



**ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

"Εφαρμογή Android για κρατήσεις δωματίων σε ξενοδοχεία και Motel"



των φοιτητών

**Ζαφειρόπουλου Κωνσταντίνου
Α.Μ. 103571**

**Ζησκάτα Αθανάσιου
Α.Μ. 103664**

Επιβλέπων καθηγητής: **Κλεφτούρης Δημήτριος**

Θεσσαλονίκη, 2017

ΠΡΟΛΟΓΟΣ

Η παρούσα πτυχιακή εργασία παρουσιάζει το θέμα της κράτησης δωματίων σε ξενοδοχεία και Motel για smartphones και tablets, χρησιμοποιώντας το περιβάλλον Android.

Έως τώρα έχουν γίνει ανάλογες εργασίες στον χώρο αυτόν όπως π.χ. το Booking.com, Trivago και άλλα παρόμοια, αλλά η εργασία αυτή καινοτομεί στο ότι δίνει έμφαση εκτός από τον ταξιδιώτη και στον ιδιοκτήτη ενός ξενοδοχείου δίνοντας του την δυνατότητα διαχείρισης της δήλωσης του ξενοδοχείου του.

Στο θεωρητικό μέρος γίνεται μία εισαγωγή στην εξέλιξη των κινητών συσκευών καθώς και στα λειτουργικά συστήματα που χρησιμοποιούν με έμφαση στο λειτουργικό σύστημα Android, βασική περιγραφή της λειτουργίας των βάσεων δεδομένων για την εφαρμογή τους στο σύστημα Android αλλά και η χρήση της γλώσσας προγραμματισμού PHP για την βοήθεια της στην επικοινωνία μεταξύ της εφαρμογής και της βάσης δεδομένων σε έναν server.

Στην παρουσίαση του πρακτικού μέρους της εργασίας γίνεται η περιγραφή της εφαρμογής με αναλυτικό τρόπο της χρήσης της τόσο εννοιολογικά όσο και με τη μορφή κώδικα, με τη χρήση του εργαλείου Android Studio και την βιβλιοθήκη του SDK (Standard Development Kit). Θα γίνει γνωστό πως η δημιουργία μίας τέτοιας εφαρμογής απο το μηδέν δεν είναι κάτι ακατόρθωτο, παρότι είναι σχετικά περίπλοκο. Βασική προϋπόθεση είναι η γνώση προγραμματισμού και ο τρόπος λειτουργίας των smartphones. Επίσης, παρουσιάζεται η χρήση της βάσης δεδομένων και η ορθή επικοινωνία της με την εφαρμογή μέσω κάποιων script χρησιμοποιώντας γλώσσα PHP.

ΠΕΡΙΛΗΨΗ

Η τεχνολογία εξελίσσεται με ταχύ ρυθμό και πλέον έχει μπει στην καθημερινότητά μας αρκετά. Σχεδόν όλοι σήμερα χρησιμοποιούμε κινητές συσκευές τελευταίας τεχνολογίας (smartphone, tablet, PDA) που μας βοηθάνε να έχουμε ένα πολυεργαλείο ανά πάσα ώρα και στιγμή. Είναι ένας συνδυασμός τηλεφώνου, φωτογραφικής μηχανής, σημειωματάρου και πολλών ακόμα. Έχει, δηλαδή, αντικαταστήσει πολλά διαφορετικά πράγματα που μπορεί κάποιος να χρησιμοποιούσε στην καθημερινότητά του και να δεσμεύανε αρκετό χώρο, σε ένα.

Όμως, έχει και δυνατότητες, οι οποίες βγήκαν στην επιφάνεια τα τελευταία χρόνια με την άνθιση της τεχνολογίας. Κάποιες από αυτές είναι η χρήση δορυφόρου (GPS) για την εντόπιση του χρήστη στον χάρτη, καθώς και η δυνατότητα σύνδεσης στο διαδίκτυο για πληροφορίες και ενημέρωση.

Η συγκεκριμένη εφαρμογή εκμεταλλεύεται αυτές τις δυνατότητες για να προσφέρει στον χρήστη μία ευκολία στην αναζήτηση και κράτηση ενός δωματίου σε ένα ξενοδοχείο ή ένα μοτέλ, ακόμη και σε μία πόλη που δεν έχει ξαναβρεθεί. Χρησιμοποιεί χάρτη, εντοπίζει τον χρήστη και κάνει αναζήτηση ξενοδοχείων στην περιοχή που αυτός βρίσκεται. Ακόμη, ο χρήστης μπορεί να δημιουργήσει λογαριασμό, όχι μόνο ως ταξιδιώτης, αλλά και ως ιδιοκτήτης ενός ξενοδοχείου. Στην περίπτωση αυτή, η εφαρμογή παρέχει τη δυνατότητα στο χρήστη διαχείρισης του ξενοδοχείου και των κρατήσεών του.

ABSTRACT

Technology is evolving rapidly and it has now come in our lives sufficiently. Today almost everyone use mobile devices (smartphone, tablet, PDA) that help us having a multi-tool at any time and everywhere. It's a combination of phone, camera, notebook and many more. It has replaced many different things that we can use in everyday life and saves us a lot of space.

But it has features that surfaced in recent years with the boom of technology. Some of them are using satellite (GPS) for the user's location on the map and the ability to connect to the internet for information and updates.

This application takes advantage of these features to provide the user with an easy search and reservation of a room in a hotel or a motel, even in a city that he has never been before. It uses map to identify the user on it and makes the hotel search in this area. Still offers the possibility to create an account not only as a traveler, but also as a hotel owner and the management of it.

ΕΥΧΑΡΙΣΤΙΕΣ

Ευχαριστούμε θερμά τον επιβλέποντα καθηγητή της εργασίας μας, κ. Κλεφτούρη Δημήτριο, που μας έδωσε την δυνατότητα εκπόνησης της παρούσας εργασίας, και μας βοήθησε στην περάτωση της. Η καθοδήγηση και η ανατροφοδότηση που μας παρείχε κατά τη διάρκεια των συναντήσεών μας ήταν ιδιαίτερα σημαντική για το τελικό περιεχόμενο και δομή της παρούσας εργασίας και εφαρμογής.

Επίσης, ευχαριστούμε όλους τους καθηγητές του Τμήματος Μηχανικών Πληροφορικής, οι οποίοι μας δίδαξαν και μας καθοδήγησαν όλα τα χρόνια των σπουδών μας, προσφέροντάς μας τις απαραίτητες γνώσεις, και βάσεις, προκειμένου να καταφέρουμε να αναπτύξουμε το δικό μας ολοκληρωμένο πληροφοριακό σύστημα και να εξελιχθούμε στον κλάδο αυτό.

Περιεχόμενα

ΠΡΟΛΟΓΟΣ	2
ΠΕΡΙΛΗΨΗ	3
ABSTRACT	4
ΕΥΧΑΡΙΣΤΙΕΣ	5
Περιεχόμενα	6
Κεφάλαιο 1ο: Εισαγωγή	11
1.1. Στόχος	11
1.2. Τεχνολογία	11
Κεφάλαιο 2: Έξυπνα Τηλέφωνα/ Smartphones	13
2.1 Τί είναι τα έξυπνα τηλέφωνα	13
2.2 Αναδρομή στην εξέλιξη των έξυπνων κινητών	13
2.3 Λειτουργικά συστήματα έξυπνων κινητών.....	14
2.3.1 <i>Android</i>	15
2.3.2 <i>IOS</i>	16
2.3.3 <i>Windows</i>	16
2.3.4 <i>Άλλα λειτουργικά συστήματα</i>	17
2.4 Υλικό (hardware) των έξυπνων κινητών	17
Κεφάλαιο 3: Το Λειτουργικό Σύστημα Android	20
3.1 Τί είναι το Android	20
3.2 Εκδόσεις Android	20
3.2.1 <i>Angel Cake (1.0)</i>	21
3.2.2 <i>Battenberg (1.1)</i>	21
3.2.3 <i>Cupcake (1.5)</i>	21
3.2.4 <i>Donut (1.6)</i>	21
3.2.5 <i>Eclair (2.0-2.1)</i>	22
3.2.6 <i>Froyo (2.2-2.2.3)</i>	22
3.2.7 <i>Gingerbread (2.3-2.3.7)</i>	22
3.2.8 <i>Honeycomb (3.0-3.2.6)</i>	22
3.2.9 <i>Ice Cream Sandwich (4.0-4.0.4)</i>	22

3.2.10 Jelly Bean (4.1-4.3.1).....	23
3.2.11 KitKat (4.4-4.4.4).....	23
3.2.12 Lollipop (5.0-5.0.2).....	23
3.2.13 Marshmallow (6.0).....	23
3.2.14 Nougat (7.0)	24
3.3 Η αρχιτεκτονική του Android.....	24
3.3.1 <i>Linux Kernel</i>	24
3.3.2 <i>Libraries</i>	25
3.3.3 <i>Android Runtime</i>	25
3.3.4 <i>Application Framework</i>	26
3.3.5 <i>Application</i>	26
3.4 Ασφάλεια στο Android	26
3.5 Δομή ενός android project στο android studio.....	28
3.5.1 <i>Δημιουργία του project</i>	28
3.5.2 <i>Επεξήγηση των στοιχείων</i>	31
Κεφάλαιο 4ο: Γλώσσα Προγραμματισμού PHP	36
4.1 Τί είναι η PHP	36
4.2 Η ιστορία της PHP	36
4.3 Βασικά χαρακτηριστικά.....	37
4.4 Πλεονεκτήματα της PHP	38
Κεφάλαιο 5: Βάσεις δεδομένων	40
5.1 Εισαγωγή.....	40
5.2 Ιστορικά στοιχεία.....	40
5.3 Εξέλιξη των βάσεων δεδομένων και των ΣΔΒΔ	41
5.3.1 Βάσεις δεδομένων.....	41
5.3.2 ΣΔΒΔ	43
5.4 Δομή βάσης δεδομένων	44
5.5 Αρχιτεκτονική ΣΔΒΔ.....	45
5.6 Ασφάλεια δεδομένων.....	46
5.7 Περιγραφή της βάσης	48

5.7.1 Διάγραμμα οντοτήτων συσχετίσεων.....	48
5.7.2 Περιγραφή της βάσης	49
Κεφάλαιο 6ο: Περιγραφή της Εφαρμογής.....	57
6.1 Αρχική εικόνα.....	57
6.2 Χαρακτηριστικά εφαρμογής ως συνδεδεμένος χρήστης	62
6.2.1 Σύνδεση ως Owner.....	62
6.2.2 Σύνδεση ως Traveler	78
6.3 Εργαλεία που χρησιμοποιήθηκαν	90
Κεφάλαιο 7 Συμπεράσματα.....	92
7.1 Εφαρμογή και περιβάλλον Android.....	92
7.2 Προτάσεις για μελλοντική έρευνα/Μελλοντική επέκταση.....	93
Βιβλιογραφία	95
Παράρτημα Α΄: Κώδικας Android.....	96

Ευρετήριο εικόνων

Εικόνα 3.1 Η ιστορία των εκδόσεων του Android	21
Εικόνα 3.2 Τα πέντε επίπεδα του Android	24
Εικόνα 3.3 Αρχική οθόνη δημιουργίας android project	28
Εικόνα 3.4 Επιλογή πλατφόρμας android	29
Εικόνα 3.5 Επιλογή του πρώτου activity	30
Εικόνα 3.6 Ονοματοδοσία activity	31
Εικόνα 3.7 Ο φάκελος res	31
Εικόνα 3.8 Android manifest	33
Εικόνα 3.9 Ο φάκελος java	34
Εικόνα 3.10 Το αρχείο build.gradle	35
Εικόνα 6.1. Το εικονίδιο εφαρμογής “Get A Room”	57
Εικόνα 6.2. Θέση του εικονιδίου εφαρμογής μεταξύ άλλων, στην επιφάνεια εργασίας.	58
Εικόνα 6.3. Αρχική οθόνη εφαρμογής	58
Εικόνα 6.4. Η αρχική οθόνη για Traveler	59
Εικόνα 6.5. Η αρχική οθόνη για Owner	59
Εικόνα 6. 6. Οθόνη δημιουργίας λογαριασμού για Owner.	60
Εικόνα 6.7. Οθόνη δημιουργίας λογαριασμού για Traveler.	60
Εικόνα 6.8. Οθόνη Forgot Password	61
Εικόνα 6.9. Αρχική οθόνη	62
Εικόνα 6.10. Μπάρα επιλογών Owner (Navigation Drawer)	63
Εικόνα 6.11. Βασικές πληροφορίες χρήστη	65
Εικόνα 6.12. Running Reservations (Owner)	66
Εικόνα 6.13. Τα ξενοδοχεία του χρήστη (Owner)	68
Εικόνα 6.14. Μήνυμα αποτροπής προσθήκης ξενοδοχείου.	69
Εικόνα 6.15. Φόρμα συμπλήρωσης με τα στοιχεία ξενοδοχείου	70
Εικόνα 6.16. Φόρμα συμπλήρωσης με τα στοιχεία ξενοδοχείου (συνέχεια)	70
Εικόνα 6.17. Οθόνη «Information» (Owner)	71
Εικόνα 6.18. Οθόνη ρυθμίσεων	74
Εικόνα 6.19. Οθόνη κατάργησης λογαριασμού	75

Εικόνα 6.20. Οθόνη διαγραφής λογαριασμού	75
Εικόνα 6.21. Dialog με τα ξενοδοχεία του χρήστη ως επιλογή για περαιτέρω ενέργειες.	76
Εικόνα 6.22. Το ξενοδοχείο όπως φαίνεται στον χάρτη στο σημείο που το «καρφισώνει» ο χρήστης.	76
Εικόνα 6.23. Alert Dialog με τις πληροφορίες του επιλεγμένου ξενοδοχείου και την δυνατότητα αλλαγής τους	77
Εικόνα 6.24. Alert Dialog με την οθόνη ενημέρωσης για οριστική διαγραφή του ξενοδοχείου	77
Εικόνα 6.25. Alert Dialog για αποσύνδεση του χρήστη	78
Εικόνα 6.26. Μενού επιλογών «Traveler» (Navigation Drawer)	79
Εικόνα 6.27. Οθόνη «Manage Booking» (Traveler)	81
Εικόνα 6.28. Επιλογή ημερομηνίας	82
Εικόνα 6.29. Οθόνη έπειτα από επιλογή ημερομηνιών check in και check out	82
Εικόνα 6.30. Alert Dialog για συμπλήρωση στοιχείων κράτησης (Traveler)	84
Εικόνα 6.31. Λίστα με τις κρατήσεις του χρήστη	85
Εικόνα 6.32. Λίστα με τα αγαπημένα ξενοδοχεία του χρήστη	85
Εικόνα 6.33. Οθόνη Map με γενική εικόνα της Ελλάδας και τα ξενοδοχεία	86
Εικόνα 6.34. Οθόνη Map με την Θεσσαλονίκη και τα ξενοδοχεία της	86
Εικόνα 6.35. Μορφή χάρτη ως Satellite	87
Εικόνα 6.36. Μορφή χάρτη ως Terrain	87
Εικόνα 6.37. Alert Dialog Traveler Guide	88
Εικόνα 6.38. Οθόνη «Setting» (Traveler)	88

Κεφάλαιο 1ο: Εισαγωγή

Πρόλογος

Στο κεφάλαιο αυτό γίνεται περιγραφή του στόχου της παρούσας εργασίας καθώς και η επεξήγηση της φιλοσοφίας της. Επίσης, περιγράφεται η τεχνολογία που χρησιμοποιήθηκε

1.1. Στόχος

Στόχος της συγκεκριμένης εργασίας είναι η δημιουργία μιας ολοκληρωμένης εφαρμογής κρατήσεων δωματίου/ων σε ξενοδοχεία και Motel χρησιμοποιώντας περιβάλλον Android. Μέσα από την εργασία αυτή φαίνεται η προσπάθεια και ο κόπος που διατέθηκε ώστε να υλοποιηθεί ένα σύστημα το οποίο θα βοηθήσει πολλούς χρήστες στην εύρεση ξενοδοχείων, αλλά θα βοηθήσει και εμάς να κατανοήσουμε καλύτερα τις απαιτήσεις για την δημιουργία μια εφαρμογής σε περιβάλλον Android.

Η γνώση και η εμπειρία που αποκτήθηκε δουλεύοντας πάνω σε κάτι που ξεκίνησε από το μηδέν είναι ανεκτίμητη και αυτό που δημιουργήθηκε ίσως καταφέρει να γίνει μία μέρα νούμερο ένα στην αγορά! Επίσης, οι γνώσεις που αποκτήθηκαν είναι η παρακαταθήκη μας για ένα καλύτερο μέλλον.

Η εργασία αυτή, σαν φιλοσοφία, δεν διαφέρει από άλλες, πλέον γνωστές, εφαρμογές όπως το Booking.com, Trivago και DiscoverRooms. Σε αντίθεση με αυτές η διαφορά της είναι στην δυνατότητα χρήσης της και από ιδιοκτήτες ξενοδοχείων εκτός από τους ταξιδιώτες. Η βασική φιλοσοφία μπορεί να είναι παρόμοια με άλλες, αλλά διαφέρει και είναι μοναδική σε λογική και εμφάνιση διότι είναι φτιαγμένη από την αρχή μέχρι το τέλος από το μηδέν .

1.2. Τεχνολογία

Αρχικά θα εξεταστούν τα χαρακτηριστικά που απαιτούνται από την πλευρά του προγραμματιστή αλλά και του χρήστη. Η εφαρμογή υλοποιήθηκε σε υπολογιστή με περιβάλλον Windows 10. Έγινε χρήση του λογισμικού Eclipse το οποίο περιλαμβάνει το περιβάλλον ανάπτυξης εφαρμογών Java

IDE, το Android Development Tool (ADT), το Android SDK και το AVD Manager δηλαδή τα εργαλεία που διευκολύνουν την ανάπτυξη και την προβολή της Android εφαρμογής σε πολλές εκδόσεις της πλατφόρμας. Απαραίτητο επίσης είναι και το Java Development Kit (JDK) για να μπορεί να γίνει σωστά η ανάπτυξη της εφαρμογής με την γλώσσα προγραμματισμού Java. Απαραίτητες γνώσεις σε Java και XML χρειάζονται για να γίνει η υλοποίηση μιας τέτοιας εφαρμογής. Η εφαρμογή είναι προγραμματισμένη για περιβάλλοντα Android 4.4.2 (KitKat) και ανωτέρων. Επίσης η εφαρμογή είναι απενεργοποιημένη για landscape mode.

Επίλογος

Στο κεφάλαιο αυτό έγινε μία σύντομη περιγραφή της γενικής ιδέας της εργασίας. Στο επόμενο γίνεται περιγραφή των κινητών συσκευών και η πιθανή εξέλιξη τους σε βάθος χρόνου.

Κεφάλαιο 2: Έξυπνα Τηλέφωνα/ Smartphones

Πρόλογος

Το κεφάλαιο αυτό περιγράφει εκτενώς την χρήση των έξυπνων τηλεφώνων και την εξέλιξή τους στο πέρασμα του χρόνου. Αναλύει τα λειτουργικά συστήματα, άλλα πιο γνωστά στο ευρύ κοινό και άλλα λιγότερο, που χρησιμοποιούνται στα έξυπνα τηλέφωνα και, τέλος, δίνει αναφορά και στο υλικό κατασκευής των συσκευών για την σωστή υποστήριξη των λειτουργικών συστημάτων.

2.1 Τί είναι τα έξυπνα τηλέφωνα

Τα smartphones αποτελούν τη φυσική εξέλιξη των κλασικών συσκευών κινητής τηλεφωνίας. Δεν υπάρχει ένας γενικά αποδεκτός ορισμός τους, αλλά γενικά θα έλεγε κανείς πως ένα smartphone είναι μια συσκευή τηλεπικοινωνίας, η οποία έχει επιπρόσθετα τη δυνατότητα να πραγματοποιήσει κάποιες από τις εργασίες που εκτελούν οι προσωπικοί υπολογιστές, όπως την λήψη και αποστολή emails, την επεξεργασία κειμένων κ.λ.π. Τα smartphones είναι αποτέλεσμα της σύζευξης των κλασικών κινητών τηλεφώνων με τα Personal Digital Assistants (PDA), τα οποία ήταν στην πράξη ηλεκτρονικές φορητές ατζέντες, που μπορούσαν να επικοινωνήσουν με τον υπολογιστή για ανταλλαγή στοιχείων.

Η διαφορά των smartphone από τα υπόλοιπα κινητά είναι η ενσωμάτωση πλήρους λειτουργικού συστήματος, η οποία έχει τη δυνατότητα να βελτιώνεται συνεχώς, εγκαθιστώντας ενημερώσεις και εφαρμογές από τα αντίστοιχα market stores. Έτσι, το smartphone δύναται να προσαρμοσθεί στα μέτρα του καθενός, ώστε να γίνει ένα εργαλείο εργασίας, διασκέδασης, ενημέρωσης και ψυχαγωγίας.

2.2 Αναδρομή στην εξέλιξη των έξυπνων κινητών

Αν ανατρέξει κανείς στην ιστορική σημασία του όρου smartphone θα διαπιστώσει ότι ήδη από το 1973 είχε γίνει αντιληπτός ο συνδυασμός τηλεφώνου και υπολογιστή.

Το πρώτο κινητό τηλέφωνο που περιείχε λειτουργίες τηλεφώνου και PDA ήταν μια συσκευή της IBM, η οποία αναπτύχθηκε το 1992 με όνομα angler. Μια ανανεωμένη έκδοση του προϊόντος ανακοινώθηκε το επόμενο έτος από την

Bellsouth Cellular Crop με την ονομασία Simon και ήταν διαθέσιμο στο εμπόριο από τον Αύγουστο του 1994. Η Simon ήταν η πρώτη συσκευή που σωστά αναφέρεται ως έξυπνο τηλέφωνο αν και ο όρος δεν είχε επινοηθεί ακόμα. Η ικανότητα του ήταν να λαμβάνει τηλεφωνικές κλήσεις, να λαμβάνει και να στέλνει φαξ, e-mail από την οθόνη αφής του. Περιλάμβανε, επίσης, αρκετές εφαρμογές όπως ημερολόγιο, βιβλίο διευθύνσεων, αριθμομηχανή, ρολόι και ηλεκτρονικά σημειώματα.

