα κρατήσεων για τους εξής σκοπούς:

- Αναζήτηση πληροφοριών για τα διαθέσιμα δωμάτια της ίδιας γεωγραφικής περιοχής ή την υπόλοιπη χώρα.
- Ενημέρωση του συστήματος για διαθεσιμότητα των δωματίων τους, τους τύπους των δωματίων, την προσφερόμενη τιμή, τους όρους συνεργασίας, πρόσθετες παροχές τους κ.λ.π.
- Δυνατότητα κράτησης σε διάφορα άλλα τμήματα του ξενοδοχείου εκτός των δωματίων π.χ. χώρους συνεδρίων, άθλησης κ.λ.π.
- Δυνατότητα κράτησης σε άλλη ξενοδοχειακή μονάδα της περιοχής τους (στη περίπτωση μη διαθεσιμότητας) ή ιδιοκτησίας τους (cross-sales).
- Δυνατότητα εφαρμογής του Yield Management.
- Άντληση πολλαπλών πληροφοριών (τουριστικές, οικονομικές, φορολογικές, στατιστικά).
- Διάθεση δωματίων σε νέες και αναπτυσσόμενες αγορές, πέραν τις υπάρχουσας πελατείας, με διαφημιστική προβολή σε αρκετά χαμηλό κόστος.

Επιπλέον, μπορεί να αποδειχθεί σημαντικό εργαλείο όσον αφορά τον υπολογισμό της προσφοράς και ζήτησης, παρακολουθώντας το βαθμό επισκεψιμότητας σε συγκρίσει με τις κρατήσεις, καθώς και να αποτελέσει σημαντικό βοήθημα για το marketing και τις πωλήσεις του ξενοδοχείου, παρακολουθώντας συγκεκριμένα στατιστικά στοιχεία της ροής πελατών.

4.2 Διαδικτυακές μηχανές κρατήσεων

Το Internet αποτελεί πλέον ένα ιδανικό μέσο για τη δραστηριοποίηση των ξενοδοχειακών μονάδων, τα τουριστικά γραφεία και τις επιχειρήσεις μαζικής εστίασης στις διαδικτυακές τεχνολογίες κρατήσεων. Οι εφαρμογές δεν εγκαθίστανται πλέον τοπικά στους υπολογιστές που υπάρχουν στο front-office ή το back-office των ξενοδοχείων, αλλά η λειτουργικότητα των υπηρεσιών προσφέρεται μέσω ιστοσελίδων στις οποίες έχουν πρόσβαση όλοι οι σχετικά ενδιαφερόμενοι, ανεξαρτήτως τοποθεσίας. Τα διαδικτυακά συστήματα κρατήσεων κυριαρχούν, προσφέροντας στους χρήστες των σελίδων πλούσιο διαδραστικό και πολυμεσικό περιεχόμενο, είναι συνεχώς ενήμερα με τη διαθεσιμότητα δωματίων της ξενοδοχειακής μονάδας, διαχειρίζονται τις κρατήσεις από την αρχή έως το τέλος με αυτοματοποιημένο τρόπο, βάσει των προτιμήσεων του πελάτη, και ενημερώνουν το σύστημα διαχείρισης και τη βάση του ξενοδοχείου σε πραγματικό χρόνο [4].

Ταυτόχρονα, προσφέρονται πολλά *συστήματα διαχείρισης περιεχομένου (Content Management Systems - CMS)*, δηλαδή εφαρμογές κατασκευής ξενοδοχειακών sites με το σχετικό περιεχόμενο την προσέλκυση δυνητικών πελατών. Οι ίδιοι οι διαχειριστές του συστήματος μπορούν να συνδέονται και να διαχειρίζονται τις λειτουργίες του ξενοδοχείου απομακρυσμένα μέσω ιστοσελίδων, ενώ στη διοίκηση της ξενοδοχειακής μονάδας προσφέρονται σύγχρονοι τρόποι συγκέντρωσης πληροφοριών και στατιστικών στοιχείων επισκεψιμότητας στις ιστοσελίδες και την αξιοποίησή τους στη βελτίωση των προσφερόμενων ξενοδοχειακών/τουριστικών προϊόντων και υπηρεσιών [9]. Ένα τέτοιο εργαλείο στατιστικής ανάλυσης για sites είναι το Google Analytics [10].

Στη συνέχεια, παρουσιάζονται συνοπτικά τα κυριότερα προϊόντα ξενοδοχειακού και συναφούς λογισμικού που χρησιμοποιούνται σήμερα στον Ελληνικό Τουριστικό Τομέα.

4.2.1 Content Inn by Amadeus

Η πλατφόρμα δημιουργίας ξενοδοχειακού περιεχομένου της Amadeus [11] κατασκευάστηκε για να προσφέρει εύκολη και γρήγορη πρόσβαση σε προμηθευτές ξενοδοχείων, καθώς και πολλές άλλες ταξιδιωτικές υπηρεσίες. Η πλατφόρμα διαθέτει μεγάλη ευελιξία, καθώς μπορεί να ενσωματωθεί πλήρως στη λύση *Amadeus Selling Platform* και προσφέρει πολλά πλεονεκτήματα όπως την πρόσβαση σε πληθώρα ξενοδοχείων, τη δυνατότητα πραγματοποίησης ομαδικών κρατήσεων, την απευθείας πρόσβαση σε μεγάλες ξενοδοχειακές αλυσίδες, τη σύγκριση τιμών των διάφορων προμηθευτών και τη προβολή της καλύτερης διαθέσιμης τιμής.

4.2.2 HIT-Hospitality

Η Hit-Hospitality [12] προσφέρει μία μεγάλη γκάμα προϊόντων για τις ξενοδοχειακές μονάδες που καλύπτουν όλο το φάσμα λειτουργιών τους τόσο στο front-office όσο και στο back-office:

- **Protel Front Office**

Το Protel καλύπτει τις ανάγκες κάθε μορφής και μεγέθους ξενοδοχείου, προσφέροντας πολλούς τρόπους για την καταχώριση μίας κράτησης, ενώ παράλληλα υποστηρίζει μία σειρά από λειτουργίες όπως, την εισαγωγή κρατήσεων, τη μεταφορά από ένα δωμάτιο σε ένα άλλο, πλάνο πληρότητας δωματίων, αφίξεων / αναχωρήσεων (check in / check out), τη δημιουργία καρτέλας πελάτη (guest profile), τη διαχείριση των τιμολογήσεων και λογαριασμών κλπ.

- **Protel for Headquarters**

Προσφέρει κεντρική διαχείριση όλης της ξενοδοχειακής αλυσίδας στο back-office, με διατήρηση αυτόνομης βάσης δεδομένων των συνδεδεμένων ξενοδοχείων, ενημερώνοντας αυτόματα τα κεντρικά για τη διαθεσιμότητα κάθε μονάδας και λαμβάνοντας από αυτά τις κρατήσεις.

- **WebHotelier**

Το WebHotelier αποτελεί ένα ολοκληρωμένο σύστημα on-line κρατήσεων για ξενοδοχειακές μονάδες και τουριστικά καταλύματα όλων των τύπων. Υποστηρίζει λειτουργίες κράτησης και διανομής του τουριστικού προϊόντος μέσα από ένα εξελιγμένο, γρήγορο και αξιόπιστο σύστημα. Είναι πλήρως παραμετροποιήσιμο και ιδιαίτερα εύκολο στη χρήση, με κύριο προσανατολισμό τη «μετατροπή» του επισκέπτη σε πελάτη.

- **WebHotel**

Το WebHotel είναι μία πληρέστατη εφαρμογή front-office και ηλεκτρονικών κρατήσεων για μικρά ξενοδοχεία και καταλύματα. Το κόστος χρήσης του WebHotel είναι ελάχιστο σε σύγκριση με το κόστος αγοράς ενός προγράμματος PMS, ενώ προσφέρει λειτουργίες όπως διαχείριση κρατήσεων, πλάνο δωματίων, αφίξεις/αναχωρήσεις πελατών, έκδοση παραστατικών χρέωσης, δυνατότητα ηλεκτρονικών κρατήσεων (μέσω του WebHotelier), κλπ.

- **WebSITE CMS**

Το WebSITE (Content Management System) είναι το κατάλληλο εργαλείο για την κατασκευή και συντήρηση του περιεχομένου του site ενός ξενοδοχείου, με επαγγελματικό τρόπο. Δίνει τη δυνατότητα ενσωμάτωσης πολυάριθμων γραφικών στοιχείων και διαδραστικού/πολυμεσικού περιεχομένου στις ιστοσελίδες του ξενοδοχείου.

4.2.3 BOOKonlinenow by IMTEC

Η IMTEC, μέσω της πλατφόρμας BOOKonlinenow [13], προσφέρει τις εξής λύσεις στις ξενοδοχειακές μονάδες:

- **BOOKonlinenow Online Booking Engine**

Ενα ολοκληρωμένο σύστημα on-line κρατήσεων που καλύπτει πλήρως τις ανάγκες και τις απαιτήσεις κάθε ξενοδοχειακής μονάδας. Από τις πολλές δυνατότητες του συστήματος αναφέρουμε ενδεικτικά τις εξής:

- *BOOKonlinenow Facebook Application*: Απευθείας on-line κρατήσεις από τη σελίδα Facebook του ξενοδοχείου, μέσω της σελίδας Facebook χωρίς ανακατεύθυνση ή μεταφορά των επισκεπτών σε άλλη σελίδα.
- *PMS Interface*: Το υποσύστημα σύνδεσης των κρατήσεων με τα ξενοδοχειακά συστήματα PMS, μέσω XML, Web Services.
- *Ασφάλεια Συναλλαγών*: Τα δεδομένα των χρηστών ανταλλάσσονται με το σύστημα μέσω SSL και 128-256bit keys.
- *Πολυγλωσσικότητα (Multilanguage)*: Το σύστημα μπορεί να εμφανίσει το περιεχόμενο των σελίδων του σε περισσότερες από 15 γλώσσες.
- *Μετατροπέας Νομίσματος (Currency Converter)*: Εμφάνιση τιμών σε πολλά εθνικά νομίσματα, πλην του ευρώ.
- *Υποστήριξη πολλαπλών τιμοκαταλόγων*, με ειδικές τιμές για συγκεκριμένες περιόδους (αργίες, τριήμερα, σαββατοκύριακα, events) και διαφορετικές πολιτικές τιμολόγησης.

- *Ειδικές Προσφορές, Πακέτα:* Υποστηρίζονται πολλοί τύποι κρατήσεων: Early Booking, Long Stay, Last Minute Booking, καθώς και ειδικές προσφορές για συγκεκριμένες κατηγορίες πελατών (π.χ., VIP), καθώς και εκπτώτικα κουπόνια (promo codes), ενώ επιτρέπει και τη δημιουργία πακέτων.
- *Διαχείριση Διαθεσιμότητας και Περιορισμών:* Ημερολόγιο διαθεσιμότητας, διαθεσιμότητα ανά τύπο δωματίου, κλείδωμα διαθεσιμότητας, περιορισμοί (μέγιστη και ελάχιστη παραμονή, κλείσιμο κατά την άφιξη, ανοικτό, κλειστό, μη διαθέσιμο).
- *Διαχείριση Κρατήσεων και Πληρωμών:* Διαχείριση πληρωμών, ενημέρωση αφίξεων-αναχωρήσεων και δυνατότητα αλλαγής/ακύρωσης μίας κράτησης.
- *Διαχείριση Πελατολογίου:* Κατηγοριοποίηση πελατών (π.χ., VIP, Gold) και χρήση ειδικών τιμοκαταλόγων και προσφορών ανά κατηγορία πελατών.
- *Πακέτα & Έξτρα Υπηρεσίες:* Απεριόριστα πακέτα (BB, HB, FB) με δυνατότητα διαφορετικής τιμολόγησης ανά περίοδο, έξτρα υπηρεσίες (μίνι μπαρ, μεταφορά με ταξί κ.λπ.), ορισμός προτιμήσεων με δυναμικό τρόπο (sea view, baby cot κ.λπ.).
- *Πολλαπλοί Τρόποι Πληρωμής:* Υποστηρίζονται όλοι οι τρόποι πληρωμής (μέσω πιστωτικής κάρτας, μέσω απευθείας διασύνδεσης με τράπεζα ή μέσω PayPal).
- *Alerts:* Το σύστημα προειδοποιεί τους χρήστες για συμβάντα σχετικά με τις κρατήσεις που πραγματοποιούν ή πρόκειται να πραγματοποιήσουν.
- *Αναφορές & Στατιστικά:* Διασύνδεση με Google Analytics, πολλαπλές αναφορές με διάφορα κριτήρια και δυνατότητα εξαγωγής των αναφορών σε φύλλα εργασίας του Excel για περαιτέρω επεξεργασία και ανάλυση.

• **BOOKonlinenow Channel Management**

Το BOOKonlinenow Channel Management επιτρέπει τη διασύνδεση του ξενοδοχείου με τα πιο γνωστά GDS. Το μεγάλο πλεονέκτημα αυτής της πλατφόρμας είναι η ύπαρξη ενός μοναδικού περιβάλλοντος για την παρακολούθηση των κρατήσεων και την αυτόματη ενημέρωση των συστημάτων κρατήσεων GDS (Sabre, Amadeus, Galileo, Worldspan και Pegasus) και των καναλιών IDS (Booking.com, Expedia, Venere, HRS, hotels.com κ.ά.) με πληροφορίες για τη διαθεσιμότητα και τις τιμές των δωματίων.

4.2.4 Travel Platform by Nelios.com

Η Nelios.com [14] έχει κατασκευάσει μία ολοκληρωμένη σουίτα διαδικτυακών λύσεων για ταξιδιωτικά γραφεία (**Online Travel Agencies - OTA**), την **Travel Platform**, μία εφαρμογή που βοηθά στην ολοκλήρωση όλων των διαδικασιών που απαιτούνται για την πραγματοποίηση μίας online κράτησης. Το Travel Platform περιλαμβάνει εργαλεία που φροντίζουν από την κατασκευή του site του ξενοδοχείου και τη realtime σύνδεση με τους προμηθευτές, έως τη σύνδεση με την εφαρμογή λογιστικής και το back-office του ταξιδιωτικού γραφείου. Συγκεκριμένα, η πλατφόρμα συνδέεται σε πραγματικό χρόνο, χρησιμοποιώντας XML-Web Services, με προμηθευτές τύπου GDS κ.λπ., οπότε οι χρήστες μπορούν να πραγματοποιούν κρατήσεις σε πραγματικό χρόνο.

4.2.5 nsTravel, nsTIS by NetSemantics

Το **nsTravel** [15] είναι ένα από τα προϊόντα της εταιρείας που απευθύνονται σε τουριστικές επιχειρήσεις. Πρόκειται για μία ολοκληρωμένη πλατφόρμα και εξασφαλίζει τη βέλτιστη ροή και εκτέλεση των καθημερινών εργασιών, την οργάνωση και τη συνεργασία μεταξύ των τμημάτων, καθώς και την παροχή πληροφοριών σε πραγματικό χρόνο. Η NetSemantics αναπτύσσει και το **nsTIS B2B & B2C Booking Engine**, ένα ολοκληρωμένο σύστημα δημιουργίας ηλεκτρονικών καταστημάτων για τη διανομή και την προώθηση του τουριστικού προϊόντος. Το σύστημα παρέχει πλήρη διαχείριση online κρατήσεων (B2B & B2C), καθώς και τη δυνατότητα διανομής τουριστικών προϊόντων τρίτων με τη χρήση μηχανισμών XML.

4.2.6 SingularLogic Hotel

Το **SingularLogic Hotel** [16], είναι ολοκληρωμένο πληροφοριακό σύστημα διαχείρισης ξενοδοχειακών μονάδων και αλυσίδων ξενοδοχείων, άριστα προσαρμοσμένο στις καθημερινές συνθήκες λειτουργίας τους. Παρέχεται ένα ενιαίο πληροφοριακό σύστημα για ολόκληρο το ξενοδοχείο και τις υπομονάδες του. Παρακολουθούνται ταυτόχρονα όλα τα τμήματα του ξενοδοχείου, χρησιμοποιώντας μία ενιαία εφαρμογή και μία κεντρική βάση δεδομένων. Έτσι, επιτυγχάνεται η μείωση του κόστους

λειτουργίας, η συνολική παρακολούθηση των διάφορων ενεργειών και λογαριασμών πελατών, η δυνατότητα εφαρμογής αποτελεσματικών ενεργειών marketing κ.ο.κ. Η εφαρμογή παρακολουθεί σε πραγματικό χρόνο όλες τις λειτουργίες ενός σύγχρονου ξενοδοχείου, παρέχοντας ταυτόχρονα άμεση συγκεντρωτική και αναλυτική πληροφόρηση. Απευθύνεται σε ξενοδοχειακές μονάδες, ανεξαρτήτως μεγέθους, λόγω των συνδυασμών που προσφέρει, αλλά και των έξτρα λειτουργιών της.

4.2.7 Sunsoft

Η Sunsoft προτείνει με τη σειρά της μία πληθώρα επαγγελματικών λύσεων στις επιχειρήσεις του κλάδου, όπως το **Αλέξανδρος FrontOffice**, το **Alexandros Consolidation** (Κεντρική Διαχείριση Στατιστικών Ξενοδοχείων Αλυσίδας), την ασύρματη κλήση σερβιτόρου (**Leanko**), καθώς επίσης και την αμφίδρομη σύνδεση με διαδικτυακές μηχανές κρατήσεων (**Web Booking Engines**) [17]. Το λογισμικό Αμφίδρομης Σύνδεσης με Web Booking Engine, αναλαμβάνει να κάνει αυτόματα την ενημέρωση της διαθεσιμότητας δωματίων και των τιμών από τα στοιχεία του Αλέξανδρος Front Office και να παραλάβει αυτόματα τις επιβεβαιωμένες κρατήσεις που στέλνει το Web Booking Engine.

4.2.8 Sysco [18]

Ενδεικτικά τα προϊόντα της είναι:

- **Sysco CRS (Central Reservation System)**

Το Sysco CRS καλύπτει την ανάγκη μίας αλυσίδας ξενοδοχείων για κεντρική διαχείριση των δεδομένων της. Μπορεί να διαχειρίζεται πληροφορίες από πολλά διαφορετικά ξενοδοχεία που λειτουργούν με το Eritome PMS (με καθένα να μπορεί να λειτουργεί και αυτόνομα), εφοδιάζοντας ταυτόχρονα το κεντρικό γραφείο κρατήσεων της επιχείρησης με όλα τα στοιχεία που απαιτούνται για την καθολική διαχείριση των κρατήσεων και την αξιοποίηση των πληροφοριών από το κεντρικό διοικητικό τμήμα.

- **Eritome PMS**

Το Eritome.NET απευθύνεται σε μεγάλο και μεσαίου μεγέθους ξενοδοχειακές μονάδες, ενώ λειτουργεί, μέσω Internet Explorer, κάτι που σημαίνει ότι δεν υπάρχει κάτι εγκατεστημένο τοπικά στους υπολογιστές του ξενοδοχείου. Τα πάντα βρίσκονται σε κάποιο κεντρικό μηχάνημα είτε στο ξενοδοχείο (Hotel Application & Database Server) είτε σε κάποιο απομακρυσμένο data center και οι χρήστες έχουν πρόσβαση στο σύστημα μέσω ιστοσελίδων με εξουσιοδοτημένα accounts.

- **Medallion PMS**

Το Medallion PMS αποτελεί σήμερα ένα κατάλληλο ξενοδοχειακό σύστημα για μεμονωμένες ξενοδοχειακές μονάδες μικρής και μεσαίας δυναμικότητας με μικρό και πολύ απασχολημένο προσωπικό.

- **e-Hermes**

Το e-Hermes είναι ένα σύστημα online κρατήσεων κατάλληλο για οποιαδήποτε ξενοδοχειακή μονάδα. Προσφέρει στον επισκέπτη του site του ξενοδοχείου απεριόριστες επιλογές ώστε γρήγορα, εύκολα και με τη μέγιστη ασφάλεια να ολοκληρώσει μία κράτηση.

4.3 Διαδικασία online κράτησης

Για να ολοκληρωθεί επιτυχώς μία κράτηση μέσω διαδικτύου, θα πρέπει η online πλατφόρμα διαχείρισης και πραγματοποίησης κρατήσεων να ολοκληρώσει διαδοχικά μία σειρά από στάδια. Σε αυτή την ενότητα παρουσιάζονται τα διαδοχικά αυτά βήματα, τόσο από την πλευρά του τελικού χρήστη όσο και από την πλευρά του συστήματος, με τη σειρά εκτέλεσής τους.

4.3.1 Βήμα 1 – Αναζήτηση βάσει προτιμήσεων

Προκειμένου να βρεθεί η οποιαδήποτε διαθεσιμότητα από τους προμηθευτές online κρατήσεων, θα πρέπει αρχικά να ζητηθούν οι προτιμήσεις και τα κριτήρια αναζήτησης του πελάτη. Για αυτό το λόγο, τα online συστήματα κρατήσεων διαθέτουν ειδικές φόρμες συμπλήρωσης με επιλογές του πελάτη, ώστε να γίνει ένα φιλτράρισμα από όλες τις διαθέσιμες επιλογές και εμφάνιση μόνο όσων πληρούν τα σχετικά κριτήρια. Οι φόρμες αυτές μπορούν να είναι από αρκετά απλές, περιέχοντας τα ελάχιστα δυνατόν πεδία, αναγκαία για μία αναζήτηση, έως αρκετά σύνθετες φόρμες με πολλαπλά διαφορετικά κριτήρια και συνδυασμό αυτών. Συνήθως, οι ελάχιστα απαραίτητες πληροφορίες που χρειάζεται ένα σύστημα προκειμένου να προβεί σε αναζήτηση ξενοδοχειακών καταλυμάτων, είναι αφενός οι ημερομηνίες άφιξης και αναχώρησης, ο προορισμός (πόλη, χώρα), και ο αριθμός και τύπος ατόμων ανά δωμάτιο (π.χ. 2

ενήλικες και 1 παιδί σε 1 δωμάτιο). Σε σύνθετες περιπτώσεις μπορούν να αναζητούνται καταλύματα βάσει κατηγορίας (αστέρια), παροχών (spa, χώροι εκδηλώσεων και συνεδρίων, parking, κ.λπ.), εύρους τιμής κ.λπ.

Στη συνέχεια, αφού ο ενδιαφερόμενος χρήστης εισάγει τις προτιμήσεις του, αυτές αποστέλλονται μέσω της ηλεκτρονικής φόρμας προς το back-end του συστήματος, για να διενεργηθεί η αναζήτηση δεδομένων των επιλεγμένων κριτηρίων.

4.3.2 Βήμα 2 – Εμφάνιση διαθεσιμότητας

Στο στάδιο αυτό, το back-end σύστημα (που αποτελεί τον πυρήνα μίας μηχανής online κρατήσεων), λαμβάνει τα κριτήρια του χρήστη και προσπαθεί να φιλτράρει τα διαθέσιμα καταλύματα και δωμάτια ώστε να επιστρέψει στο χρήστη μόνο τις επιλογές εκείνες που ικανοποιούν τις απαιτήσεις του. Σε αυτό το σημείο, ανάλογα τον τρόπο με τον οποίο λειτουργεί το σύστημα, δύναται είτε να αναζητήσει τοπικά στην βάση δεδομένων του ποια δωμάτια ξενοδοχείων καλύπτουν τις απαιτήσεις μέσω των κατάλληλων ερωτημάτων, είτε σε πιο σύνθετες εφαρμογές, να στείλει απομακρυσμένα το ερώτημα με ειδικά διαμορφωμένο μήνυμα (π.χ. SOAP/XML, JSON) σε μία ή περισσότερες υπηρεσίες-προμηθευτές online κρατήσεων. Οι προμηθευτές αυτοί συλλέγουν πληροφορίες διαθεσιμότητας από διάφορες πηγές GDS, και στις περισσότερες περιπτώσεις εκτελούν και τις επόμενες υπηρεσίες καταχώρησης στοιχείων, κράτησης και έκδοσης ηλεκτρονικής κράτησης. Επομένως, αποτελούν ένα middleware μεταξύ της εκάστοτε online εφαρμογής και της διασύνδεσής της με το αυξημένο πλήθος GDS παρόχων, με ένα διαφανώς ίδιο μοντέλο επικοινωνίας (π.χ. Web Services), χωρίς την απαίτηση διαμόρφωσης του συστήματος ανάλογα με τις απαιτήσεις του εκάστοτε GDS.

Αφού εξάγει το σύστημα τα διαθέσιμα αποτελέσματα κλινών από το σύνολο της διαθεσιμότητάς του, μορφοποιεί και οργανώνει τα αποτελέσματα αυτά όπως ορίζονται από την εφαρμογή. Για παράδειγμα, κάποιες online μηχανής κρατήσεων επιστρέφουν τα αποτελέσματα με αύξουσα σειρά τιμής, συνεπώς η βέλτιστη επιλογή θεωρείται εκείνη με την μικρότερη τιμή. Σε άλλα συστήματα τα αποτελέσματα ομαδοποιούνται ανάλογα τη κατηγορία του ξενοδοχείου ή την απόσταση από το κέντρο της πόλης κλπ. Σε όποια περίπτωση, τα μορφοποιημένα και ομαδοποιημένα αποτελέσματα αυτά, σε συνδυασμό με συμπληρωματικές πληροφορίες για τα καταλύματα, επιστρέφονται μέσω ιστοσελίδας στον υποψήφιο επισκέπτη, προς επιλογή του.

4.3.3 Βήμα 3 – Επιλογή χρήστη

Στη σελίδα αποτελεσμάτων διαθεσιμότητας, ο χρήστης μπορεί να επιλέξει σε ποια δωμάτια επιθυμεί να κάνει κράτηση, αφού επεξεργαστεί τις διαθέσιμες πληροφορίες για τον κάθε τύπο δωματίου, όπως φωτογραφίες, παροχές, περιγραφές, σχόλια προηγούμενων επισκεπτών κ.ά. Στα πιο εξελιγμένα συστήματα κρατήσεων, δίνεται συνήθως η επιπλέον δυνατότητα φιλτραρίσματος, σύγκρισης και αφαίρεσης δωματίων από την υπάρχουσα λίστα αποτελεσμάτων, χωρίς να διενεργηθεί νέα αναζήτηση. Η μέθοδος αυτή βοηθά το χρήστη να καταλήξει στο(α) επιθυμητό(ά) δωμάτιο(α).

4.3.4 Βήμα 4 – Καταχώρηση στοιχείων επισκεπτών

Αφού επιλέξει ο χρήστης τα δωμάτια για τα οποία επιθυμεί να κάνει κράτηση, το σύστημα κρατά προσωρινά τις επιλογές αυτές και μεταφέρει τον χρήστη στο επόμενο στάδιο, την καταχώρηση των απαραίτητων προσωπικών στοιχείων που απαιτούνται για να ολοκληρωθεί η κράτηση. Εδώ καταχωρούνται τα στοιχεία επικοινωνίας του χρήστη που διενεργεί την κράτηση, όπως επίσης τα βασικά στοιχεία των επισκεπτών, όπως όνομα, γένος, ηλικία για παιδιά κ.ά. (ανάλογα τις απαιτήσεις της εφαρμογής ή των προμηθευτών), ανά δωμάτιο. Αφού συμπληρωθεί η φόρμα αυτή και ελεγχθεί, αποστέλλεται στο back-office σύστημα και αποθηκεύεται προσωρινά.

4.3.5 Βήμα 5 – Πληρωμή και ολοκλήρωση κράτησης

Στο βήμα αυτό, εμφανίζεται στον χρήστη μία επισκόπηση των επιλογών του προς κράτηση, καθώς και των στοιχείων που έχει συμπληρώσει για τους επισκέπτες. Σε αυτό το σημείο δίδονται οι επιλογές πληρωμής για την συγκεκριμένη κράτηση. Οι διαθέσιμοι τρόποι πληρωμής ποικίλλουν ανάλογα την εφαρμογή. Αφού ο πελάτης συμφωνήσει με τα στοιχεία και τους όρους κράτησης, συμπληρώνει τον τρόπο πληρωμής (για παράδειγμα τα στοιχεία της πιστωτικής κάρτας μέσω SSL καναλιού) και τα στοιχεία αυτά αποστέλλονται στο κεντρικό σύστημα.

Το back-office σύστημα εκτελεί με τη σειρά του μία αλληλουχία ενεργειών προκειμένου να ολοκληρώσει την κράτηση. Αρχικά, ελέγχει ότι τα συγκεκριμένα δωμάτια είναι ακόμα διαθέσιμα, προτού γίνει η χρέωση της κάρτας. Καθώς υπάρχει η δυνατότητα ταυτόχρονης σύνδεσης πολλαπλών χρηστών, υπάρχει η πιθανότητα κάποιο δωμάτιο που προηγουμένως ήταν διαθέσιμο, να έχει κρατηθεί πλέον από άλλο επισκέπτη. Συνεπώς, ο έλεγχος αυτός πριν την χρέωση εξυπηρετεί στην αποφυγή λανθασμένων χρεώσεων του συστήματος σε δωμάτια που δεν είναι δυνατό να κρατηθούν πλέον (περίπτωση over-booking). Αμέσως μετά, και αφού τα δωμάτια παραμένουν διαθέσιμα, γίνεται μία «προ-κράτηση» των δωματίων, σε προσωρινό επίπεδο, έως ότου ολοκληρωθεί η χρέωση, γνωστή και ως διαδικασία room-blocking. Στη συνέχεια, γίνεται η χρέωση της κάρτας (ή όποιου άλλου μηχανισμού πληρωμής), και στην περίπτωση και μόνο όπου ολοκληρωθεί επιτυχώς η χρέωση, τότε ολοκληρώνεται η κράτηση. Η ολοκλήρωση της κράτησης γίνεται συλλέγοντας και αποθηκεύοντας όλα τα παραπάνω στοιχεία του χρήστη, τις επιλογές δωματίων και τα στοιχεία χρέωσης και συνθέτοντας τη τελική κράτηση. Τα δωμάτια που προηγουμένως είχαν προ-κρατηθεί (blocked), δεσμεύονται πλέον μόνιμα για τη συγκεκριμένη κράτηση (reserved). Στο τέλος του βήματος αυτού, ενημερώνεται η βάση δεδομένων και αποθηκεύονται σε αυτήν τα στοιχεία κράτησης. Στα σύνθετα συστήματα, ο ενδιαμέσος «μεσάζοντας» αναλαμβάνει το ρόλο της προ-κράτησης και τελικής κράτησης, μέσω των κατάλληλων μηνυμάτων με τη σχετική πληροφορία.

4.3.6 Βήμα 6 – Έκδοση ηλεκτρονικής κράτησης και επιβεβαίωση

Κατά την ολοκλήρωση της κράτησης, εκδίδεται ένας μοναδικός αριθμός κράτησης, με τον οποίο συσχετίζεται η κράτηση αυτή. Ο αριθμός αυτός είναι σημαντικός τόσο για τον χρήστη όσο και για το σύστημα και το ξενοδοχείο, καθώς αποτελεί σημείο αναφοράς είτε κατά την άφιξη του πελάτη στο κατάλυμα, είτε σε περίπτωση επιλογής ακύρωσης από την πλευρά του χρήστη, είτε επισκόπησης ή/και οποιασδήποτε μεταβολής στα δεδομένα της κράτησης.

Συνεπώς, τα περισσότερα συστήματα, εμφανίζουν κατόπιν ολοκλήρωσης της κράτησης μία σελίδα επιβεβαίωσης της κράτησης που πραγματοποίησε ο χρήστης, με όλα τα απαραίτητα στοιχεία αυτής και τον αριθμό κράτησης. Παράλληλα, τα περισσότερα συστήματα αποστέλλουν την επιβεβαίωση και μέσω ηλεκτρονικού ταχυδρομείου στη διεύθυνση που έχει καταχωρήσει ο χρήστης για επικοινωνία, ή/και επιτρέπουν την εκτύπωση της online σελίδας και τη μετατροπή της σε PDF. Κατά αυτό τον τρόπο, ο χρήστης διατηρεί το αντίγραφο της κράτησης που διεκπεραίωσε, το οποίο και χρησιμοποιεί κατά την άφιξή του στο αντίστοιχο ξενοδοχειακό κατάλυμα.

5 Ανάλυση Εφαρμογής

Στόχος της παρούσας εργασίας είναι η ανάπτυξη της εφαρμογής διαχείρισης ξενοδοχείων με την αρχιτεκτονική browser/server, και η απρόσκοπτη λειτουργία της για τους χρήστες. Τα βασικά στοιχεία που αφορούν τα ξενοδοχεία, τις κρατήσεις και τη λειτουργικότητα του συστήματος βρίσκονται αποθηκευμένα σε μία βάση δεδομένων στον server, παρέχοντας τον συσχετισμό των στοιχείων, τον έλεγχο ακεραιότητας των δεδομένων, και το συγχρονισμό αυτών. Σε επίπεδο εφαρμογής, χρησιμοποιώντας την τεχνολογία δυναμικών σελίδων στους browser για τους χρήστες, επιτυγχάνεται η ταυτόχρονη σύνδεση πολλαπλών χρηστών παράλληλα, καθένας από τους οποίους μπορεί να πραγματοποιήσει μία κράτηση ξενοδοχείου, διαφανώς ως προς τους υπόλοιπους συνδεδεμένους χρήστες. Στις σελίδες των χρηστών, πραγματοποιείται ο πρωταρχικός έλεγχος των στοιχείων που υποβάλλονται κατά την κράτηση, πριν από τον δεύτερο έλεγχο που εκτελείται αφού ληφθούν τα στοιχεία αυτά από τον server.

Σε αυτή την ενότητα παρουσιάζονται οι τεχνολογίες που χρησιμοποιήθηκαν κατά την εκπόνηση της πτυχιακής εργασίας, κατά την υλοποίηση της εφαρμογής.

5.1 Τεχνολογίες που χρησιμοποιήθηκαν

5.1.1 MySQL

Το σημαντικότερο επίπεδο της εφαρμογής είναι η βάση δεδομένων, όπου βρίσκονται αποθηκευμένα τα στοιχεία των ξενοδοχείων, κρατήσεων, χρεώσεων και πληροφορίες χρηστών. Από την πληθώρα των σχεσιακών βάσεων δεδομένων που είναι διαθέσιμες, επελέγη η βάση MySQL. Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων [19] της οποίας ο πηγαίος κώδικας είναι διαθέσιμος με ελεύθερη άδεια χρήσης. Το SQL αναφέρεται στη γλώσσα *SQL (Structured Query Language)* [20], μια γλώσσα ερωτημάτων που σχεδιάστηκε ανεξάρτητα από την MySQL για τη διαχείριση δεδομένων, σε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (MySQL, PostgreSQL, Oracle κ.λπ.). Τα ερωτήματα SQL μπορούν να κληθούν μέσα από δυναμικές σελίδες, όπως για παράδειγμα από κώδικα της PHP για την αποθήκευση και ανάκτηση στοιχείων για το περιεχόμενο της ιστοσελίδας, η οποία δημιουργείται δυναμικά στον τελικό χρήστη.

Η RDB MySQL θεωρείται ως αρκετά δημοφιλής βάση δεδομένων λόγω της αυξημένης απόδοσης, της ευελιξίας της κατά το σχεδιασμό και την υλοποίηση βάσεων, και της υψηλής αξιοπιστίας που παρέχει. Επιπλέον, χρησιμοποιείται από μεγάλες εταιρείες με μεγάλο όγκο δεδομένων, μέχρι εξειδικευμένες εφαρμογές με ενσωματωμένες βάσεις δεδομένων παγκοσμίως. Επειδή είναι ανοικτού πηγαίου κώδικα (open source), δίνει τη δυνατότητα διαμόρφωσής της σύμφωνα με τις ανάγκες του εκάστοτε συστήματος. Έχει βρεθεί ότι δουλεύει καλύτερα στην διαχείριση περιεχομένου και όχι κατά την εκτέλεση συναλλαγών. Μπορεί να εγκατασταθεί σε οποιοδήποτε λειτουργικό σύστημα (Windows, Linux, Unix) χωρίς προβλήματα.

5.1.2 PHP

Η PHP, όπου τα αρχικά σημαίνουν Hypertext PreProcessor, είναι μια γλώσσα συγγραφής σεναρίων (scripting language) που ενσωματώνεται μέσα στον κώδικα των σελίδων HTML και εκτελείται στην πλευρά του server (server-side scripting) [21]. Ανταγωνιστικές της τεχνολογίας PHP είναι οι εξής γλώσσες προγραμματισμού: ASP (Active Server Pages), CFML (ColdFusion Markup Language) και JSP (JavaServer Pages). Ο σκοπός της γλώσσας είναι να δώσει τη δυνατότητα στους web developers να δημιουργούν δυναμικά παραγόμενες ιστοσελίδες. Ο κώδικας της PHP περικλείεται με ειδικά tags αρχής και τέλους για να μπορούμε να εισερχόμαστε και να εξερχόμαστε από το PHP mode, εντός της HTML σελίδας. Αυτό που ξεχωρίζει την PHP από μια γλώσσα όπως η JavaScript, η οποία εκτελείται στην πλευρά του χρήστη (client-side), είναι ότι ο κώδικας της εκτελείται στον server. Ο χρήστης (client) λαμβάνει το αποτέλεσμα της εκτέλεσης αυτού του script ως μία ενιαία σελίδα HTML, χωρίς να είναι σε θέση να γνωρίζει ποιος μπορεί να είναι ο αρχικός κώδικας.

Η PHP μπορεί να κάνει ό, τι και τα άλλα προγράμματα της τεχνολογίας CGI [22], όπως επεξεργασία των δεδομένων μιας φόρμας, δημιουργία δυναμικού περιεχομένου ιστοσελίδων ή αποστολή και λήψη cookies και μεταβλητών του session. Ίσως το δυνατότερο και πιο σημαντικό χαρακτηριστικό της PHP είναι η υποστήριξη που παρέχει σε μια ευρεία γκάμα από βάσεις δεδομένων. Έτσι, η δημιουργία

μιας ιστοσελίδας που να παρέχει υποστήριξη σε βάσεις δεδομένων γίνεται με απλό τρόπο. Υποστηρίζει βάσεις δεδομένων, όπως τις: Informix, MySQL, Oracle, PostgreSQL, Sybase, κ.ά. Υποστηρίζει επίσης πρωτόκολλα όπως τα IMAP, SNMP, NNTP, POP3 ή και το HTTP.

5.1.3 HTML και CSS

Η **HTML (HyperText Markup Language)** είναι η κύρια γλώσσα παρουσίασης υπερκειμένου σελίδων, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων που παρουσιάζονται στους σύγχρονους web browsers [23]. Ο σκοπός ενός web browser είναι να διαβάζει τα αρχεία HTML και να τα συνθέτει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο browser δεν εμφανίζει τον πηγαίο κώδικα HTML, αλλά τον χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της σελίδας. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων (δηλαδή εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα. Μπορούν επίσης να ενσωματώνονται σενάρια εντολών σε γλώσσες όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML.

Οι Web browsers μπορούν επίσης να αναφέρονται σε στυλ μορφοποίησης CSS για να ορίζουν την εμφάνιση και τη διάταξη του κειμένου και του υπόλοιπου υλικού. Τα **CSS (Cascading Style Sheets)** αποτελούν ένα πολύ καλό εργαλείο για την διαμόρφωση της εμφάνισης και τη διάταξη (layout) των ιστοσελίδων [24]. Τα CSS έχουν να κάνουν με γραμματοσειρές (fonts), με χρώματα (colours), με περιθώρια (margins), με εικόνες φόντου (background images) και με πολλά άλλα. Επιπλέον, υποστηρίζονται απ' όλους τους browsers.

Στο ερώτημα ποια είναι η διαφορά ανάμεσα στα CSS και την HTML, μπορούμε να πούμε ότι η HTML χρησιμοποιείται για να δομήσει το περιεχόμενο (content), ενώ τα CSS χρησιμοποιούνται για τη διαμόρφωση ή μορφοποίηση (formatting) του δομημένου περιεχομένου. Η χρήση των CSS προσφέρει πλεονεκτήματα, όπως:

- Δυνατότητα για τον ταυτόχρονο έλεγχο της διάταξης (layout) πολλών εγγράφων από ένα μόνο φύλλο στυλ (style sheet).
- Πιο ακριβής έλεγχος της διάταξης (layout) των ιστοσελίδων.
- Εφαρμογή διαφορετικής διάταξης σε διαφορετικές εξόδους, όπως screen, print κ.ά.
- Αναρίθμητες προηγμένες και εξεζητημένες τεχνικές.

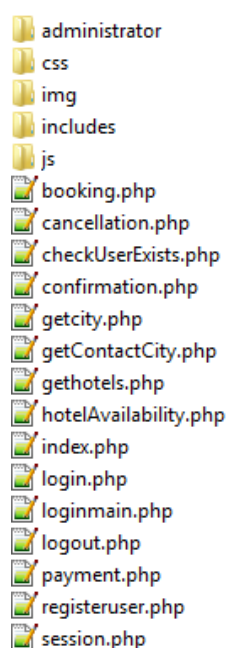
5.1.4 JavaScript

Η **JavaScript (JS)** είναι μία γλώσσα προγραμματισμού, ώστε τα σενάρια από την πλευρά του πελάτη (client-side scripts) να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου που εμφανίζεται [25]. Η JavaScript χρησιμοποιείται και σε εφαρμογές εκτός ιστοσελίδων. Τέτοια παραδείγματα είναι τα έγγραφα PDF, οι εξειδικευμένοι φυλλομετρητές (site-specific browsers) και οι μικρές εφαρμογές της επιφάνειας εργασίας (desktop widgets). Οι νεότερες εικονικές μηχανές και πλαίσια ανάπτυξης για JavaScript (όπως το Node.js [26]) έχουν επίσης κάνει τη JavaScript πιο δημοφιλή για την ανάπτυξη εφαρμογών Ιστού στην πλευρά του διακομιστή (server-side).

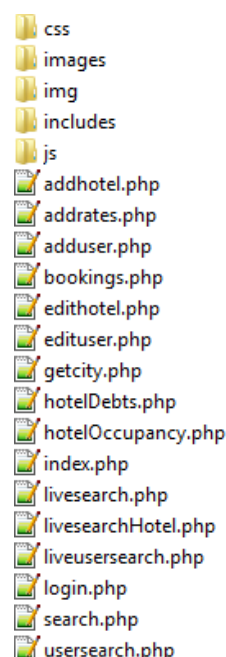
6 Δομή εφαρμογής

6.1 Απαιτήσεις συστήματος

Το σύστημα κρατήσεων που υλοποιήθηκε στην παρούσα εργασία αποτελείται από δύο διακριτά μέρη, την πλευρά του server καθώς και την πλευρά του client που εκτελείται μέσω web browser. Οι σελίδες που παρουσιάζονται στο χρήστη παράγονται με δυναμικό τρόπο στη πλευρά του server, χρησιμοποιώντας κώδικα PHP σε συνδυασμό με HTML markup. Η έκδοση της PHP που χρησιμοποιήθηκε η 5.3, η οποία θεωρείται stable κατά τη χρονική στιγμή που δημιουργήθηκε η παρούσα εφαρμογή. Χρησιμοποιήθηκε ένας remote web server που υποστηρίζει την PHP 5.3, στον οποίο εγκαταστάθηκαν τα απαραίτητα αρχεία και οι σελίδες PHP. Στον web server υπάρχουν δύο σημεία αναφοράς, οι σελίδες που εμφανίζονται στους χρήστες και στις οποίες έχουν καθολική πρόσβαση όλοι οι χρήστες, όπως επίσης και ο φάκελος /administrator/, στον οποίο υπάρχουν οι σελίδες πρόσβασης στο διαχειριστικό τμήμα της εφαρμογής.



Εικόνα 3 – Δομή αρχείων web server



Εικόνα 4 – Δομή αρχείων back-end διαχείρισης

Η δημιουργία και επεξεργασία των σελίδων PHP μπορεί να πραγματοποιηθεί μέσω ενός οποιουδήποτε editor κειμένου, όπως το Notepad++ [27]. Στη συγκεκριμένη περίπτωση, χρησιμοποιήθηκε το Netbeans IDE 7.4 [28], δημιουργώντας ένα νέο project PHP, καθώς έχει αρκετές δυνατότητες κατά την συγγραφή τέτοιων σελίδων.

Οι σελίδες αυτές απαιτούν την ύπαρξη μίας σχεσιακής βάσης δεδομένων, για την ανάκτηση και αποθήκευση της εκάστοτε πληροφορίας. Για τους σκοπούς της εργασίας αυτής, χρησιμοποιήθηκε η βάση δεδομένων MySQL, έκδοση 5.5, η οποία επίσης βρίσκεται σε remote server. Για την προβολή, επεξεργασία, και διαγραμματική απεικόνιση των στοιχείων της βάσης κατά το στάδιο της υλοποίησης χρησιμοποιήθηκε επιπροσθέτως το εργαλείο MySQL Workbench [29]. Με το εργαλείο αυτό, συνδεδεμένοι στην σχετική βάση ενδιαφέροντος, έγινε η δημιουργία των πινάκων με τους τύπους και τους περιορισμούς των πεδίων τους, η δημιουργία και διασύνδεση κλειδιών και συσχετίσεων των πινάκων, και οι δοκιμές και επαλήθευση των ερωτημάτων που χρησιμοποιήθηκαν στις σελίδες PHP. Επίσης, μέσω του MySQL Workbench δημιουργήθηκε το μοντέλο οντοτήτων – συσχετίσεων της βάσης δεδομένων, όπως παρουσιάζεται στην ενότητα 6.3. Τέλος, στην πλευρά του χρήστη χρησιμοποιήθηκε κώδικας Javascript για την διαχείριση της διάδρασης του χρήστη με τη σελίδα, σε συνδυασμό με έτοιμες βιβλιοθήκες Javascript, όπως το jQuery [30], που παρέχουν επιπλέον δυνατότητες προγραμματισμού πάνω στην

Javascript του browser. Οι επιπλέον βιβλιοθήκες που χρησιμοποιήθηκαν παρουσιάζονται στα σημεία εφαρμογής τους, σε επόμενες ενότητες.

6.2 Επικοινωνία με τη Βάση Δεδομένων

Για να επικοινωνήσει η σελίδα PHP με την βάση δεδομένων, απαιτείται η σύνδεση με αυτήν, προτού εκτελεστούν ερωτήματα. Η PHP διαθέτει έτοιμες εντολές για την διασύνδεσή της με την MySQL, οι οποίες και χρησιμοποιήθηκαν στη παρούσα εργασία. Αρχικά, γίνεται σύνδεση με τη βάση με την εντολή `mysql_connect`, δεδομένου του `hostname` της βάσης και του `username` και `password` πρόσβασης. Στη συνέχεια χρησιμοποιείται η εντολή `mysql_select_db` με όρισμα την βάση στην οποία επιθυμείται η πρόσβαση (ένα `hostname` αφορά σε ένα ολόκληρο DBMS, που μπορεί να περιέχει πολλές βάσεις). Τέλος, θέτουμε το `character-set encoding` σε UTF-8 για την ακεραιότητα των δεδομένων, μέσω την εντολής `mysql_set_charset`.

```
include "includes/configuration.php";
$con=mysql_connect($location,$username,$password);
mysql_select_db($database_name);
mysql_set_charset('utf8',$con);
```

Σχήμα 1 – Κώδικας σύνδεσης στη βάση

Οι μεταβλητές που ορίζονται για το `hostname`, `username`, `password`, και `database_name` έχουν δηλωθεί σε ειδικό αρχείο PHP στην σελίδα `configuration.php` στον φάκελο `/includes/`.

```
<?php
$location="ihotelier.db.11771338.hostedresource.com";
$database_name="ihotelier";
$username="ihotelier";
$password="*****";
?>
```

Σχήμα 2 – Μεταβλητές σύνδεσης στη βάση

Η δημιουργία σύνδεσης γίνεται στην αρχή κάθε σελίδας αυτόνομα, ενώ στο τέλος της σελίδας, εφόσον έχει ολοκληρωθεί η δημιουργία του HTML markup, φροντίζουμε να γίνεται αποσύνδεση από τη βάση, ώστε να μην υπάρχουν εκκρεμείς συνδέσεις με τη βάση και να αυξάνεται άσκοπα το bandwidth επικοινωνίας μεταξύ web server και DB server. Η αποσύνδεση γίνεται μέσω της εντολής `mysql_close` και όρισμα την σύνδεση που ορίστηκε στην αρχή της σελίδας.