Το 1996 η Nokia κυκλοφόρησε το Nokia 9000 το οποίο ήταν το πρώτο smartphone της εταιρίας. Συνδύαζε PDA με κινητό τηλέφωνο σε σχεδίαση clamshell. Η συσκευή αυτή διέθετε μια οθόνη στο εξωτερικό μέρος της και ένα αριθμητικό πληκτρολόγιο. Στο εσωτερικό πάνω μέρος του τηλεφώνου υπήρχε μια οθόνη ανάλυσης 640X200 pixel, ενώ στο κάτω μέρος υπήρχε ένα πληκτρολόγιο τύπου qwerty 1 . Με το μοντέλο αυτό ο χρήστης μπορούσε να έχει πρόσβαση στο διαδίκτυο να στέλνει και να λαμβάνει τα mail του. Ακολούθησε το 1997 η ericsson όπου πρώτη ονομάζει το κινητό “Penelope” ως smartphone.

Στις αρχές του 2001 η palm inc. εισήγαγε το Kyocera 6035, το πρώτο smartphone που χρησιμοποιήθηκε ευρέως στις ΗΠΑ. Αυτή η συσκευή συνδύαζε λειτουργίες PDA με αυτές ενός τηλεφώνου που λειτουργούσε στο ασύρματο δίκτυο της Verizon. Ο χρήστης είχε τη δυνατότητα να επιλέξει ένα όνομα από τη λίστα επαφών του PDA και η συσκευή θα καλούσε τον αριθμό της συσκευής αυτής.

Τα επόμενα χρόνια, καθώς η τεχνολογία εξελισσόταν, οι συσκευές γινόντουσαν μικρότερες σε μέγεθος, ενώ, παράλληλα, αυξανόταν η επεξεργαστική τους ισχύ και προσθέτονταν περισσότερες λειτουργικές δυνατότητες. Έτσι, φτάνοντας στα τέλη της δεκαετίας του 2000 εμφανίστηκαν νέα λειτουργικά συστήματα, τα οποία αναπτύχθηκαν για να είναι συμβατά με τα smartphone τελευταίας τεχνολογίας, όπου έδωσαν μια διαφορετική τροπή στην έννοια τους.

Από τις αρχές της δεκαετίας τους 2010 έως σήμερα τα smartphone έχουν κατακλύσει την αγορά κινητών τηλεφώνων και αποτελούν αναπόσπαστο κομμάτι της καθημερινής ζωής μας.

2.3 Λειτουργικά συστήματα έξυπνων κινητών

Ίσως το κυριότερο χαρακτηριστικό που ξεχωρίζει τα smartphones, πέραν της εμφάνισης, είναι το λειτουργικό σύστημα που χρησιμοποιούν. Ίσως να φαίνεται λίγο

περίεργο για τους χρήστες οι οποίοι δεν είχαν συνηθίσει να επιλέγουν κάποιο κινητό με κριτήριο το λογισμικό, εντούτοις αποτελεί μια σημαντική παράμετρο, η οποία προδιαγράφει τις πραγματικές δυνατότητες της συσκευής.

Το λειτουργικό σύστημα μπορεί να παρέχεται από εταιρία διαφορετική από την κατασκευάστρια της συσκευής. Μέσω αυτού, ο χρήστης μπορεί να εγκαταστήσει στην συσκευή του εφαρμογές, οι οποίες θα είναι συμβατές με το λειτουργικό. Αυτό σίγουρα είναι μια βελτίωση από άλλες τεχνολογίες στα κινητά, όπου λ.χ. κάποιες εφαρμογές σε Java μπορεί να μην έτρεχαν σε διαφορετικά μοντέλα κινητών. Μέσω του λειτουργικού συστήματος λοιπόν, μπορούν οι εφαρμογές να εκμεταλλευτούν την τεχνολογία που χρησιμοποιεί η συσκευή smartphone. Κάποιες κατασκευάστριες εταιρίες, όπως η Apple και RIM, χρησιμοποιούν τα δικά τους, «κλειστά» λειτουργικά, με αποτέλεσμα οι εφαρμογές που μπορούν να εγκατασταθούν να έχουν περιορισμούς. Αντίθετα, άλλες εταιρίες που χρησιμοποιούν λειτουργικά ανοικτού κώδικα όπως είναι το Android, Linux (το Symbian θα εξελιχθεί σύντομα σε λειτουργικό ανοικτού κώδικα), μπορούν να δώσουν περισσότερη δύναμη στον προγραμματιστή εφαρμογών για το smartphone.

2.3.1 Android

Το Android είναι λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας, το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Αρχικά, αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας τη συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google. Το Android είναι κατά κύριο λόγο σχεδιασμένο για συσκευές με οθόνη αφής, όπως τα έξυπνα τηλέφωνα και τα τάμπλετ, με διαφορετικό περιβάλλον χρήσης για τηλεοράσεις (Android TV), αυτοκίνητα (Android Auto) και ρολόγια χειρός (Android Wear). Παρόλο που έχει αναπτυχθεί για συσκευές με οθόνη αφής, έχει χρησιμοποιηθεί σε κονσόλες παιχνιδιών, ψηφιακές φωτογραφικές μηχανές, συνηθισμένους Η/Υ (π.χ. το HP Slate 21) και σε άλλες ηλεκτρονικές συσκευές.

Θα μελετήσουμε το android αναλυτικά στο 3^ο κεφάλαιο.

2.3.2 IOS

Το iOS (προηγουμένως iPhone OS) είναι ένα λογισμικό για κινητά, το οποίο αναπτύχθηκε και διανέμεται από την Apple Inc. Αρχικά, παρουσιάστηκε το 2007 για το iPhone, ενώ υποστηρίζει και άλλες συσκευές της Apple, όπως το iPod touch (Σεπτέμβριος 2007), το iPad (Ιανουάριος 2010) και το Apple TV, δεύτερης γενιάς, (Σεπτέμβριος 2010). Η Apple δε δίνει την άδεια για την εγκατάσταση του λογισμικού iOS σε συσκευές που δεν είναι κατασκευής Apple.

Αυτή τη στιγμή βρίσκεται στη δέκατη έκδοση του λογισμικού και έχει ανακοινωθεί η ενδέκατη που αναμένεται να κυκλοφορήσει το τελευταίο τρίμηνο του 2017. Μία σημαντική στιγμή ήταν το 2010 με την κυκλοφορία της έκδοσης 4.0, καθώς έδωσε 1500 νέα APIs στους δημιουργούς εφαρμογών και στις συσκευές που το έτρεχαν υπήρχε η δυνατότητα του multitasking.

Τα επόμενα χρόνια αφιερώθηκαν περισσότερο στη δημιουργία του λογισμικού για τα tablet και, έτσι, δεν υπήρχε κάποια σημαντική ενημέρωση μέχρι που ανακοινώθηκε η έκδοση 9.0 όπου έδωσε τη δυνατότητα στους χρήστες να παραμετροποιήσουν σε μεγάλο βαθμό το user interface.

2.3.3 Windows

Windows Phone είναι η έκδοση του Λειτουργικού Συστήματος Windows της Microsoft για έξυπνα κινητά τηλέφωνα. Αναπτύχθηκε στις αρχές του 2000, ενώ στις πρώτες εκδόσεις του είχε το όνομα Windows Mobile.

Η έναρξη είναι αρχική οθόνη των Windows Phone και αποτελείται από πλακίδια (live tiles) των οποίων τον αριθμό, το μέγεθος, το χρώμα και την εφαρμογή που ανοίγουν ορίζει ο χρήστης, ενώ επίσης μερικά πλακίδια ανανεώνονται με τις τελευταίες εξελίξεις, όπως είναι το Facebook, το email, ο καιρός και οι ειδήσεις.

Η μέτρηση στις εκδόσεις ξεκίνησε από το Windows Phone 7 και αυτή τη στιγμή βρίσκεται στην έκδοση 10. Η σημαντικότερη έκδοση ήταν η 8.1 καθώς η οθόνη έναρξης μπορεί να δεχτεί περισσότερα live tiles, όπως επίσης είναι δυνατό στα live tiles να μπει κάποια εικόνα για φόντο ώστε να αφαιρείται το μονότονο χρώμα, τα παιχνίδια του Xbox έχουν μπει στην λίστα της έναρξης για να τα βρίσκει ο χρήστης πιο γρήγορα, έχουν προστεθεί παραπάνω λειτουργίες στις ρυθμίσεις και

προστέθηκε κι ένα notification center που το οποίο ήταν μια σημαντική έλλειψη στις προηγούμενες εκδόσεις.

2.3.4 Άλλα λειτουργικά συστήματα

Σε αυτήν την υποενότητα γίνεται αναφορά σε κάποια λειτουργικά που πλέον δεν υποστηρίζονται από τις εταιρίες παραγωγής τους ή έχουν χάσει μεγάλο μερίδιο της αγοράς, αλλά είχαν κάνει αίσθηση όταν πρωτοεμφανίστηκαν.

Το BlackBerry OS είναι το λειτουργικό που δημιούργησε η ομώνυμη εταιρία για τις κινητές της συσκευές. Το σύστημα ήταν το πρώτο που παρείχε multitasking δυνατότητες και υποστήριζε εξειδικευμένες συσκευές εισόδου. Επίσης, οι αναβαθμίσεις του λογισμικού γίνονται μέσω της τεχνολογίας OTASL (over the air software loading).

Το Firefox OS ήταν ένα ελεύθερο λειτουργικό σύστημα για κινητά τηλέφωνα και ταμπλέτες (tablets). Αρχικά, ήταν γνωστό με την κωδική ονομασία: B2G - Boot to Gecko. Ήταν βασισμένο στο Linux και η ανάπτυξή του υποστηριζόταν από τη Mozilla, ένα μη κερδοσκοπικό οργανισμό, μέχρι τα μέσα του Ιανουαρίου του έτους 2017. Το Firefox OS είχε σχεδιαστεί για να παρέχει ένα εναλλακτικό λειτουργικό σύστημα για φορητές συσκευές, βασισμένο στην κοινότητα, χρησιμοποιώντας ανοιχτά πρότυπα και προσεγγίσεις όπως εφαρμογές HTML5, Javascript και ανοιχτά APIs για την επικοινωνία με το υλικό των συσκευών.

Τέλος, το λειτουργικό σύστημα Ubuntu Touch είναι η έκδοση για κινητά τηλέφωνα από τους δημιουργούς του ομώνυμου λειτουργικού για τους υπολογιστές. Σχεδιάστηκε, κυρίως, για συσκευές με οθόνη αφής.

2.4 Υλικό (hardware) των έξυπνων κινητών

Σήμερα, τα smartphones είναι εξοπλισμένα με περισσότερες δυνατότητες από ότι οι συσκευές της προηγούμενης γενιάς. Η προσθήκη επιπλέον χαρακτηριστικών έκαναν τα κινητά τηλέφωνα χρήσιμα στην καθημερινή μας ζωή και αυτό δικαιολογεί την αύξηση στις πωλήσεις των συγκεκριμένων συσκευών. Παρακάτω αναφέρονται αρκετά από τα χαρακτηριστικά τους και οι λειτουργίες που επιτελούν.

Βασικό χαρακτηριστικό είναι η οθόνης αφής, όπου λειτουργεί σαν πληκτρολόγιο και οθόνη μαζί. Η οθόνη είναι μεγαλύτερη και παρέχει καλύτερα

χρώματα και μεγαλύτερη ανάλυση. Η προσθήκη γραφικών τις κάνουν να ξεχωρίζουν και οι απεικονίσεις θυμίζουν πολύ ηλεκτρονικό υπολογιστή. Επίσης, δίνεται η δυνατότητα στους προγραμματιστές να δημιουργήσουν τρισδιάστατα γραφικά βασισμένα σε κάποιες βιβλιοθήκες.

Από τα smartphones δε θα μπορούσε να λείπει η ανάγκη για αποθήκευση πληροφοριών. Για το λόγο αυτό υπάρχει εγκατεστημένη μια ελαφριάς μορφής βάση δεδομένων, ώστε η αποθήκευση των δεδομένων να αποτελεί εύκολη υπόθεση για τον προγραμματιστή.

Η συνδεσιμότητα είναι εξίσου βασικό χαρακτηριστικό. Τα smartphones είναι εξοπλισμένα με τεχνολογίες Bluetooth, WiFi, GSM/UMTS, όπου παρέχουν στο χρήστη τη δυνατότητα διασύνδεσης με το διαδίκτυο.

Η υποστήριξη πολυμέσων είναι, επίσης, πολύ σημαντική σε μια συσκευή που είναι "όλα σε ένα", για αυτό υποστηρίζει κάποιες από τις πιο γνωστές μορφές εικόνας, ήχου και βίντεο όπως MP3, WAV, JPEG, PNG, GIF, H.263, H.264. Η εγγραφή ήχου και βίντεο είναι από τα βασικά κομμάτια ενός smartphone.

Ακόμα ένα χαρακτηριστικό που δε θα μπορούσε να λείπει από ένα τέτοιο κινητό τηλέφωνο είναι η κάμερα, η οποία λειτουργεί σα ψηφιακή μηχανή. Σε πολλά από τα νέα μοντέλα υπάρχει κάμερα και στο μπροστινό μέρος, ώστε η επικοινωνία να γίνεται καλύτερη από ποτέ.

Τα smartphones διαθέτουν, μεταξύ άλλων, και επιπλέον υποστήριξη hardware. Μάλιστα, έχουν ενσωματωμένο GPS για τον εύκολο προσανατολισμό του χρήστη, ψηφιακή πυξίδα, βαρόμετρο, όπου μετριέται η ατμοσφαιρική πίεση, καθώς και αισθητήρες επιτάχυνσης, όπου λαμβάνουν τις κινήσεις της συσκευής και εκτελούν κάποιες λειτουργίες.

Ιδιαίτερα σημαντικό χαρακτηριστικό των smartphones είναι πως ο χρήστης έχει τη δυνατότητα να "κατεβάσει" εφαρμογές της επιλογής του. Οι εφαρμογές αυτές μπορούν να "ενισχύσουν" τη συσκευή, ώστε να αποδίδει τα μέγιστα σε κάθε υλικό μέρος της, κάνοντας την, έτσι, ένα πολύτιμο εργαλείο της καθημερινότητας του.

Προκειμένου να λειτουργήσουν όλα τα παραπάνω απαραίτητη προϋπόθεση είναι η ύπαρξη καλών επεξεργαστών, όπως, επίσης, και αρκετής μνήμης (ram), ώστε να μην παρουσιάζονται προβλήματα στη συσκευή.

Επίλογος

Σε αυτό το κεφάλαιο αυτό περιεγράφηκε η χρήση και η τεχνολογία των έξυπνων τηλεφώνων και αναλύθηκαν τα λειτουργικά συστήματα που αυτά χρησιμοποιούν. Στο επόμενο κεφάλαιο επιχειρείται μία επεξήγηση των λειτουργικών συστημάτων με έμφαση στο λειτουργικό σύστημα Android.

Κεφάλαιο 3: Το Λειτουργικό Σύστημα Android

Πρόλογος

Το κεφάλαιο αυτό εστιάζει και εμβαθύνει στην περιγραφή του λειτουργικού συστήματος Android ξεκινώντας με ιστορική ανάλυση και επεξηγώντας την αρχιτεκτονική που υπάρχει πίσω από αυτό, σε επίπεδο επεξεργαστή. Επίσης, εξηγεί την ασφάλεια που υπάρχει πίσω από κάθε εφαρμογή μέσω της δικής της εικονικής μηχανής.

3.1 Τί είναι το Android

Το Android είναι ένα λειτουργικό σύστημα που ενσωματώνεται σε συσκευές κινητής τηλεφωνίας, τα οποία διαθέτουν οθόνη αφής, τρέχουν τον πυρήνα (kernel) του λειτουργικού Linux. Όπως ήδη προαναφέρθηκε, επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με τη χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας τη συσκευή μέσω βιβλιοθηκών λογισμικού της Google.

Ένα από τα θετικά με τις συσκευές Android είναι ότι είναι multimedia και multitasking. Αυτό σημαίνει πως δίνουν τη δυνατότητα αφενός να γίνει αναπαραγωγή πολλαπλών μέσων και αφετέρου εκτέλεσης πολλών εφαρμογών ταυτόχρονα. Επιπλέον, το web browsing στο Android είναι ταχύτατο, υποστηρίζεται από flash και υπάρχουν πολλοί browsers ακολουθώντας τα βήματα στους προσωπικούς υπολογιστές. Ανεξάρτητα από το κόστος, όλες οι συσκευές Android διαθέτουν GPS και Wi-fi, δικαιώνοντας, έτσι, το βασικό λόγο δημιουργίας του εν λόγω λειτουργικού συστήματος που δεν είναι άλλος παρά η ανεμπόδιστη και εύκολη πρόσβαση στο διαδίκτυο, σε συνδυασμό με ένα πλήθος εφαρμογών (apps), όπως χάρτες, αναζήτηση, chat και e-mail, που επιτρέπουν στο χρήστη να μένει διαρκώς δικτυωμένος και ενημερωμένος.

3.2 Εκδόσεις Android

Το Android έχει δημιουργηθεί πάνω στο λειτουργικό σύστημα Linux, αλλά προσαρμοσμένο πάνω σε κινητές συσκευές. Ξεκινώντας από το 2008 με την κυκλοφορία του πρώτου Android smartphone μέχρι σήμερα έχουν βγει αρκετές εκδόσεις. Χαρακτηριστικό γνώρισμα όλων είναι ότι έχουν πάντα ονόματα κάποιου

γλυκού είτε επιδόρπιου και τα πρώτα γράμματα ακολουθούν την αλφαβητική σειρά, όπως φαίνεται στην επόμενη εικόνα:



Εικόνα 3. 1 : Η ιστορία των εκδόσεων του Android

3.2.1 Angel Cake (1.0)

Η πρώτη έκδοση του λειτουργικού συστήματος κυκλοφόρησε στις 23 Σεπτεμβρίου το 2008 μαζί με την προωθητική συσκευή HTC Hero.

3.2.2 Battenberg (1.1)

Κυκλοφόρησε το Φεβρουάριο του 2009. Δεν ήταν μεγάλη αναβάθμιση καθώς διορθώθηκαν μερικά σφάλματα. Επίσης, το λογισμικό βελτιστοποιήθηκε για να τρέχει πιο ομαλά.

3.2.3 Cupcake (1.5)

Κυκλοφόρησε την άνοιξη του 2009 και ήταν η αρχή των ονομάτων με γλυκά. Είναι μια μεγάλη αναβαθμισμένη πλατφόρμα για τις συσκευές με λειτουργικό Android που κυκλοφόρησε το Μάιο του 2009. Η έκδοση περιλαμβάνει νέα χαρακτηριστικά για τους χρήστες και τους προγραμματιστές, καθώς και οι αλλαγές στο Android API.

3.2.4 Donut (1.6)

Η συγκεκριμένη έκδοση κυκλοφόρησε τον Σεπτέμβριο του 2009. Σε αυτή την αναβάθμιση εισήχθη η υποστήριξη CDMA smartphones, επιπλέον μεγέθη οθόνης, περιστροφή οθόνης και ένας δείκτης μπαταρίας.

3.2.5 Eclair (2.0-2.1)

Τον Οκτώβριο του 2009 κυκλοφόρησε, μόλις ένα μήνα μετά το Donut. Οι σημαντικές αλλαγές που ενσωματώθηκαν σε αυτή την έκδοση ήταν η υποστήριξη της HTML5, νέο user interface, ψηφιακό ζουμ και υποστήριξη της χρήσης φλας από κατασκευής στην κάμερα, δυνατότητα αντίληψης multi-touch, εξελιγμένο πληκτρολόγιο που μαθαίνει από την χρήση λέξεων και Bluetooth 2.1 .

3.2.6 Froyo (2.2-2.2.3)

Τα αποκαλυπτήρια έγιναν στις 20 Μαΐου του 2010, κατά τη διάρκεια του συνεδρίου Google I/O 2010. Μία από τις σημαντικότερες αλλαγές ήταν η υποστήριξη σύνδεσης της συσκευής με USB. Επίσης, προστέθηκε η δυνατότητα δημιουργίας Wi-Fi Hotspot και η υποστήριξη για το Android Cloud στα Μηνύματα (C2DM) υπηρεσία συσκευής, επιτρέποντας τα Push Notifications.

3.2.7 Gingerbread (2.3-2.3.7)

Κυκλοφόρησε τον Δεκέμβριο του 2010 και είναι μία από τις πιο διαδεδομένες εκδόσεις. Οι βελτιώσεις περιλάμβαναν υποστήριξη πολλών καμερών και μεγαλύτερη ανάλυση οθόνης. Επίσης προστέθηκε προεγκατεστημένη υποστήριξη για τηλεφωνικές κλήσεις μέσω ίντερνετ (VoIP), λειτουργίες copy-paste σε όλο το λειτουργικό και έδωσε στους προγραμματιστές μεγαλύτερες δυνατότητες.

3.2.8 Honeycomb (3.0-3.2.6)

Αυτή η έκδοση κυκλοφόρησε τον Ιούλιο του 2011 και ήταν σχεδιασμένη αποκλειστικά για tablets. Οι βελτιώσεις της περιλαμβάνουν γρήγορη πρόσβαση σε χαρακτηριστικά της κάμερας, καλύτερο πληκτρολόγιο κατάλληλο για μεγάλες οθόνες, εκτέλεση πολλαπλών λειτουργιών και εύκολη μετάβαση από την μια στην άλλη.

3.2.9 Ice Cream Sandwich (4.0-4.0.4)

Στα μέσα του Οκτώβρη του 2011 δημοσιεύτηκε η έκδοση 4.0.1 με το όνομα Ice Cream Sandwich. Θεωρητικά αυτή η έκδοση είναι συμβατή με οποιοδήποτε συσκευή υποστηρίζει το Android 2.3.x. Η δυνατότητα χρήσης “μαλακών” κουμπιών δηλαδή των κουμπιών πάνω στην οθόνη (πίσω, αρχική, ενεργές εφαρμογές) είναι

πλέον πραγματικότητα καθώς μέχρι τότε όλα τα κινητά είχαν εξωτερικά κουμπιά για αυτές τις λειτουργίες.

3.2.10 Jelly Bean (4.1-4.3.1)

Τον Ιούνιο του 2012 κυκλοφόρησε και αποτελεί την καλύτερη έκδοση του Android μέχρι σήμερα. Το περιβάλλον χρήσης και η απόκρισή του είναι πιο γρήγορη και καλοφτιαγμένη από ποτέ ενώ περιλαμβάνει πάρα πολλές μικρές βελτιώσεις σε όλο το σύστημα, όπως για παράδειγμα στην κάμερα και στην χρήση φωνής για υπαγόρευση κειμένου.

3.2.11 KitKat (4.4-4.4.4)

Η συγκεκριμένη έκδοση δημοσιεύτηκε τον Οκτώβριο του 2013. Το σημαντικό σε αυτήν την έκδοση είναι η μειωμένη κατανάλωση της μνήμης RAM κάτι που επέτρεψε low-end μοντέλα να πάρουν τη συγκεκριμένη αναβάθμιση.

3.2.12 Lollipop (5.0-5.0.2)

Κυκλοφόρησε το Νοέμβριο του 2014 και ήταν διαθέσιμο για συγκεκριμένες συσκευές που εξυπηρετούνταν από την google όπως τα Nexus. Μία βασική αλλαγή που έγινε είναι το πλήρως επανασχεδιασμένο Material Design και ανανεωμένο User interface έτσι ώστε να παρέχει την βέλτιστη απόδοση για χρήστες με μικρές ποσότητες μνήμης RAM.

3.2.13 Marshmallow (6.0)

Το Android 6.0 Marshmallow κυκλοφόρησε τον Οκτώβριο του 2015 υπάρχουν ενδιαφέρουσες προσθήκες και αλλαγές τόσο στην εμφάνιση, όσο και στη λειτουργικότητα. εισήγαγε μια νέα αρχιτεκτονική στα δικαιώματα των εφαρμογών, νέα APIs για συμφραζόμενα, ένα νέο σύστημα διαχείρισης ενέργειας που μειώνει την δραστηριότητα του παρασκηνίου όταν μια συσκευή δεν χρησιμοποιείται, εγγενή υποστήριξη για την αναγνώριση δακτυλικών αποτυπωμάτων και USB τύπου-C, τη δυνατότητα να μεταφοράς των δεδομένων και τις εφαρμογές σε μια κάρτα microSD.

3.2.14 Nougat (7.0)

Κυκλοφόρησε επίσημα τον Αύγουστο του 2016 και οι συσκευές τις Nexus ήταν οι πρώτες που πήραν το καινούργιο λογισμικό. Η έκδοση αυτή εισάγει μια λειτουργία split-screen για κινητά τηλέφωνα, όπου δύο εφαρμογές μπορούν ταυτόχρονα να μοιράζονται την οθόνη. Μια κρυφή λειτουργία για παραπάνω απο δύο εφαρμογές ταυτόχρονα στην οθόνη είναι επίσης διαθέσιμη.

3.3 Η αρχιτεκτονική του Android

Το λειτουργικό σύστημα Android λειτουργεί σαν μια στοίβα, η οποία σε κάθε επίπεδό της περιέχει μια πληθώρα προγραμμάτων για την ορθή και σωστή χρήση της συσκευής και των εξαρτημάτων της. Συγκεκριμένα, αποτελείται από πέντε επίπεδα τα οποία είναι: Linux Kernel, Libraries, Android Runtime, Application framework, Application.



Εικόνα 3.2 Τα πέντε επίπεδα του Android

3.3.1 Linux Kernel

Το Android έχει χτιστεί πάνω στον Linux Kernel και επομένως χρησιμοποιείται Linux Power Management για την επικοινωνία με το hardware της συσκευής. Συγκεκριμένα χρησιμοποιείται Linux για τους drivers της συσκευής, την διαχείριση

μνήμης, την διαχείριση διεργασιών και τη δικτύωση. Ωστόσο, ο προγραμματιστής εφαρμογών δεν θα χρειαστεί ποτέ να προγραμματίσει σε αυτό το επίπεδο.

3.3.2 Libraries

Στο επόμενο επίπεδο βρίσκονται οι βιβλιοθήκες. Το Android περιλαμβάνει ένα σετ από βιβλιοθήκες, που χρησιμοποιούνται από διάφορα components του συστήματος και είναι γραμμένες σε C/C++. Αυτές οι δυνατότητες διαθέτονται στους developers μέσα από το Android Application Framework.

Μερικές από τις βασικές βιβλιοθήκες είναι οι παρακάτω: α) System C library – μια ενσωμάτωση της standard βιβλιοθήκης συστήματος της C(libc) τροποποιημένη για κινητές συσκευές βασισμένες στο Linux, β) Βιβλιοθήκες Πολυμέσων – υποστηρίζουν αναπαραγωγή και εγγραφή πολλών δημοφιλών μέσων ήχου και εικόνας, όπως: MPEG4, H.264, MP3, AAC, AMR, JPG και PNG, γ) Surface Manager – διαχειρίζεται την πρόσβαση στο υποσύστημα προβολής, και συνθέτει πρόσκοπτα δισδιάστατα και τρισδιάστατα επίπεδα γραφικών τα οποία προέρχονται από πολλαπλές εφαρμογές, δ) LibWebCore – μια μοντέρνα μηχανή υποστήριξης πλοήγησης στο διαδίκτυο (browserengine) η οποία χρησιμοποιείτε και από τον ενσωματωμένο browser του Android αλλά και από τις WebViews που ενσωματώνονται στις εφαρμογές, ε) SGL – η γνωστή μηχανή δισδιάστατων γραφικών, στ) Βιβλιοθήκες 3D – μια υλοποίηση βασισμένη στα APIs του OpenGL ES 1. Οι βιβλιοθήκες χρησιμοποιούν είτε τρισδιάστατη επιτάχυνση υλικού, όπου αυτή είναι διαθέσιμη, είτε μια υψηλά βελτιωμένη τρισδιάστατη επιτάχυνση λογισμικού σε περίπτωση που η πρώτη δεν είναι διαθέσιμη. FreeType – παρέχει ευκρίνεια γραφικών στα bitmaps και τις γραμματοσειρές των εφαρμογών του συστήματος., ζ) SQLite – μια πανίσχυρη και συνάμα πολύ ελαφριά σχεσιακή βάση δεδομένων Android Runtime.

3.3.3 Android Runtime

Στο αμέσως επόμενο επίπεδο βρίσκεται το Android Runtime, το οποίο αποτελείται από μια εικονική μηχανή (Dalvik Virtual Machine) και τις βιβλιοθήκες πυρήνα.

Dalvik είναι μια εικονική μηχανή (VM) που σχεδιάστηκε και γράφτηκε από τον Dan Bornstrein στη Google. Ο κώδικας της εφαρμογής μεταγλωττίζεται σε εντολές

μηχανής που ονομάζονται `bytecodes` και στη συνέχεια εκτελούνται από την VM στο κινητό τηλέφωνο.

Παρόλο που το `format` των `bytecodes` είναι λίγο διαφορετικό, η Dalvik είναι ουσιαστικά μια Java VM σχεδιασμένη για χαμηλές απαιτήσεις σε μνήμη. Επιτρέπει πολλαπλά `instances` της VM να τρέχουν ταυτόχρονα και εκμεταλλεύεται το Linux για τον τομέα της ασφάλειας και της διαχείρισης εφαρμογών.

Η Dalvik VM διαφέρει από την παραδοσιακή Java σε δυο σημεία. Η πρώτη τρέχει `.dex` αρχεία, τα οποία έχουν προέλθει κατά την ώρα της μεταγλώττισης από τα `.class` και `.jar` αρχεία. Τα `.dex` αρχεία είναι πιο συμπαγή και αποδοτικά από τα `.class` το οποίο είναι στόχος για συσκευές περιορισμένης μνήμης μπαταρίας. Η δεύτερη διαφορά έχει να κάνει με το ότι οι Java βιβλιοθήκες πυρήνα του Android είναι διαφορετικές από τις Java Standard Edition (Java SE) βιβλιοθήκες, αλλά και από τις Java Mobile Edition (Java ME) βιβλιοθήκες.

3.3.4 Application Framework

Το επόμενο επίπεδο είναι το Application framework, το οποίο περιέχει κώδικες υψηλότερου επιπέδου, που διαχειρίζονται όλες τις βασικές λειτουργίες της κινητής συσκευής και μπορούν να χρησιμοποιηθούν από εφαρμογές. Οι προγραμματιστές εφαρμογών έχουν πλήρη πρόσβαση σε αυτούς τους κώδικες όπως επίσης και στο υλικό μέρος και τις δυνατότητές του. Στην ουσία οι εφαρμογές είναι εξελεγμένα εργαλεία κατασκευασμένα από τα βασικά εργαλεία.

3.3.5 Application

Το τελευταίο στρώμα στην αρχιτεκτονική του Android είναι τα applications και είναι αυτά που αντιλαμβάνεται ο χρήστης. Πρόκειται για εφαρμογές όπως email client, πρόγραμμα για SMS, ημερολόγιο, χάρτες, browser και άλλα. Όλες οι εφαρμογές είναι προγραμματισμένες σε Java. Μια Java εφαρμογή για Android δεν είναι συμβατή με προγράμματα Java γραμμένα για Java SE και Java ME πλατφόρμες.

3.4 Ασφάλεια στο Android

Τη στιγμή που μία εφαρμογή εγκαθίσταται στη συσκευή λειτουργεί στη δικιά της εικονική μηχανή, η οποία αποτελεί και το πλαίσιο ασφαλείας (sandbox) της

εφαρμογής. Το android είναι ένα λειτουργικό σύστημα πολλών χρηστών στο οποίο η κάθε εφαρμογή αντιμετωπίζεται σα διαφορετικός χρήστης. Από προεπιλογή το σύστημα δίνει ένα μοναδικό αριθμό ID στην εφαρμογή ο οποίος της είναι άγνωστος. Το σύστημα αναθέτει συγκεκριμένες άδειες χρήσης στα αρχεία της εφαρμογής και μόνο εφαρμογές με το ίδιο ID έχουν πρόσβαση σε αυτά. Επιπλέον, κάθε εφαρμογή τρέχει τη δική της εικονική μηχανή (VM) απομονωμένη από τις υπόλοιπες εφαρμογές. Η κάθε VM εκκινεί μόλις ζητηθεί από το σύστημα και κλείνει είτε επειδή δεν χρησιμοποιείται πλέον είτε το σύστημα θέλει να ελευθερώσει πόρους του συστήματος για να χρησιμοποιηθούν από κάποια άλλη εφαρμογή.

Με αυτό τον τρόπο το Android χρησιμοποιεί την αρχή των ελάχιστων δικαιωμάτων. Η κάθε εφαρμογή έχει πρόσβαση μόνο σε όσους πόρους χρειάζεται και αυτό καθορίζεται από το AndroidManifest. Οι πόροι και τα δικαιώματα που χρειάζονται γνωστοποιούνται στο χρήστη πριν την εγκατάσταση και μπορεί να επιλέξει εάν συμφωνεί ή όχι.

Δυστυχώς, όπως και με κάθε άλλο παραγωγικό σύστημα, η ασφάλεια του Android είναι ένα μεγάλο ζήτημα που παραμένει ανοικτό και η θωράκισή του από εξωγενείς κακοήθεις παράγοντες δεν έχει καταστεί απόλυτα επιτυχής. Αποτέλεσμα αυτού είναι να έχουν υπάρξει διάφορα και σημαντικά κρούσματα υποκλοπής στοιχείων και ανεξέλεγκτης χρέωσης των τελικών χρηστών τα τελευταία χρόνια.

Οι πηγές του κακού εντοπίζονται σε τέσσερα σημεία. Αρχικά, στην έλλειψη ελέγχου κατά τη διάρκεια υποβολής ενός προγράμματος στο Play Store. Ενώ η πολιτική αυτή βοηθάει στο να μειωθούν οι απαιτήσεις δημοσίευσης εφαρμογών από του νομότυπους προγραμματιστές/ εκδότες, ταυτόχρονα μπορεί να αποτελέσει σημείο εκμετάλλευσης για όσους θέλουν να διαθέσουν κακόβουλες και ζημιογόνες εφαρμογές. Ίσως μια πολύ γρήγορη και τυπική διαδικασία ανθρωπίνου ελέγχου από την Google να βοηθούσε στο να μειωθεί η έκταση αυτού του φαινομένου. Σημειώνεται ότι το Play Store έχει ήδη ένα αυτοματοποιημένο σύστημα ελέγχου εφαρμογών, γνωστό ως "Bouncer", εντούτοις, εκ των πραγμάτων η ικανότητά του στην πρόληψη αμφισβητείται. Επιπλέον, ο μηχανισμός ασφαλείας του Android μερικές φορές δεν γίνεται απόλυτα κατανοητός από τους προγραμματιστές ή γίνεται κατάχρηση του μηχανισμού των δικαιωμάτων, με αποτέλεσμα ο χρήστης να συμφωνεί και να παραχωρεί πρόσβαση σε περισσότερα πράγματα απ' ό,τι θα επιθυμούσε. Ένα ακόμα τέτοιο σημείο είναι οι «Κερκόπορτες» (backdoors) από κατασκευαστές κινητών Android, οι οποίες ανακαλύπτονται και αξιοποιούνται από

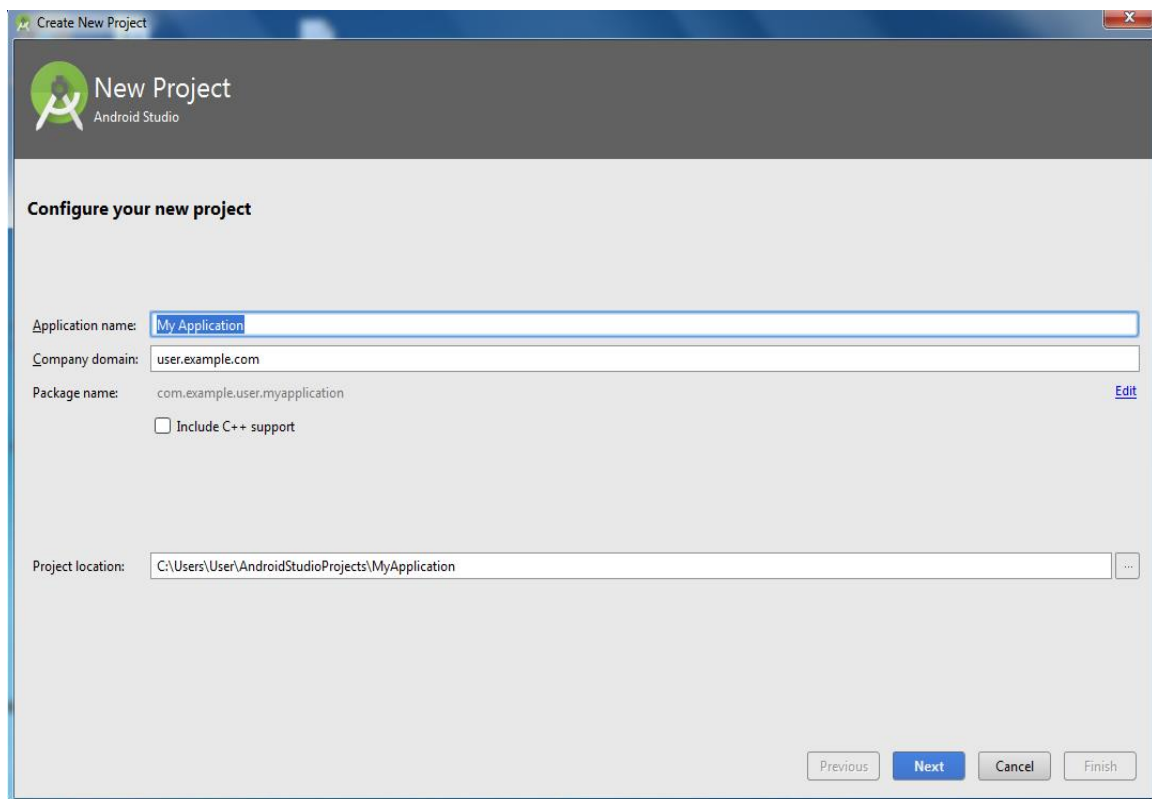
διάφορους χάκερ, ενώ, τέλος, και τα «κενά ασφαλείας» του Android που επιτρέπουν στους χρήστες να γίνουν υπερχρήστες (root) και να αποκτήσουν ελεύθερη πρόσβαση σε όλο το λειτουργικό σύστημα.

3.5 Δομή ενός android project στο android studio

Σε αυτή την υποενότητα θα μελετηθεί η διαδικασία δημιουργίας ενός android project και τι σημαίνουν οι φάκελοι που δημιουργούνται μέσα από το περιβάλλον του android studio το οποίο αποτελεί πλέον το επίσημο λογισμικό ανάπτυξης android.

3.5.1 Δημιουργία του project

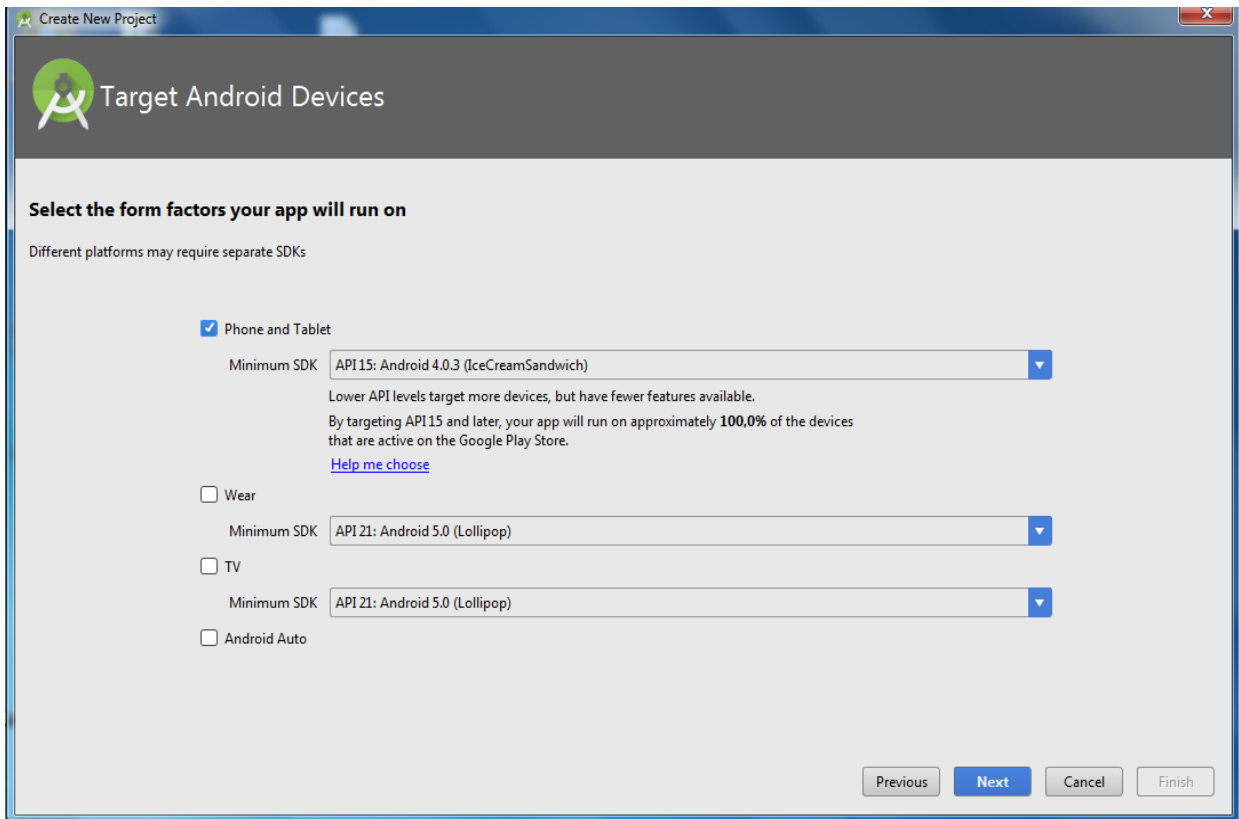
Όταν ο προγραμματιστής θα επιλέξει τη δημιουργία ενός android project θα δει την παρακάτω εικόνα:



Εικόνα 3.3 Αρχική οθόνη δημιουργίας android

- Στο πεδίο Application name μπαίνει το όνομα της εφαρμογής.
- Στο πεδίο Company name μπαίνει το όνομα της εταιρείας ή του οργανισμού.
- Το πεδίο package name δεν αλλάζει καθώς δημιουργείται αυτόματα σε συνάρτηση με το application name.
- Στο checkbox εάν το επιλέξουμε μπορούμε να προσθέσουμε την υποστήριξη κώδικα στη γλώσσα C++.
- Στο πεδίο Project location τοποθετείται η διαδρομή στην οποία θα αποθηκευτεί το android project μέσα στον υπολογιστή στον οποίο δημιουργείται.