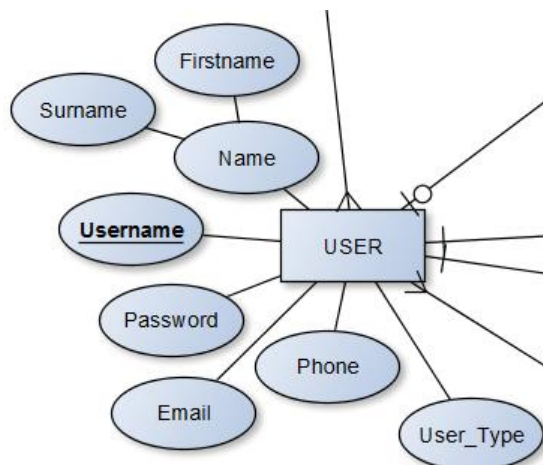
```
mysql_close($con);
```

6.3 Δομή βάσης δεδομένων

6.3.1 Οντότητες και Συσχετίσεις

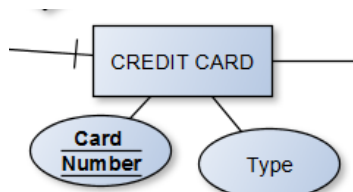
Στην παρούσα εργασία διακρίνονται τρεις βασικές οντότητες πληροφορίας, οι **χρήστες** του συστήματος, οι **ξενοδοχειακές μονάδες** και τα **δωμάτια** κάθε μονάδας.

Σε ότι αφορά τους χρήστες, δημιουργείται μία οντότητα, η **USER**, η οποία περιλαμβάνει τις ιδιότητες (attributes) *όνομα*, *επίθετο*, *email*, *τηλέφωνο επικοινωνίας*, *username* και *password*. Επίσης, κατά την ανάλυση της εφαρμογής προκύπτει και μία επιπλέον ιδιότητα, αυτή του *ρόλου του χρήστη* (απλός χρήστης ή διαχειριστής). Από τις παραπάνω ιδιότητες επιλέγεται το *username* να έχει μοναδική τιμή ανά χρήστη (δεν μπορούν δύο χρήστες να έχουν το ίδιο username), επομένως και θεωρείται πρωτεύον κλειδί αναγνώρισης των εγγραφών.



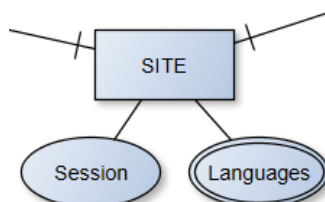
Εικόνα 5 – Σχεδίαση οντότητας USER

Ταυτόχρονα, κάθε χρήστης διαθέτει μία πιστωτική κάρτα, επομένως δημιουργείται ακόμα μία οντότητα, αυτή της **ΠΙΣΤΩΤΙΚΗΣ ΚΑΡΤΑΣ**, με ιδιότητες τον *αριθμό* της κάρτας και τον *τύπο* της. Εδώ, ο αριθμός της κάρτας είναι μοναδικός ανά εγγραφή, συνεπώς και πρωτεύον κλειδί.



Εικόνα 6 – Σχεδίαση οντότητας CREDIT CARD

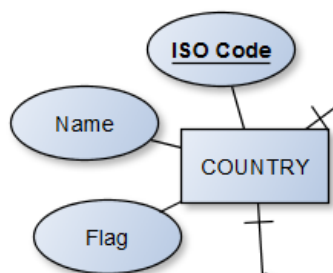
Παράλληλα, ο χρήστης κάνει εγγραφή στο σύστημα, συνεπώς δημιουργείται μία ακόμα οντότητα, αυτή του **SITE**, με δύο ιδιότητες, αυτές του *Session* διαχείρισης χρηστών και των *γλωσσών* του site, που είναι πλειότιμη ιδιότητα.



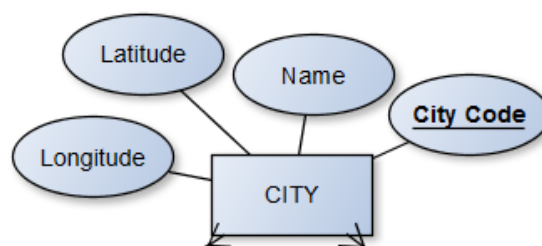
Εικόνα 7 – Σχεδίαση οντότητας SITE

Οι δύο αυτές ιδιότητες συσχετίζονται με την ιδιότητα του **USER**, όπου ο μεν χρήστης διαθέτει μία **ΠΙΣΤΩΤΙΚΗ ΚΑΡΤΑ**, ενώ πολλοί χρήστες μπορούν να κάνουν εγγραφή στο **SITE**.

Ακόμα, κάθε χρήστης εγγράφεται από κάποια χώρα διαμονής. Εδώ, δημιουργείται η οντότητα **ΧΩΡΑΣ** και **ΠΟΛΗΣ**, για τα γεωγραφικά στοιχεία της εφαρμογής. Στην οντότητα **ΧΩΡΑ** διακρίνονται οι ιδιότητες *όνομα χώρας*, *κωδικός χώρας* και για την συγκεκριμένη εφαρμογή, η *σημαία* της χώρας. Ο κωδικός της χώρας θεωρείται μοναδικός, άρα και πρωτεύον κλειδί για την οντότητα. Για την **ΠΟΛΗ**, διακρίνονται οι ιδιότητες *όνομα πόλεως*, *κωδικός πόλεως*, *γεωγραφικό μήκος* και *πλάτος*. Επιλέγεται ο κωδικός της πόλης ως πρωτεύον κλειδί. Σε ότι αφορά την συσχέτισή τους, μία **ΧΩΡΑ** περιλαμβάνει πολλές **ΠΟΛΕΙΣ**, αλλά κάθε **ΠΟΛΗ** ανήκει σε μία και μόνο **ΧΩΡΑ** (σχέση 1:N). Επίσης, ένας **USER** διαμένει σε μία και μόνο **ΧΩΡΑ**, αλλά κάθε **ΧΩΡΑ** έχει πολλούς **USERS** (σχέση 1:N).

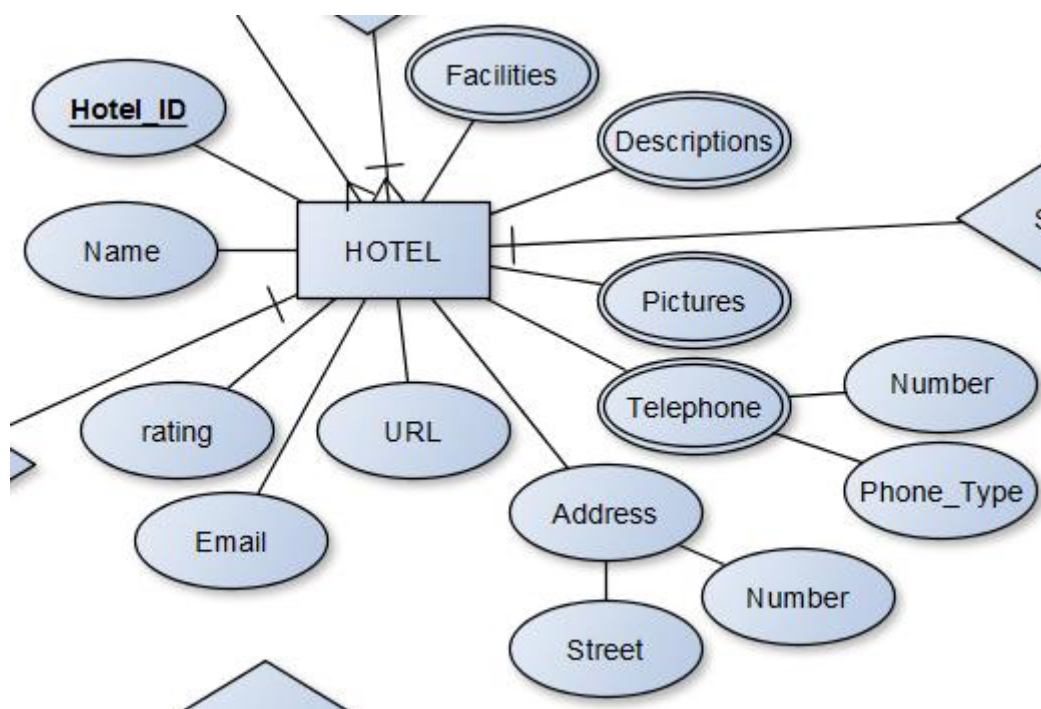


Εικόνα 8 – Σχεδίαση οντότητας COUNTRY



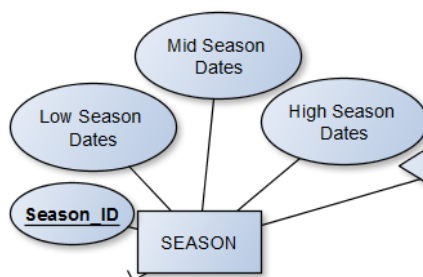
Εικόνα 9 – Σχεδίαση οντότητας CITY

Οι οντότητες αυτές χρησιμοποιούνται και σε ότι αφορά τα **ΞΕΝΟΔΟΧΕΙΑ**, τα οποία με τη σειρά τους ανήκουν σε κάποια **ΠΟΛΗ** μίας χώρας. Η οντότητα **ΞΕΝΟΔΟΧΕΙΟ** διαθέτει τις ιδιότητες *όνομα ξενοδοχείου, κατηγορία ξενοδοχείου, διεύθυνση ξενοδοχείου (σύνθετη ιδιότητα), ηλεκτρονική διεύθυνση ξενοδοχείου (URL), τηλέφωνα ξενοδοχείου (πλειότιμη ιδιότητα), παροχές (πλειότιμη ιδιότητα), περιγραφές (πλειότιμη ιδιότητα) και φωτογραφίες (πλειότιμη ιδιότητα)*. Επίσης, τίθεται ένας μοναδικός κωδικός σε κάθε ξενοδοχείο, ο οποίος είναι και το πρωτεύον κλειδί της οντότητας.



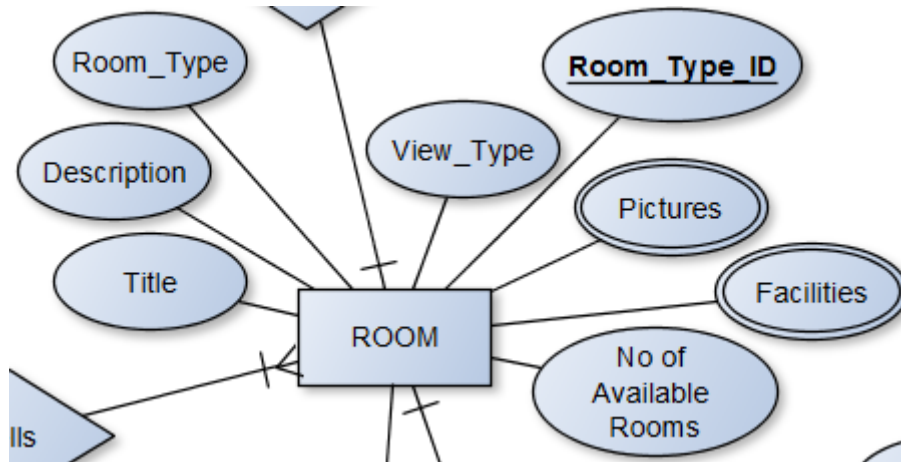
Εικόνα 10 – Σχεδίαση οντότητας HOTEL

Κάθε **ΞΕΝΟΔΟΧΕΙΟ** λειτουργεί σε συγκεκριμένο χρονικό διάστημα, γνωστό ως σεζόν λειτουργίας. Δημιουργείται λοιπόν μία ξεχωριστή οντότητα για τις **ΣΕΖΟΝ**, με ιδιότητες τις *χρονικές περιόδους χαμηλής, μέσης και υψηλής σεζόν*.



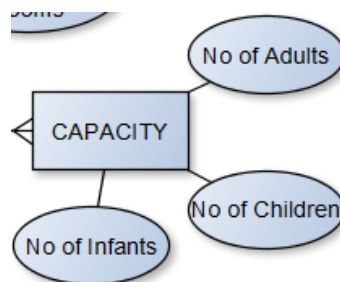
Εικόνα 11 – Σχεδίαση οντότητας SEASON

Κάθε **ΞΕΝΟΔΟΧΕΙΟ** πουλάει ένα αριθμό διαθέσιμων δωματίων, διαφορετικού τύπου και χωρητικότητας. Συνεπώς, διακρίνεται η οντότητα **ΔΩΜΑΤΙΟ**, με ιδιότητες τον *τύπο δωματίου*, το *όνομα του τύπου δωματίου*, την *περιγραφή του τύπου* δωματίου, την *κατηγορία θέας* του, τον *αριθμό διαθέσιμων δωματίων*, τις *παροχές* του δωματίου (πλειότιμη ιδιότητα) και τις *φωτογραφίες* του δωματίου (πλειότιμη ιδιότητα). Στον κάθε τύπο δωματίου αντιστοιχεί ένας μοναδικός κωδικός, ο οποίος λειτουργεί ως πρωτεύον κλειδί για την οντότητα.



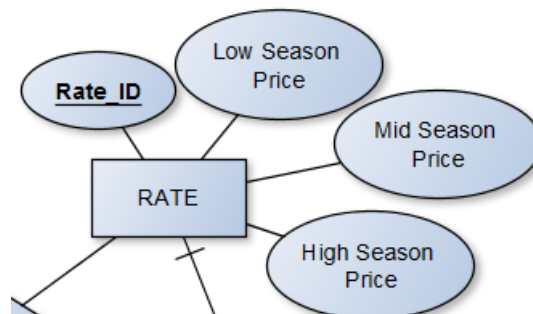
Εικόνα 12 – Σχεδίαση οντότητας ROOM

Παράλληλα, διακρίνεται η οντότητα **ΧΩΡΗΤΙΚΟΤΗΤΑΣ** κάθε δωματίου, με ιδιότητες το *πλήθος ενήλικων*, *παιδιών* και *βρεφών* που μπορούν να διαμείνουν στον κάθε τύπο δωματίου.



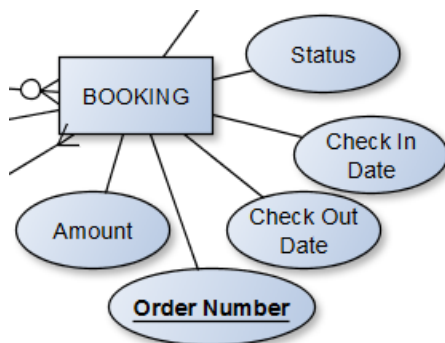
Εικόνα 13 – Σχεδίαση οντότητας CAPACITY

Επίσης, κάθε **ΔΩΜΑΤΙΟ** πωλείται σε διαφορετική **ΤΙΜΗ**, και η οποία επηρεάζεται ανάλογα με τη **ΣΕΖΟΝ** λειτουργίας του ξενοδοχείου. Συνεπώς, διακρίνεται μία νέα οντότητα, αυτή της **ΤΙΜΗΣ** δωματίου, με ιδιότητες τις *τιμές πώλησης* σε χαμηλή, μέση και υψηλή περίοδο.



Εικόνα 14 – Σχεδίαση οντότητας RATE

Τέλος, σε κάθε **ΔΩΜΑΤΙΟ** μπορούν να γίνουν κρατήσεις από τους **USERS**, εφόσον υπάρχει η κατάλληλη διαθεσιμότητα. Επομένως, διακρίνεται και μία τελευταία οντότητα, αυτή της **ΚΡΑΤΗΣΗΣ**, με ιδιότητες τον *κωδικό κράτησης* (που εξυπηρετεί ως πρωτεύον κλειδί της οντότητας), τις *ημερομηνίες άφιξης* και *αναχώρησης* των επισκεπτών, το *συνολικό ποσό πληρωμής* της κράτησης, και το *στάδιο εξέλιξης* της.

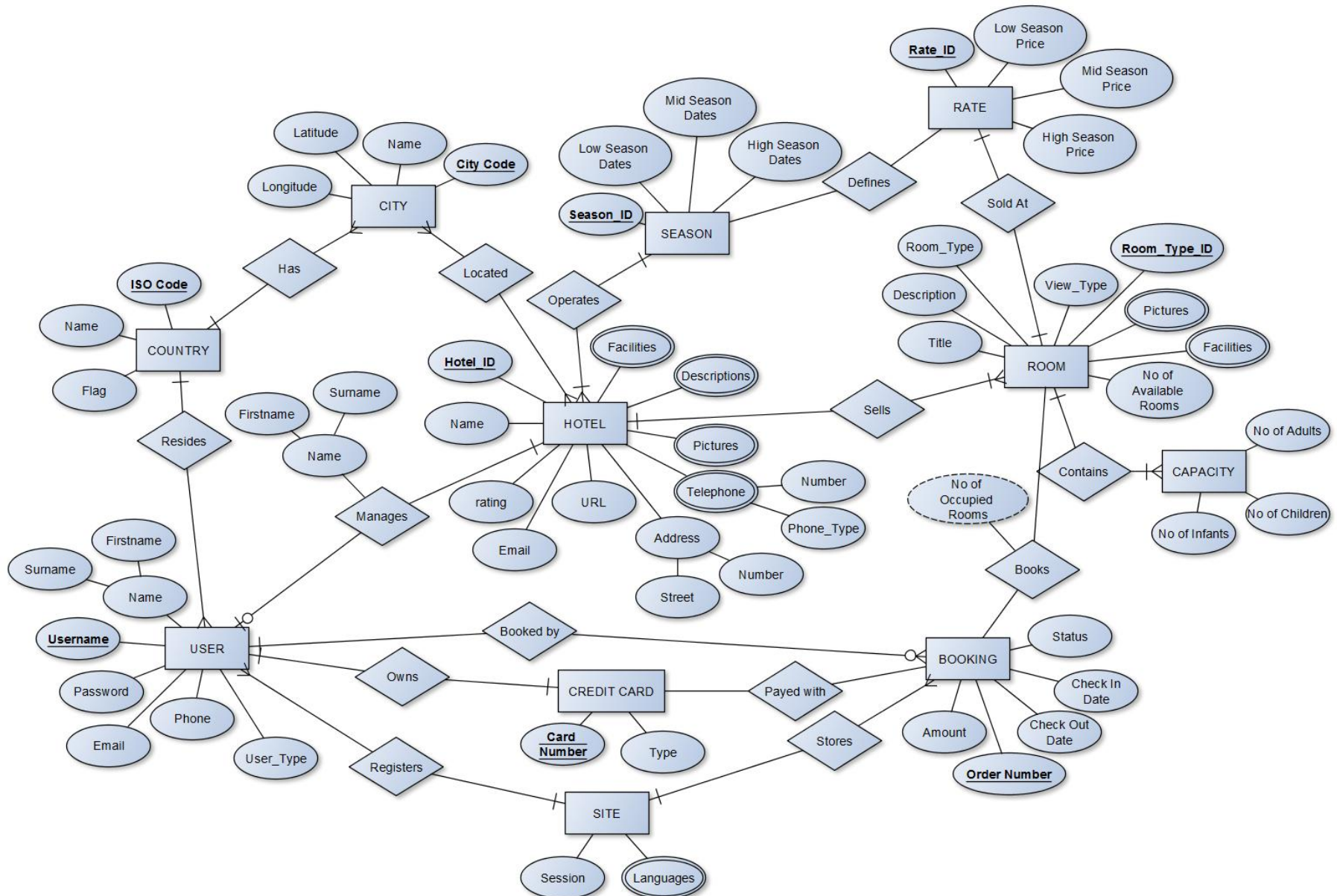


Εικόνα 15 – Σχεδίαση οντότητας BOOKING

Κάθε **ΚΡΑΤΗΣΗ** μπορεί να περιλαμβάνει ένα (1) ή περισσότερα **ΔΩΜΑΤΙΑ**. Κάθε **USER** του συστήματος μπορεί να κάνει μηδέν ή παραπάνω **ΚΡΑΤΗΣΕΙΣ**. Κάθε **ΚΡΑΤΗΣΗ** αποθηκεύεται στο **SITE**, και το **SITE** αποθηκεύει πολλές **ΚΡΑΤΗΣΕΙΣ**. Κάθε **ΚΡΑΤΗΣΗ** πληρώνεται με μία **ΠΙΣΤΩΤΙΚΗ ΚΑΡΤΑ**. Η ίδια **ΠΙΣΤΩΤΙΚΗ ΚΑΡΤΑ** μπορεί να χρησιμοποιηθεί σε πολλές **ΚΡΑΤΗΣΕΙΣ**.

6.3.2 Διάγραμμα Οντοτήτων – Συσχετίσεων Βάσης Δεδομένων

Από την παραπάνω ανάλυση, προκύπτει ένα συνολικό διάγραμμα οντοτήτων – συσχετίσεων, το οποίο απεικονίζεται στην επόμενη σελίδα.

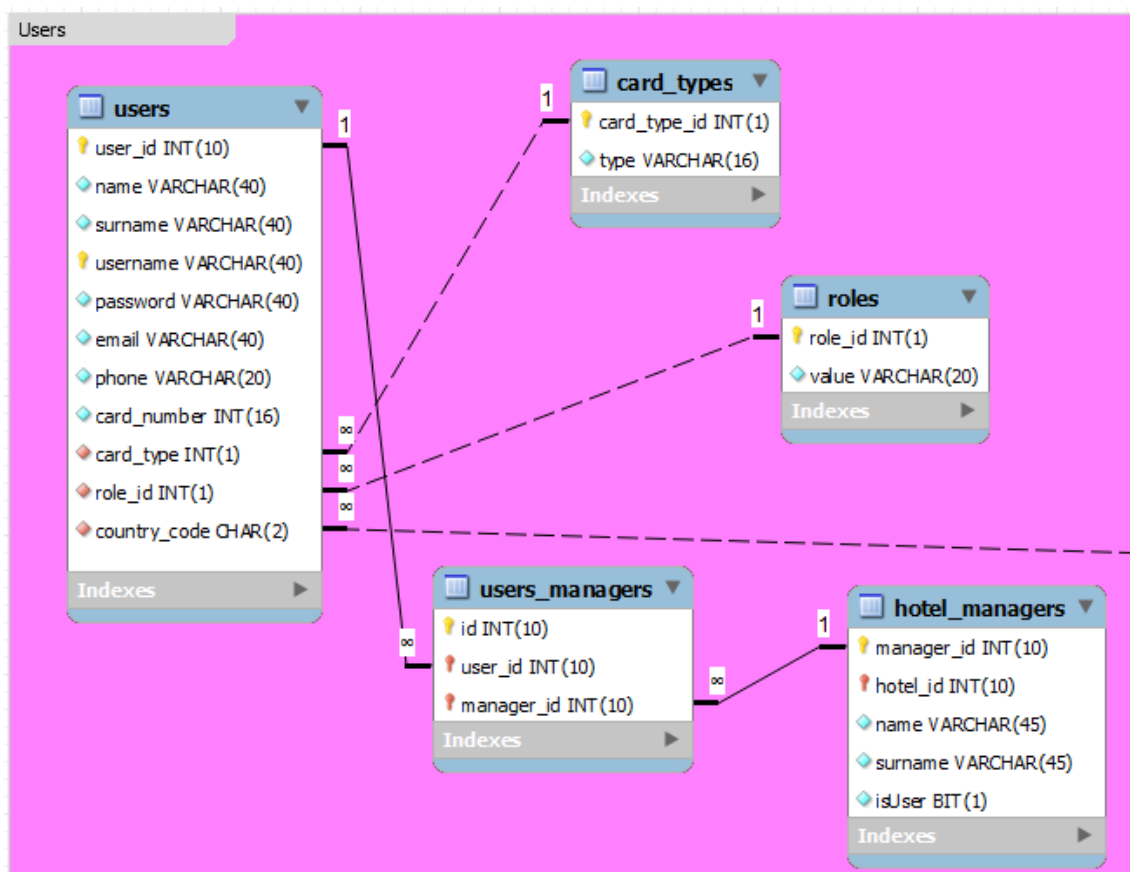


Εικόνα 16 – Καθολικό διάγραμμα οντοτήτων – συσχετίσεων του συστήματος

6.3.3 Σχεσιακό μοντέλο βάσης

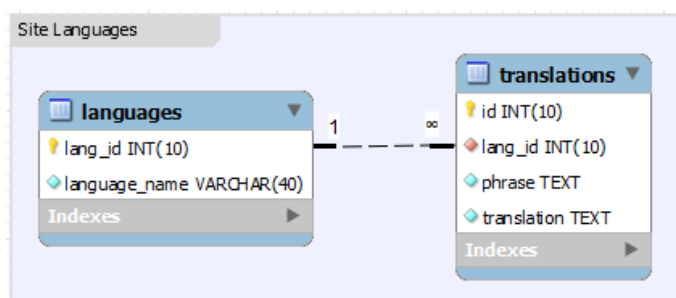
Προκειμένου να κατασκευαστούν οι πίνακες της βάσης δεδομένων, το παραπάνω διάγραμμα οντοτήτων – συσχετίσεων μετατράπηκε στο αντίστοιχο σχεσιακό μοντέλο της βάσης, παρουσιάζοντας ποιοι πίνακες θα δημιουργηθούν, με ποια πεδία και τον τύπο κάθε πεδίου. Να σημειωθεί ότι το παρακάτω σχεσιακό μοντέλο βρίσκεται ήδη σε **Τρίτη Κανονική Μορφή (3NF)**.