Αφού επιλέξει τα επιθυμητά στοιχεία ο χρήστης πρέπει να πατήσει στο πλήκτρο Next. Κάνοντας κλικ στο πλήκτρο next εμφανίζεται η παρακάτω εικόνα:



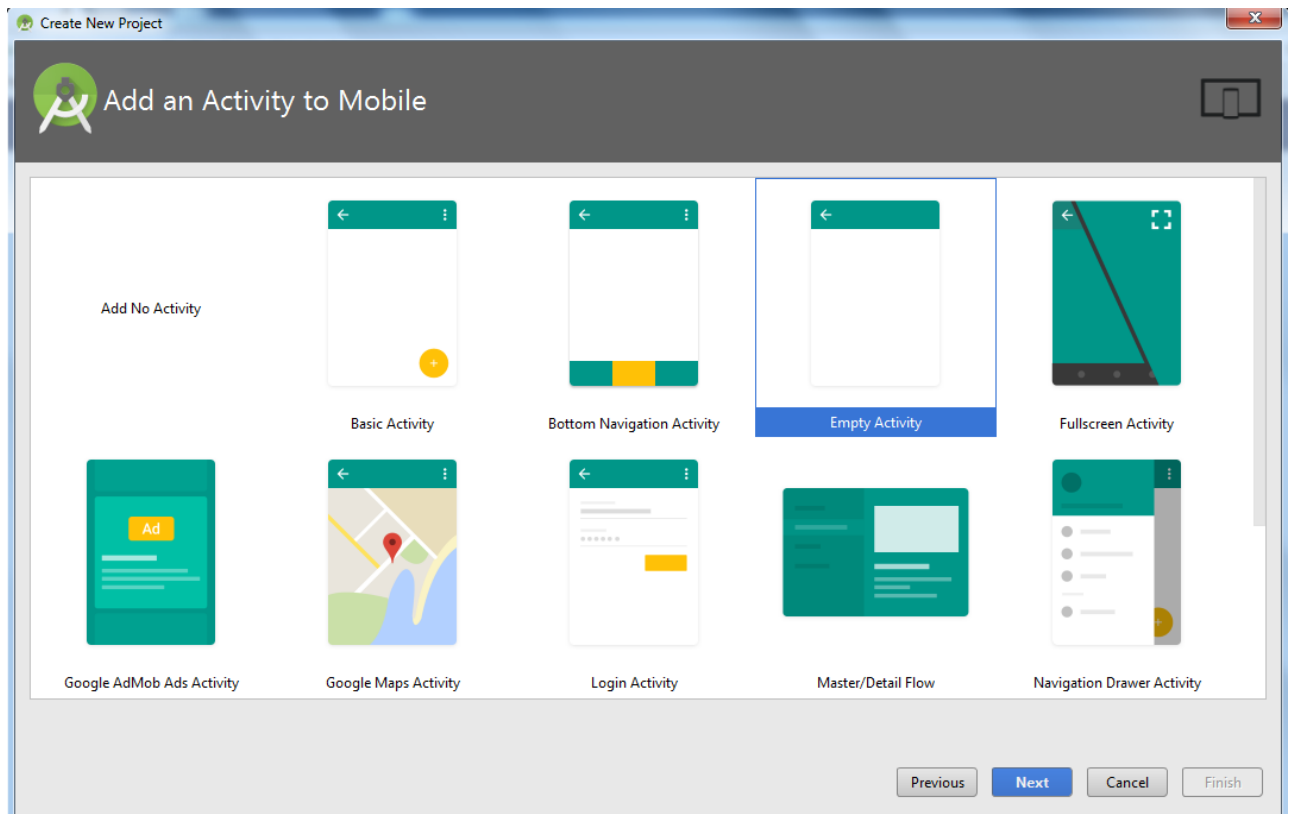
Εικόνα 3.4 Επιλογή πλατφόρμας android

Σε αυτή την εικόνα ο χρήστης μπορεί να επιλέξει σε ποια πλατφόρμα android θα αναπτύξει την εφαρμογή του. Έχει τη δυνατότητα να διαλέξει μεταξύ :

- Κινητά και tablet
- Ρολόγια android
- Τηλεοράσεις android
- Αυτοκίνητα android

Σε αυτές τις επιλογές διαλέγει και την ελάχιστη έκδοση στην οποία θα μπορεί να δουλέψει η εφαρμογή.

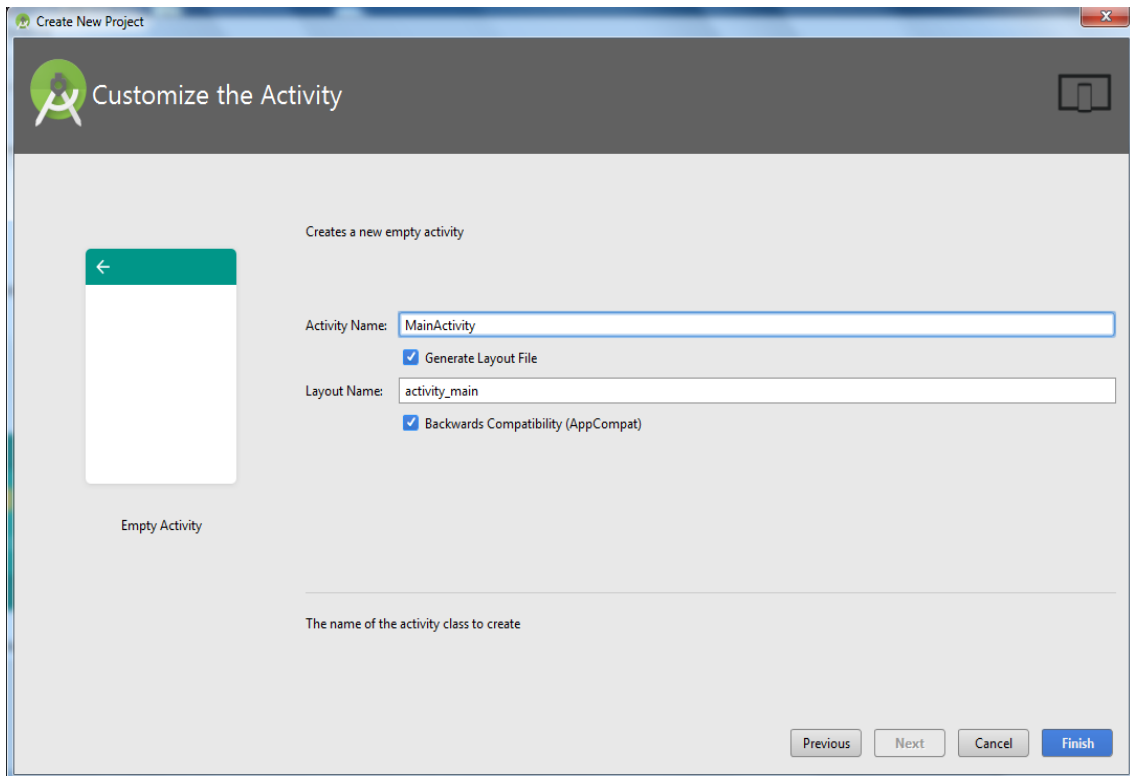
Αφού επιλέξει τα επιθυμητά στοιχεία ο χρήστης πρέπει να πατήσει στο κουμπί Next. Κάνοντας κλικ στο κουμπί next εμφανίζεται η παρακάτω εικόνα:



Εικόνα 3.5 Επιλογή του πρώτου activity

Εδώ ο προγραμματιστής διαλέγει ένα αρχικό activity για να διευκολύνει τη σχεδίαση της εφαρμογής καθώς υπάρχουν αρκετά activities που έχουν μία προσχεδιασμένη λειτουργικότητα. Ως επί των πλείστον επιλέγεται το Empty activity όπως φαίνεται στην Εικόνα 3.5.

Τέλος, όπως φαίνεται ακολούθως ο χρήστης θα πρέπει να ονομάσει την activity που επέλεξε πιο πριν για να δημιουργηθεί η κλάση.

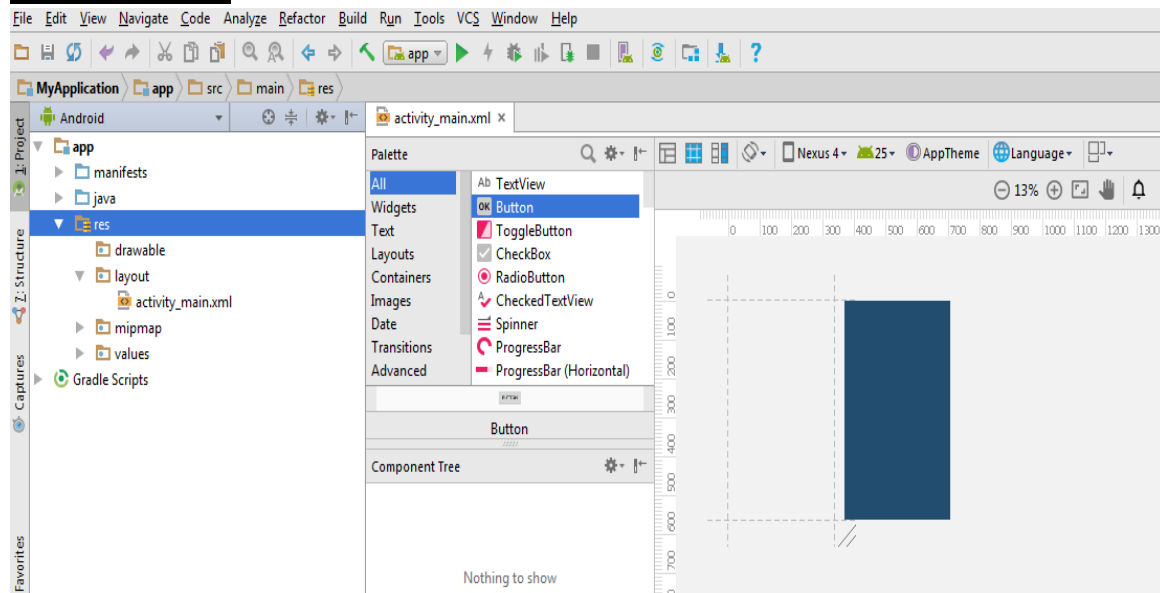


Εικόνα 3.6 Ονοματοδοσία activity

3.5.2 Επεξήγηση των στοιχείων

Σε αυτή την υποενότητα θα αναλυθούν τα στοιχεία που δημιουργήθηκαν από την προηγούμενη διαδικασία στο περιβάλλον του android studio.

Ο φάκελος res



Εικόνα 3.7 Ο φάκελος res 1

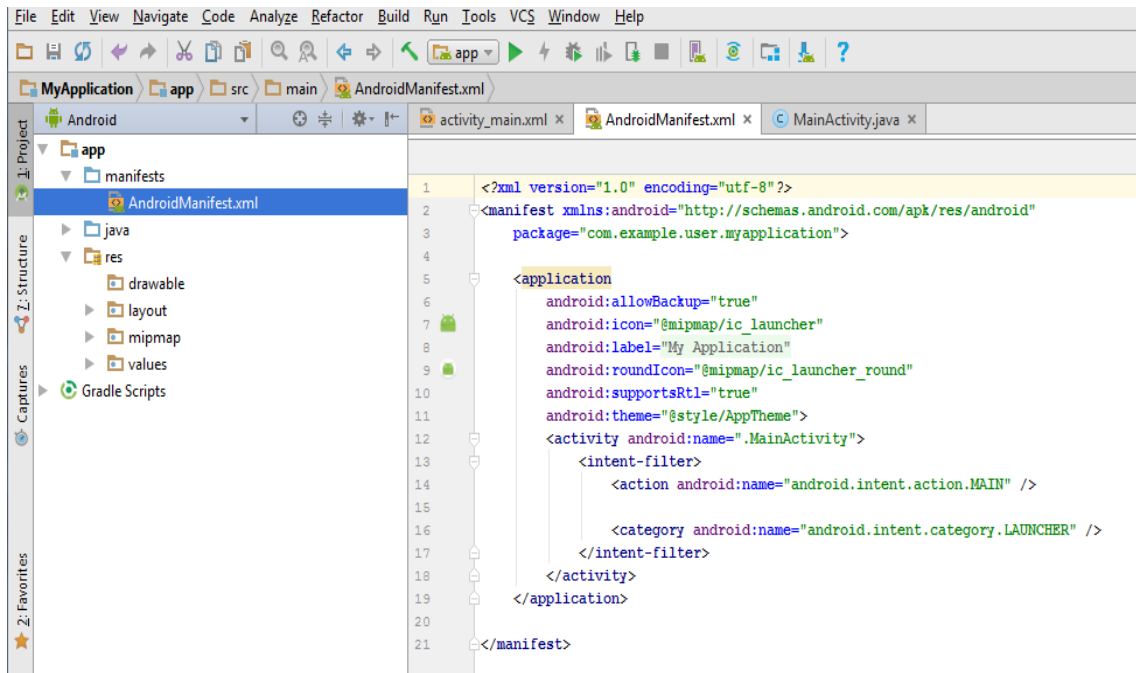
Ο φάκελος res (resources) περιέχει τους πόρους της εφαρμογής οι οποίοι μπορεί να είναι αρχεία XML, εικόνες, ήχοι, animation και string τιμές. Ο μεταγλωττιστής πόρων (resource compiler) κατηγοριοποιεί τους πόρους βάσει του τύπου τους, για

παράδειγμα οι εικόνες βρίσκονται στον υποφάκελο drawable. Σε περίπτωση που ο προγραμματιστής χρειάζεται να δημιουργήσει νέο υποφάκελο για τις ανάγκες της εφαρμογής του είναι ελεύθερος να το κάνει, καθώς το android studio δεν έχει κανένα περιορισμό.

Στον υποφάκελο res/drawable βρίσκονται κυρίως τα αρχεία που είναι στατικά γραφικά, όπως για παράδειγμα εικόνες, σχήματα και animation. Το Android υποστηρίζει αρχεία εικόνων τύπου PNG, JPG και GIF, και αρχεία XML τα οποία ορίζουν σχήματα, χρώματα, διαγράμματα, σχέδια που αλλάζουν το μέγεθός τους ανάλογα με την οθόνη και εικόνες που εναλλάσσουν την κατάσταση τους σε καθορισμένα γεγονότα.

Στον υποφάκελο res/layout αποθηκεύονται όλα τα αρχεία που είναι υπεύθυνα για την απεικόνιση και την μορφοποίηση της εφαρμογής. Αυτά τα αρχεία είναι τα .xml (eXtended Markup Language) . Η χρήση των αρχείων XML είναι ο ευκολότερος τρόπος για την απεικόνιση γραφικών στοιχείων στην εφαρμογή. Μεταγλωττίζονται σε διατάξεις οθόνης ή μέρους της οθόνης. Αυτός ο τρόπος, υπερτερεί της κλασσικής δημιουργίας αντικειμένων προγραμματιστικά επειδή διαχωρίζει τον πηγαίο κώδικα που αφορά τις λειτουργίες της εφαρμογής με τον σχεδιασμό της και γίνεται εύκολη η αλλαγή της εμφάνισης της εφαρμογής χωρίς να χρειάζεται αλλαγή στον πηγαίο κώδικα.

To αρχείο Android Manifest

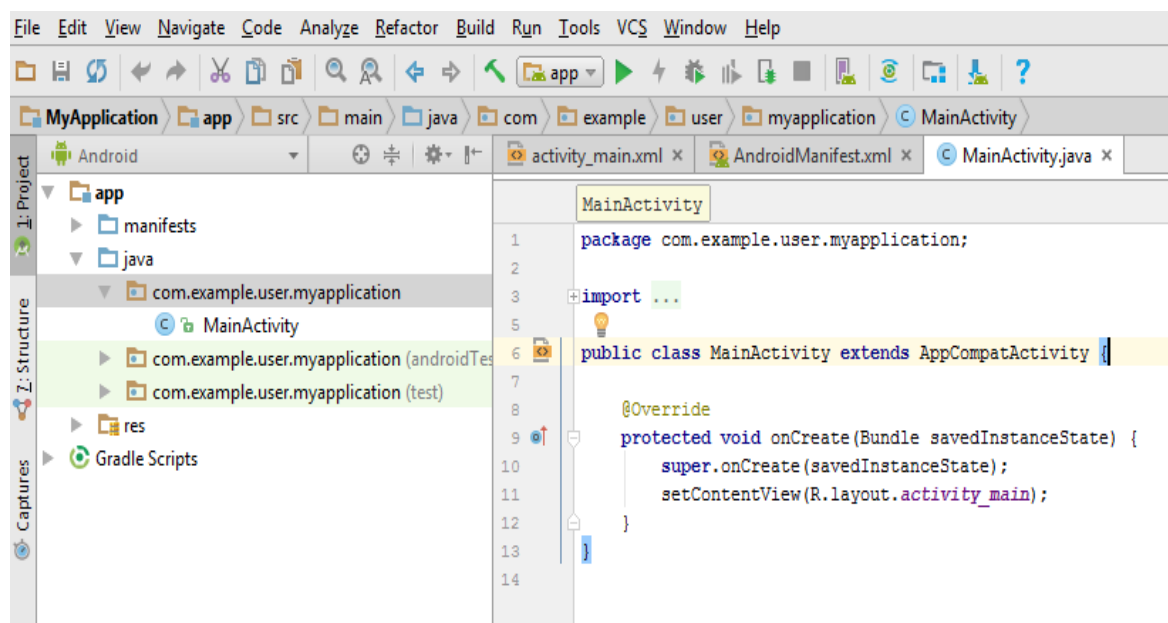


Εικόνα 3.8 Android manifest

Το αρχείο AndroidManifest αποτελεί το άλφα και το ωμέγα της εφαρμογής. Σε αυτό ο προγραμματιστής ορίζει τις απαιτήσεις της, δηλώνει τα δικαιώματα που απαιτούνται και καθορίζει ποιες εκδόσεις θα μπορούν να την τρέχουν. Συγκεκριμένα, δηλώνονται ποιά χαρακτηριστικά hardware ή software θα χρησιμοποιηθούν (camera, Bluetooth κλπ.), δηλώνονται τα δικαιώματα πρόσβασης που θα έχει η εφαρμογή στα δεδομένα του χρήστη και τέλος ορίζεται η ελάχιστη απαιτούμενη έκδοση.

Υποχρεωτικά, κάθε εφαρμογή πρέπει να έχει αυτό το αρχείο για να μπορεί να γίνει εκτελέσιμη. Εδώ, δηλώνονται επίσης οι δραστηριότητες (activities) μαζί με τις ενέργειές τους. Με την δημιουργία κάθε νέου project, δηλώνεται αυτόματα η activity της κεντρικής οθόνης (main screen).

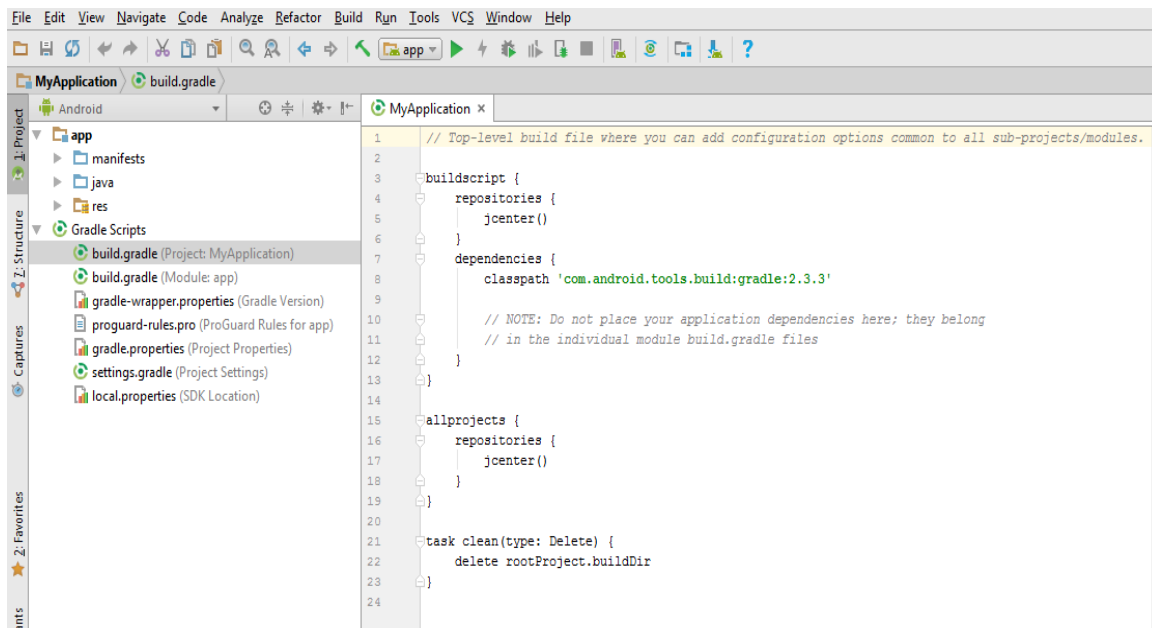
Ο φάκελος java



Εικόνα 3.9 Ο φάκελος java

Σε αυτόν τον φάκελο αποθηκεύονται όλα τα αρχεία java της εφαρμογής μας οργανωμένες σε πακέτα.

Το αρχείο build.gradle



Εικόνα 3.10 Το αρχείο build.gradle

Το σύστημα διαθέτει ένα toolkit το οποίο παρέχει κάποιες συγκεκριμένες δυνατότητες(Τροποποίηση, Ρύθμιση, Επέκταση) της διαδικασίας του build. Παρέχεται η δυνατότητα δημιουργίας πολλών και πολύπλοκων εφαρμογών με την χρήση ιδίων προσθέτων (plugins), βιβλιοθηκών (libraries, dependencies) διαφόρων επεκτάσεων (modules).

Το αρχείο build.gradle είναι υπεύθυνο για την προσθήκη όλων των παραπάνω, έτσι ώστε εάν ο προγραμματιστής θέλει μια custom library να μην κάνει εισαγωγή όλο το project στην εφαρμογή παρα μόνο μια αναφορά στο groupId, ArtifactId της βιβλιοθήκης που θέλει να χρησιμοποιήσει.

Επίλογος

Στο κεφάλαιο αυτό εξηγήθηκε πώς λειτουργεί ένα λειτουργικό σύστημα με την αρχιτεκτονική μιας κινητής συσκευής, τι ασφάλεια υπάρχει πίσω από κάθε λειτουργία αλλά και κατά πόσο ευπαθή είναι σε εξωγενείς κακοήθεις παράγοντες. Στο επόμενο κεφάλαιο αναλύεται η γλώσσα προγραμματισμού PHP καθώς και ο ρόλος της στην σωστή επικοινωνία μίας εφαρμογής και μίας απομακρυσμένης βάσης δεδομένων.

Κεφάλαιο 4ο: Γλώσσα Προγραμματισμού PHP

Πρόλογος

Στο κεφάλαιο αυτό περιγράφεται η γλώσσα προγραμματισμού PHP, όπως η ιστορία της, τα βασικά της χαρακτηριστικά και τα πλεονεκτήματά της συγκριτικά με άλλες client-side γλώσσες.

4.1 Τί είναι η PHP

Τα αρχικά PHP αντιστοιχούν σε “Personal home page Hypertext Preprocessor”. Η PHP είναι μια γλώσσα προγραμματισμού ανοικτού κώδικα, που χρησιμοποιείται κυρίως για την ανάπτυξη server-side εφαρμογών και δυναμικού περιεχομένου στον Παγκόσμιο Ιστό.

Αυτό που διαχωρίζει την PHP από τα client-side JavaScripts είναι ότι ο κώδικας εκτελείται στον server (εξυπηρετητή). Αν υπήρχε ένα script PHP, ο browser θα έπαιρνε τα αποτελέσματα της εκτέλεσης αυτού του script, χωρίς να μπορεί να καταλάβει με κανένα τρόπο τι κώδικας υπάρχει από κάτω.