Για τους χρήστες του συστήματος, τους διαχειριστές, τους διευθυντές ξενοδοχείων και τις πιστωτικές κάρτες, δημιουργείται το εξής μοντέλο:



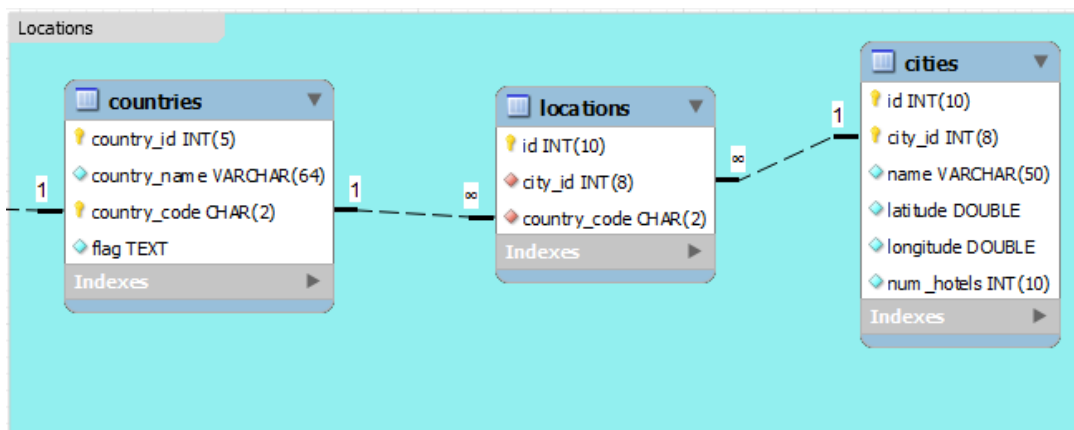
Εικόνα 17 – Σχεσιακό μοντέλο χρηστών

Για τις γλώσσες του site δημιουργείται το εξής μοντέλο:



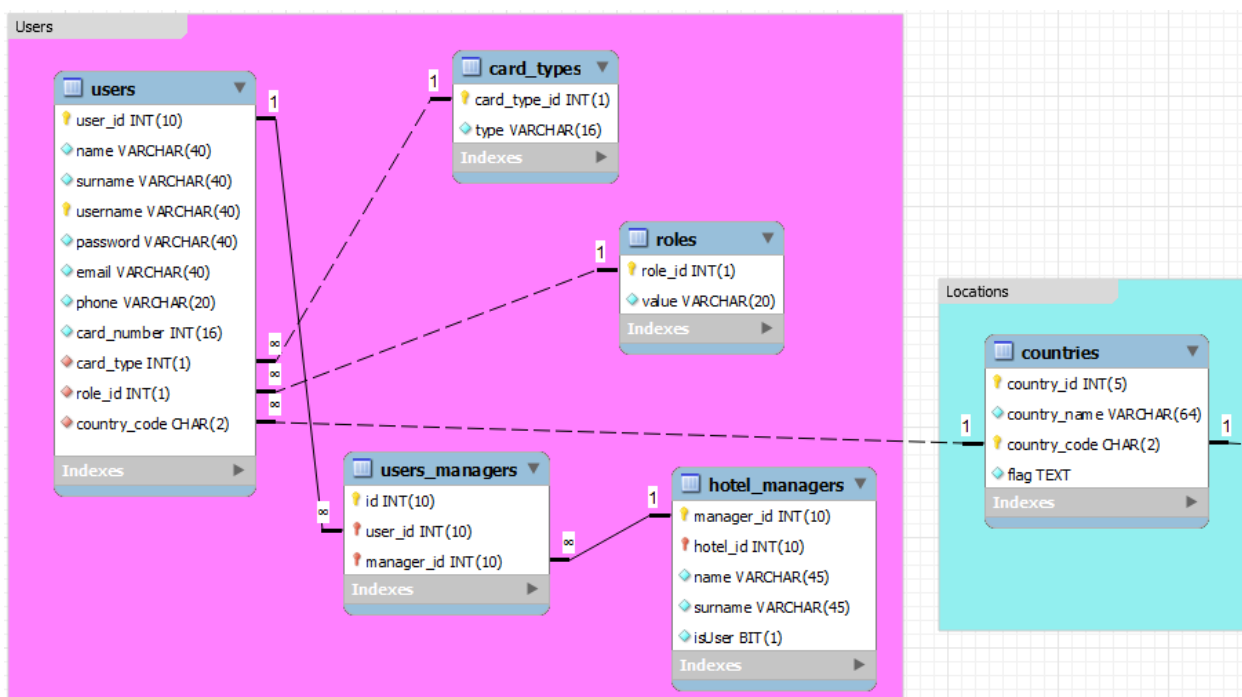
Εικόνα 18 – Σχεσιακό μοντέλο γλωσσών site

Για τις γεωγραφικές περιοχές που σχετίζονται τόσο με τους χρήστες όσο και με τις ξενοδοχειακές μονάδες, σχηματίζεται το εξής μοντέλο:



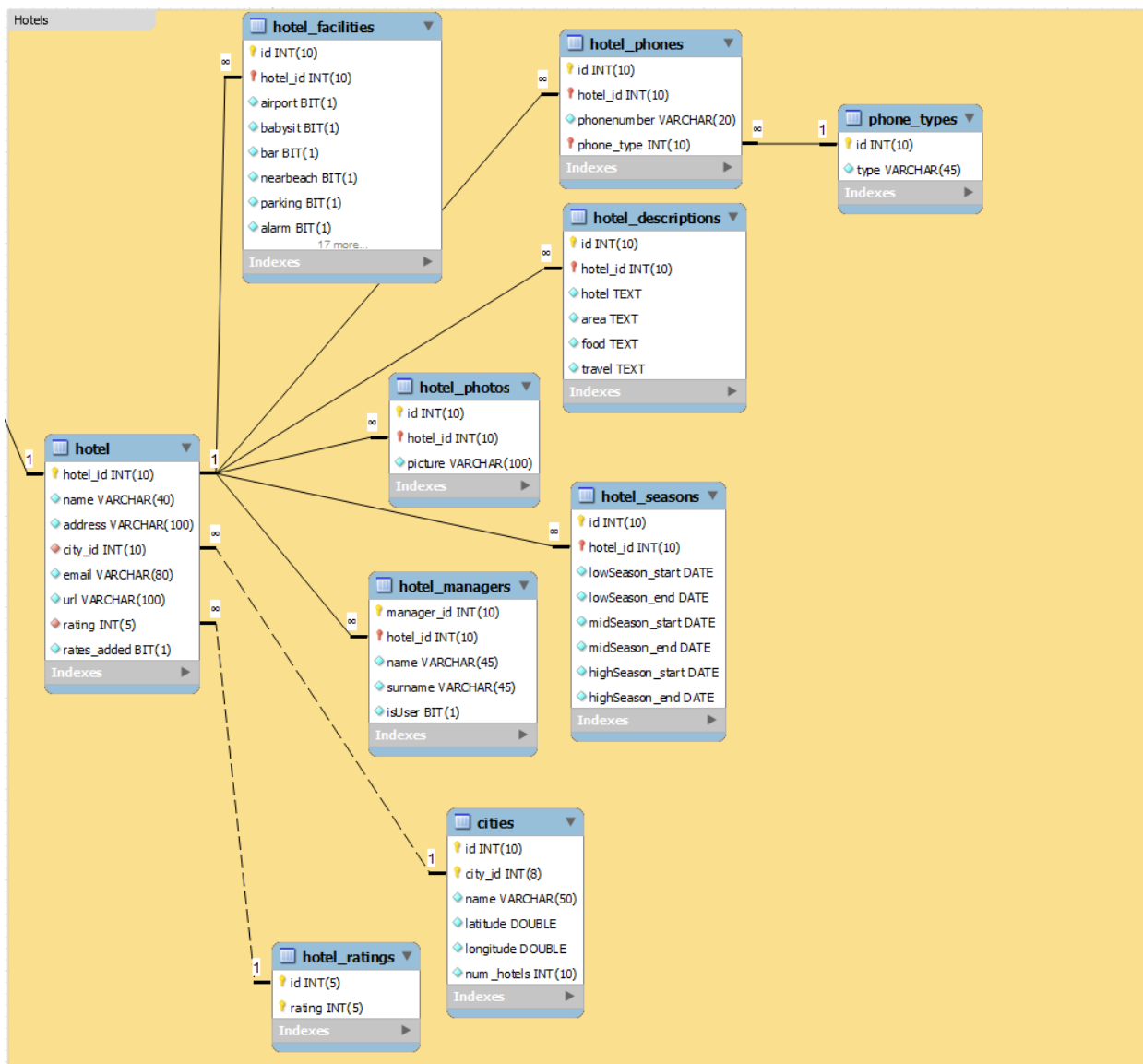
Εικόνα 19 – Σχεσιακό μοντέλο γεωγραφικών τοποθεσιών

Οι πίνακας χρηστών συνδέεται με τον πίνακα χωρών, ως εξής:



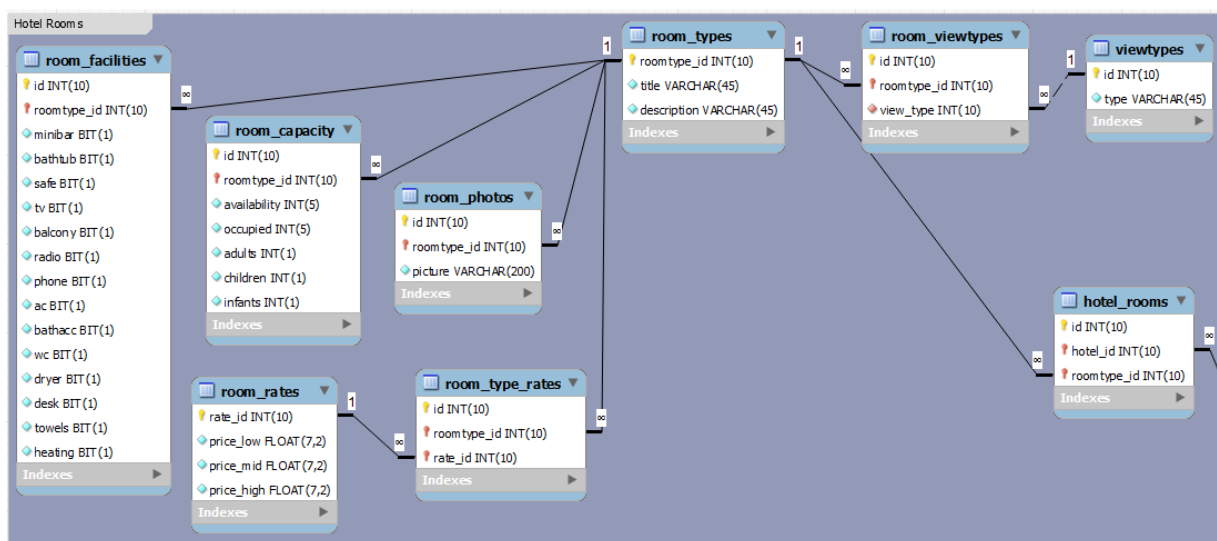
Εικόνα 20 – Απεικόνιση συσχέτισης χωρών και χρηστών

Η ανάλυση των απαραίτητων πινάκων για τα ξενοδοχεία παρουσιάζονται στο επόμενο μοντέλο:



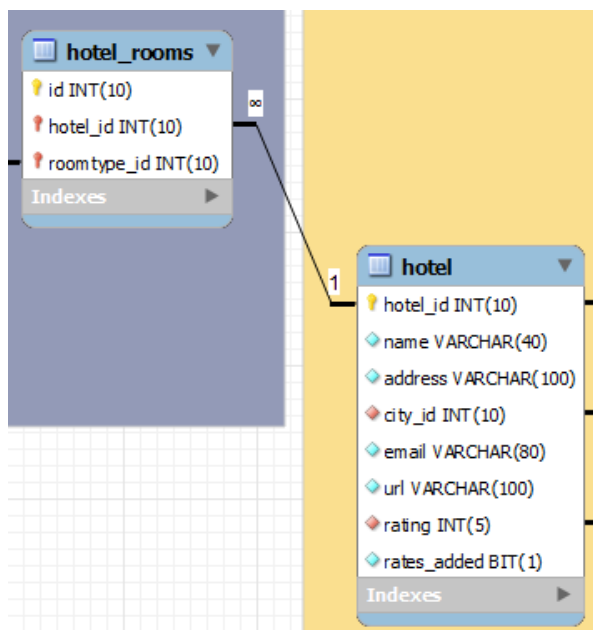
Εικόνα 21 – Σχεσιακό μοντέλο ξενοδοχείου

Αντίστοιχα, το μοντέλο που απεικονίζει τους τύπους δωματίων και τις συσχετίσεις τους είναι:



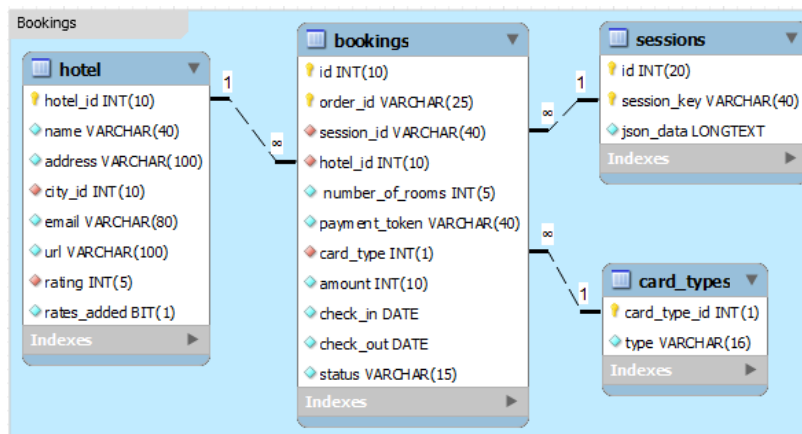
Εικόνα 22 – Σχεσιακό μοντέλο δωματίων ξενοδοχείου

Τα ξενοδοχεία συνδέονται με τα δωμάτιά τους μέσω της εξής σχέσης:



Εικόνα 23 – Απεικόνιση συσχέτισης ξενοδοχείων και δωματίων

Τέλος, ο πίνακας των κρατήσεων και οι συσχετίσεις του με τους σχετικούς πίνακες διαμορφώνεται ως εξής:



Εικόνα 24 – Σχεσιακό μοντέλο κρατήσεων

Στο Παράρτημα Α παρουσιάζονται τα SQL queries δημιουργίας των παραπάνω πινάκων στην βάση MySQL.

6.4 Ανάλυση διαδικασίας κράτησης

Στην ενότητα αυτή παρουσιάζεται αναλυτικά η υλοποίηση του διαδικτυακού συστήματος και η ροή πραγματοποίησης μίας κράτησης δωματίων.

6.4.1 Σελίδα αναζήτησης καταλυμάτων

Η κεντρική σελίδα του συστήματος περιέχει την φόρμα αναζήτησης ξενοδοχειακών καταλυμάτων και των δωματίων τους. Τα απαραίτητα πεδία που πρέπει να συμπληρώσει ο χρήστης είναι η χώρα και πόλη την οποία επιθυμεί να επισκεφτεί, το συγκεκριμένο ξενοδοχείο της πόλης που επέλεξε, τις ημερομηνίες άφιξης και αναχώρησης καθώς και τους επισκέπτες ανά δωμάτιο.

Συγκεκριμένα, ο χρήστης πρέπει αρχικώς να επιλέξει μία από τις διαθέσιμες χώρες του συστήματος. Το drop-down αυτό συμπληρώνεται από τον πίνακα countries μέσω του εξής ερωτήματος:

```
SELECT * FROM countries;
```

Αφού επιλεγεί η χώρα, αποστέλλεται μέσω Ajax request ο κωδικός της επιλεγμένης χώρας στην σελίδα **getcity.php**, η οποία μέσω του επόμενου SQL query επιστρέφει τις πόλεις της χώρας που βρίσκονται στο σύστημα:

```
SELECT cities.city_id, cities.name FROM cities
JOIN locations on cities.city_id = locations.city_id
WHERE country_code='".$_POST['country']."
ORDER BY cities.name;
```

Επιλέγοντας μία από τις διαθέσιμες πόλεις, αποστέλλεται ομοίως μέσω Ajax request ο κωδικός της πόλης στην σελίδα **gethotels.php**, η οποία επιστρέφει όλα τα ξενοδοχεία του συστήματος που βρίσκονται γεωγραφικά στην πόλη αυτή. Το SQL query εύρεσης ξενοδοχείων είναι το εξής:

```
SELECT * FROM hotel WHERE city_id=".$_POST['city']
```

Αμέσως μετά επιλέγονται μέσω javascript calendar οι ημερομηνίες άφιξης και αναχώρησης. Στη συνέχεια, ο πελάτης επιλέγει πόσοι επισκέπτες θα διαμένουν σε κάθε δωμάτιο, καθώς και τον τύπο κάθε επισκέπτη (για παράδειγμα, Δωμάτιο 1: 2 Ενήλικες, 1 παιδί, Δωμάτιο 2, 2 Ενήλικες). Το σύστημα επιτρέπει την κράτηση έως και 5 δωματίων ταυτόχρονα. Η συμπληρωμένη φόρμα φαίνεται ως εξής:

Country:	Greece	
City:	Heraklion	
Hotel:	Galaxy Hotel	
From:	09-09-2014	
To:	13-09-2014	
Room 1	Children	Infants
Adults	1	0
2	1	0
Room 2	Children	Infants
Adults	0	0
2	0	0
Add More	Remove Room	
Search		

Εικόνα 25 – Παράδειγμα συμπληρωμένης φόρμας αναζήτησης

Αφού κάνει τις απαραίτητες επιλογές ο χρήστης, αποστέλλει τα στοιχεία αυτά με POST στην σελίδα αποτελεσμάτων αναζήτησης.

6.4.2 Σελίδα αποτελεσμάτων διαθεσιμότητας

Τα αποτελέσματα της διαθεσιμότητας προκύπτουν από τα στοιχεία αναζήτησης του χρήστη, τα οποία επιστρέφονται μέσω της δυναμικής σελίδας **hotelAvailability.php**.

Στη σελίδα αυτή γίνεται αρχικά ένας έλεγχος στην βάση δεδομένων, και ιδιαίτερα στον πίνακα **room_capacity** για να διαπιστωθεί αν το συγκεκριμένο ξενοδοχείο πληροί τα κριτήρια διαθεσιμότητας. Το SQL query προς τη βάση εκτελείται μία φορά για κάθε επιλεγμένο δωμάτιο, και είναι το εξής:

```
SELECT * FROM room_capacity
JOIN hotel_rooms ON room_capacity.roomtype_id = hotel_rooms.roomtype_id WHERE
hotel_id=".$_req_hotelId." AND (availability > occupied)
AND adults=".$_req_adt[$i]."
AND children>=".$_req_chi[$i]." AND infants>=".$_req_inf[$i];
```


Σε περίπτωση που ικανοποιούνται οι προϋποθέσεις για κάθε επιλογή δωματίου στο συγκεκριμένο ξενοδοχείο, τότε επιστρέφονται τα αποτελέσματα για αυτό το ξενοδοχείο. Αντίθετα, σε περίπτωση αστοχίας έστω και για ένα τύπο δωματίου, θεωρείται ότι το ξενοδοχείο αυτό δεν μπορεί να εξυπηρετήσει τη συγκεκριμένη κράτηση. Σε αυτή την περίπτωση, γίνεται νέα αναζήτηση από την βάση, επιστρέφοντας όλα τα υπόλοιπα ξενοδοχεία που έχουν καταχωρηθεί στην ίδια πόλη.

```
SELECT * FROM hotel
WHERE city_id = ".$req_city."
AND hotel_id <>".$req_hotelId
```

Για κάθε τέτοιο ξενοδοχείο διενεργείται το ίδιο ερώτημα εύρεσης χωρητικότητας ώστε να ελεγχθεί αν το ξενοδοχείο αυτό μπορεί να εξυπηρετήσει τα κριτήρια κράτησης. Σε περίπτωση που δεν καλύπτονται τα κριτήρια, ο αλγόριθμος προχωράει στο επόμενο ξενοδοχείο. Σε περίπτωση που καλύπτονται τα κριτήρια, γίνεται ο έλεγχος κόστους, δηλαδή των οφειλών του συγκεκριμένου ξενοδοχείου (HotelB) ως προς αυτό που επέλεξε ο πελάτης (HotelA).

Ο αλγόριθμος αυτός υπαγορεύει να γίνουν τα εξής:

A = Υπολογισμός του ποσού που χρωστάει το HotelA στο HotelB

B = Υπολογισμός του ποσού που χρωστάει το HotelB στο HotelA

C = A-B

Το κόστος C συγκρίνεται με το αντίστοιχο κόστος που προκύπτει για κάθε άλλη δυάδα ξενοδοχείου που καλύπτει τα κριτήρια σε σχέση με το επιλεγμένο του χρήστη. Αφού ολοκληρωθούν οι επαναλήψεις, το τελικό κόστος FC, είναι το μεγαλύτερο κόστος που βρέθηκε, έχοντας ως σκοπό την μείωση των χρεώσεων μεταξύ των ξενοδοχείων όσο το δυνατό.

Οι οφειλές του ενός ξενοδοχείου προς το άλλο βρίσκονται καταχωρημένες στον πίνακα costs της βάσης δεδομένων. Αναλυτικά, ο κώδικας που υλοποιεί τον παραπάνω αλγόριθμο είναι ο εξής:

```
if (!$isAvailable) {
    $finalHotel = 0;
    $a = 0;
    $b = 0;
    $finalCost = 0;
    $rooms_avail2 = array();
    $query = "SELECT * FROM hotel
            WHERE city_id = ".$req_city."
            AND hotel_id <>".$req_hotelId;
    $result = mysql_query($query);
    if (mysql_num_rows($result) >0) {
        while($hotelrow=mysql_fetch_array($result)) {
            for ($i = 1; $i < 5; $i++) {
                $query2 = "SELECT * FROM room_capacity
                        JOIN hotel_rooms
                        ON room_capacity.roomtype_id =
                        hotel_rooms.roomtype_id
                        WHERE hotel_id=".$hotelrow['hotel_id']."
                        AND (availability > occupied)
                        AND adults=".$req_adt[$i]."
                        AND children>=".$req_chi[$i]."
                        AND infants>=".$req_inf[$i];
                $result2 = mysql_query($query2);
                if (mysql_num_rows($result2) > 0) {
                    $roomtype=mysql_fetch_array($result2);
                    $rooms_avail2[$i] = $roomtype['availability'];
                } else {
                    break; //no matching rooms in hotel - go to next hotel
                }
            }
        }
        if (count($rooms_avail2) == $roomsNum) {
```

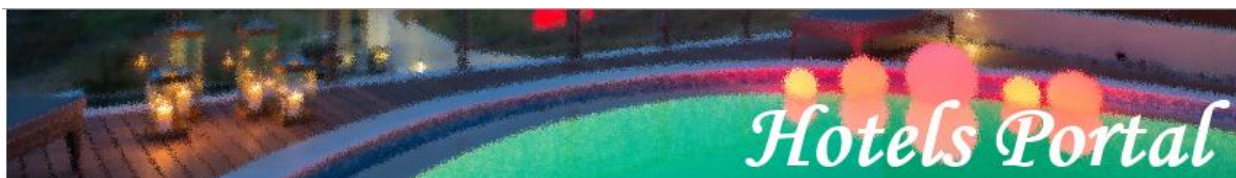
```

$curHotel = $hotelrow['hotel_id'];
//find costings for this hotel and compare
$queryForCosts = "SELECT * FROM costs
                    WHERE hotelA=". $req_hotelId. "
                    AND hotelB=". $curHotel;
$costResult = mysql_query($queryForCosts);
if (mysql_num_rows($costResult) > 0) {
    $costRow = mysql_fetch_array($costResult);
    $h1h2 = $costRow['amount_due'];
}
else{
    $h1h2 = 0;
}
$queryForCosts = "SELECT * FROM costs
                    WHERE hotelB=". $req_hotelId. "
                    AND hotelA=". $curHotel;
$costResult = mysql_query($queryForCosts);
if (mysql_num_rows($costResult) > 0) {
    $costRow = mysql_fetch_array($costResult);
    $h2h1 = $costRow['amount_due'];
}
else{
    $h2h1 = 0;
}
$cost = $h1h2 - $h2h1;
if ($cost > $finalCost) {
    $finalCost = $cost;
    $finalHotel = $curHotel;
}
}
}
if ($finalHotel > 0) {
    $query = "SELECT * FROM hotel WHERE hotel_id = ". $finalHotel;
    $result = mysql_query($query);
    $hotelInfo = mysql_fetch_array($result);
}
else{
    $noMatchFoundInDB = TRUE;
}
}
else{
    $noMatchFoundInDB = TRUE;
}
}

```

Κατά την ολοκλήρωση του παραπάνω τμήματος κώδικα, η μεταβλητή **\$finalCost** διατηρεί την τιμή οφειλής του ξενοδοχείου που καλύπτει τα κριτήρια διαθεσιμότητας σε σχέση με το αρχικά επιλεγμένο, ενώ ο πίνακας **\$hotelInfo** διατηρεί τα στοιχεία του νέου ξενοδοχείου, όπως ανακτήθηκαν από την βάση.