4.2 Η ιστορία της PHP

Η ιστορία της PHP ξεκινά από το 1994, όταν ένας φοιτητής, ο Rasmus Lerdorf δημιούργησε χρησιμοποιώντας τη γλώσσα προγραμματισμού C[1] ένα απλό script με όνομα php.cgi, για προσωπική χρήση. Το script αυτό είχε σαν σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημείωμα. Αργότερα αυτό το script το διέθεσε και σε φίλους του, οι οποίοι άρχισαν να του ζητούν να προσθέσει περισσότερες δυνατότητες. Η γλώσσα τότε ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter. Το 1997 η PHP/FI έφθασε στην έκδοση 2.0 αριθμώντας περισσότερους από 50.000 ιστότοπους που τη χρησιμοποιούσαν, ενώ αργότερα την ίδια χρονιά οι Andi Gutmans και Zeev Suraski ξαναέγραψαν τη γλώσσα από την αρχή, βασιζόμενοι, όμως, αρκετά στην PHP/FI 2.0. Έτσι, η PHP έφθασε στην έκδοση 3.0 η οποία θύμιζε περισσότερο τη σημερινή μορφή της. Στη συνέχεια, οι Zeev και Andi δημιούργησαν την εταιρεία Zend (από τα αρχικά των ονομάτων τους), η οποία συνεχίζει μέχρι και σήμερα την ανάπτυξη και εξέλιξη της γλώσσας PHP. Ακολούθησε το 1998 η έκδοση 4 της PHP, τον Ιούλιο του 2004 διατέθηκε η έκδοση 5, ενώ αυτή τη στιγμή έχουν ήδη διατεθεί και οι πρώτες δοκιμαστικές εκδόσεις της επερχόμενης PHP 6, για οποιονδήποτε προγραμματιστή

θέλει να τη χρησιμοποιήσει. Οι περισσότεροι ιστότοποι επί του παρόντος χρησιμοποιούν κυρίως τις εκδόσεις 4 και 5 της PHP.

4.3 Βασικά χαρακτηριστικά

Έπειτα από τα εισαγωγικά στοιχεία για την PHP, σε αυτή την υποενότητα θα εμβαθύνουμε σε κάποια χαρακτηριστικά της. Η PHP μπορεί να χρησιμοποιηθεί σε τρεις τομείς για την παραγωγή εφαρμογών και ιστοσελίδων: α)Server – side scripting, β)Command line scripting και γ)Συγγραφή client – side GUI εφαρμογών (Γραφικά περιβάλλοντα χρήστη).

Ο πρώτος τομέας αποτελεί το βασικό πεδίο δράσης της PHP. Τα αποτελέσματα ενός script, μπορούν να προσπελαστούν με τη χρήση του browser, βλέποντας την αντίστοιχη ιστοσελίδα μέσα από το web server, ο οποίος είναι συνδεδεμένος με το μεταγλωττιστή της PHP. Όσον αφορά στο δεύτερο τομέα, ένα script, μπορεί να γραφεί για να λειτουργεί χωρίς server ή browser. Απαραίτητος είναι μόνο ο μεταγλωττιστής, ενώ αυτός ο τύπος script είναι ο ιδανικός για λειτουργίες που εκτελούνται συχνά. Τέλος, αναφορικά με τα γραφικά περιβάλλοντα χρήστη, αν και η PHP δεν είναι η κατάλληλη γλώσσα για παραθυρικές (Windows) εφαρμογές, μπορεί να χρησιμοποιηθεί για τη συγγραφή τέτοιου είδους προγραμμάτων. Για τέτοια script, είναι απαραίτητη η ύπαρξη της επέκτασης της PHP, PHP – GTK, η οποία και διανέμεται ξεχωριστά από το κύριο πακέτο.

Η PHP, σε γενικές γραμμές, παρέχει μεγάλη ελευθερία και στην επιλογή του web server, καθώς υποστηρίζει μεγάλο πλήθος από όσους υπάρχουν σήμερα. Ονομαστικά, αναφέρονται μερικοί μόνο από αυτούς: Microsoft Internet Information Server, Microsoft Personal Web Server, Netscape και iPlanet servers, Apache, OmniHTTPd, κ.λ.π. Επιπλέον, η PHP δίνει τη δυνατότητα επιλογής στον τρόπο προγραμματισμού των εφαρμογών. Έτσι, λοιπόν, μπορεί να χρησιμοποιηθεί συναρτησιακός (procedural) ή αντικειμενοστρεφής (object – oriented) προγραμματισμός, ή ακόμα περισσότερο και τα δυο μαζί.

Ένα τελευταίο αλλά εξίσου σημαντικό χαρακτηριστικό της PHP, είναι η δυνατότητα που παρέχει για εισαγωγή στοιχείων σε ιστοσελίδες, τα οποία αντλούνται από βάσεις δεδομένων. Αυτό έχει ως αποτέλεσμα, η PHP να είναι ιδιαίτερα χρήσιμη, ακόμα και για απαιτητικές εφαρμογές. Το εύρος των πακέτων βάσεων δεδομένων που υποστηρίζεται είναι αρκετά μεγάλο και περιλαμβάνει τα εξής

: MySQL, ODBC, Oracle, dBase, mSQL, InterBase, Sybase, Unix dbm, PostgreSQL, IBM DB2, Informix, Direct MS – SQL, καθώς και πολλά ακόμη.

4.4 Πλεονεκτήματα της PHP

Υπάρχουν σημαντικές διαφορές μεταξύ της PHP και των άλλων γλωσσών προγραμματισμού. Πολλές φορές η PHP είναι προτιμότερη έναντι των άλλων, ανάλογα πάντοτε και με τη φύση της διαδικτυακής εφαρμογής που πρόκειται να αναπτυχθεί. Παρακάτω, αναφέρονται και αναλύονται οι σημαντικότεροι λόγοι, για τους οποίους η PHP είναι προτιμότερη, τόσο σε αυτή την εργασία, όσο και γενικότερα.

Αρχικά, βασικό της πλεονέκτημα είναι η *υψηλή απόδοση*. Η PHP είναι πολύ αποτελεσματική. Με ένα φθηνό διακομιστή μπορούν να εξυπηρετηθούν εκατομμύρια επισκέψεων καθημερινά. Οι δοκιμές που δημοσιεύθηκαν από την Zend Technologies (<http://www.zend.com>), δείχνουν ότι η PHP ξεπερνά τους ανταγωνιστές της.

Ένα ακόμα πλεονέκτημα είναι οι *διασυνδέσεις* που παρέχει με πολλά διαφορετικά συστήματα βάσεων δεδομένων. Η PHP έχει εγγενείς συνδέσεις για πολλά συστήματα βάσεων δεδομένων. Εκτός από την MySQL μπορεί να συνδεθεί με τις βάσεις δεδομένων PostgreSQL, mSQL Oracle, dbm, filePro, Informix, InterBase, Sybase, κ.α. Χρησιμοποιώντας το Open Database Connectivity Standard (ODBC) μπορεί να συνδεθεί σε οποιαδήποτε βάση δεδομένων παρέχει ένα πρόγραμμα οδήγησης ODBC . Αυτό περιλαμβάνει και τα προϊόντα της Microsoft, μεταξύ άλλων.

Οι *ενσωματωμένες βιβλιοθήκες* για πολλές συνηθισμένες Web διαδικασίες είναι ένα εξίσου σημαντικό πλεονέκτημα της PHP. Επειδή η PHP σχεδιάστηκε για να χρησιμοποιείται στο Web , έχει πολλές ενσωματωμένες βιβλιοθήκες, που εκτελούν πολλές χρήσιμες λειτουργίες σχετικές με το Web . Μπορείτε να δημιουργήσετε εικόνες GIF δυναμικά, να συνδεθείτε με άλλες υπηρεσίες δικτύων , να στείλετε ηλεκτρονικό ταχυδρομείο, να δουλέψετε με cookies και να δημιουργήσετε PDF έγγραφα: όλα αυτά με λίγες γραμμές κώδικα.

Αξίζει, επίσης, να αναφερθεί πως η PHP είναι *δωρεάν και εύκολη στη χρήση και τη μάθηση*. Μπορεί κανείς να κατεβάσει την τελευταία έκδοση από το <http://www.php.net> , χωρίς χρέωση, ενώ η σύνταξή της βασίζεται σε άλλες γλώσσες προγραμματισμού, κυρίως στην C και στην Perl .

Ένα επιπλέον πλεονέκτημα είναι η *μεταφερσιμότητα*. Η PHP είναι διαθέσιμη για πολλά λειτουργικά συστήματα . Μπορείτε να γράψετε κώδικα PHP για δωρεάν συστήματα τύπου Unix, όπως LINUX και FreeBSD, για εμπορικές εκδόσεις του UNIX, όπως το Solaris και το IRIX ή για διαφορετικές εκδόσεις των Microsoft Windows. Ο κώδικας σας συνήθως θα δουλεύει χωρίς αλλαγές στα συστήματα που τρέχουν την PHP.

Τέλος, η διαθεσιμότητα του κώδικα προέλευσης αποτελεί ένα ακόμα πλεονέκτημα της PHP. Οποιοσδήποτε έχει πρόσβαση στον κώδικα προέλευσης της PHP. Αντίθετα με εμπορικά, κλειστά προγράμματα, αν υπάρχει κάτι που θα ήθελε κανείς να αλλάξει ή να προσθέσει στη γλώσσα, μπορεί να το κάνει. Δεν χρειάζεται να περιμένει τον κατασκευαστή να εμφανίσει διορθώσεις. Έτσι, δε θα ανησυχεί αν ο κατασκευαστής θα σταματήσει να υπάρχει ή αν θα σταματήσει να υποστηρίζει το προϊόν.

Επίλογος

Στο κεφάλαιο αυτό αναλύθηκαν τα χαρακτηριστικά της γλώσσας προγραμματισμού PHP, η ιστορία της και τα πλεονεκτήματά της. Στο επόμενο κεφάλαιο περιγράφεται η χρήση των βάσεων δεδομένων και η ανάλυση της τεχνολογίας τους.

Κεφάλαιο 5: Βάσεις δεδομένων

Πρόλογος

Το κεφάλαιο αυτό εξηγεί την λειτουργία και την ανάπτυξη των συστημάτων διαχείρισης βάσεων δεδομένων στην πάροδο του χρόνου, την τεχνολογία τους και αναλύει τη δομή και την αρχιτεκτονική μίας βάσης δεδομένων.

5.1 Εισαγωγή

Η αλματώδης ανάπτυξη της επιστήμης της πληροφορικής και των επικοινωνιών τα τελευταία χρόνια έχει καταστήσει την πληροφορία ως ένα από τα βασικότερα και πολυτιμότερα αγαθά. Είναι κοινός τόπος σήμερα η εκτίμηση ότι το αγαθό της πληροφορίας είναι επιθυμητό απ' όλους τους εργαζόμενους αλλά και τους εκπαιδευόμενους, ώστε να είναι πιο αποδοτικοί, ανταγωνιστικοί αλλά και παραγωγικοί στην εργασία τους.

Οι βάσεις δεδομένων και η τεχνολογία βάσεων δεδομένων εξασκούν σημαντική επίδραση στην αυξανόμενη χρήση των υπολογιστών. Είναι εύλογο να ειπωθεί ότι οι βάσεις δεδομένων διαδραματίζουν ένα σημαντικό ρόλο σε όλες τις περιοχές όπου χρησιμοποιούνται υπολογιστές, συμπεριλαμβανομένων των επιχειρήσεων, του ηλεκτρονικού εμπορίου, της μηχανικής, της ιατρικής, της νομικής, της εκπαίδευσης και της βιβλιοθηκονομίας, για να αναφέρουμε μερικές μόνο από αυτές.

Τα συστήματα βάσεων δεδομένων τα χρησιμοποιούμε για να μπορούμε να αποθηκεύσουμε, να επεξεργαστούμε αλλά και να εκμεταλλευτούμε αποδοτικά αυτόν τον τεράστιο όγκο των πληροφοριών που αυξάνονται με αλματώδεις ρυθμούς καθημερινά.

5.2 Ιστορικά στοιχεία

Η έννοια «Βάση Δεδομένων» έχει εισαχθεί από τη δεκαετία του 1960 για να διευκολύνει τις αυξανόμενες δυσκολίες στο σχεδιασμό, την κατασκευή και την συντήρηση πολύπλοκων συστημάτων πληροφορίας (συνήθως με πολλούς ταυτόχρονους τελικούς χρήστες και ένα ευρύ και μεγάλο όγκο δεδομένων). Έχει εξελιχθεί από κοινού με τα Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ ή DBMS) που επιτρέπουν την αποτελεσματική διαχείριση των βάσεων δεδομένων. Αν

και οι όροι Βάση Δεδομένων και ΣΔΒΔ καθορίζουν δύο διαφορετικές οντότητες, είναι άρρηκτα συνδεδεμένοι: οι ιδιότητες μιας Βάσης Δεδομένων καθορίζονται από το ΣΔΒΔ που την υποστηρίζει και το αντίστροφο.

Με την πρόοδο της τεχνολογίας στους τομείς των επεξεργαστών, της μνήμης των υπολογιστών, του χώρου αποθήκευσης στους υπολογιστές και των δικτύων υπολογιστών, τα μεγέθη, οι δυνατότητες και η απόδοση των Βάσεων Δεδομένων και των ΣΔΒΔ έχουν αυξηθεί κατά μεγάλες τάξεις μεγέθους. Για δεκαετίες θεωρούταν απίθανο το γεγονός ότι ένα πολύπλοκο σύστημα πληροφοριών μπορεί να χτιστεί αποτελεσματικά χωρίς μία σωστή βάση δεδομένων που να υποστηρίζεται από ένα ΣΔΒΔ. Η αξιοποίηση των βάσεων δεδομένων έχει εξαπλωθεί πολύ μεγάλο βαθμό. Πολλά εμπορικά προϊόντα βασίζονται στις Βάσεις Δεδομένων και τα ΣΔΒΔ για την ανάπτυξη και την εμπορευματοποίησή τους. Επίσης, οργανισμοί και εταιρείες, από τις πιο μικρές έως τις πιο μεγάλες, εξαρτώνται σε μεγάλο βαθμό από τις βάσεις δεδομένων για την λειτουργία τους.

Δεν υπάρχει ένας ευρέως αποδεκτός ακριβής ορισμός για το ΣΔΒΔ. Ωστόσο, ένα σύστημα πρέπει να παρέχει σημαντικές λειτουργίες για να χαρακτηριστεί ως ένα ΣΔΒΔ. Ανάλογα με την υποστηριζόμενη συλλογή των δεδομένων του, πρέπει να πληρούνται και οι αντίστοιχες απαιτήσεις ευχρηστίας για να χαρακτηριστεί ως μια Βάση Δεδομένων. Έτσι, μια Βάση Δεδομένων και το ΣΔΒΔ που την υποστηρίζει ορίζονται εδώ με μια σειρά από γενικές απαιτήσεις που απαριθμούνται κατωτέρω. Σχεδόν όλα τα υπάρχοντα «ώριμα» προϊόντα ΣΔΒΔ ανταποκρίνονται στις απαιτήσεις αυτές σε μεγάλο βαθμό, ενώ τα λιγότερο «ώριμα», είτε καλύπτουν μερικώς αυτές τις απαιτήσεις, είτε κλίνουν προς αυτές.

5.3 Εξέλιξη των βάσεων δεδομένων και των ΣΔΒΔ

5.3.1 Βάσεις δεδομένων

Η εισαγωγή του όρου «Βάση Δεδομένων» συνέπεσε με τη διαθεσιμότητα της άμεσης αποθήκευσης σε κατάλληλα μέσα (σκληρούς δίσκους και δισκέτες) από τα μέσα της δεκαετίας του 1960 και μετά. Ο όρος αντιπροσώπευε μία αντίθεση με τα *taped-based* συστήματα του παρελθόντος, επιτρέποντας την κοινή διαδραστική χρήση παρά την καθημερινή μαζική επεξεργασία.

Στα πρώτα συστήματα βάσεων δεδομένων, η αποτελεσματικότητα ήταν ίσως το κυριότερο μέλημα, αλλά είχε ήδη αναγνωριστεί ότι υπήρχαν κι άλλοι στόχοι. Ένας από τους βασικούς στόχους ήταν να γίνουν τα δεδομένα ανεξάρτητα από τη λογική της εφαρμογής αποκλειστικά σε κάποια προγράμματα, έτσι ώστε να μπορούν να διατεθούν σε διαφορετικές εφαρμογές.

Η πρώτη γενιά των συστημάτων βάσεων δεδομένων ήταν βάσεις πλοήγησης (navigational), όπου οι εφαρμογές συνήθως είχαν πρόσβαση σε δεδομένα ακολουθώντας δείκτες από μία εγγραφή σε μία άλλη. Τα δύο κύρια μοντέλα αυτή την εποχή ήταν το ιεραρχικό μοντέλο (Hierarchical Model) που αποτελεί την επιτομή του συστήματος IMS της IBM και το Codesyl μοντέλο (μοντέλο δικτύου – Network Model), που εφαρμόζονται σε ορισμένα προϊόντα, όπως το IDMS.

Το Σχεσιακό Μοντέλο (Relational Model), προτάθηκε για πρώτη φορά το 1970 από τον Edgar F. Codd, ο οποίος επέμενε ότι οι αιτήσεις (ενός χρήστη) θα πρέπει να αναζητούν δεδομένα με βάση το περιεχόμενο και όχι ακολουθώντας συνδέσμους. Αυτό θεωρήθηκε αναγκαίο, για να επιτραπεί στο περιεχόμενο μίας βάσης δεδομένων να εξελιχθεί χωρίς τη συνεχή αναμόρφωση των εφαρμογών. Τα σχεσιακά συστήματα έθεσαν σοβαρές απαιτήσεις όσον αφορά τους πόρους επεξεργασίας, και δεν ήταν μέχρι τα μέσα της δεκαετίας του 1980 όπου το υλικό των υπολογιστών έγινε αρκετά δυνατό έτσι ώστε να μπορούν να χρησιμοποιηθούν ευρέως. Από τις αρχές του 1990, ωστόσο, τα σχεσιακά συστήματα ήταν κυρίαρχα σε όλες τις εφαρμογές μεγάλης κλίμακας επεξεργασίας στοιχείων και παραμένουν κυρίαρχα μέχρι και σήμερα, εκτός από εξειδικευμένες περιοχές. Η κυρίαρχη γλώσσα των Βάσεων Δεδομένων είναι το πρότυπο SQL (Structured Query Language) ή Δομημένη Γλώσσα Ερωτημάτων για το Σχεσιακό μοντέλο, το οποίο έχει επηρεάσει γλώσσες βάσεων δεδομένων και για άλλα μοντέλα δεδομένων.

Επειδή το Σχεσιακό μοντέλο δίνει έμφαση στην αναζήτηση παρά στην πλοήγηση, δεν παριστάνει τις σχέσεις μεταξύ διαφόρων οντοτήτων αποκλειστικά με τη μορφή δεικτών, αλλά τους αντιπροσωπεύει αντί να χρησιμοποιεί πρωτεύοντα (primary) και ξένα (foreign) κλειδιά. Ενώ αυτό είναι μία καλή βάση για μία γλώσσα ερωταπαντήσεων, είναι λιγότερο κατάλληλο ως γλώσσα μοντελοποίησης. Για το λόγο αυτό, ένα διαφορετικό μοντέλο, το μοντέλο Οντοτήτων-Συσχετίσεων (Entity-Relationship model) που αναπτύχθηκε λίγο αργότερα (1976), κέρδισε τη δημοτικότητα για το σχεδιασμό βάσεων δεδομένων.

Από το 1970 και μετά, η τεχνολογία βάσεων δεδομένων έχει συμβαδίσει με τους αυξανόμενους πόρους που θα καταστούν διαθέσιμοι μέσω της τεχνολογίας των υπολογιστών και συγκεκριμένα μέσω της ταχείας αύξησης της χωρητικότητας και της ταχύτητας (με ταυτόχρονη μείωση των τιμών) των μέσων αποθήκευσης και της αύξησης της χωρητικότητας της κύριας μνήμης.

Η ακαμψία του Σχεσιακού μοντέλου, στο οποίο όλα τα δεδομένα αποθηκεύονται σε πίνακες με συγκεκριμένη δομή γραμμών και στηλών, όλο και περισσότερο θεωρείται ως περιορισμός κατά τον χειρισμό πληροφοριών, η οποία είναι πιο πλούσια ή ποικίλει στην δομή από το παραδοσιακό «καθολικό βιβλίο» δεδομένων των εταιρικών πληροφοριακών συστημάτων : για παράδειγμα, βάσεις δεδομένων εγγράφου, βάσεις δεδομένων μηχανικής, βάσεις δεδομένων πολυμέσων ή βάσεις δεδομένων που χρησιμοποιούνται στις μοριακές επιστήμες. Διάφορες προσπάθειες έχουν γίνει για να αντιμετωπιστεί αυτό το πρόβλημα, πολλές από αυτές οδηγούν σε λύσεις όπως οι «μετά-σχεσιακές» ή η NoSQL. Δύο εξελίξεις που αξίζει να σημειωθούν είναι η Βάση Δεδομένων Αντικειμένων – Object Database και η Βάση Δεδομένων XML – eXtensible Markup Language. Οι προμηθευτές- πωλητές των σχεσιακών βάσεων δεδομένων έχουν πολεμήσει τον ανταγωνισμό από τα νεότερα μοντέλα, επεκτείνοντας τις δυνατότητες των προϊόντων τους ώστε να υποστηρίξουν μια ερύτερη ποικιλία δεδομένων.

5.3.2 ΣΔΒΔ

Ένα ΣΔΒΔ έχει εξελχθεί σε ένα πολύπλοκο σύστημα λογισμικού και η ανάπτυξη του απαιτεί συνήθως χιλιάδες ανθρωπο-έτη αναπτυξιακής προσπάθειας. Ορισμένα γενικού-σκοπού ΣΔΒΔ, όπως η Oracle, η Microsoft SQL Server και η IBM DB2 έχουν υποστεί αναβαθμίσεις για τριάντα χρόνια ή και περισσότερο. Τα γενικού-σκοπού ΣΔΒΔ έχουν στόχο να ικανοποιήσουν όσο το δυνατόν περισσότερες εφαρμογές, που συνήθως τις καθιστούν ακόμη πιο πολύπλοκες απ' ότι οι ειδικού σκοπού βάσεις δεδομένων. Ωστόσο, το γεγονός ότι μπορούν να χρησιμοποιηθούν «από το ράφι», καθώς και το αναπόσβεστο κόστος τους σε πολλές εφαρμογές και περιπτώσεις, τα κάνει μια ελκυστική εναλλακτική λύση κάθε φορά που πληρούν τις απαιτήσεις μιας εφαρμογής.

Αν και ελκυστικό σε πολλές περιπτώσεις, ένα γενικού-σκοπού ΣΔΒΔ δεν είναι πάντα η καλύτερη λύση, δηλαδή, όταν ορισμένες εφαρμογές είναι ευρέως

διαδεδομένες, με πολλές δυνατότητες λειτουργίας και η καθεμία με πολλούς χρήστες, ένα γενικού-σκοπού ΣΔΒΔ μπορεί να εισάγει περιττή επιβάρυνση (overhead) και πολύ μεγάλο ποσό περιττού και άχρηστου κώδικα λογισμικού. Οι εφαρμογές αυτές δικαιολογούν συνήθως αποκλειστική ανάπτυξη. Τυπικά παραδείγματα είναι τα συστήματα ηλεκτρονικού ταχυδρομείου (e-mail systems), τα οποία πρέπει να διαθέτουν ορισμένες ιδιότητες ενός ΣΔΒΔ : τα συστήματα ηλεκτρονικού ταχυδρομείου είναι χτισμένα με τρόπο που βελτιστοποιούν τη διαχείριση μηνυμάτων ηλεκτρονικού ταχυδρομείου και δεν χρειάζονται σημαντικά μέρη της λειτουργικότητας ενός γενικού-σκοπού ΣΔΒΔ.

5.4 Δομή βάσης δεδομένων

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 με το οποίο μπορούμε να αποθηκεύσουμε και να αξιοποιήσουμε δεδομένα με τη βοήθεια ηλεκτρονικού υπολογιστή. Ειδικότερα:

- Υλικό:

Το υλικό (hardware) αποτελείται όπως είναι γνωστό από τους ηλεκτρονικούς υπολογιστές, τα περιφερειακά, τους σκληρούς δίσκους, τις μαγνητικές ταινίες κ.ά., όπου είναι αποθηκευμένα τα αρχεία της βάσης δεδομένων αλλά και τα προγράμματα που χρησιμοποιούνται για την επεξεργασία τους.

- Λογισμικό :

Το λογισμικό (software) είναι τα προγράμματα που χρησιμοποιούνται για την επεξεργασία των δεδομένων (στοιχείων) της βάσης δεδομένων.

- Βάση 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 η εντύπωση ότι βρίσκονται συγκεντρωμένα στον ίδιο υπολογιστή. Τα δεδομένα των αρχείων αυτών είναι ενοποιημένα (data integration), δηλαδή, δεν υπάρχει πλεονασμός (άσκοπη επανάληψη) δεδομένων και διαμερισμένα (data sharing), δηλαδή, υπάρχει

δυνατότητα ταυτόχρονης προσπέλασης των δεδομένων από πολλούς χρήστες. 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 χωρίζονται στις εξής κατηγορίες :

- Προγραμματιστές του ΣΔΒΔ :

Αυτοί είναι οι άνθρωποι, οι οποίοι σχεδιάζουν και αναπτύσσουν το ΣΔΒΔ και οι μόνοι που θα «αγγίξουν» τον κώδικά του. Είναι συνήθως οι εργαζόμενοι ενός ΣΔΒΔ προμηθευτή (π.χ. η Oracle, η IBM, η Microsoft, η Sybase), ή στην περίπτωση των open-source ΣΔΒΔ (π.χ. η MySQL) εθελοντές ή άτομα που υποστηρίζονται από τις ενδιαφερόμενες επιχειρήσεις και οργανισμούς. Τα άτομα αυτά έχουν τυπικά προσόντα προγραμματιστών συστημάτων. Η ανάπτυξη ενός ΣΔΒΔ είναι πολύπλοκη και δύσκολη εργασία και μερικά από τα πιο δημοφιλή ΣΔΒΔ είναι υπό συνεχή ανάπτυξη και βελτίωση (ταυτόχρονα με την παρακολούθηση της προόδου της τεχνολογίας) για δεκαετίες

- Διαχειριστές Δεδομένων :

Έχουν τη διοικητική αρμοδιότητα και ευθύνη για την οργάνωση της βάσης δεδομένων και την απόδοση δικαιωμάτων πρόσβασης στους χρήστες.

- Τελικοί χρήστες :

Χρησιμοποιούν κάποια εφαρμογή για να παίρνουν στοιχεία από μια βάση δεδομένων, έχουν τις λιγότερες δυνατότητες επέμβασης στα στοιχεία της βάσης δεδομένων, χρησιμοποιούν ειδικούς κωδικούς πρόσβασης και το σύστημα τους επιτρέπει ανάλογη πρόσβαση σε συγκεκριμένο κομμάτι της βάσης δεδομένων.

5.5 Αρχιτεκτονική ΣΔΒΔ

Ένα ΣΔΒΔ (Σύστημα Διαχείρισης Βάσης Δεδομένων) έχει σαν αποστολή τη διαχείριση των δεδομένων των αρχείων της βάσης, δηλ. την προσθήκη, διαγραφή, τροποποίηση εγγραφών, την αναζήτηση μέσα στις εγγραφές κ.ά.). Το ΣΔΒΔ δέχεται αιτήσεις από τους χρήστες των εφαρμογών και επικοινωνεί με τα αρχεία της βάσης δεδομένων για να τις διεκπεραιώσει.

Αυτή η κοινή διεπαφή (interface) των εφαρμογών με τα αρχεία αποκαλείται λογική διεπαφή. Οι εφαρμογές που δημιουργούμε δεν απασχολούνται με τον τρόπο που είναι αποθηκευμένα τα δεδομένα, πόσο χώρο καταλαμβάνουν κ.ο.κ. και αυτή η ιδιότητα είναι γνωστή ως ανεξαρτησία δεδομένων.

Αυτό σημαίνει πρακτικά ότι οποιαδήποτε αλλαγή στον τρόπο οργάνωσης των αρχείων της βάσης δεδομένων δεν θα συνεπάγεται και αλλαγή στις εφαρμογές· ένα πρόβλημα που ταλαιπωρούσε πολύ τους προγραμματιστές παλαιότερων εποχών. Ακόμη, η προσθήκη, η κατάργηση ή και η τροποποίηση κάποιων εφαρμογών δεν θα έχει καμία επίπτωση στον τρόπο οργάνωσης των αρχείων της βάσης δεδομένων. Στα ΣΔΒΔ έχει επικρατήσει η λεγόμενη αρχιτεκτονική των τριών επιπέδων (βαθμίδων), όπου τα τρία επίπεδα είναι τα εξής :

- **Εσωτερικό επίπεδο** (internal level), έχει να κάνει με την αποθήκευση των αρχείων στον σκληρό δίσκο, δηλ. την πραγματική ή φυσική κατάστασή τους. Ασχολείται με το κόστος, την απόδοση, την επεκτασιμότητα, με τη διάταξη αποθήκευσης του εννοιολογικού επιπέδου και παρέχει υποστήριξη αποθήκευσης-δομών (όπως π.χ. ευρετήρια όπου θα επικεντρωθούμε στο επόμενο κεφάλαιο).
- **Εξωτερικό επίπεδο** (external level), έχει να κάνει με τους χρήστες είτε αυτοί είναι απλοί χειριστές, είτε προγραμματιστές ή και διαχειριστές της βάσης δεδομένων και καθορίζει το πως κάθε τελικός χρήστης κατανοεί την οργάνωση των αντίστοιχων σχετικών στοιχείων του στη βάση δεδομένων.
- **Εννοιολογικό επίπεδο** (conceptual level), είναι ένα ενδιάμεσο επίπεδο που διασυνδέει τα δύο άλλα επίπεδα και έχει να κάνει με τη λογική σχεδίαση των αρχείων της βάσης δεδομένων. Παρέχεται με τον απλούστερο δυνατό τρόπο και αποτελεί τη σπονδυλική στήλη της βάσης δεδομένων. Είναι έξω από το πεδίο εφαρμογής των διάφορων τελικών χρηστών, εξυπηρετεί τους προγραμματιστές εφαρμογών και ορίζεται από τους διαχειριστές της βάσης δεδομένων.

5.6 Ασφάλεια δεδομένων

Η ασφάλεια των βάσεων δεδομένων ασχολείται με όλες τις διαφορετικές πτυχές της προστασίας του περιεχομένου της, των ιδιοκτητών της και των χρηστών της. Συμβάλλει στην προστασία από εσκεμμένη μη- εξουσιοδοτημένη χρήση της

βάσης δεδομένων καθώς και την ακούσια πρόσβαση από μη-εξουσιοδοτημένες οντότητες (π.χ. ένα άτομο ή ένα πρόγραμμα). Παρακάτω αναφέρονται δύο σημαντικοί τομείς της ασφάλεια σε μια βάση δεδομένων.

- Έλεγχος πρόσβασης (Access Control)

Ο έλεγχος πρόσβασης σε μία βάση δεδομένων ασχολείται με το να ελέγχει ποιος (ένα άτομο ή κάποιο πρόγραμμα του υπολογιστή) επιτρέπεται να έχει πρόσβαση σε αντίστοιχες πληροφορίες στην βάση. Οι πληροφορίες μπορεί να περιλαμβάνουν συγκεκριμένα αντικείμενα της βάσης (π.χ. τύπους εγγραφών, συγκεκριμένες εγγραφές, δομές δεδομένων), υπολογισμούς σε ορισμένα αντικείμενα (π.χ. τύπους ερωτημάτων ή και συγκεκριμένα ερωτήματα) ή να χρησιμοποιούν συγκεκριμένους δρόμους πρόσβασης σε αυτά τα αντικείμενα (π.χ. χρησιμοποιώντας συγκεκριμένα ευρετήρια ή άλλες δομές δεδομένων για να αποκτήσουν πρόσβαση σε πληροφορίες).

Οι έλεγχοι πρόσβασης σε μία βάση δεδομένων ορίζονται από ειδικό εξουσιοδοτημένο προσωπικό (από τον ιδιοκτήτη της βάσης), το οποίο χρησιμοποιεί αποκλειστικά προστατευμένα ΣΔΒΔ interfaces ασφαλείας.

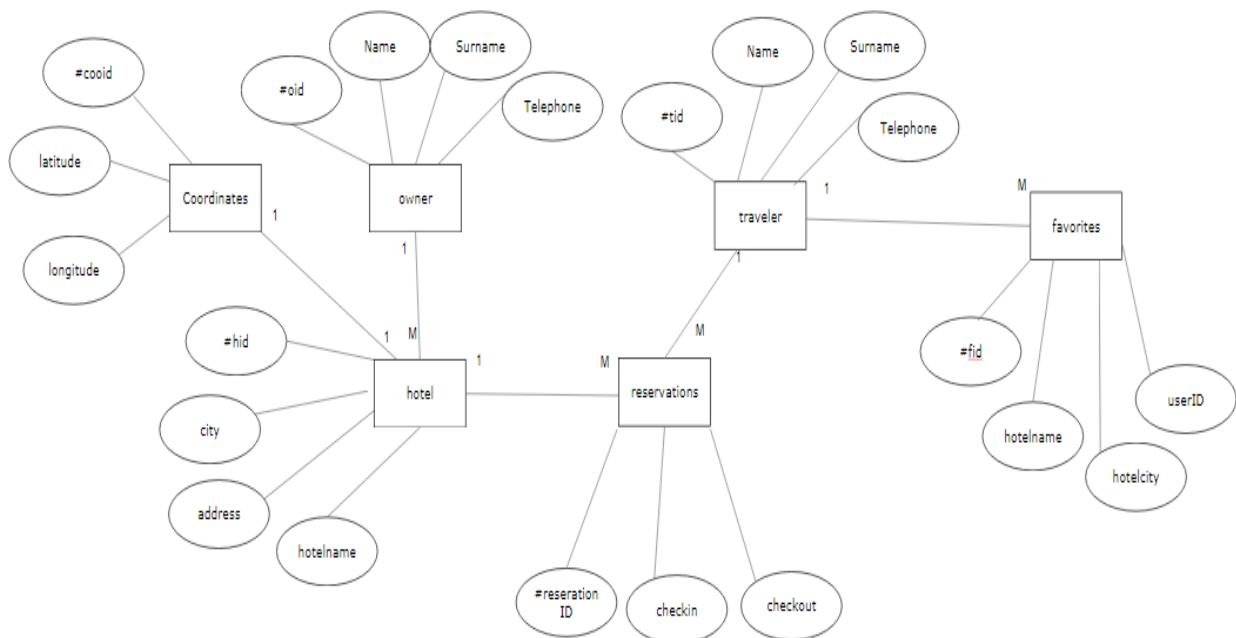
- Ασφάλεια Δεδομένων (Data Security)

Ο ορισμός της ασφάλειας δεδομένων ποικίλει και μπορεί να επικαλύπτεται από διάφορες πλευρές της ασφάλειας της βάσης δεδομένων. Σε γενικές γραμμές, ασχολείται με την προστασία συγκεκριμένων κομματιών δεδομένων, τόσο υλικά (Υλική Ασφάλεια – Physical Security), δηλαδή, από φθορά, καταστροφή ή αφαίρεση, όσο και από την ερμηνεία των δεδομένων ή μέρους αυτών (Κρυπτογράφηση Δεδομένων – Data Encryption), π.χ. εξετάζοντας τις σειρές των bits που αποτελούν ειδικούς έγκυρους αριθμούς πιστωτικών καρτών.

5.7 Περιγραφή της βάσης

5.7.1 Διάγραμμα οντοτήτων συσχετίσεων

Από τις απαιτήσεις, τους περιορισμούς αλλά και τα χαρακτηριστικά της βάσης, δημιουργείται ένα διάγραμμα το οποίο αναπαριστά τις μεταξύ τους σχέσεις. Εφαρμόζοντας στο διάγραμμα αυτό κάποιους κανόνες, προκύπτουν σχέσεις οι οποίες βρίσκονται σε τρίτη κανονική μορφή. Αυτός ο τρόπος σχεδίασης και παρουσίασης της βάσης δεδομένων ονομάζεται διάγραμμα οντοτήτων-συσχετίσεων. Στην επόμενη εικόνα (εικόνα 5.1) φαίνεται το μοντέλο οντοτήτων-συσχετίσεων για τη νέα βάση δεδομένων :



Εικόνα 5.1: Μοντέλο οντοτήτων-συσχετίσεων

Χρήση του διαγράμματος οντοτήτων – συσχετίσεων

Χρησιμοποιείται στο πρώτο στάδιο σχεδίασης ενός συστήματος πληροφοριών, κατά την ανάλυση των απαιτήσεων του. Σκοπός του είναι να περιγράψει τις αναγκαίες πληροφορίες οι οποίες πρόκειται να αποθηκευτούν στη βάση δεδομένων ή τον τύπο τους. Η μοντελοποίηση δεδομένων γίνεται για την περιγραφή των χρησιμοποιούμενων όρων και των σχέσεών τους σε έναν ορισμένο τομέα ενδιαφέροντος. Στην περίπτωση σχεδιασμού ενός συστήματος πληροφοριών, που στηρίζεται σε μια βάση δεδομένων, το εννοιολογικό μοντέλο δεδομένων

χαρτογραφείται σε προχωρημένο στάδιο σε ένα λογικό μοντέλο δεδομένων, όπως το σχεσιακό μοντέλο δεδομένων. Το στάδιο αυτό ονομάζεται συνήθως στάδιο λογικού σχεδιασμού. Ύστερα, κατά τη διάρκεια του φυσικού σχεδιασμού το λογικό μοντέλο χαρτογραφείται σε κάποιο φυσικό μοντέλο. Ας σημειωθεί ότι ορισμένες φορές και οι δύο φάσεις αναφέρονται ως "φυσικός σχεδιασμός".

Σημειογραφία του διαγράμματος οντοτήτων – συσχετίσεων

Υπάρχουν διάφορες συμβάσεις για τη σημειογραφία που χρησιμοποιείται στα διαγράμματα οντοτήτων-συσχετίσεων (ERDs). Αρκετές φορές η σημειογραφία που υιοθετείται κατά το λογικό και φυσικό σχεδιασμό μιας βάσης δεδομένων διαφέρει ως προς τη σαφήνεια, τις δυνατότητες που έχει η γραφική γλώσσα, την υποστήριξη από πρότυπα και τα εργαλεία. Συνήθως για τις συσχετίσεις χρησιμοποιούνται ευθείες με διαφορετικές άκρες, ή με σημειώσεις που παριστάνουν την πληθικότητα του τύπου συσχέτισης.

Ιδιαίτερη - κυρίως ιστορική - σημασία έχουν οι παρακάτω σημειογραφίες:

- Η σημειογραφία Chen, του Peter Chen που δημιούργησε τα διαγράμματα το 1976.
- Η σημειογραφία IDEF1X ως de facto πρότυπο, χρησιμοποιούμενο για χρόνια από τις αρχές των ΗΠΑ.
- Η σημειογραφία Martin (πόδι του κόρακα) διαδεδομένη σε εργαλεία για διαγράμματα.
- Η σημειογραφία του Charles Bachman, σε εργαλεία διαγραμμάτων.
- Η σημειογραφία (Min, max), του Jean Raymond Abrial το 1974.
- Η γλώσσα UML, πρότυπο που χρησιμοποιείται ως αντικαταστάτης των διαγραμμάτων Ο/Σ.

5.7.2 Περιγραφή της βάσης

Η υπάρχουσα Βάση Δεδομένων, υλοποιημένη σε MySQL, είναι κατασκευασμένη έτσι ώστε να προσαρμόζεται στις ανάγκες της εφαρμογής. Η Βάση Δεδομένων απαρτίζεται από τους εξής πίνακες:

- owner
- hotel
- coordinates

- reservations
- traveler
- favorites

Ο κώδικας που χρησιμοποιήθηκε είναι ο παρακάτω:

```
CREATE DATABASE IF NOT EXISTS `hotelReservation`;
USE `hotelReservation`;
CREATE TABLE `coordinates` (
  `cooid` int(11) NOT NULL PRIMARY_KEY AUTO_INCREMENT,
  `hotelname` varchar(30) NOT NULL,
  `oidFK` varchar(11) NOT NULL,
  `latitude` varchar(50) NOT NULL,
  `longitude` varchar(50) NOT NULL
PRIMARY_KEY
)
CREATE TABLE `favorites` (
  `fid` int(10) NOT NULL PRIMARY_KEY AUTO_INCREMENT,
  `hotelname` varchar(30) NOT NULL,
  `hotelcity` varchar(30) NOT NULL,
  `userID` varchar(10) NOT NULL
)
```

```
CREATE TABLE `reservations` (  
  `reservationID` int(11) NOT NULL PRIMARY_KEY AUTO_INCREMENT,  
  `hotelname` varchar(30) NOT NULL,  
  `checkin` varchar(20) DEFAULT NULL,  
  `checkout` varchar(20) DEFAULT NULL,  
  `tidFK` varchar(11) DEFAULT NULL,  
  `adults` varchar(10) DEFAULT NULL,  
  `children` varchar(10) DEFAULT NULL,  
  `bedtype` varchar(10) NOT NULL,  
  `roomtype` varchar(10) NOT NULL  
)  
CREATE TABLE `traveler` (  
  `tid` int(11) NOT NULL PRIMARY_KEY AUTO_INCREMENT,  
  `name` varchar(20) DEFAULT NULL,  
  `surname` varchar(30) DEFAULT NULL,  
  `country` varchar(30) NOT NULL,  
  `email` varchar(30) DEFAULT NULL,  
  `password` varchar(80) DEFAULT NULL,  
  `salt` varchar(80) NOT NULL,  
  `telephone` varchar(15) NOT NULL  
)
```

```
CREATE TABLE `hotel` (  
  `hid` int(11) NOT NULL PRIMARY_KEY AUTO_INCREMENT,  
  `hotelname` varchar(20) DEFAULT NULL,  
  `city` varchar(30) DEFAULT NULL,  
  `address` varchar(50) DEFAULT NULL,  
  `tk` varchar(11) DEFAULT NULL,  
  `telephone` varchar(20) DEFAULT NULL,  
  `stars` varchar(1) DEFAULT NULL,  
  `oidFK` varchar(11) DEFAULT NULL,  
  `website` varchar(50) NOT NULL,  
  `swimmingpool` varchar(5) DEFAULT NULL  
)  
CREATE TABLE `owner` (  
  `oid` int(11) NOT NULL PRIMARY_KEY AUTO_INCREMENT,  
  `name` varchar(20) DEFAULT NULL,  
  `surname` varchar(30) DEFAULT NULL,  
  `country` varchar(30) DEFAULT NULL,  
  `email` varchar(30) DEFAULT NULL,  
  `password` varchar(80) DEFAULT NULL,  
  `salt` varchar(80) NOT NULL,  
  `telephone` varchar(15) NOT NULL  
)
```

Κατωτέρω αναλύουμε τον ρόλο και τα πεδία κάθε χαρακτηριστικού γνωρίσματος.

Ο πίνακας owner

Σε αυτόν τον πίνακα καταχωρούνται οι ιδιοκτήτες των ξενοδοχείων όταν δημιουργούν τον χρήστη τους. Πέρα από τα προσωπικά τους στοιχεία σε αυτόν τον πίνακα αποθηκεύονται τα στοιχεία εισόδου και η κρυπτασφάλιση του κωδικού. Παρακάτω ακολουθούν αναλυτικά τα πεδία του πίνακα.

owner	
ΟΝΟΜΑ ΠΕΔΙΟΥ	ΠΕΡΙΓΡΑΦΗ ΠΕΔΙΟΥ
<u>oid</u>	Ο μοναδικός κωδικός κάθε ιδιοκτήτη. Κύριο κλειδί του πίνακα.
name	Το όνομα του ιδιοκτήτη
surname	Το επώνυμο του ιδιοκτήτη
email	Η διεύθυνση e-mail του ιδιοκτήτη
country	Η χώρα καταγωγής του ιδιοκτήτη
telephone	Το προσωπικό τηλέφωνο του ιδιοκτήτη
password	Ο κωδικός για την είσοδο στην εφαρμογή
hash	Η κρυπτασφάλιση του κωδικού για να μην αποστέλεται ως απλό κείμενο

Ο πίνακας hotel

Σε αυτόν τον πίνακα αποθηκεύεται κάθε ξενοδοχείο που δημιουργεί ένας ιδιοκτήτης. Τα πεδία που υπάρχουν είναι τα γενικά στοιχεία του ξενοδοχείου. Παρακάτω ακολουθούν αναλυτικά τα πεδία του πίνακα.

hotel	
ΟΝΟΜΑ ΠΕΔΙΟΥ	ΠΕΡΙΓΡΑΦΗ ΠΕΔΙΟΥ
<u>hid</u>	Ο μοναδικός κωδικός του ξενοδοχείου. Κύριο κλειδί του πίνακα
hotelname	Το όνομα του ξενοδοχείου
stars	Η κατηγορία του ξενοδοχείου με το

	σύστημα των αστεριών
website	Η ιστοσελίδα του ξενοδοχείου
telephone	Το τηλέφωνο του ξενοδοχείου
city	Η πόλη στην οποία βρίσκεται
address	Η διεύθυνση του ξενοδοχείου
tk	Ο ταχυδρομικός κώδικας του ξενοδοχείου
oidFK	Το κυριο κλειδί του ιδιοκτήτη ως συνδετικός κρίκος.