Παράλληλα, ο κώδικας της σελίδας αποθηκεύει προσωρινά στο Session της PHP, τα στοιχεία αναζήτησης όπως τα έλαβε από την κεντρική φόρμα αναζήτησης. Τέλος, επιστρέφεται το HTML markup της σελίδας με τα αποτελέσματα του κατάλληλου ξενοδοχείου, όπως φαίνεται στην παρακάτω εικόνα. Στην συγκεκριμένη εικόνα δεν βρέθηκε το αρχικό ξενοδοχείο που επέλεξε ο χρήστης, αλλά κάποιο άλλο στην ίδια πόλη. Ο χρήστης ενημερώνεται με το κατάλληλο μήνυμα στην αρχή της σελίδας.



The hotel you searched for has no availability. We suggest the following closest matching hotel.

Candia Maris Resort and Spa ★★★★★

Andreas Papandreou Street 72 71414 Heraklion - Gazi - Greece | T: +30 2810 377 000 | F: +30 2810 250 669 | E: resort@candiamaris.gr



The Hotel

The Area

How to arrive

Food

Club Magic Life Candia Maris Imperial is built on the beach of Amoudara, very close to the city center of Heraklion and the International airport. The location is perfect for guests who wish to explore both the East & West coast of the island Crete.

Facilities



Bar



Near Beach



Parking



Alarm Service



Conference Hall



Elevator Available



Entertainment Hall



Gym facilities



In room safe



Internet



Laundry Service



Spa



Swimming pool







Air Conditioning

ROOMS

Εικόνα 26 – Σελίδα αποτελεσμάτων – ξενοδοχείο

ROOMS

Arrival: Tuesday, 09 Sep 2014, Departure: Saturday, 13 Sep 2014 (4 nights)

Room 1 (Adults: 2, Children: 1, Infants: 0)		
<input type="radio"/> Superior Room - Pool View		€ 96.99 / per night = € 387.96
Description	Photos	Facilities
<ul style="list-style-type: none"> Air Conditioning Minibar Bath tub In room Safe Balcony Telephone Bath towels Hair dryer 		
<input type="radio"/> Pool Deluxe Room - Main Building		€ 65.00 / per night = € 260
Description	Photos	Facilities
The Pool Deluxe are located on the ground flo		
<input type="radio"/> Sea View Suite - Main Building		€ 127.99 / per night = € 511.96
Description	Photos	Facilities
 		
<input type="radio"/> Pool Deluxe Suite - Pool View		€ 135.00 / per night = € 540
Description	Photos	Facilities
These luxurious suites are located on the gro		
Room 2 (Adults: 2, Children: 0, Infants: 0)		
<input type="radio"/> Superior Room - Pool View		€ 96.99 / per night = € 387.96
Description	Photos	Facilities
 		
<input type="radio"/> Pool Deluxe Room - Main Building		€ 65.00 / per night = € 260
Description	Photos	Facilities
The Pool Deluxe are located on the ground flo		
<input type="radio"/> Sea View Suite - Main Building		€ 127.99 / per night = € 511.96
Description	Photos	Facilities
The Sea View Suite, spanning 50 sqm, is locat		
<input type="radio"/> Pool Deluxe Suite - Pool View		€ 135.00 / per night = € 540
Description	Photos	Facilities
These luxurious suites are located on the gro		

[Go back](#) [Book](#)

Εικόνα 27 – Σελίδα αποτελεσμάτων - δωμάτια

Στη σελίδα αποτελεσμάτων, ο χρήστης μπορεί να κάνει επισκόπηση των διαθέσιμων δωματίων, για τα οποία παρέχονται οι σχετικές ενημερωτικές πληροφορίες, αλλά και μία γενικότερη επισκόπηση του ξενοδοχείου που βρέθηκε, με τα συνοδευτικά στοιχεία του.

6.4.3 Σελίδα στοιχείων κράτησης

Επιλέγοντας τους τύπους δωματίων που τον ενδιαφέρουν και πατώντας το κουμπί Book, ο χρήστης μεταφέρεται στην σελίδα **booking.php**, προκειμένου να συμπληρώσει τα απαραίτητα στοιχεία της κράτησης. Απαραίτητα στοιχεία θεωρούνται το Όνομα, Επίθετο και γένος κάθε επισκέπτη, καθώς επίσης και τα στοιχεία επικοινωνίας με τον χρήστη που διενεργεί την κράτηση. Ως στοιχεία επικοινωνίας ορίστηκαν το Όνομα και Επίθετο του χρήστη, ένα email επικοινωνίας (στο οποίο θα λάβει και το αντίτυπο κράτησης), η διεύθυνσή του, η χώρα και πόλη διαμονής του και ένα τηλέφωνο επικοινωνίας. Τα στοιχεία επικοινωνίας μπορούν να συμπληρωθούν και αυτόματα για τους εγγεγραμμένους χρήστες του συστήματος, όπως περιγράφεται στην ενότητα 7.2.3.

ROOMS

Arrival: Tuesday, 09 Sep 2014, Departure: Saturday, 13 Sep 2014 (4 nights)

Room 1 (Adults: 2, Children: 1, Infants: 0)			
<input checked="" type="radio"/> Superior Room - Pool View		€ 96.99 / per night = € 387.96	
Adult 1:	Gender: <input type="text"/>	Name: <input type="text"/>	Surname: <input type="text"/>
Adult 2:	Gender: <input type="text"/>	Name: <input type="text"/>	Surname: <input type="text"/>
Child 1:	Gender: <input type="text"/>	Name: <input type="text"/>	Surname: <input type="text"/>
Room 2 (Adults: 2, Children: 0, Infants: 0)			
<input checked="" type="radio"/> Pool Deluxe Suite - Pool View		€ 135.00 / per night = € 540	
Adult 1:	Gender: <input type="text"/>	Name: <input type="text"/>	Surname: <input type="text"/>
Adult 2:	Gender: <input type="text"/>	Name: <input type="text"/>	Surname: <input type="text"/>
Total Price: € 927.96			

Contact Information			
Already a user?	Username: <input type="text"/>	password: <input type="text"/>	<input type="button" value="Sign In"/>
Name: <input type="text"/>	Surname: <input type="text"/>	E-mail: <input type="text"/>	
Address: <input type="text"/>	Country: <input type="text" value="Select"/>	City: <input type="text" value="No Cities Available"/>	
Phone: <input type="text"/>			

Εικόνα 28 – Φόρμα συμπλήρωσης στοιχείων κράτησης

6.4.4 Σελίδα πληρωμής

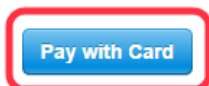
Ο χρήστης μεταφέρεται στην σελίδα πληρωμής αφού συμπληρώσει όλα τα στοιχεία στη σελίδα κράτησης και πατήσει το κουμπί Next. Τα στοιχεία αυτά αποστέλλονται στην σελίδα **payment.php** μέσω POST και καταχωρούνται στο Session του χρήστη που έχει ενεργοποιηθεί προηγουμένως, μαζί με τις υπόλοιπες πληροφορίες που έχουν συλλεχθεί. Τα στοιχεία αυτά διατηρούνται σε μορφή JSON array σύμφωνα με το ID του συγκεκριμένου Session για εύκολη ανάκτηση και αναζήτησή τους. Στο **Παράρτημα Β**, δίνεται ένα παράδειγμα τέτοιου JSON array στο Session.

Στην σελίδα πληρωμής, ο χρήστης μπορεί να κάνει μία τελευταία επισκόπηση των επιλογών του καθώς και των στοιχείων που έχει συμπληρώσει. Αφού ελέγξει τα στοιχεία αυτά μπορεί να προχωρήσει σε πληρωμή του ποσού που αναγράφεται, μέσω πιστωτικής κάρτας.

ROOMS

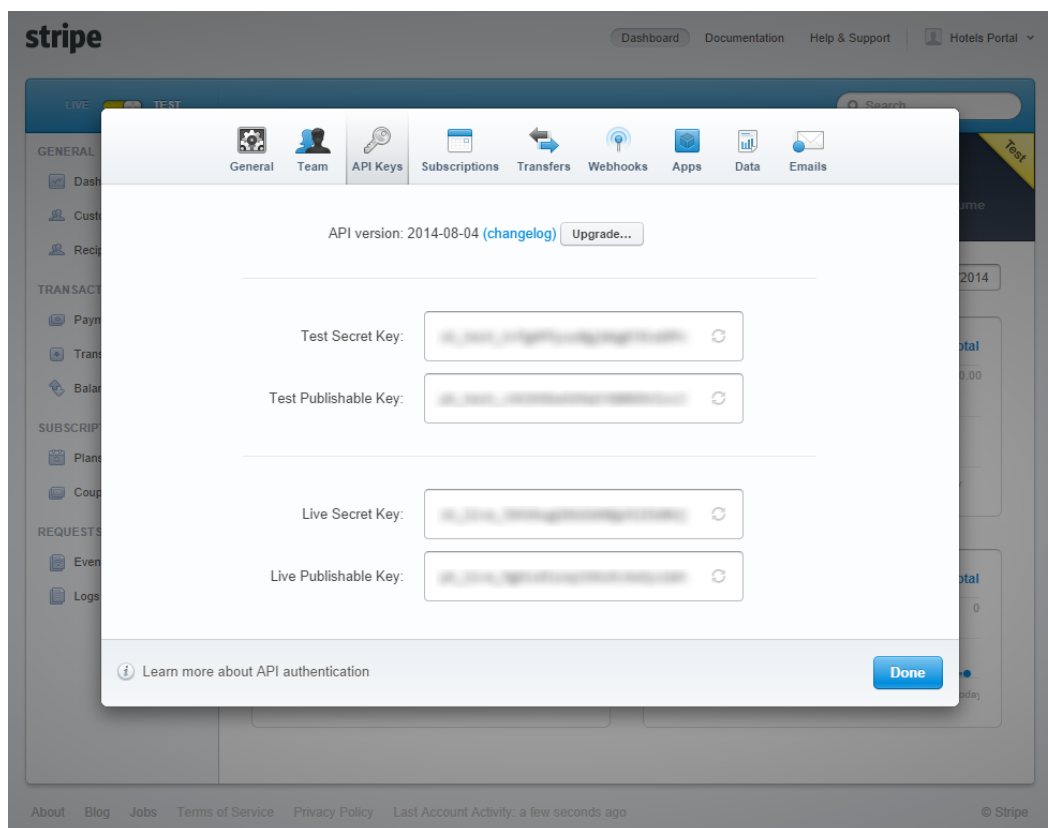
Arrival: Tuesday, 09 Sep 2014, Departure: Saturday, 13 Sep 2014 (4 nights)

Room 1 (Adults: 2, Children: 1, Infants: 0)
● Superior Room - Pool View
Room 2 (Adults: 2, Children: 0, Infants: 0)
● Pool Deluxe Suite - Pool View
Total Price: € 927.96



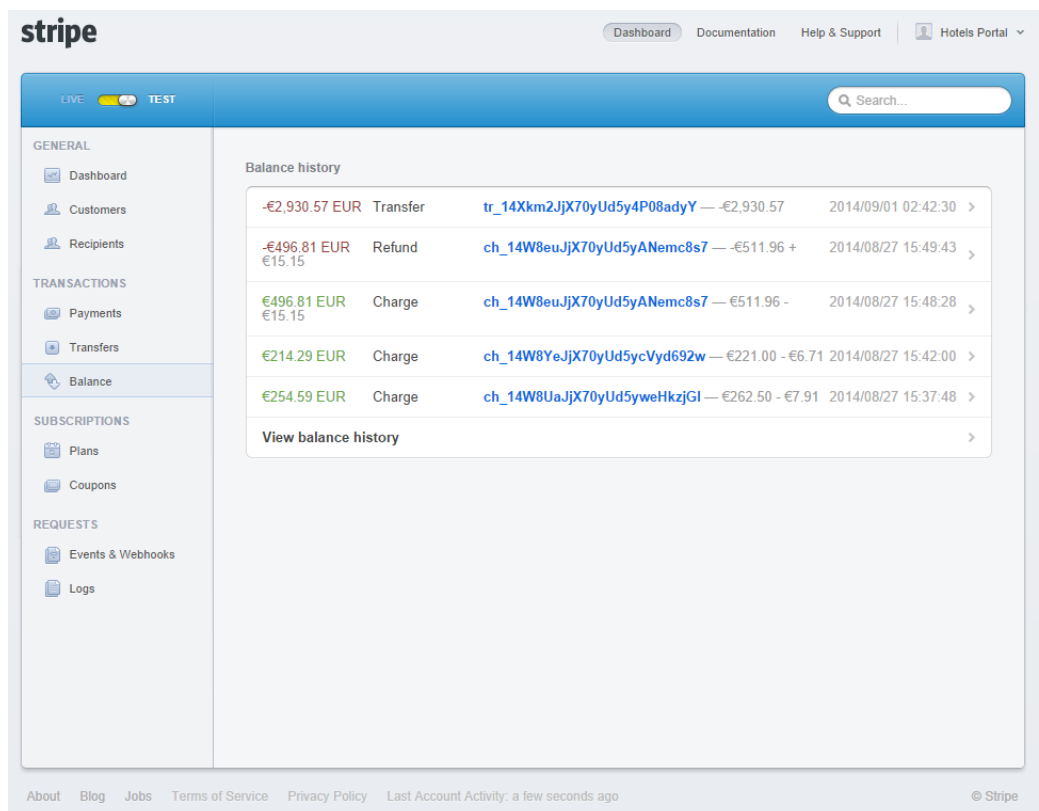
Εικόνα 29 – Σελίδα πληρωμής με πιστ. κάρτα

Για την διαδικασία πληρωμής χρησιμοποιήθηκε το PHP API framework του **Stripe Checkout** [31], το οποίο παρέχει μία ολοκληρωμένη λύση χρέωσης πιστωτικών καρτών σε ένα ή περισσότερα website με μικρή χρέωση προμήθειας για την χρήση του (με παρόμοιο τρόπο λειτουργεί το PayPal). Ουσιαστικά, ο ιδιοκτήτης του εκάστοτε site δημιουργεί ένα λογαριασμό στο Stripe και συμπληρώνει τα στοιχεία του site, όπως το URL του. Βάσει των στοιχείων αυτών, λαμβάνει δύο κλειδιά πιστοποίησης με το σύστημα Stripe. Τα δύο αυτά κλειδιά, μυστικό και δημόσιο, χρησιμοποιούνται για την δημιουργία ενός SSL καναλιού επικοινωνίας μεταξύ site και Stripe, μέσω του οποίου εκτελείται η διαδικασία χρέωσης της κάρτας. Στο διαχειριστικό πάνελ του Stripe μπορεί ο developer ή διαχειριστής να δει τα κλειδιά που οφείλει να χρησιμοποιήσει τόσο στο δοκιμαστικό περιβάλλον ανάπτυξης, όσο και κατά την live λειτουργία του εκάστοτε site.



Εικόνα 30 – Κλειδιά επικοινωνίας Stripe

Επίσης, το διαχειριστικό σύστημα του Stripe προσφέρει ανάλυση όλων των χρεώσεων (πληρωμές, ακυρώσεις) που έχουν εκτελεστεί με το συγκεκριμένο λογαριασμό.

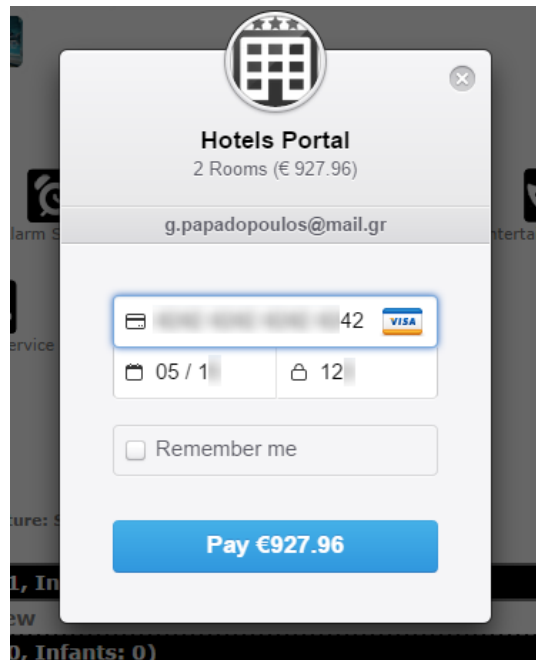


Εικόνα 31 – Ανάλυση χρεώσεων μέσω Stripe

Για την προσθήκη της διεπαφής του Stripe στην σελίδα πληρωμής, χρησιμοποιείται ο παρακάτω Javascript κώδικας εντός μίας απλής φόρμας, όπως αναφέρεται στο API του framework:

```
<form action="" method="POST">
  <script src="https://checkout.stripe.com/checkout.js"
    class="stripe-button"
    data-key="pk_test_vVKSh5boK0MaIYG8R0hVlxxJ"
    data-amount="<?php echo $totalPrice*100?>"
    data-currency="EUR"
    data-name="Hotels Portal"
    data-email="<?php echo $session['contactDetails']['mail']?>"
    data-image="img/city-hotel-icon.png">
  </script>
</form>
```

Πατώντας το κουμπί πληρωμής του Stripe, εμφανίζεται η παρακάτω φόρμα, στην οποία συμπληρώνει ο χρήστης τα στοιχεία της πιστωτικής κάρτας με την οποία επιθυμεί να κάνει την πληρωμή.



Εικόνα 32 – Φόρμα πληρωμής Stripe

Τα στοιχεία αυτά αποστέλλονται κρυπτογραφημένα μέσω SSL στο σύστημα του Stripe, το οποίο ελέγχει τη πιστότητά τους, και επιστρέφει ένα μοναδικό *token* αναγνώρισης. Στο σημείο αυτό δεν έχει πραγματοποιηθεί η χρέωση. Το Stripe token επιστρέφεται στον web server της εφαρμογής για να πραγματοποιήσει ο Server την χρέωση στο back-end.

Στο σημείο αυτό, εκτελείται ο αλγόριθμος της κράτησης με τα εξής βήματα:

1. Έλεγχος διαθεσιμότητας των επιλεγμένων δωματίων ξανά. Υπάρχει πιθανότητα στο χρονικό διάστημα που μεσολαβεί για την συμπλήρωση των στοιχείων κράτησης από τον χρήστη, κάποιος άλλος χρήστης της εφαρμογής που έχει συνδεθεί παράλληλα, να έχει ήδη κλείσει τα δωμάτια. Συνεπώς, πρέπει να ελέγξουμε ότι υπάρχει ακόμα διαθεσιμότητα, ειδάλλως να μην προχωρήσει το σύστημα σε μία λανθασμένη χρέωση κάρτας. Ο έλεγχος αυτός πραγματοποιείται με το εξής ερώτημα στην βάση, για κάθε επιλεγμένο δωμάτιο:

```
SELECT * FROM room_capacity
WHERE roomtype_id= ". $selRoom[' roomType' ]. "
AND adults=" . $selRoom[' adt' ][' num' ] . "
AND children>=" . $selRoom[' chi' ][' num' ]. "
AND infants>=" . $selRoom[' inf' ][' num' ];
```

2. Χρέωση πιστωτικής κάρτας μέσω Stripe. Αυτό γίνεται μέσα από το API του Stripe με τον εξής κώδικα PHP:

```
require_once(' includes/lib/Stripe.php' );
Stripe::setApiKey("*****DPn");
$charge = Stripe_Charge::create(array(
    "amount" => $totalAmount, // amount in cents
    "currency" => "eur",
    "card" => $token,
    "description" => $_POST[' stripeEmail' ])
);
```

3. Ανανέωση πίνακα χωρητικότητας δωματίων **room_capacity**, προσθέτοντας τα κατειλημμένα δωμάτια στο κελί occupied κάθε τύπου δωματίου.


```
UPDATE room_capacity
SET occupied = (occupied + 1)
WHERE roomtype_id = ". $selRoom[' roomType' ]. " AND
adults = ". $selRoom[' adt' ][' num' ]. "
AND children >= ". $selRoom[' chi' ][' num' ]. "
AND infants >= ". $selRoom[' inf' ][' num' ] ;
```

4. Προσθήκη μίας εγγραφής κράτησης στον πίνακα **bookings** με *status* "complete".

```
INSERT INTO bookings VALUES (
NULL, ". $orderID. ", ". $session[' guid' ]. ", $selectedHotelId, "
count($session[' rooms' ]). ", ". $charge[' id' ]. " " "
. ", ". $cardtype[' card_type_id' ]. ", $totalAmount,
STR_TO_DATE( ". $session[' dateFrom' ]. ", ' %d-%m-%Y' ), "
STR_TO_DATE( ". $session[' dateTo' ]. ", ' %d-%m-%Y' ), ' complete' );
```

5. Σε περίπτωση όπου έγινε επιλογή άλλου ξενοδοχείου από αυτό της αρχικής επιλογής του χρήστη, τότε ανανεώνεται ο πίνακας **costs**, χρεώνοντας το 10% της συγκεκριμένης κράτησης στο πρώτο ξενοδοχείο.

```
UPDATE costs
SET amount_due = ". $amountDue. "
WHERE hotelA = ". $session[' finalHotel' ]. "
AND hotelB = ". $session[' reqHotel' ]
```

Αφού ολοκληρωθούν τα παραπάνω βήματα επιτυχώς, ανανεώνεται το Session του χρήστη με τις περαιτέρω πληροφορίες πληρωμής, και μεταφέρεται ο χρήστης στην σελίδα επιβεβαίωσης.

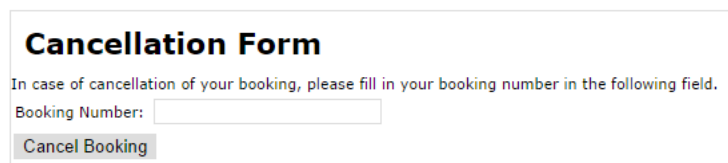
6.4.5 Σελίδα επιβεβαίωσης και email

Το τελικό στάδιο της κράτησης είναι η επιβεβαίωση του πελάτη και η επισκόπηση των τελικών στοιχείων της κράτησης. Η σελίδα **confirmation.php** εκτελεί αυτόν ακριβώς το σκοπό, εξάγοντας από τις μεταβλητές του Session array του χρήστη τα απαραίτητα στοιχεία, τα οποία σε συνδυασμό με ερωτήματα ανάκτησης επιπλέον στοιχείων περιγραφής από την βάση (SELECT statements) παράγει το τελικό HTML markup επιβεβαίωσης του χρήστη. Παράλληλα, απεικονίζεται στην σελίδα ο **μοναδικός κωδικός κράτησης (booking number ή PNR)** μέσω του οποίου γίνεται όποια αναφορά στη κράτηση (σε περίπτωση για παράδειγμα αλλαγής, ακύρωσης ή επικοινωνίας με το ξενοδοχείο).

Ταυτόχρονα, αποστέλλεται το ίδιο HTML μέσω email στην ηλεκτρονική διεύθυνση επικοινωνίας του χρήστη, ώστε να διατηρήσει ένα αντίτυπο της ηλεκτρονικής του κράτησης. Η αποστολή email γίνεται μέσω του PHP framework **PHPMailer** [32], το οποίο προσφέρει μία ολοκληρωμένη λύση στην παραμετροποίηση SMTP mail εφαρμογών και την αποστολή των σχετικών email από αυτές.

6.4.6 Σελίδα ακύρωσης κράτησης

Στην περίπτωση όπου ο χρήστης που έκανε την κράτηση επιθυμεί την ακύρωσή της για τον οποιονδήποτε λόγο, μπορεί να προβεί σε ακύρωση από την κεντρική σελίδα της εφαρμογής. Στη σελίδα αυτή υπάρχει η κατάλληλη φόρμα, με μοναδικό πεδίο που πρέπει να συμπληρωθεί, αυτό του κωδικού της συγκεκριμένης κράτησης.



Εικόνα 33 – Φόρμα ακυρώσεων

Συμπληρώνοντας το πεδίο αυτό, ο χρήστης μεταφέρεται στην δυναμική σελίδα **cancelation.php**, στην οποία αντιστρέφονται οι διαδικασίες που εκτελέστηκαν στη σελίδα πληρωμής. Συγκεκριμένα, γίνεται πίστωση της κάρτας με τον εξής PHP κώδικα του Stripe API:

```
$ch = Stripe_Charge::retrieve($token);  
$re = $ch->refunds->create();
```

Εφόσον ακυρωθεί η χρέωση, μειώνονται τα κατειλημμένα δωμάτια στη βάση, σύμφωνα με τον αριθμό και τον τύπο δωματίων της κράτησης. Αμέσως μετά, τίθεται το *status* της συγκεκριμένης κράτησης ως “*cancelled*”, στην αντίστοιχη εγγραφή του πίνακα **bookings** της βάσης. Στη συνέχεια, μειώνεται κατά 10% το χρέος του ξενοδοχείου για το οποίο έγινε η κράτηση, σε σχέση με το αρχικά επιλεγμένο ξενοδοχείο, αν και εφόσον υπάρχει τέτοιο χρέος.

Αφού ολοκληρωθούν τα συγκεκριμένα βήματα, επιστρέφεται κατάλληλο HTML markup στη σελίδα του χρήστη, ενημερώνοντας τον ότι η κράτηση ακυρώθηκε. Ταυτόχρονα, αποστέλλεται και σε αυτή τη περίπτωση ένα σχετικό email ενημέρωσης στην ηλεκτρονική διεύθυνση του χρήστη, μέσω του PHPMailer.

7 Χρήστες συστήματος

7.1 Είδη χρηστών

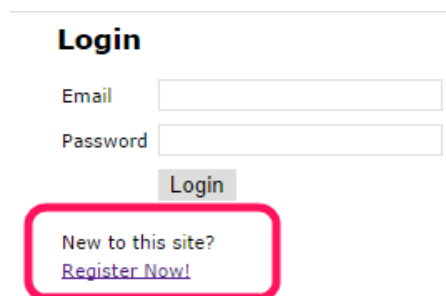
Στο σύστημα που υλοποιήθηκε μπορούν να εγγραφούν δύο διακριτές κατηγορίες χρηστών, αναλόγως τα δικαιώματα επέμβασης που διαθέτουν πάνω στο σύστημα. Οι μεν **απλοί χρήστες (users)** έχουν το δικαίωμα εγγραφής στο σύστημα (registration), συμπληρώνοντας τα απαραίτητα στοιχεία που ζητούνται, εισόδου (login) και εξόδου (logout) ως εγγεγραμμένοι χρήστες στο front-end της εφαρμογής, καθώς επίσης και ανάκτησης των στοιχείων τους στην σελίδα συμπλήρωσης στοιχείων επισκεπτών, για αυτόματη συμπλήρωση των πεδίων.

Συμπληρωματικά με τα δικαιώματα των απλών χρηστών, η κατηγορία των **διαχειριστών (administrators)** έχει επιπλέον δυνατότητες διαχείρισης των στοιχείων του συστήματος, έχοντας πρόσβαση στο back-end της πλατφόρμας. Τα δικαιώματα αυτά παρουσιάζονται αναλυτικά παρακάτω, ανά βαθμό πρόσβασης.

7.2 Απλοί χρήστες

7.2.1 Εγγραφή νέου χρήστη

Στην κεντρική σελίδα της εφαρμογής, υπάρχει η περιοχή εισόδου χρηστών. Στην περίπτωση όπου ένας χρήστης δεν έχει κάνει εγγραφή στον site, υπάρχει η δυνατότητα εγγραφής του από τον σχετικό σύνδεσμο.



The image shows a login form with the following elements:

- Title: **Login**
- Input field: Email
- Input field: Password
- Button: Login
- Link: [New to this site? Register Now!](#) (highlighted with a red box)

Εικόνα 34 – Σημείο εγγραφής χρήστη στη κεντρική σελίδα

Στην περίπτωση όπου επιλέξει να κάνει εγγραφή, ο χρήστης μεταφέρεται στη δυναμική ιστοσελίδα **registeruser.php**, ώστε να συμπληρώσει τα απαραίτητα στοιχεία του. Απαραίτητα στοιχεία εγγραφής ενός νέου χρήστη στο σύστημα θεωρούνται το Ονοματεπώνυμό του, το τηλέφωνο επικοινωνίας, η χώρα διαμονής του, ένα email επικοινωνίας, τα στοιχεία της πιστωτικής του κάρτας (αριθμός και τύπος), όπως επίσης ένα μοναδικό username και κωδικός εισόδου στο σύστημα.

Add User

Name

Surname

Phone

Country

Username

Password

Repeat Password

Email

Credit Card Number

Credit Card Type

Εικόνα 35 – Φόρμα εγγραφής νέου χρήστη

Add User

Name

Surname

Phone

Country

Username

Password

Repeat Password

Email

Credit Card Number

Credit Card Type

Εικόνα 36 – Φόρμα εγγραφής με συμπληρωμένα πεδία

Ο έλεγχος ακεραιότητας και ύπαρξης τιμής των πεδίων γίνεται αρχικά στη πλευρά του χρήστη μέσω Javascript. Αμέσως μετά τον έλεγχο αυτό και εφόσον τα στοιχεία είναι συμπληρωμένα, αποστέλλονται μέσω της φόρμας στο back-end της εφαρμογής. Σε αυτό το σημείο τα στοιχεία ελέγχονται ξανά σε επίπεδο PHP, ώστε να υπάρχει ακεραιότητα των στοιχείων πριν την αποθήκευσή τους στην βάση δεδομένων. Αμέσως μετά, ελέγχεται αν ο χρήστης έχει ήδη εγγραφεί, μέσω της μεθόδου `check_user_email`. Η μέθοδος αυτή εκτελεί ένα SQL ερώτημα στην βάση, και συγκεκριμένα στον πίνακα `users`.

```
function check_user_email($user,$email) {
    $query="SELECT * FROM users WHERE username='".$user.'" OR email='".$email.'";
    $result=mysql_query($query);
    if(mysql_num_rows($result)==0)
        return true;
    else
        return false;
}
```

Στην περίπτωση όπου δεν βρεθεί εγγραφή με το ίδιο username ή email, τότε το σύστημα μπορεί να προχωρήσει στην εγγραφή του νέου αυτού χρήστη.

```
$query = "INSERT INTO users VALUES (NULL, '".$_POST['name']. "', '".$_POST['surname']. "',
'".$_POST['username']. "', '".$_POST['password']. "',
'".$_POST['email']. "', '".$_POST['phone']. "', '".$_POST['ccard']. "',
'".$_POST['ccardtype']. "', 2, '".$_POST['country']. "')";
```

Όπως φαίνεται από το SQL ερώτημα, καταχωρούνται στον πίνακα `users` τα στοιχεία του χρήστη, όπως φαίνεται παρακάτω:

1 • `SELECT * FROM ihotelier.users;`

user_id	name	surname	username	password	email	phone	card_number	card_type	role_id	country_code
1								1	2	GR
2								1	1	GR
3								1	1	GR
11	George	Papadopoulos	gpapadop		g.papadopoulos@mail.gr	+30 210 1234 123		1	2	GR

Εικόνα 37 – Πίνακας βάσης καταχωρημένων χρηστών

Στην περίπτωση επιτυχούς αποθήκευσης, ο χρήστης ενημερώνεται με το κατάλληλο μήνυμα στη σελίδα:

Add User

Name	<input type="text"/>
Surname	<input type="text"/>
Phone	<input type="text"/>
Country	<input type="text" value="Afghanistan"/>
Username	<input type="text"/>
Password	<input type="text"/>
Repeat Password	<input type="text"/>
Email	<input type="text"/>
Credit Card Number	<input type="text"/>
Credit Card Type	<input type="text" value="VISA"/>

User added succesfully!

Save Cancel

Εικόνα 38 – Μήνυμα επιβεβαίωσης εγγραφής χρήστη

7.2.2 Είσοδος/Εξοδος χρήστη

Η φόρμα εισόδου χρηστών βρίσκεται στην κεντρική σελίδα της εφαρμογής. Για την είσοδο ενός χρήστη απαιτείται η συμπλήρωση του email επικοινωνίας που έχει δηλώσει κατά την εγγραφή του καθώς και ο μοναδικός του κωδικός (password). Τα στοιχεία αυτά στέλνονται στη συνέχεια μέσω ασύγχρονης επικοινωνίας Ajax στην σελίδα `loginmain.php`. Στη σελίδα αυτή εκτελείται το παρακάτω SQL query στη βάση, και συγκεκριμένα στον πίνακα users, ώστε να βρεθεί αν ο χρήστης είναι ήδη εγγεγραμμένος στο σύστημα.

```
SELECT * FROM users  
WHERE email='".$_POST['email']."' ;  
AND password='".$_POST['password']."' ;
```

Στην περίπτωση όπου είναι εγγεγραμμένος, τότε η σελίδα επιστρέφει HTML κώδικα με το όνομα και το επίθετο του χρήστη, καλωσορίζοντάς τον στον site. Σε αντίθετη περίπτωση, επιστρέφεται μήνυμα ενημέρωσης του χρήστη ότι δεν βρέθηκε η εγγραφή του στο σύστημα.

Login

Email	<input type="text"/>
Password	<input type="text"/>
	<input type="button" value="Login"/>

Wrong email or password.

Εικόνα 39 – Μήνυμα σφάλματος εισόδου χρήστη στο σύστημα

Παράλληλα, τίθεται μία μεταβλητή του PHP Session με όνομα “email” ίση με το email του χρήστη που έχει κάνει είσοδο.

```
$_SESSION['email']=$_POST['email'];
```

Αυτή η μεταβλητή παραμένει ενεργή καθόλη τη διάρκεια που χρησιμοποιεί ο χρήστης τις σελίδες κρατήσεων και είναι προσβάσιμη από όλες τις σελίδες του site (μέσω PHP).

Για την έξοδο του χρήστη από το σύστημα, πατώντας στο σύνδεσμο logout, απενεργοποιείται η μεταβλητή που είχε προηγουμένως οριστεί στο PHP Session με όνομα "email". Συνεπώς, κάθε άλλη σελίδα μπορεί να ελέγξει την ύπαρξη ή όχι της μεταβλητής αυτής, για να διακρίνει αν υπάρχει ενεργός χρήστης που έχει κάνει είσοδο στο σύστημα.

```
unset($_SESSION['email']);
```

7.2.3 Αυτόματη συμπλήρωση στοιχείων

Οι εγγεγραμμένοι χρήστες του συστήματος έχουν το πλεονέκτημα του ότι εφόσον τα στοιχεία τους είναι ήδη καταχωρημένα στην βάση του συστήματος, δεν χρειάζεται να τα συμπληρώσουν χειροκίνητα εκ νέου κατά τη διαδικασία της κράτησης, εξυπηρετώντας έτσι στη μείωση του χρόνου που απαιτείται για την ολοκλήρωση μίας κράτησης καθώς και στην ευχρηστία της εφαρμογής, διευκολύνοντας όσο περισσότερο γίνεται τον τελικό χρήστη.

Η εξής λειτουργία επιτυγχάνεται στη σελίδα συμπλήρωσης στοιχείων των επισκεπτών **booking.php**, μέσω των προαιρετικών πεδίων συμπλήρωσης του username και password του χρήστη. Πατώντας το κουμπί "Sign In", ταυτοποιούνται τα στοιχεία αυτά βάσει των στοιχείων του πίνακα users στη βάση. Σε προγραμματιστικό επίπεδο, αποστέλλονται τα στοιχεία πρόσβασης μέσω ajax request στη δυναμική σελίδα **checkUserExists.php**. Στη σελίδα αυτή εκτελείται το εξής SQL query προς τη βάση:

```
SELECT * FROM users
WHERE username='".$_POST['user']."'
AND password='".$_POST['pass']."' ;
```

Αν πράγματι ταυτοποιηθεί η ύπαρξη του χρήστη στο σύστημα, επιστρέφονται μέσω JSON object και συμπληρώνονται τα διαθέσιμα στοιχεία στην φόρμα επικοινωνίας της κράτησης.

```
[
  {
    "user_id": "11",
    "name": "George",
    "surname": "Papadopoulos",
    "username": "gpapadop",
    "password": "*****",
    "email": "g.papadopoulos@mail.gr",
    "phone": "+30 210 1234 123",
    "card_number": "*****",
    "card_type": "1",
    "role_id": "2",
    "country_code": "GR"
  }
]
```

Contact Information			
Already a user?	Username: gpapadop	password: *****	Sign In
Name: George	Surname: Papadopoulos	E-mail: g.papadopoulos@mail.gr	
Address:	Country: Greece	City: Alexandroupolis	
Phone: +30 210 1234 123			

Εικόνα 40 – Φόρμα επικοινωνίας με αυτομάτως συμπληρωμένα στοιχεία

7.3 Διαχειριστές

7.3.1 Δικαιώματα διαχειριστών

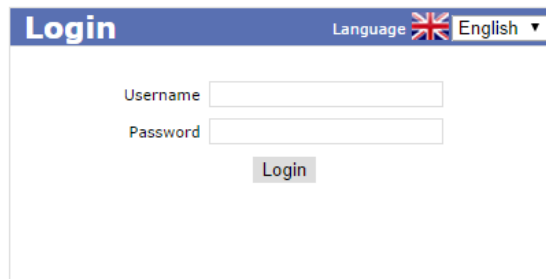
Η κατηγορία των διαχειριστών διαθέτει επιπλέον δικαιώματα από τους απλούς χρήστες. Στο front-end της εφαρμογής, δεν υπάρχει ουσιώδης διαφορά, δηλαδή οι περιπτώσεις εισόδου / εξόδου από το σύστημα και η αυτόματη συμπλήρωση στοιχείων παραμένουν ίδιες, όπως έχουν αναλυθεί προηγουμένως. Εδώ θα πρέπει να σημειωθεί ότι κατά την εγγραφή ενός νέου χρήστη στο σύστημα, αυτός ο χρήστης ορίζεται αυτόματα με ρόλο απλού χρήστη (user role). Μόνο ένας ήδη υπάρχων διαχειριστής μπορεί να προσθέσει ένα νέο διαχειριστή ή να αυξήσει τα δικαιώματα ενός απλού χρήστη σε ρόλο διαχειριστή. Αυτές οι λειτουργίες πραγματοποιούνται από τις διαχειριστικές σελίδες στο back-end της εφαρμογής.

Σε επίπεδο back-end, οι διαχειριστές έχουν τη δυνατότητα επέμβασης και τροποποίησης των δεδομένων της εφαρμογής, στις εξής θεματικές κατηγορίες:

- **Χρήστες:** Δικαιώματα προσθήκης, διαγραφής και τροποποίησης στοιχείων και ρόλου χρηστών.
- **Ξενοδοχειακές Μονάδες:** Δικαιώματα προσθήκης νέου ξενοδοχείου, αναζήτησης ξενοδοχείων ανά περιοχή, τροποποίηση στοιχείων ξενοδοχείου, προσθήκη και τροποποίηση τιμών διανυκτέρευσης ανά ξενοδοχείο.
- **Κρατήσεις:** εμφάνιση όλων των κρατήσεων στο σύστημα, πρόσβαση στις συγκεντρωτικές αναφορές του συστήματος κρατήσεων.

7.3.2 Διαχειριστικό πάνελ

Η είσοδος των διαχειριστών στο διαχειριστικό πάνελ της εφαρμογής γίνεται από την σελίδα `login.php` στον φάκελο `/administrator/`.

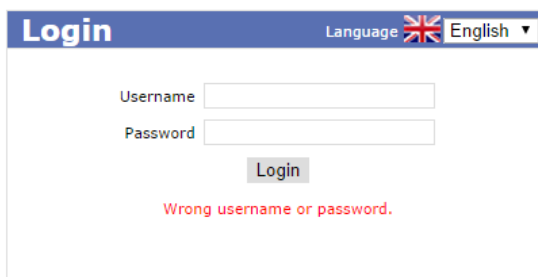


Εικόνα 41 – Φόρμα εισόδου διαχειριστών

Στην σελίδα αυτή, ζητούνται τα μοναδικά στοιχεία πρόσβασης του διαχειριστή, username και password. Το SQL ερώτημα που εκτελείται για την πιστοποίηση του χρήστη είναι το εξής:

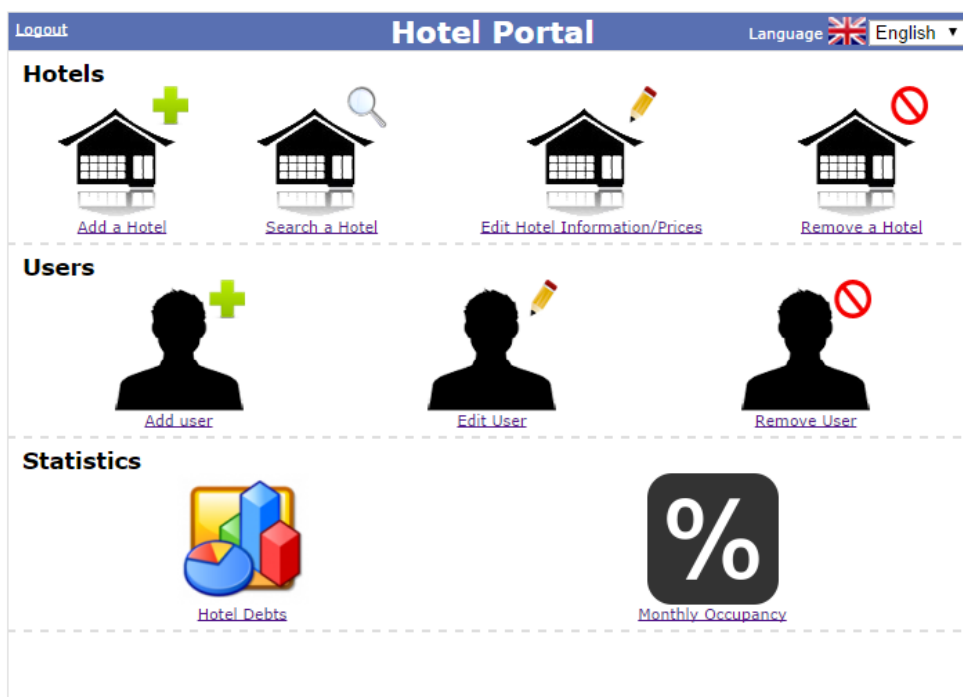
```
SELECT * FROM users
JOIN roles on users.role_id=roles.role_id
WHERE username='".$_POST['username']."'
AND password='".$_POST['password']."'
AND value='administrator';
```

Σε περίπτωση όπου ένας χρήστης με ρόλο απλού χρήστη και όχι διαχειριστή προσπαθεί να εισέλθει στο σύστημα, τότε εμφανίζεται το παρακάτω μήνυμα σφάλματος και αποτρέπεται η είσοδος:



Εικόνα 42 – Σφάλμα εισόδου διαχειριστή

Εφόσον γίνει ταυτοποίηση των στοιχείων του διαχειριστή από τα στοιχεία που έχουν συμπληρωθεί, εμφανίζεται το πάνελ διαχείρισης της εφαρμογής.



Εικόνα 43 – Διαχειριστικό πάνελ

7.3.2.1 Διαχείριση ξενοδοχείων

Η προσθήκη ενός ξενοδοχείου πραγματοποιείται από τον σύνδεσμο:



Εικόνα 44 – Σύνδεσμος εισαγωγής ξενοδοχείου

Ο σύνδεσμος αυτός μεταφέρει τον διαχειριστή στην σελίδα `addhotel.php`.

The screenshot shows a web application interface for adding a hotel. The top navigation bar includes 'Hotels', 'Bookings', 'Statistics', 'Users', and 'Logout'. A language selector is set to 'English'. The main heading is 'Add a Hotel'. The form contains the following fields: Country (dropdown menu with 'Afghanistan' selected), City (dropdown menu with 'No Cities Available' selected), Name, Address, Phone 1, Phone 2, Fax, Mobile Phone, Email, URL, Rating (dropdown menu), and Hotel Manager (Name and Surname text boxes). Below the form are tabs for 'Description', 'Rooms', 'Area', 'How to arrive', 'Food', 'Pictures', and 'Facilities'. The 'Description' tab is active, showing a text area with the prompt 'Please write the hotel description below'. At the bottom of the form are 'Save' and 'Cancel' buttons.

Εικόνα 45 – Σελίδα προσθήκης στοιχείων ξενοδοχείου

Αφού συμπληρωθούν τα γενικά στοιχεία του ξενοδοχείου καθώς και οι πληροφορίες των επιμέρους τύπων δωματίων, αποστέλλονται πίσω στη σελίδα μέσω POST. Στο σημείο αυτό εκτελείται η εξής διαδικασία αποθήκευσης στη βάση:

1. Έλεγχος ότι το συγκεκριμένο ξενοδοχείο δεν έχει ήδη καταχωρηθεί. Αυτό γίνεται μέσω της μεθόδου `check_hotel_exists`.

```
function check_hotel_exists($hotel_name, $cityID) {
    $query="SELECT name
            FROM hotel
            WHERE name=' ".$hotel_name."'
            AND city_id=' ".$cityID."' ";
    $result=mysql_query($query);
    if(mysql_num_rows($result)==0) {
        return false;
    }else{
        return true;
    }
    return false;
}
```

2. Ανανέωση πίνακα `cities`, προσθέτοντας 1 ακόμα στο πλήθος των ξενοδοχείων (πεδίο `num_hotels`) της επιλεγμένης πόλης.

```
UPDATE cities
SET num_hotels=(num_hotels + 1)
WHERE city_id=" ".$cityID
```

3. Προσθήκη στοιχείων ξενοδοχείου στον πίνακα **hotel**

```
INSERT INTO hotel VALUES (NULL, '". addslashes($_POST['hotel_name'])."',
'. addslashes($_POST['address'])."', '". $cityID."',
'. $_POST['email']."', '". $_POST['url']."',
'. $ratings['rating']."', 0)
```

4. Προσθήκη πληροφοριών ξενοδοχείου στον πίνακα **hotel_descriptions**

```
INSERT INTO hotel_descriptions VALUES
(NULL, '". $hotelID."',
'. addslashes($_POST['hotel_description'])."',
'. addslashes($_POST['area_description'])."',
'. addslashes($_POST['food_description'])."',
'. addslashes($_POST['travel_description'])."')
```

5. Προσθήκη παροχών ξενοδοχείου στον πίνακα **hotel_facilities**

```
$query="INSERT INTO hotel_facilities VALUES (NULL, '". $hotelID."', ";
for ($j=0;$j<23;$j++) {
    $query.="$_POST['facilitiesvalues'][$j];
    if ($j!=22)
        $query.=",";
}
$query.=")";
```

6. Προσθήκη φωτογραφιών ξενοδοχείου στον πίνακα **hotel_photos**

```
INSERT INTO hotel_photos VALUES (NULL, '". $hotelID.$queryPic."")
```

7. Προσθήκη τηλεφώνων ξενοδοχείου στον πίνακα **hotel_phones**

```
INSERT INTO hotel_phones VALUES (NULL, '". $hotelID."',
'. $_POST['phone1']."', 1)";
INSERT INTO hotel_phones VALUES (NULL, '". $hotelID."',
'. $_POST['phone2']."', 2)";
INSERT INTO hotel_phones VALUES (NULL, '". $hotelID."',
'. $_POST['mobile']."', 3)";
INSERT INTO hotel_phones VALUES (NULL, '". $hotelID."',
'. $_POST['fax']."', 4)";
```

8. Προσθήκη διευθυντή ξενοδοχείου στον πίνακα **hotel_managers**

```
INSERT INTO hotel_managers VALUES (NULL, '". $hotelID."',
'. $_POST['mgr_name']."',
'. $_POST['mgr_surname']."',
$isRegistered)
```

9. Έλεγχος αν ο διευθυντής είναι και εγγεγραμμένος χρήστης του συστήματος, και προσθήκη συσχετίσής τους στον πίνακα **users_managers**

```
SELECT * FROM hotel_managers
WHERE hotel_id='". $hotelID."'
```

```
AND name=' ".$_POST['mgr_name']. "'
AND surname=' ".$_POST['mgr_surname']. "' )
```

10. Για κάθε τύπο δωματίου, προσθήκη των στοιχείων του στον πίνακα **room_types**

```
INSERT INTO room_types VALUES
(NULL, ' ".$room. "' , ' ". addslashes($_POST['room_description' . $i]). "' )
```

11. Προσθήκη χωρητικότητας δωματίου στον πίνακα **room_capacity**

```
INSERT INTO room_capacity VALUES (NULL, ".$roomtype_id.", ".$roomsnum[$i].", 0,
".$adt[$i].", ".$chi[$i].", ".$inf[$i].")
```

12. Προσθήκη τύπου θέας δωματίου στον πίνακα **room_viewtypes**

```
INSERT INTO room_viewtypes VALUES (NULL, ".$roomtype_id.", ".$view[$i].")
```

13. Προσθήκη παροχών ξενοδοχείου στον πίνακα **room_facilities**

```
$facilitiesroom=array("minibar"=>0,
                    "bathtub"=>0,
                    "inroomsafe"=>0,
                    "tv"=>0,
                    "balcony"=>0,
                    "radio"=>0,
                    "phone"=>0,
                    "ac"=>0,
                    "bathacc"=>0,
                    "wc"=>0,
                    "hairdryer"=>0,
                    "desk"=>0,
                    "towels"=>0,
                    "heating"=>0);
foreach ($_POST['things' . $i] as $things) {
    $facilitiesroom[$things]=1;
}
$facroom="";
foreach($facilitiesroom as $key=>$value) {
    $facroom.="' ".$value. "' , ";
}
INSERT INTO room_facilities VALUES (NULL, ".$roomtype_id.", ".$facroom.")
```

14. Προσθήκη φωτογραφιών δωματίου στον πίνακα **room_photos**

```
INSERT INTO room_photos VALUES
(NULL, ".$roomtype_id.", ".$roompic.")
```

15. Προσθήκη νέας εγγραφής στον πίνακα συσχέτισης ξενοδοχείου και τύπου δωματίου **hotel_rooms**

```
INSERT INTO hotel_rooms VALUES
(NULL, ".$hotelID.", ".$roomtype_id.")
```

Εφόσον εκτελεστούν απρόσκοπτα τα παραπάνω βήματα, ο διαχειριστής μεταφέρεται στην σελίδα **addrates.php**, ώστε να συμπληρώσει τις τιμές κάθε τύπου δωματίου και τις ημερομηνίες έναρξης και λήξης κάθε επιμέρους σεζόν λειτουργίας του ξενοδοχείου.