Ο πίνακας coordinates

Σε αυτόν τον πίνακα αποθηκεύονται οι συντεταγμένες του κάθε ξενοδοχείου. Χρησιμοποιείται για την αναζήτηση των ξενοδοχείων στους χάρτες μέσω της εφαρμογής. Παρακάτω ακολουθούν αναλυτικά τα πεδία του πίνακα.

coordinates	
ΟΝΟΜΑ ΠΕΔΙΟΥ	ΠΕΡΙΓΡΑΦΗ ΠΕΔΙΟΥ
<u>cooid</u>	Ο μοναδικός κωδικός των συντεταγμένων στην εφαρμογή. Κύριο κλειδί του πίνακα.
hotelname	Το όνομα του ξενοδοχείου
latitude	Το γεωγραφικό πλάτος στο οποίο βρίσκεται το ξενοδοχείο
longitude	Το γεωγραφικό μήκος στο οποίο βρίσκεται το ξενοδοχείο
hidFK	Το κυριο κλειδί του ξενοδοχείου ως συνδετικός κρίκος.

Ο πίνακας traveler

Σε αυτόν τον πίνακα καταχωρούνται οι ταξιδιώτες που θέλουν να κάνουν κράτηση όταν δημιουργούν τον χρήστη τους. Πέρα από τα προσωπικά τους στοιχεία σε αυτόν τον πίνακα αποθηκεύονται τα στοιχεία εισόδου και η κρυπτασφάλιση του κωδικού. Παρακάτω ακολουθούν αναλυτικά τα πεδία του πίνακα.

owner	
ΟΝΟΜΑ ΠΕΔΙΟΥ	ΠΕΡΙΓΡΑΦΗ ΠΕΔΙΟΥ
<u>tid</u>	Ο μοναδικός κωδικός κάθε ταξιδιώτη. Κύριο κλειδί του πίνακα.
name	Το όνομα του ταξιδιώτη
surname	Το επώνυμο του ταξιδιώτη
email	Η διεύθυνση e-mail του ταξιδιώτη
country	Η χώρα καταγωγής του ταξιδιώτη
telephone	Το προσωπικό τηλέφωνο του ταξιδιώτη
password	Ο κωδικός για την είσοδο στην εφαρμογή
hash	Η κρυπτασφάλιση του κωδικού για να μην αποστέλεται ως απλό κείμενο

Ο πίνακας favorites

Σε αυτόν τον πίνακα ο ταξιδιώτης μπορεί να αποθηκεύει τα αγαπημένα του ξενοδοχεία. Παρακάτω ακολουθούν αναλυτικά τα πεδία του πίνακα.

favorites	
ΟΝΟΜΑ ΠΕΔΙΟΥ	ΠΕΡΙΓΡΑΦΗ ΠΕΔΙΟΥ
<u>fid</u>	Ο μοναδικός κωδικός για κάθε προσθήκη στα αγαπημένα. Κύριο κλειδί του πίνακα.
hotelcity	Η πόλη που βρίσκεται το ξενοδοχείο
hotelname	Το όνομα του ξενοδοχείου
userID	Το κύριο κλειδί του ταξιδιώτη ως συνδετικός κρίκος.

Ο πίνακας reservations

Σε αυτόν τον πίνακα καταχωρούνται όλες οι κρατήσεις που έχουν γίνει. Κατά την χρήση της εφαρμογής μπορεί ο ενδιαφερόμενος να δει τις κρατήσεις που

λέχει κάνει όσο και αυτές που έρχονται στο μέλλον. Παρακάτω ακολουθούν αναλυτικά τα πεδία του πίνακα.

reservations	
ΟΝΟΜΑ ΠΕΔΙΟΥ	ΠΕΡΙΓΡΑΦΗ ΠΕΔΙΟΥ
<u>reservationID</u>	Ο μοναδικός κωδικός κάθε κράτησης. Κύριο κλειδί ου πίνακα
checkin	Ημερομηνία έναρξης της κράτησης
checkout	Ημερομηνία λήξης της κράτησης
adults	Πόσοι ενήλικες θα είναι
children	Πόσα παιδιά θα είναι
bedtype	Τύπος κρεβατιών
hotelname	Το όνομα του ξενοδοχείου
roomtype	Τύπος δωματίου
tidFK	Το κυριο κλειδί του ταξιδιώτη ως συνδετικός κρίκος.

Επίλογος

Στο κεφάλαιο αυτό περιεγράφηκε αρχιτεκτονική των συστημάτων διαχείρισης βάσεων δεδομένων και μέσα από παραδείγματα παρουσιάστηκε η εφαρμογή μίας βάσης δεδομένων. Στο επόμενο κεφάλαιο γίνεται εκτενής περιγραφή της εφαρμογής την οποία πραγματεύεται η εργασία, καθώς και η χρήση της από την μεριά ενός ιδιοκτήτη ξενοδοχείων και ενός ταξιδιώτη.

Κεφάλαιο 6ο: Περιγραφή της Εφαρμογής

Πρόλογος

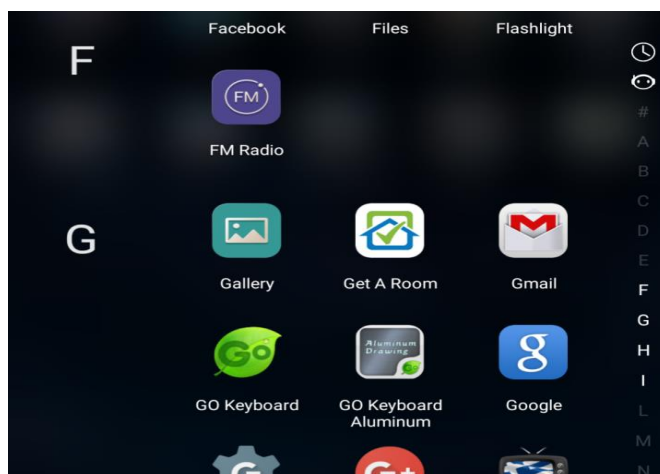
Το κεφάλαιο αυτό εξηγεί και περιγράφει με λεπτομέρεια την χρήση της παρούσας εφαρμογής, από την σκοπιά του χρήστη, αναλύει βασικά σημεία και λειτουργίες της και είναι και ένας οδηγός χρήσης της. Περιγράφει την αρχική εικόνα και την εντύπωση που δέχεται ένας νέος χρήστης όταν την δει για πρώτη φορά. Γίνεται ανάλυση ως προς τον ιδιοκτήτη (Owner) αλλά και ως προς τον ταξιδιώτη (Traveler) και εξηγείται πώς έγινε δυνατή η επικοινωνία της εφαρμογής με έναν απομακρυσμένο server για την διαχείριση των δεδομένων του κάθε χρήστη και την ορθή λειτουργία της. Επίσης, γίνεται αναφορά στην τεχνολογία που χρησιμοποιήθηκε για την επίτευξη της εφαρμογής.

6.1 Αρχική εικόνα

Αρχικά, λαμβάνοντας υπόψιν την εντύπωση που δημιουργεί η εφαρμογή στον χρήστη που την εγκαθιστά για πρώτη φορά, δίνεται έμφαση στην λειτουργία της εφαρμογής από την πλευρά του ταξιδιώτη καθώς και του ιδιοκτήτη ενός ξενοδοχείου. Τα βασικά χαρακτηριστικά της εφαρμογής περιγράφονται παρακάτω. Όπως όλες οι εφαρμογές αυτού του είδους, έτσι και η παρούσα έχει ένα εικονίδιο και τον τίτλο του. Το αρχικό εικονίδιο στην επιφάνεια εργασίας είναι αυτό που ακολουθεί (Εικόνα 6.1) και ο τίτλος της είναι «Get A Room».

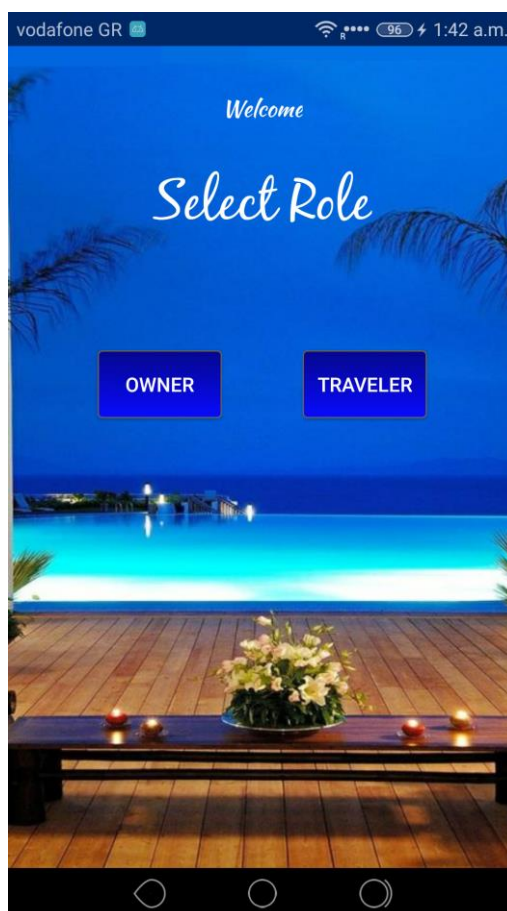


Εικόνα 6.1. Το εικονίδιο εφαρμογής “Get A Room”



Εικόνα 6.2. Θέση του εικονιδίου εφαρμογής μεταξύ άλλων, στην επιφάνεια εργασίας.

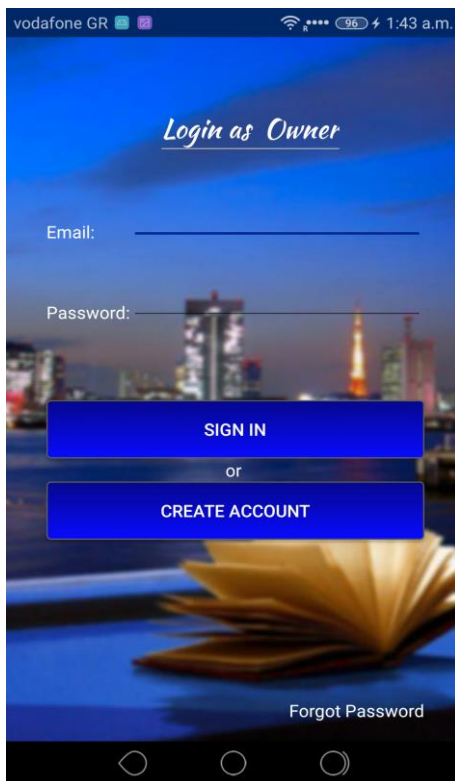
Όταν επιλεγεί το συγκεκριμένο εικονίδιο από τον χρήστη, θα εμφανιστεί η αρχική εικόνα της εφαρμογής, η οποία φαίνεται παρακάτω. (Εικόνα 6.3).



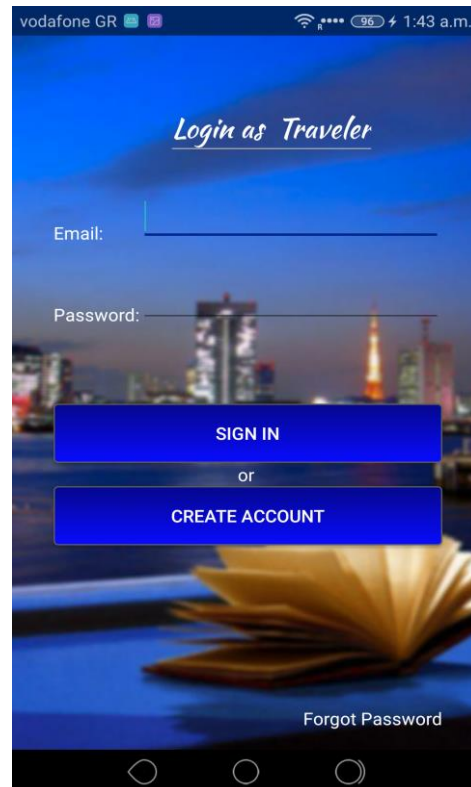
Εικόνα 6.3. Αρχική οθόνη εφαρμογής

Όπως φαίνεται στην Εικόνα 6.3, υπάρχουν δύο επιλογές/κουμπιά, στη διάθεση του χρήστη: η επιλογή “Owner” και η επιλογή “Traveler”. Η κάθε μία από τις επιλογές αυτές θα τον οδηγήσει στην αντίστοιχη οθόνη. Η διαδικασία αλλαγής μεταξύ των οθονών (Activity) γίνεται με συγκεκριμένη εντολή κώδικα, η οποία ενεργεί σαν Event με την εκτέλεση κάποιας ενέργειας που στην παρούσα περίπτωση είναι το πάτημα ενός κουμπιού.

Θεωρώντας ότι ο χρήστης επιλέγει ένα από τα δύο κουμπιά, οι οθόνες που θα εμφανιστούν θα είναι αντίστοιχα οι παρακάτω (Εικόνες 6.4 και 6.5). Η αριστερή εικόνα είναι η οθόνη που εμφανίζεται όταν ο χρήστης επιλέξει να συνδεθεί ως Owner ενώ η δεξιά όταν επιλέξει να συνδεθεί ως Traveler.



Εικόνα 6.5. Η αρχική οθόνη για Owner

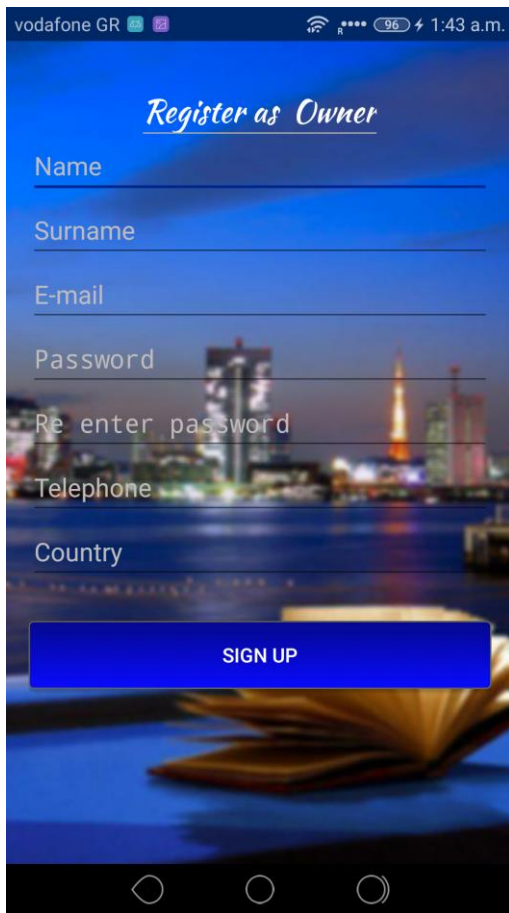


Εικόνα 6.4. Η αρχική οθόνη για Traveler

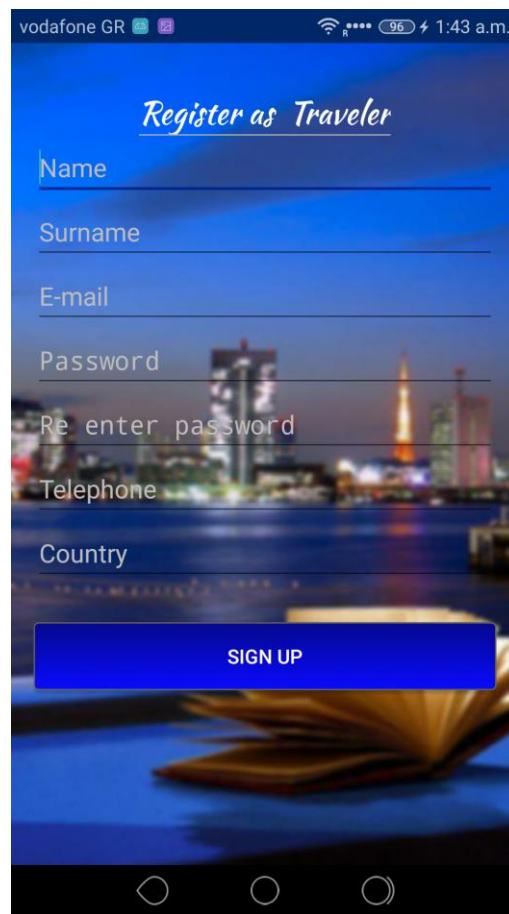
Η μόνη διαφορά στην εμφάνιση (Εικόνες 6.4 και 6.5) είναι το κείμενο: (“Login as Owner” για τον Owner και “Login as Traveler” για τον Traveler). Αυτό γίνεται με την χρήση μίας boolean μεταβλητής, η οποία ανάλογα με το κουμπί που πατάει ο χρήστης θα καταχωρείται και η αντίστοιχη τιμή. Ουσιαστικά, το layout είναι το ίδιο, απλά αλλάζει το string που έχει το κείμενο.

Στην αρχική οθόνη εμφανίζονται στον χρήστη τρεις επιλογές, “Sign in”, “Create account” και “Forgot Password”. Επιλέγοντας μία από αυτές, εμφανίζεται μία άλλη οθόνη ακολουθώντας την ίδια διαδικασία που ακολουθήθηκε και στην προηγούμενη οθόνη.

Θεωρώντας ότι ο χρήστης επιλέγει “Create Account” για να δημιουργήσει έναν λογαριασμό, οι οθόνες για “Owner” και “Traveler” είναι αντίστοιχα οι παρακάτω (Εικόνες 6.6 και 6.7).

The screenshot shows a mobile application interface for registering as an 'Owner'. The title is 'Register as Owner' in a white, cursive font. Below the title are seven text input fields: 'Name', 'Surname', 'E-mail', 'Password', 'Re enter password', 'Telephone', and 'Country'. At the bottom of the form is a blue button with the text 'SIGN UP' in white capital letters. The background of the form is a blurred image of a city skyline at night with a lotus flower in the foreground.

Εικόνα 6. 6. Οθόνη δημιουργίας λογαριασμού για Owner.

The screenshot shows a mobile application interface for registering as a 'Traveler'. The title is 'Register as Traveler' in a white, cursive font. Below the title are seven text input fields: 'Name', 'Surname', 'E-mail', 'Password', 'Re enter password', 'Telephone', and 'Country'. At the bottom of the form is a blue button with the text 'SIGN UP' in white capital letters. The background of the form is a blurred image of a city skyline at night with a lotus flower in the foreground.

Εικόνα 6.7. Οθόνη δημιουργίας λογαριασμού για Traveler.

Για την εμφάνιση του κειμένου “Register as Owner” και “Register as Traveler” η διαδικασία είναι η ίδια όπως προηγουμένως. Εδώ υπάρχουν κάποια πεδία (Text-Field), τα οποία είναι υποχρεωτικά προς συμπλήρωση. Τα “Password” και “Re enter password” πεδία πρέπει να είναι τα ίδια, ώστε να γίνεται επαλήθευση του κωδικού που θα πληκτρολογηθεί από τον χρήστη. Αυτό γίνεται με έναν έλεγχο των δύο Text-Fields ως προς το περιεχόμενό τους. Σε περίπτωση πληκτρολόγησης διαφορετικών κωδικών ή απώλειας συμπλήρωσης κάποιου πεδίου, όταν πατηθεί το κουμπί “Sign

up” θα εμφανιστεί ένα μήνυμα “Toast” το οποίο θα αναφέρει τί λάθος υπάρχει. Σε διαφορετική περίπτωση, θα γίνει επιτυχής εγγραφή του χρήστη στη βάση δεδομένων. Στο σημείο αυτό, πρέπει να αναφερθεί πως για τη διαδικασία αυτή απαιτείται σύνδεση στο διαδίκτυο. Ακολουθως, εξετάζουμε την περίπτωση που ο χρήστης έχει ξεχάσει τον κωδικό, ενώ έχει προηγουμένως κάνει εγγραφή. Επιλέγοντας “Forgot Password” θα οδηγηθεί στην παρακάτω εικόνα (Εικόνα 6.8).



Εικόνα 6.8. Οθόνη *Forgot Password*

Επιλέγοντας “Sent an e-mail” οδηγεί τον χρήστη στο default mail πρόγραμμα που έχει στο κινητό του τηλέφωνο για email (gmail, hotmail κτλ). Εκεί διαπιστώνει πως έχει αυτόματα συμπληρωμένα τα πεδία “Θέμα” και “Κείμενο”, καθώς και τα email που θα στείλει, με την μορφή:

Θέμα: **Forgot Password!**

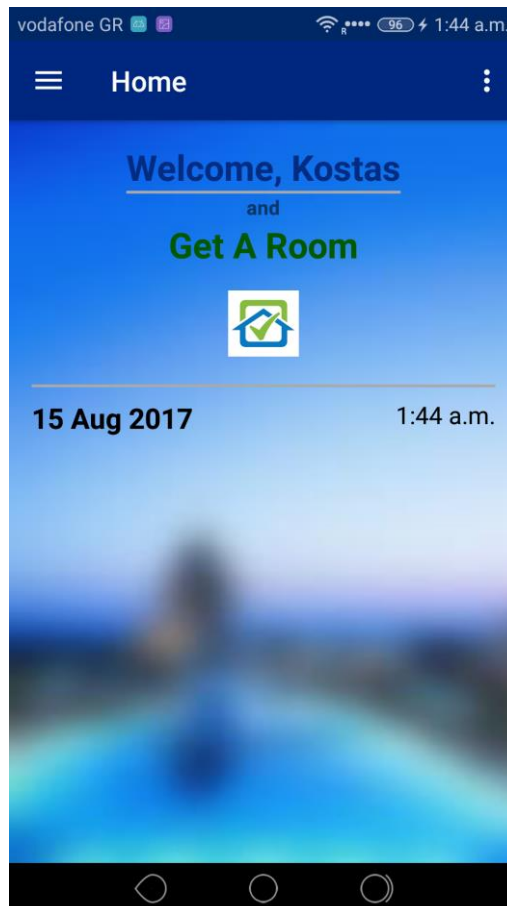
Κείμενο: I forgot my account password at Hotel Reservation app, please give me a new one. My account name is: **εδώ πρέπει να πληκτρολογήσει ο χρήστης το όνομά του καθώς και το email του.*

Τα default email τα οποία η φόρμα έχει συμπληρωμένα και στα οποία στέλνει τις πληροφορίες είναι τα kzaf@it.teithe.gr και ziskatas@it.teithe.gr .