Εικόνα 46 – Σελίδα διαχείρισης τιμών και σεζόν

Κατά το submit της σελίδας, εκτελούνται οι εξής προσθήκες στην βάση δεδομένων:

1. Προσθήκη των σεζόν στον πίνακα **hotel_seasons**

```
INSERT INTO hotel_seasons VALUES (NULL, ".$hotelID.",
STR_TO_DATE('".$_POST['date_low_start']."', '%d-%m-%Y'),
STR_TO_DATE('".$_POST['date_low_end']."', '%d-%m-%Y'),
STR_TO_DATE('".$_POST['date_mid_start']."', '%d-%m-%Y'),
STR_TO_DATE('".$_POST['date_mid_end']."', '%d-%m-%Y'),
STR_TO_DATE('".$_POST['date_high_start']."', '%d-%m-%Y'),
STR_TO_DATE('".$_POST['date_high_end']."', '%d-%m-%Y'));
```

2. Προσθήκη μίας εγγραφής τιμών στον πίνακα **room_rates** για κάθε τύπο δωματίου του ξενοδοχείου

```
INSERT INTO room_rates VALUES (NULL, ".$lowrate.", ".$midrate.", ".$highrate.");
```

3. Προσθήκη μίας εγγραφής συσχέτισης του πίνακα τύπων δωματίων `room_types` με τον πίνακα τιμών του `room_rates`, στον πίνακα **room_type_rates**

```
INSERT INTO room_type_rates VALUES (NULL, ".$roomtypeID.", ".$rate_id.")
```

ΣΗΜΕΙΩΣΗ: Καθώς η ίδια σελίδα χρησιμοποιείται και κατά την ανανέωση των στοιχείων ενός ξενοδοχείου (*Edit Hotel*), ο αλγόριθμος ελέγχει αν υπάρχουν ήδη εγγραφές στον πίνακα συσχέτισης `room_type_rates` και σε αυτή την περίπτωση εκτελείται η ίδια λογική αλλά με ερωτήματα **UPDATE** και όχι **INSERT**.

Η **τροποποίηση** ενός ξενοδοχείου γίνεται από την σελίδα `edithotel.php`, πατώντας τον σχετικό σύνδεσμο από το κεντρικό πάνελ διαχείρισης.



[Edit Hotel Information/Prices](#)

Εικόνα 47 – Σύνδεσμος τροποποίησης ξενοδοχείου/τιμών

Στην σελίδα αυτή εκτελείται η ίδια διαδικασία με αυτήν εισαγωγής νέου ξενοδοχείου, όπως αναλύθηκε προηγουμένως, με τη διαφορά ότι τα ερωτήματα που εκτελούνται είναι ανανέωσης των ίδιων πεδίων, δηλαδή **UPDATE** queries και όχι νέας προσθήκης (**INSERT**). Δίνεται ένα σχετικό παράδειγμα, για την ανανέωση του πρώτου βήματος που αναλύθηκε προηγουμένως (πίνακας **hotel**).

```
UPDATE hotel SET  
name=' ".$addslashes($_POST['hotel_name'])."', address=' ".$addslashes($_POST['address'])."',  
city_id=".$cityID.",  
email=' ".$_POST['email']."',  
url=' ".$_POST['url']."',  
rating=' ".$_POST['rating']. "'  
WHERE hotel_id=".$hotelID;
```

Αντίστοιχα, για την **διαγραφή** ενός ξενοδοχείου από την βάση, χρησιμοποιείται ο σύνδεσμος:



[Remove a Hotel](#)

Εικόνα 48 – Σύνδεσμος διαγραφής ξενοδοχείου

Στην περίπτωση αυτή, εκτελείται το SQL statement **DELETE** στους πίνακες που αναφέρθηκαν στο σενάριο προσθήκης, χρησιμοποιώντας τα κατάλληλα κλειδιά των εγγραφών που διαγράφονται. Δίνεται ένα σχετικό παράδειγμα διαγραφής του ξενοδοχείου από τον πίνακα **hotel**.

```
DELETE FROM hotel WHERE hotel_id=".$hotelID;
```

7.3.2.2 Διαχείριση χρηστών

Ομοίως με τα ξενοδοχεία, ο διαχειριστής έχει τη δυνατότητα προσθήκης, τροποποίησης και διαγραφής των χρηστών του συστήματος, χρησιμοποιώντας τους αντίστοιχους συνδέσμους του πάνελ:



Εικόνα 49 – Σύνδεσμοι διαχείρισης χρηστών

Η δημιουργία νέου χρήστη εκτελείται με την ίδια λογική όπως παρουσιάστηκε στην ενότητα 7.2.1, εγγραφής νέου χρήστη. Η μόνη διαφορά έγκειται στο γεγονός ότι ο διαχειριστής έχει την δυνατότητα να εγγράψει ένα χρήστη σε οποιοδήποτε ρόλο χρηστών, είτε ως απλό user είτε ως administrator. Η επιλογή αυτή δίνεται στη φόρμα εγγραφής με radiobutton επιλογής ρόλου, και εισάγεται κατά το SQL INSERT στον πίνακα **users**.

The image shows a form titled "Add User". It contains several input fields: Name, Surname, Phone, Country (dropdown menu with "Afghanistan" selected), Username, Password, Repeat Password, Email, Credit Card Number, Credit Card Type (dropdown menu with "VISA" selected), and Type (radio buttons for "User" and "Administrator", with "User" selected). The "Type" section is highlighted with a red box. At the bottom right, there are "Save" and "Cancel" buttons.

Εικόνα 50 – Φόρμα δημιουργίας χρήστη στο διαχειριστικό

Ομοίως, με τα προηγούμενα, κατά την τροποποίηση των στοιχείων ενός χρήστη, εκτελείται UPDATE στον πίνακα users, ενώ κατά την διαγραφή DELETE. Στην περίπτωση της τροποποίησης, δεν επιτρέπεται αλλαγή στο πεδίο username, καθώς πρέπει να παραμένει πάντα το ίδιο για τον συγκεκριμένο χρήστη.

The image shows a form titled "Edit User". It contains several input fields: Username (highlighted with a red box and containing "gpapadop"), Name (containing "George"), Surname (containing "Papadopoulos"), Phone (containing "+30 210 1234 123"), Country (dropdown menu with "Greece" selected), New Password, Repeat New Password, Email (containing "g.papadopoulos@mail.gr"), Credit Card Number (blurred), Credit Card Type (dropdown menu with "VISA" selected), and Type (radio buttons for "User" and "Administrator", with "User" selected). At the bottom right, there are "Save" and "Cancel" buttons.

Εικόνα 51 – Τροποποίηση χρήστη στο διαχειριστικό

7.3.3 Αναφορές συστήματος

Το σύστημα που υλοποιήθηκε στα πλαίσια της εργασίας αυτής, εξάγει δύο διαφορετικές κατηγορίες αναφορών από τα δεδομένα του συστήματος, και παρουσιάζει τα στοιχεία με γραφικό τρόπο στις σελίδες διαχείρισης της εφαρμογής. Η μία κατηγορία αφορά στις οφειλές του ενός ξενοδοχείου προς τα υπόλοιπα, αν και εφόσον υπάρχουν. Η δεύτερη αφορά στην απεικόνιση της πληρότητας του εκάστοτε ξενοδοχείου ανά χρονική σεζόν λειτουργίας (Low, Mid, High Season). Η πρόσβαση στις αναφορές αυτές γίνεται από τις δυναμικές σελίδες **hotelDebts.php** και **hotelOccupancy.php** αντίστοιχα. Οι σελίδες αυτές αντιστοιχούν στους συνδέσμους που βρίσκονται στην περιοχή των Στατιστικών του διαχειριστικού πάνελ.

Statistics



Hotel Debts



Monthly Occupancy

Εικόνα 52 – Περιοχή στατιστικών αναφορών στο διαχειριστικό πάνελ

7.3.3.1 Οφειλές ξενοδοχείου

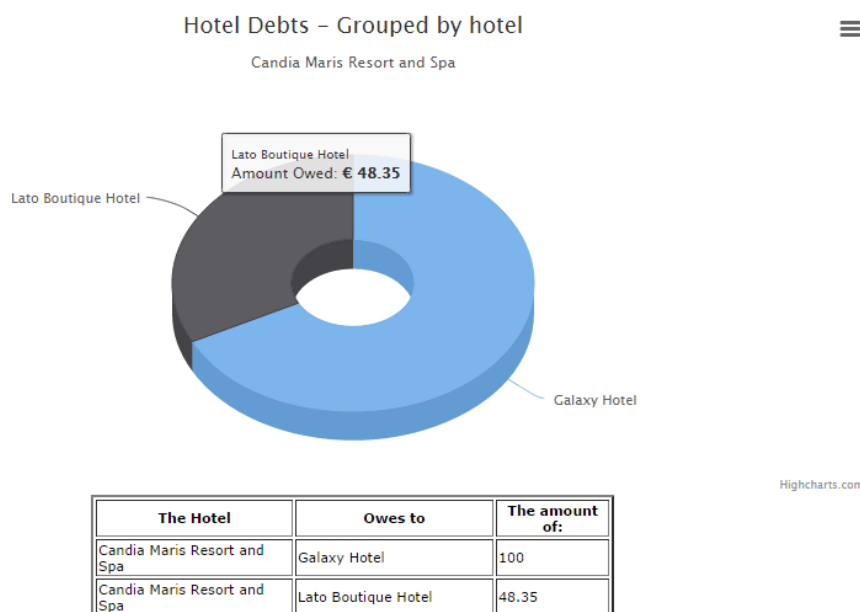
Η σελίδα **hotelDebts.php** παρέχει στον διαχειριστή ένα drop-down πεδίο με όλα τα καταχωρημένα ξενοδοχεία της βάσης. Επιλέγοντας το ξενοδοχείο που τον ενδιαφέρει, εμφανίζεται στη σελίδα η γραφική απεικόνιση με τη γνωστή μορφή pie chart, όπου κάθε τμήμα αφορά το ποσοστό κόστους προς κάποιο άλλο ξενοδοχείο. Με mouseover σε κάθε τμήμα, εμφανίζεται ένα popup με επιπλέον πληροφορίες για το εκάστοτε κόστος. Επίσης, παρατίθεται και συγκεντρωτικός πίνακας με τα κόστη στο τέλος της σελίδας.

Search Hotel

Select the hotel to view its debts to other hotels

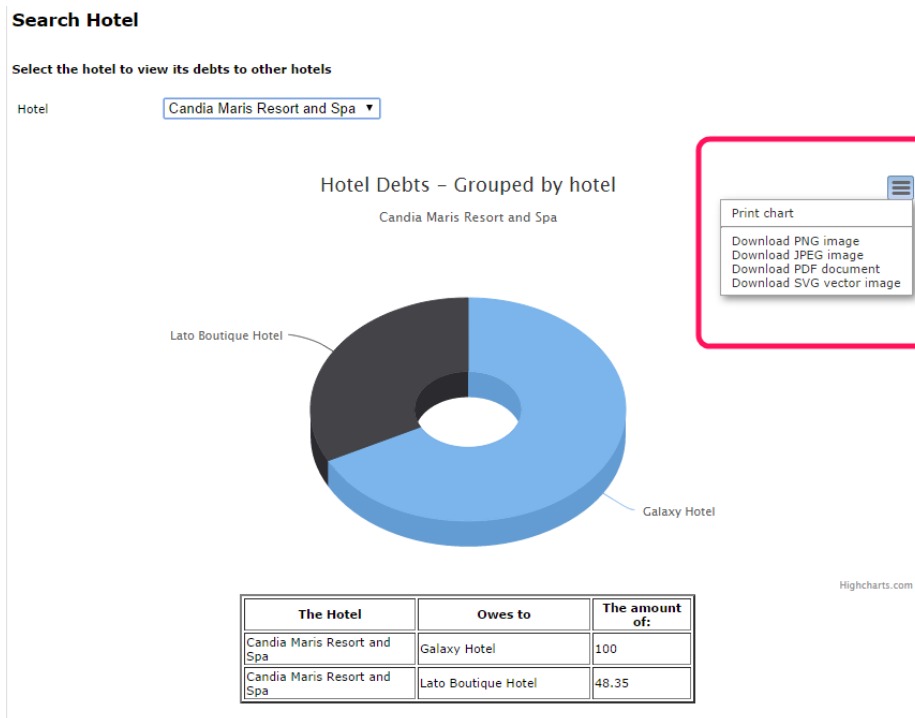
Hotel

Candia Maris Resort and Spa ▼



Εικόνα 53 – Σελίδα ανάλυσης οφειλών ξενοδοχείου

Επιπλέον, ο διαχειριστής μπορεί να εκτυπώσει το εκάστοτε γραφικό ή να το αποθηκεύσει τοπικά σε διάφορα format, όπως PNG, JPEG ή SVG-vector εικόνα, ή ακόμα και ως αρχείο PDF.



Εικόνα 54 – Δυνατότητες εκτύπωσης και αποθήκευσης γραφημάτων

Για την ανάκτηση της συγκεκριμένης πληροφορίας χρησιμοποιείται ο πίνακας costs από τη βάση δεδομένων του συστήματος. Συγκεκριμένα, το SQL query που εκτελείται προς τη βάση είναι το

```
SELECT hotel.name, amount_due FROM costs
JOIN hotel on costs.hotelID=hotel.hotel_id
WHERE costs.amount_due > 0 AND costs.hotelID=".$hotelID;
```

εξής:

Τα αποτελέσματα του ερωτήματος επιστρέφονται σε κατάλληλα διαμορφωμένο JSON object ώστε να τα χειριστεί κατάλληλα η Javascript της σελίδας. Εδώ, επιλέχθηκε η χρήση της δυναμικής Javascript βιβλιοθήκης απεικόνισης γραφημάτων, ονόματι **Highcharts.js** [33], η οποία παρέχει μία ευρεία γκάμα λειτουργιών και διαδραστικής εμφάνισης γραφημάτων στην πλευρά του χρήστη.

7.3.3.2 Πληρότητα ξενοδοχείου

Η σελίδα **hotelOccupancy.php** λειτουργεί με την ίδια λογική όπως τη σελίδα ανάλυσης κόστους των ξενοδοχείων. Και εδώ, ο διαχειριστής επιλέγει το ξενοδοχείο που τον ενδιαφέρει ώστε να προβάλλει την ανάλυση πληρότητάς του ανά σεζόν. Ομοίως, το γράφημα εμφανίζεται μέσω της Javascript βιβλιοθήκης Highcharts.js, και παρέχονται λειτουργίες εκτύπωσης και αποθήκευσης του γραφήματος. Επίσης, εμφανίζεται και ο συγκεντρωτικός πίνακας ανάλυσης στο τέλος της σελίδας. Οι ουσιώδεις διαφορές με τα προηγούμενα, είναι ότι σε αυτή τη περίπτωση το γράφημα απεικονίζεται με τη μορφή απλού γράφου γραμμής (line chart), όπου στη μία γραμμή εκφράζεται το ποσοστό (%) πληρότητας ανά σεζόν, ενώ στην άλλη γραμμή οι κατειλημμένες κλίνες. Η ανάκτηση των στοιχείων αυτών προκύπτει από τον συνδυασμό των εξής τριών (3) SQL queries, ένα για την εύρεση της διαθεσιμότητας, ένα για την εύρεση των κρατήσεων που βρίσκονται σε στάδιο "complete" και όχι "cancelled", και ένα για τις χρονικές περιόδους κάθε σεζόν:

```
SELECT sum(room_capacity.availability) as sum
FROM room_capacity
```



```
JOIN hotel_rooms on room_capacity.roomtype_id=hotel_rooms.roomtype_id
WHERE hotel_rooms.hotel_id=". $hotelID
```

```
SELECT number_of_rooms, check_in, check_out
FROM bookings
WHERE status='complete' AND hotel_id=". $hotelID
```

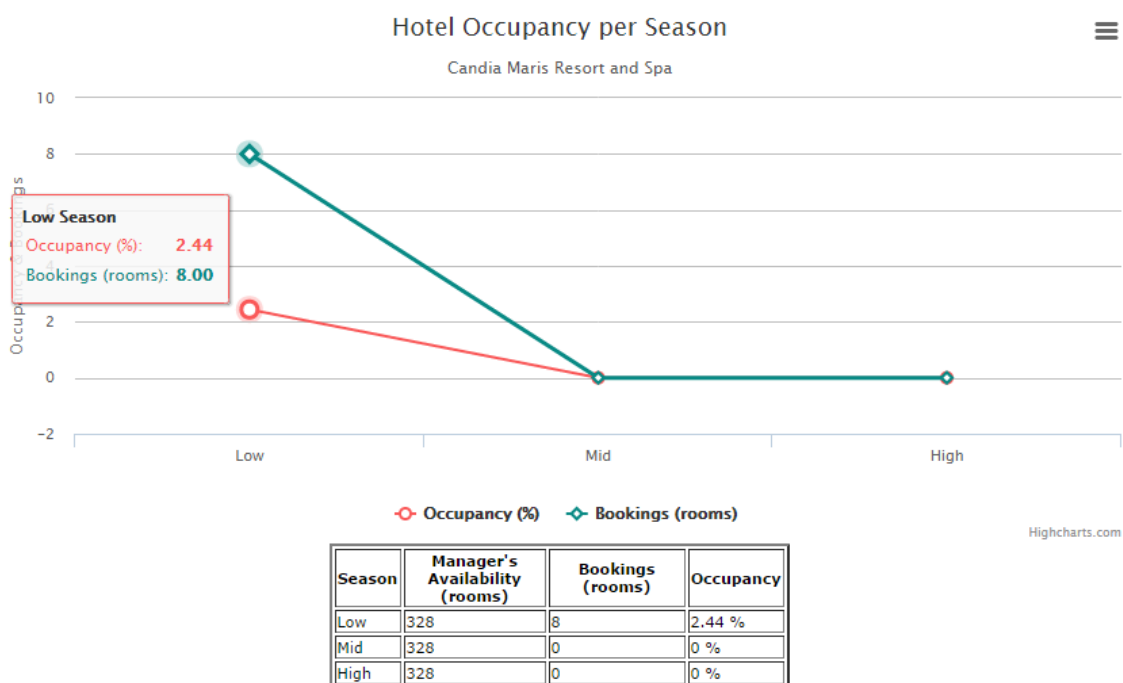
```
SELECT * from hotel_seasons where hotel_id=". $hotelID
```

Τα αποτελέσματα των ερωτημάτων αυτών απεικονίζονται γραφικά στη σελίδα ως εξής:

Search Hotel

Select the hotel to view its seasonal occupancy

Hotel




Εικόνα 55- Σελίδα αναφορών πληρότητας ξενοδοχείου

Τέλος, στη σελίδα **bookings.php**, παρουσιάζεται η συγκεντρωτική κατάσταση όλων των κρατήσεων που έχουν διενεργηθεί στο σύστημά μας.



Εικόνα 56 – Μενού προβολής κρατήσεων συστήματος

Hotels Bookings Statistics Users Logout										Language  English ▼	
id	PNR	Session Key	Hotel	Rooms	Card Type	Amount	Check In Date	Check Out Date	Status		
1	HP_O2-U1810399262	ed64a1e8-19cf-4cb8-9f5b-10606ea52108	Candia Maris Resort and Spa	1	VISA	772	2014-08-26	2014-08-30	complete		
2	HP_O2-U1351390303	3b8cd5bb-8724-4f9a-ae7b-97c1c7fe10e7	Candia Maris Resort and Spa	1	VISA	640	2014-08-27	2014-08-31	complete		
3	HP_O2-U2845040205	6e60ed11-ea82-4e69-8904-6a93058f0335	Candia Maris Resort and Spa	1	VISA	960	2014-08-25	2014-08-31	complete		
4	HP_O2-U8486446166	9aba596c-dc8e-43fd-bf39-01a7415beebe	Candia Maris Resort and Spa	1	VISA	960	2014-08-25	2014-08-31	complete		
5	HP_O2-U1766373274	d8b4c944-bdae-4fd6-aa65-ff4291914eb3	Candia Maris Resort and Spa	1	VISA	1544	2014-08-21	2014-08-29	complete		
6	HP_O2-U2799137387	cb41f7d3-ba16-4b76-b956-91b4240a546e	Candia Maris Resort and Spa	1	VISA	772	2014-08-27	2014-08-31	complete		
7	HP_O2-U2791396010	009ba682-14d6-44f5-bd5b-3af7645c2bb9	Candia Maris Resort and Spa	1	VISA	965	2014-08-26	2014-08-31	cancelled		
8	HP_O2-U1076376557	1c0f2729-8811-44cb-ab6a-c7eea7889271	Candia Maris Resort and Spa	1	VISA	1158	2014-08-25	2014-08-31	cancelled		
9	HP_O2-U2147702955	62abc566-a3ff-4a84-8838-efb44bbe8f0d	Candia Maris Resort and Spa	1	VISA	640	2014-08-26	2014-08-30	complete		
10	HP_O10-U1122192570	0364ea13-a921-4c0b-8754-3f70fb4bb282	Candia Maris Resort and Spa	1	VISA	221	2014-08-29	2014-08-31	complete		
11	HP_O11-U7572022136	c55dfbe8-b309-4a08-bbac-e3fe568e115b	Candia Maris Resort and Spa	1	VISA	511.96	2014-08-29	2014-09-02	cancelled		

Εικόνα 57 – Κατάσταση κρατήσεων στο σύστημα

8 Συμπεράσματα και Μελλοντικές Επεκτάσεις

Αδιαμφισβήτητα, η χρήση ολοκληρωμένων συστημάτων κρατήσεων μέσω διαδικτύου, θα συνεχίσει να παίζει σημαντικό ρόλο στην εξέλιξη του τουριστικού τομέα μέσα στα επόμενα χρόνια. Ολοένα και περισσότερες επιχειρήσεις στρέφονται σε νέα δυναμικά προϊόντα, που μπορούν να αυτοματοποιήσουν χρονοβόρες διαδικασίες, να προβάλουν τα προϊόντα και τις υπηρεσίες τους σε νέες αγορές, με σκοπό την μείωση του κόστους και την αύξηση των κερδών.

Τα συστήματα online διαχείρισης κρατήσεων μπορούν να επιτελέσουν αυτό το σκοπό, παρουσιάζοντας τα βέλτιστα δυνατά αποτελέσματα στην σύγχρονη εποχή. Τα συστήματα αυτά είναι ικανά να υποστηρίξουν μεγάλο όγκο δεδομένων και διεργασιών, σε συνδυασμό με την εξυπηρέτηση πολλών πελατών παράλληλα. Το δε κόστος κατασκευής τέτοιων συστημάτων εξαρτάται από τις δυνατότητες που πρέπει να παρέχει, και το είδος της επιχείρησης που εξυπηρετεί.

Σε επίπεδο σχεδίασης και ανάπτυξης του συστήματος, απαιτείται αυξημένη τεχνογνωσία στην διαχείριση της πληροφορίας, καθώς ο τουριστικός τομέας έχει αρκετά πολύπλοκους περιορισμούς και διαδικασίες. Παράλληλα, χρειάζεται ιδιαίτερη προσοχή στην ανάπτυξη της εφαρμογής, για να ελέγχονται όλα τα πιθανά σενάρια καθώς επίσης να αποφεύγονται λογικά λάθη.

Κλείνοντας την εργασία αυτή και έχοντας υλοποιήσει την συγκεκριμένη εφαρμογή κρατήσεων, επισημαίνονται κάποιες μελλοντικές αλλαγές και προσθήκες που θεωρούνται απαραίτητες. Καταρχάς, σημειώνεται ότι η αποθήκευση προσωπικών δεδομένων των χρηστών θα ήταν καλύτερο να γίνεται με κάποια κρυπτογράφηση της μορφής MD5 hashing των στοιχείων, προς αποφυγή διαρροών προσωπικών στοιχείων. Επίσης, προτείνεται η δημιουργία επιπλέον πεδίων στην φόρμα αναζήτησης, όπως την κατηγορία του ξενοδοχείου, επιλογή ή συνδυασμών προσφερόμενων παροχών, κ.λπ., και αφαίρεση της επιλογής μεμονωμένου ξενοδοχείου. Κατά αυτό τον τρόπο, ο δυνητικός πελάτης αποκτά μεγαλύτερη πληροφόρηση σχετικά με όλα τα παρεχόμενα καταλύματα της περιοχής. Με την ίδια λογική, θα μπορούσε να υλοποιηθεί και η λειτουργία φίλτρων των σχετικών αποτελεσμάτων, για την καλύτερη εξυπηρέτηση του πελάτη. Τέλος, είναι εμφανές ότι καθώς προστίθενται πληροφορίες ξενοδοχείων και δωματίων, αυξάνεται κατά πολύ η ανάγκη αποθήκευσης και ενός ισχυρού συστήματος διαχείρισης βάσης δεδομένων για την λειτουργία της εφαρμογής. Προτείνεται λοιπόν η αντικατάσταση των τοπικά αποθηκευμένων δεδομένων, που σε καμία περίπτωση δεν είναι πλήρη, με την χρήση online συστημάτων διανομής GDS της σχετικής πληροφορίας, που είναι πάντα κατάλληλα ενημερωμένα, μέσω υλοποίησης τεχνολογιών Web Services.

Βιβλιογραφία

- [1] Κ. Π. Ν. ΦΛΩΡΙΝΑΣ, «Η Θεωρία των Βάσεων Δεδομένων» [Ηλεκτρονικό]. Available: <http://dide.flo.sch.gr/Plinet/Tutorials/Tutorials-DataBasesTheory.html>.
- [2] «Κεφάλαιο 1 : Γενικές αρχές Βάσεων Δεδομένων» [Ηλεκτρονικό]. Available: http://anamorfosi.teicm.gr/ekp_yliko/e-notes/Data/database/main.htm.
- [3] «Κανονικοποίηση Σχέσεων» [Ηλεκτρονικό]. Available: http://www.math.uoi.gr/~istamat/database_ioannina/book/dbases/k6/k6_3.html.
- [4] Ν. Theoharis, «Reservation Systems in tourist section» [Ηλεκτρονικό]. Available: <http://traveldailynews.gr/pdf/Epth3/NikosTheoharis.pdf>.
- [5] «Sabre» [Ηλεκτρονικό]. Available: <http://www.sabre.com/>.
- [6] «Travelport» [Ηλεκτρονικό]. Available: <http://www.travelport.com/Solutions/Travel-Suppliers/Overview>.
- [7] «Travelport Galileo Web Services» [Ηλεκτρονικό]. Available: <http://www.travelport.com/Products/Galileo-Web-Services>.
- [8] «Amadeus Greece» [Ηλεκτρονικό]. Available: http://www.amadeus.com/web/amadeus/en_GR-GR/Amadeus-Home/1259077846833-Page-AMAD_HomePpal.
- [9] S. Gkigas, «Ξενοδοχειακό Λογισμικό» Touristiki Agora, [Ηλεκτρονικό]. Available: <http://www.touristiki-agera.gr/article.asp?ID=2620>.
- [10] «Google Analytics» [Ηλεκτρονικό]. Available: <http://www.google.com/analytics/>.
- [11] «Amadeus Content Inn» [Ηλεκτρονικό]. Available: http://www.amadeus.com/web/amadeus/en_LT-LT/Travel-agencies/Solutions-and-services/Single-site-travel-agencies/LT_Content-Inn/1319526383493-Solution_C-AMAD_ProductDetailPpal-1259082732594?industrySegment=1259068356166.
- [12] «Hit Hospitality» [Ηλεκτρονικό]. Available: <http://www.hit.com.gr/>.
- [13] «BOOKOnlinenow» [Ηλεκτρονικό]. Available: <http://book-onlinenow.com/>.
- [14] «Nelios.com» [Ηλεκτρονικό]. Available: <http://www.nelios.com/etourism-solutions-hoteliars-92.htm>.
- [15] «NetSemantics nsTravel» [Ηλεκτρονικό]. Available: <http://www.netsemantics.gr/ns/HomePage/Products.aspx>.
- [16] «SingularLogic Hospitality System» [Ηλεκτρονικό]. Available: <http://portal.singularlogic.eu/product/266/hospitality-system-hotel>.
- [17] «Sunsoft» [Ηλεκτρονικό]. Available: <http://www.sunsoft.gr/load/content/category/view/%CE%BB%CF%8D%CF%83%CE%B5%CE%B9%CF%82#.VAdsBjKSzqo>.
- [18] «Sysco» [Ηλεκτρονικό]. Available: <http://www.sysco.gr/content/%CE%BE%CE%B5%CE%BD%CE%BF%CE%B4%CE%BF%CF%87%CE%B5%CE%AF%CE%B1>.
- [19] «MySQL» [Ηλεκτρονικό]. Available: <http://www.mysql.com/>.
- [20] Wikipedia, «SQL» [Ηλεκτρονικό]. Available: <http://en.wikipedia.org/wiki/SQL>.
- [21] «PHP.net» [Ηλεκτρονικό]. Available: <http://php.net/>.
- [22] «CGI» [Ηλεκτρονικό]. Available: <https://metacpan.org/pod/CGI>.
- [23] «W3C HTML» [Ηλεκτρονικό]. Available: <http://www.w3.org/html/>.
- [24] «W3C CSS» [Ηλεκτρονικό]. Available: <http://www.w3.org/Style/CSS/Overview.en.html>.
- [25] Wikipedia, «JavaScript» [Ηλεκτρονικό]. Available: <http://en.wikipedia.org/wiki/JavaScript>.
- [26] «Node.js» [Ηλεκτρονικό]. Available: <http://nodejs.org/>.
- [27] «Notepad ++» [Ηλεκτρονικό]. Available: <http://notepad-plus-plus.org/>.
- [28] «Netbeans» [Ηλεκτρονικό]. Available: <https://netbeans.org/>.
- [29] «MySQL Workbench» [Ηλεκτρονικό]. Available: <http://www.mysql.com/products/workbench/>.

- [30] «jQuery,» [Ηλεκτρονικό]. Available: <http://jquery.com/>.
- [31] «Stripe Checkout PHP API Documentation,» [Ηλεκτρονικό]. Available: <https://stripe.com/docs/api/php>.
- [32] «PHPMailer API,» [Ηλεκτρονικό]. Available: <https://github.com/Synchro/PHPMailer>.
- [33] «Highcharts.js,» [Ηλεκτρονικό]. Available: <http://www.highcharts.com/>.

ΠΑΡΑΡΤΗΜΑ Α

ΚΩΔΙΚΑΣ ΔΗΜΙΟΥΡΓΙΑΣ ΠΙΝΑΚΩΝ ΒΑΣΗΣ

Στο παράρτημα αυτό καταγράφονται τα SQL ερωτήματα δημιουργίας κάθε πίνακα, με αλφαβητική σειρά:

bookings

```
CREATE TABLE `bookings` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `order_id` varchar(25) NOT NULL,  
  `session_id` varchar(40) NOT NULL,  
  `hotel_id` int(10) unsigned NOT NULL,  
  `number_of_rooms` int(5) unsigned NOT NULL,  
  `payment_token` varchar(40) NOT NULL,  
  `card_type` int(1) unsigned NOT NULL,  
  `amount` int(10) unsigned NOT NULL,  
  `check_in` date NOT NULL,  
  `check_out` date NOT NULL,  
  `status` varchar(15) NOT NULL,  
  PRIMARY KEY (`id`,`order_id`),  
  UNIQUE KEY `id_UNIQUE` (`id`),  
  UNIQUE KEY `order_id_UNIQUE` (`order_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8;
```

card_types

```
CREATE TABLE `card_types` (  
  `card_type_id` int(1) unsigned NOT NULL AUTO_INCREMENT,  
  `type` varchar(16) NOT NULL,  
  PRIMARY KEY (`card_type_id`),  
  UNIQUE KEY `card_type_id_UNIQUE` (`card_type_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8;
```

cities

```
CREATE TABLE `cities` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `city_id` int(8) unsigned NOT NULL,  
  `name` varchar(50) NOT NULL,  
  `latitude` double NOT NULL,  
  `longitude` double NOT NULL,  
  `num_hotels` int(10) unsigned NOT NULL,  
  PRIMARY KEY (`id`,`city_id`),  
  UNIQUE KEY `id_UNIQUE` (`id`),  
  UNIQUE KEY `city_id_UNIQUE` (`city_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=102689 DEFAULT CHARSET=utf8;
```

costs

```
CREATE TABLE `costs` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `hotelA` int(10) unsigned NOT NULL,  
  `hotelB` int(10) unsigned NOT NULL,  
  `amount_due` float(7,2) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `id_UNIQUE` (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8;
```

countries

```
CREATE TABLE `countries` (  
  `country_id` int(5) NOT NULL AUTO_INCREMENT,  
  `country_name` varchar(64) NOT NULL,  
  `country_code` char(2) NOT NULL,  
  `flag` text NOT NULL,  
  PRIMARY KEY (`country_id`, `country_code`),  
  UNIQUE KEY `country_id_UNIQUE` (`country_id`),  
  UNIQUE KEY `country_name_UNIQUE` (`country_name`),  
  UNIQUE KEY `country_code_UNIQUE` (`country_code`)  
) ENGINE=InnoDB AUTO_INCREMENT=240 DEFAULT CHARSET=utf8;
```

hotel

```
CREATE TABLE `hotel` (  
  `hotel_id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `name` varchar(40) NOT NULL,  
  `address` varchar(100) NOT NULL,  
  `city_id` int(10) unsigned NOT NULL,  
  `email` varchar(80) NOT NULL,  
  `url` varchar(100) NOT NULL,  
  `rating` int(5) unsigned NOT NULL,  
  `rates_added` bit(1) NOT NULL,  
  PRIMARY KEY (`hotel_id`),  
  UNIQUE KEY `hotel_id_UNIQUE` (`hotel_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8;
```

hotel_descriptions

```
CREATE TABLE `hotel_descriptions` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `hotel_id` int(10) unsigned NOT NULL,  
  `hotel` text NOT NULL,  
  `area` text NOT NULL,  
  `food` text NOT NULL,  
  `travel` text NOT NULL,  
  PRIMARY KEY (`id`, `hotel_id`),  
  UNIQUE KEY `id_UNIQUE` (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8;
```

hotel_facilities

```
CREATE TABLE `hotel_facilities` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `hotel_id` int(10) unsigned NOT NULL,  
  `airport` bit(1) NOT NULL,  
  `babysit` bit(1) NOT NULL,  
  `bar` bit(1) NOT NULL,  
  `nearbeach` bit(1) NOT NULL,  
  `parking` bit(1) NOT NULL,  
  `alarm` bit(1) NOT NULL,  
  `conference` bit(1) NOT NULL,  
  `elevator` bit(1) NOT NULL,  
  `entertainment` bit(1) NOT NULL,  
  `gym` bit(1) NOT NULL,  
  `safe` bit(1) NOT NULL,  
  `internet` bit(1) NOT NULL,  
  `iron` bit(1) NOT NULL,  
  `laundry` bit(1) NOT NULL,
```

```
`mailbox` bit(1) NOT NULL,  
`moneyexchange` bit(1) NOT NULL,  
`satelite` bit(1) NOT NULL,  
`smoking` bit(1) NOT NULL,  
`spa` bit(1) NOT NULL,  
`pool` bit(1) NOT NULL,  
`heating` bit(1) NOT NULL,  
`ac` bit(1) NOT NULL,  
`pets` bit(1) NOT NULL,  
PRIMARY KEY (`id`,`hotel_id`),  
UNIQUE KEY `id_UNIQUE` (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8;
```

hotel_managers

```
CREATE TABLE `hotel_managers` (  
  `manager_id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `hotel_id` int(10) unsigned NOT NULL,  
  `name` varchar(45) NOT NULL,  
  `surname` varchar(45) NOT NULL,  
  `isUser` bit(1) NOT NULL,  
  PRIMARY KEY (`manager_id`,`hotel_id`),  
  UNIQUE KEY `id_UNIQUE` (`manager_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8;
```

hotel_phones

```
CREATE TABLE `hotel_phones` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `hotel_id` int(10) NOT NULL,  
  `phonenumber` varchar(20) NOT NULL,  
  `phone_type` int(10) NOT NULL,  
  PRIMARY KEY (`id`,`hotel_id`,`phone_type`),  
  UNIQUE KEY `id_UNIQUE` (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=65 DEFAULT CHARSET=utf8;
```

hotel_photos

```
CREATE TABLE `hotel_photos` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `hotel_id` int(10) unsigned NOT NULL,  
  `picture` varchar(100) NOT NULL,  
  PRIMARY KEY (`id`,`hotel_id`),  
  UNIQUE KEY `id_UNIQUE` (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=41 DEFAULT CHARSET=utf8;
```

hotel_ratings

```
CREATE TABLE `hotel_ratings` (  
  `id` int(5) unsigned NOT NULL AUTO_INCREMENT,  
  `rating` int(5) unsigned NOT NULL,  
  PRIMARY KEY (`id`,`rating`),  
  UNIQUE KEY `id_UNIQUE` (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;
```

hotel_rooms

```
CREATE TABLE `hotel_rooms` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `hotel_id` int(10) unsigned NOT NULL,  
  `roomtype_id` int(10) unsigned NOT NULL,
```



```
PRIMARY KEY (`id`,`hotel_id`,`roomtype_id`),  
UNIQUE KEY `id_UNIQUE` (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=15 DEFAULT CHARSET=utf8;
```

hotel_seasons

```
CREATE TABLE `hotel_seasons` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `hotel_id` int(10) unsigned NOT NULL,  
  `lowSeason_start` date NOT NULL,  
  `lowSeason_end` date NOT NULL,  
  `midSeason_start` date NOT NULL,  
  `midSeason_end` date NOT NULL,  
  `highSeason_start` date NOT NULL,  
  `highSeason_end` date NOT NULL,  
  PRIMARY KEY (`id`,`hotel_id`),  
  UNIQUE KEY `id_UNIQUE` (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8;
```

languages

```
CREATE TABLE `languages` (  
  `lang_id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `language_name` varchar(40) NOT NULL,  
  PRIMARY KEY (`lang_id`),  
  UNIQUE KEY `id_UNIQUE` (`lang_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8;
```

locations

```
CREATE TABLE `locations` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `city_id` int(8) unsigned NOT NULL,  
  `country_code` char(2) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `id_UNIQUE` (`id`),  
  UNIQUE KEY `city_id_UNIQUE` (`city_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=82904 DEFAULT CHARSET=utf8;
```

phone_types

```
CREATE TABLE `phone_types` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `type` varchar(45) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `id_UNIQUE` (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8;
```

roles

```
CREATE TABLE `roles` (  
  `role_id` int(1) unsigned NOT NULL AUTO_INCREMENT,  
  `value` varchar(20) NOT NULL,  
  PRIMARY KEY (`role_id`),  
  UNIQUE KEY `role_id_UNIQUE` (`role_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8;
```

room_capacity

```
CREATE TABLE `room_capacity` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
```

```
`roomtype_id` int(10) unsigned NOT NULL,  
`availability` int(5) unsigned NOT NULL,  
`adults` int(1) unsigned NOT NULL,  
`children` int(1) unsigned NOT NULL,  
`infants` int(1) unsigned NOT NULL,  
PRIMARY KEY (`id`,`roomtype_id`),  
UNIQUE KEY `id_UNIQUE` (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=20 DEFAULT CHARSET=utf8;
```

room_facilities

```
CREATE TABLE `room_facilities` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `roomtype_id` int(10) unsigned NOT NULL,  
  `minibar` bit(1) NOT NULL,  
  `bathtub` bit(1) NOT NULL,  
  `safe` bit(1) NOT NULL,  
  `tv` bit(1) NOT NULL,  
  `balcony` bit(1) NOT NULL,  
  `radio` bit(1) NOT NULL,  
  `phone` bit(1) NOT NULL,  
  `ac` bit(1) NOT NULL,  
  `bathacc` bit(1) NOT NULL,  
  `wc` bit(1) NOT NULL,  
  `dryer` bit(1) NOT NULL,  
  `desk` bit(1) NOT NULL,  
  `towels` bit(1) NOT NULL,  
  `heating` bit(1) NOT NULL,  
  PRIMARY KEY (`id`,`roomtype_id`),  
  UNIQUE KEY `id_UNIQUE` (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=20 DEFAULT CHARSET=utf8;
```

room_photos

```
CREATE TABLE `room_photos` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `roomtype_id` int(10) unsigned NOT NULL,  
  `picture` varchar(200) NOT NULL,  
  PRIMARY KEY (`id`,`roomtype_id`),  
  UNIQUE KEY `id_UNIQUE` (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=29 DEFAULT CHARSET=utf8;
```

room_rates

```
CREATE TABLE `room_rates` (  
  `rate_id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `price_low` float(7,2) unsigned NOT NULL,  
  `price_mid` float(7,2) unsigned NOT NULL,  
  `price_high` float(7,2) unsigned NOT NULL,  
  PRIMARY KEY (`rate_id`),  
  UNIQUE KEY `id_UNIQUE` (`rate_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=15 DEFAULT CHARSET=utf8;
```

room_type_rates

```
CREATE TABLE `room_type_rates` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `roomtype_id` int(10) unsigned NOT NULL,  
  `rate_id` int(10) unsigned NOT NULL,  
  PRIMARY KEY (`id`,`roomtype_id`,`rate_id`),  
  UNIQUE KEY `id_UNIQUE` (`id`)
```

```
) ENGINE=InnoDB AUTO_INCREMENT=15 DEFAULT CHARSET=utf8;
```

room_types

```
CREATE TABLE `room_types` (  
  `roomtype_id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `title` varchar(45) NOT NULL,  
  `description` varchar(45) NOT NULL,  
  PRIMARY KEY (`roomtype_id`),  
  UNIQUE KEY `roomtype_id_UNIQUE` (`roomtype_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=15 DEFAULT CHARSET=utf8;
```

room_viewtypes

```
CREATE TABLE `room_viewtypes` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `roomtype_id` int(10) unsigned NOT NULL,  
  `view_type` int(10) unsigned NOT NULL,  
  PRIMARY KEY (`id`,`roomtype_id`),  
  UNIQUE KEY `id_UNIQUE` (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=20 DEFAULT CHARSET=utf8;
```

sessions

```
CREATE TABLE `sessions` (  
  `id` int(20) unsigned NOT NULL AUTO_INCREMENT,  
  `session_key` varchar(40) NOT NULL,  
  `json_data` longtext NOT NULL,  
  PRIMARY KEY (`id`,`session_key`),  
  UNIQUE KEY `id_UNIQUE` (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;
```

translations

```
CREATE TABLE `translations` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `lang_id` int(10) unsigned NOT NULL,  
  `phrase` text NOT NULL,  
  `translation` text NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `id_UNIQUE` (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=129 DEFAULT CHARSET=utf8;
```

users

```
CREATE TABLE `users` (  
  `user_id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `name` varchar(40) NOT NULL,  
  `surname` varchar(40) NOT NULL,  
  `username` varchar(40) NOT NULL,  
  `password` varchar(40) NOT NULL,  
  `email` varchar(40) NOT NULL,  
  `phone` varchar(20) NOT NULL,  
  `card_number` varchar(16) NOT NULL,  
  `card_type` int(1) unsigned NOT NULL,  
  `role_id` int(1) unsigned NOT NULL,  
  `country_code` char(2) NOT NULL,  
  PRIMARY KEY (`user_id`,`username`),  
  UNIQUE KEY `username_UNIQUE` (`username`)  
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;
```

users_managers

```
CREATE TABLE `users_managers` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `user_id` int(10) unsigned NOT NULL,  
  `manager_id` int(10) unsigned NOT NULL,  
  PRIMARY KEY (`id`,`user_id`,`manager_id`),  
  UNIQUE KEY `id_UNIQUE` (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;
```

viewtypes

```
CREATE TABLE `viewtypes` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `type` varchar(45) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `id_UNIQUE` (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;
```

ΠΑΡΑΡΤΗΜΑ Β

ΔΕΙΓΜΑ SESSION ARRAY ΕΦΑΡΜΟΓΗΣ

```
'94d6a856-d3f4-489a-a578-98a4bb97227d' =>
array (
  'guid' => '94d6a856-d3f4-489a-a578-98a4bb97227d',
  'dateFrom' => '09-09-2014',
  'dateTo' => '13-09-2014',
  'days' => 4,
  'city' => '124329',
  'country' => 'GR',
  'reqHotel' => '4',
  'finalHotel' => '3',
  'rooms' =>
array (
  1 =>
array (
  'roomNumber' => 1,
  'adt' =>
array (
  'num' => '2',
  'details' =>
array (
  1 =>
array (
  'name' => 'Maria',
  'surname' => 'Koumi',
  'gender' => 'F',
),
2 =>
array (
  'name' => 'Giorgos',
  'surname' => 'Papadopoulos',
  'gender' => 'M',
),
),
),
'chi' =>
array (
  'num' => '1',
  'details' =>
array (
  1 =>
array (
  'name' => 'Giannis',
  'surname' => 'Papadopoulos',
  'gender' => 'F',
),
),
),
'inf' =>
array (
  'num' => '0',
  'details' =>
array (
),
),
'roomType' => '7',
```

```
),
2 =>
array (
  'roomNumber' => 2,
  'adt' =>
array (
  'num' => '2',
  'details' =>
array (
  1 =>
array (
  'name' => 'Jane',
  'surname' => 'Smith',
  'gender' => 'F',
),
2 =>
array (
  'name' => 'John',
  'surname' => 'Doe',
  'gender' => 'M',
),
),
),
'chi' =>
array (
  'num' => '0',
  'details' =>
array (
),
),
'inf' =>
array (
  'num' => '0',
  'details' =>
array (
),
),
'roomType' => '10',
),
),
'contactDetails' =>
array (
  'name' => 'George',
  'surname' => 'Papadopoulos',
  'mail' => 'g.papadopoulos@mail.gr',
  'address' => 'Eth. Antistasews 2, Athens',
  'cityID' => '114071',
  'countryID' => 'GR',
  'phone' => '+30 210 1234 123',
),
),
)
```