6.2 Χαρακτηριστικά εφαρμογής ως συνδεδεμένος χρήστης

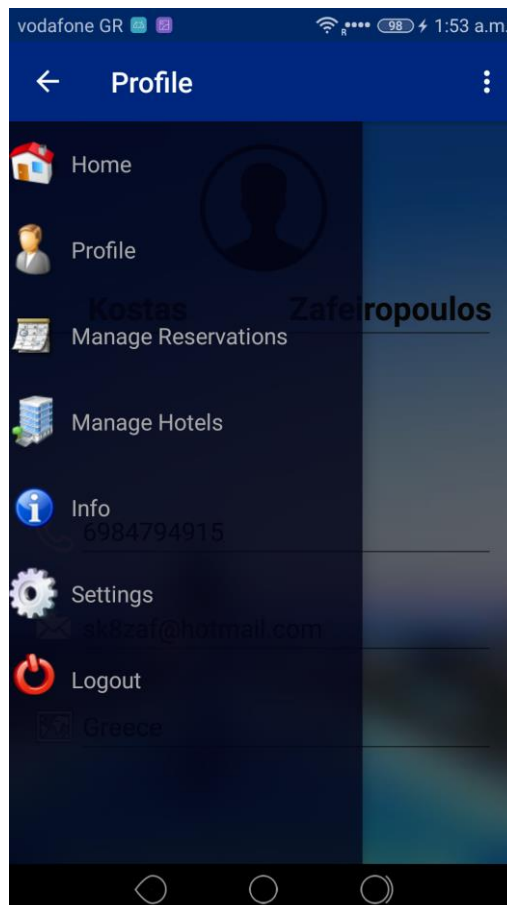
6.2.1 Σύνδεση ως Owner

Έως τώρα περιεγράφηκε το αρχικό interface που θα δει ο χρήστης όταν κατεβάσει την εφαρμογή για πρώτη φορά. Στην παρούσα ενότητα γίνεται λόγος για τα χαρακτηριστικά της εφαρμογής όταν ο χρήστης έχει πλέον συνδεθεί. Στην αρχή, εξετάζονται οι λειτουργίες της εφαρμογής από την πλευρά του ιδιοκτήτη (Owner) ενός ξενοδοχείου. (*Η σύνδεση έγινε με προσωπικό λογαριασμό και τα στοιχεία που ακολουθούν είναι παραδειγματικά*).



Εικόνα 6.9. Αρχική οθόνη

Στην ανωτέρω εικόνα (Εικόνα 6.9) φαίνεται η αρχική εμφάνιση (το Home της εφαρμογής) όταν ο χρήστης συνδεθεί στο λογαριασμό του. Η εφαρμογή δείχνει βασικές πληροφορίες ημερομηνίας και ώρας. Στο πάνω μέρος της οθόνης υπάρχει μία μπάρα (Action-Bar) η οποία δείχνει την κατηγορία στην οποία βρίσκεται ο χρήστης (Home), τρεις γραμμές στα αριστερά και τρεις κατακόρυφες τελείες στα δεξιά. Οι γραμμές δείχνουν πως υπάρχει menu με βασικές επιλογές, ενώ οι τελείες δηλώνουν περαιτέρω επιλογές όπως αποσύνδεση, ρυθμίσεις και λοιπά. Αν ο χρήστης πατήσει το menu, δηλαδή τις τρεις γραμμές, θα εμφανιστεί από τα αριστερά η μπάρα με τις επιλογές, όπως φαίνεται παρακάτω (Εικόνα 6.10). Με μία οποιαδήποτε επιλογή από τις εμφανιζόμενες στο navigation drawer, θα οδηγηθεί και στην αντίστοιχη οθόνη (fragment), η οποία θα αντικαταστήσει αυτήν που φαίνεται με τη νέα. Κάνοντας χρήση της εφαρμογής διαπιστώνει κανείς πως η μπάρα (Action-Bar) δεν αλλάζει, αλλά αλλάζει μόνο η οθόνη που περιέχεται.

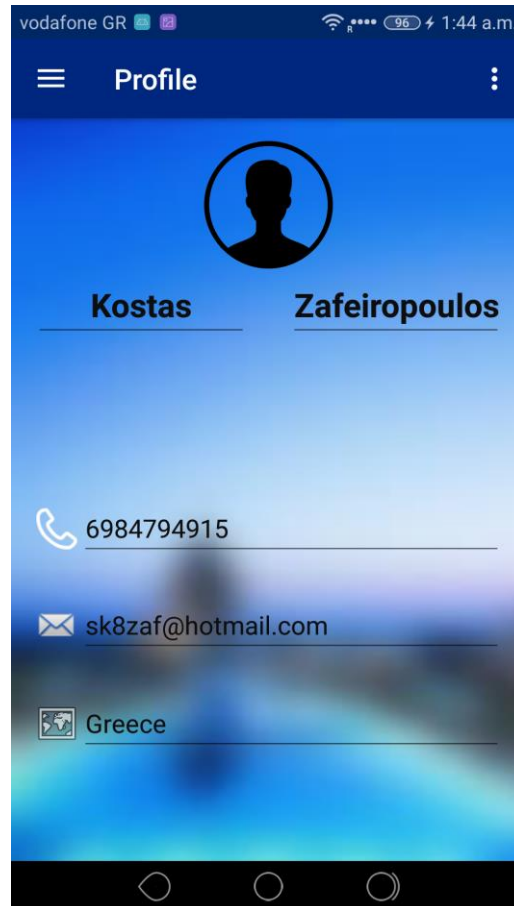


Εικόνα 6.10. Μπάρα επιλογών Owner (Navigation Drawer)

Είναι σημαντικό να γίνει αντιληπτό ότι οι οθόνες αυτές δεν είναι ξεχωριστά Activities, αλλά Fragments τα οποία “πατάνε” πάνω σε ένα γενικό Activity που τα περιέχει. Το Activity αυτό περιέχει την μπάρα (Action-Bar) και το menu, έχοντας έναν container, με τον οποίο εναλλάσσει τα Fragment ανάλογα με την επιλογή από το menu (Navigation Drawer). Είναι parent-child φόρμες, με parent το Activity και child τα Fragments. Οι αρχικές οθόνες (Εικόνες 6.3 έως 6.8) πριν γίνει σύνδεση του χρήστη, είναι ξεχωριστά Activity στα οποία όπως φαίνεται, έχει απενεργοποιηθεί η μπάρα (Action-Bar). Ο προγραμματισμός σε περιβάλλον Android χωρίς τη χρήση Fragment είναι πιο εύκολος, αλλά στην περίπτωση που θέλουμε να χρησιμοποιήσουμε κάποιο menu (και όχι μόνο) όπως στη δική μας, η χρήση των Fragment είναι απαραίτητη. Η δυσκολία έγκειται στη μεταφορά των μεταβλητών και στην επικοινωνία μεταξύ των Activity και των Fragment. Οι επιλογές που έχει ο χρήστης στο menu όταν συνδέεται ως “Owner”, είναι.

- Home
- Profile
- Manage Reservations
- Manage Hotels
- Info
- Settings
- Logout

Θεωρώντας ότι ο χρήστης επιλέγει τη δεύτερη επιλογή (Profile), το περιεχόμενο Fragment θα αλλάξει σε αυτό που αντιστοιχεί στην επιλογή αυτή. Όπως φαίνεται στην Εικόνα 6.11 που ακολουθεί, η επιλογή αυτή περιέχει βασικές πληροφορίες του λογαριασμού του χρήστη. Εδώ έχει τη δυνατότητα να προσθέσει μία φωτογραφία στο προφίλ πατώντας στην εικόνα στη μέση η οποία στη συνέχεια τον οδηγεί στο menu του κινητού ή τάμπλετ για την επιλογή μιας εικόνας. Η εικόνα θα αποθηκευτεί για το συγκεκριμένο λογαριασμό, ωστόσο μπορεί να αλλαχθεί ανά πάσα στιγμή απλά πατώντας πάνω της.



Εικόνα 6.11. Βασικές πληροφορίες χρήστη

Η επιλογή “Home” οδηγεί τον χρήστη στην αρχική οθόνη (Εικόνα 6.9) περιέχοντας ένα εισαγωγικό μήνυμα, καθώς και την ημερομηνία και την ώρα. Η δεύτερη επιλογή, “Profile” (Εικόνα 6.11) περιέχει τις βασικές πληροφορίες του λογαριασμού, όπως όνομα, επώνυμο, τηλέφωνο, email και χώρα. Ο χρήστης και εδώ έχει την επιλογή να ορίσει φωτογραφία στο προφίλ του όπως προαναφέρθηκε.



Εικόνα 6.12. Running Reservations (Owner)

Η τρίτη επιλογή, “Manage Reservations”, είναι μία από τις βασικές λειτουργίες της εφαρμογής, δηλαδή της κατάστασης των κρατήσεων που έχουν γίνει για τα ξενοδοχεία του συγκεκριμένου χρήστη. Όπως φαίνεται, υπάρχουν τρία tabs χωρισμένα σε “Running”, “Upcoming” και “Past”. Με βάση την εκάστοτε τρέχουσα ημερομηνία, οι κρατήσεις χωρίζονται σε αυτές τις κατηγορίες.



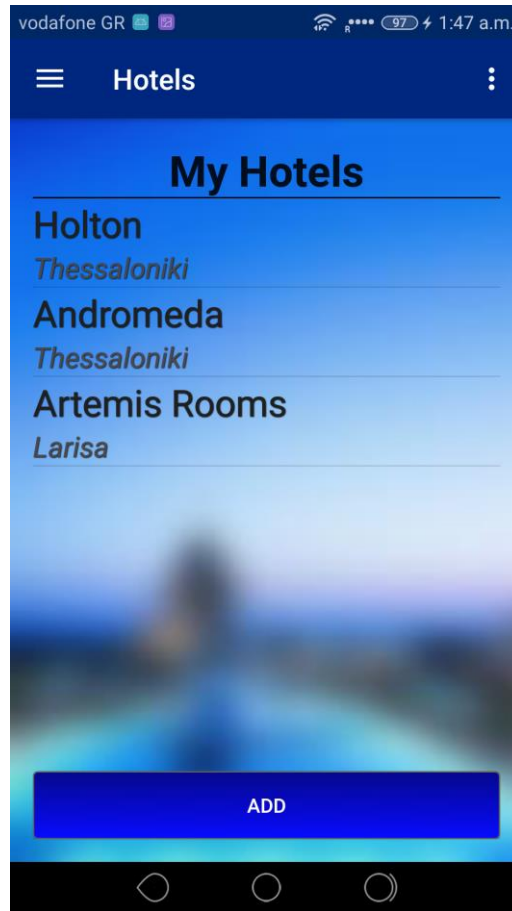
Εικόνα 6.13. *Upcoming Reservations (Owner)*



Εικόνα 6.14. *Past Reservations (Owner)*

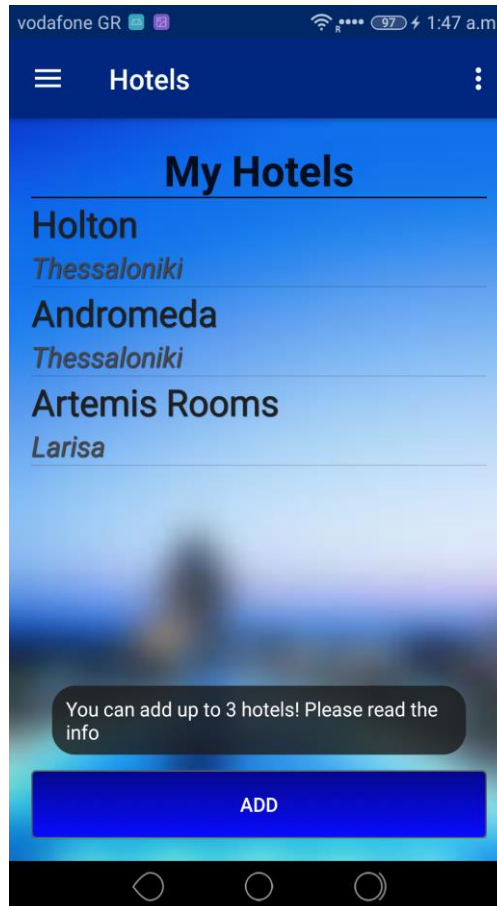
Στην Εικόνα 6.13 φαίνονται οι κρατήσεις που πρόκειται να πραγματοποιηθούν στο κοντινό χρονικό διάστημα, ενώ στην Εικόνα 6.14 διακρίνουμε τις κρατήσεις που ανήκουν πλέον στο παρελθόν. Ο χρήστης έχει τη δυνατότητα να βλέπει την κατάσταση των κρατήσεων στα ξενοδοχεία που του ανήκουν. Η λίστα περιέχει το όνομα του ξενοδοχείου στο οποίο έγινε η κράτηση, το όνομα του χρήστη (Traveler) που έκανε την κράτηση και τις ημερομηνίες check in και check out.

Επόμενη επιλογή από το menu επιλογών του Owner είναι το “Manage Hotels”. Η επιλογή αυτή δείχνει όλα τα ξενοδοχεία που ανήκουν στο συνδεδεμένο χρήστη, καθώς και την πόλη στην οποία βρίσκονται. Για παράδειγμα, στην Εικόνα 6.15 διακρίνονται τα ξενοδοχεία του συνδεδεμένου χρήστη.



Εικόνα 6.13. Τα ξενοδοχεία του χρήστη

Ακολουθως, στις επιλογές της εφαρμογής, ο κάθε ^(Owner) Owner έχει τη δυνατότητα να προσθέσει έως τρία ξενοδοχεία στο λογαριασμό του. Αυτός ο περιορισμός υπάρχει διότι η δυνατότητα προσθήκης απεριόριστων ξενοδοχείων κάνει την εφαρμογή πολύπλοκη στη δημιουργία της. Σε περίπτωση κατά την οποία κάποιος προσπαθήσει να προσθέσει παραπάνω ξενοδοχεία το πρόγραμμα του αποτρέπει την κίνηση αυτή με ένα μήνυμα όπως φαίνεται στην Εικόνα 6.16.



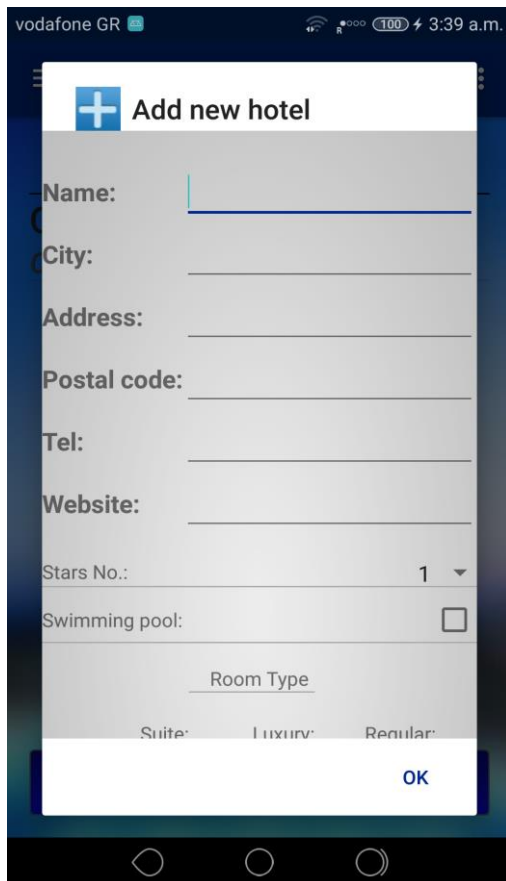
Εικόνα 6.14. Μήνυμα αποτροπής προσθήκης ξενοδοχείου.

Σε αντίθετη περίπτωση εμφανίζεται μια φόρμα (custom Alert Dialog) που απαιτεί από τον Owner να τοποθετήσει στα πεδία προς συμπλήρωση, τα στοιχεία του ξενοδοχείου, (Εικόνα 6.17 και 618).

Τα υποχρεωτικά πεδία του ξενοδοχείου είναι:

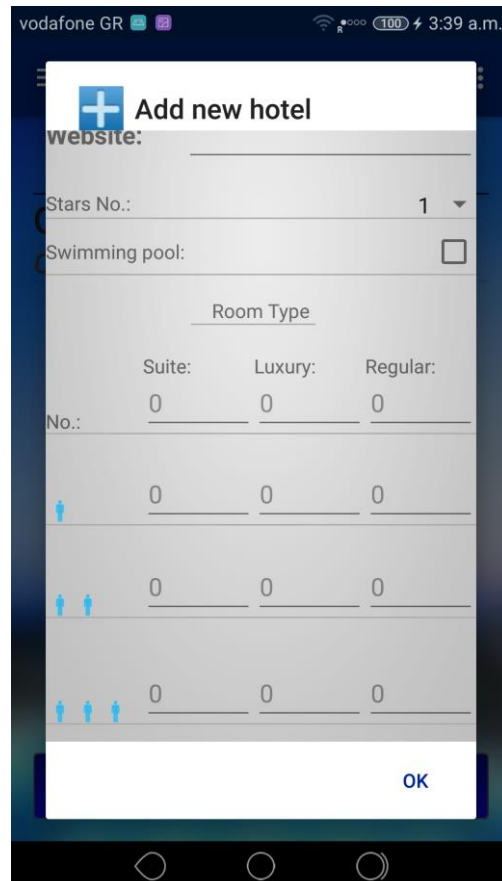
- Name
- City
- Address
- Postal Code
- Tel
- Website
- Stars No
- Swimming Pool

Σε περίπτωση απώλειας ενός από τα πεδία αυτά, το πρόγραμμα αποτρέπει την εγγραφή του ξενοδοχείου και εμφανίζει μήνυμα Toast : “Please fill in all of the fields”.



The screenshot shows a mobile application interface for adding a new hotel. The form is titled "Add new hotel" and contains several input fields: "Name:", "City:", "Address:", "Postal code:", "Tel:", "Website:", "Stars No." (with a dropdown menu set to 1), and "Swimming pool:" (with a checkbox). The form is partially filled out, and an "OK" button is visible at the bottom right.

Εικόνα 6.15. Φόρμα συμπλήρωσης με τα στοιχεία ξενοδοχείου



The screenshot shows the second section of the "Add new hotel" form. It features a table for "Room Type" with columns for "Suite:", "Luxury:", and "Regular:". Each column has a "No.:" label and a text input field containing the number "0". There are also three blue person icons in the left margin of the table. Below the table is an "OK" button.

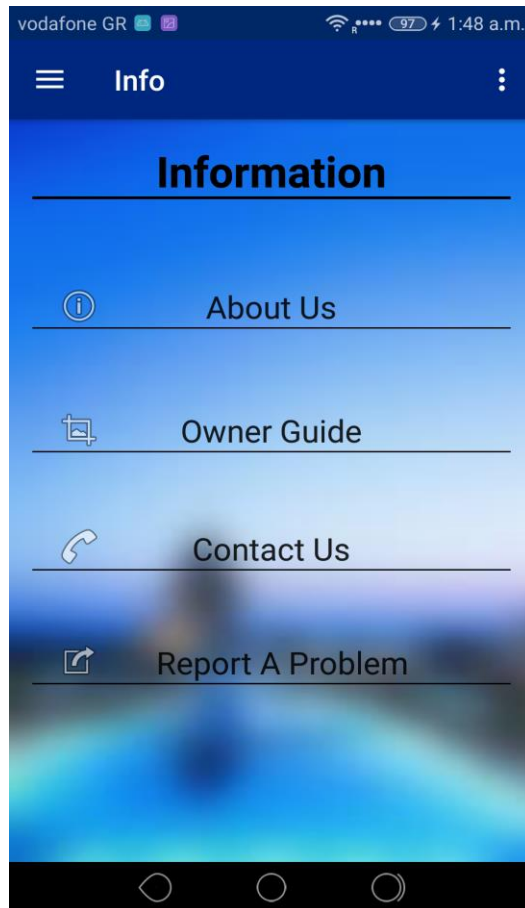
Εικόνα 6.16. Φόρμα συμπλήρωσης με τα στοιχεία ξενοδοχείου (συνέχεια)

Όπως φαίνεται και στις εικόνες, το “Stars No” και το “Swimming Pool” διαφέρουν από τα υπόλοιπα, επειδή δεν απαιτούν κάποια γραπτή συμπλήρωση. Το “Stars No” είναι ένα drop-down menu με επιλογές από το 1 (default) έως το 5 και το “Swimming pool” είναι ένα check box που δίνει μια boolean τιμή true ή false (default) αν έχει πισίνα ή όχι. Η Εικόνα 6.18, η οποία δείχνει το “Room Type”, είναι μια φόρμα συμπλήρωσης με τον αριθμό των διαθέσιμων δωματίων για το κάθε ξενοδοχείο, χωρισμένα σε μονόκλινα, δίκλινα και τρίκλινα καθώς και σε τρεις βασικές κατηγορίες δωματίων οι οποίες είναι Suite, Luxury και Regular. Η δυνατότητα συμπλήρωσης της φόρμας “Room Type” υπάρχει, χωρίς να είναι υποχρεωτική, διότι δε γίνεται εγγραφή των στοιχείων στη βάση δεδομένων. Η επιλογή αυτή είναι μέρος του μελλοντικού σχεδιασμού και επέκτασης της εφαρμογής. (Κεφάλαιο 7.3).

Κατόπιν, εξετάζοντας τα στοιχεία της λίστας του “Owner” βλέπουμε την επιλογή “Info”. Η επιλογή αυτή είναι κοινή για τον Owner και τον Traveler, έχοντας όμως διαφορά στο Guide, όπως θα δούμε και παρακάτω.

Στην επιλογή "Info" (Εικόνα 6.19) έχουμε τέσσερις επιλογές:

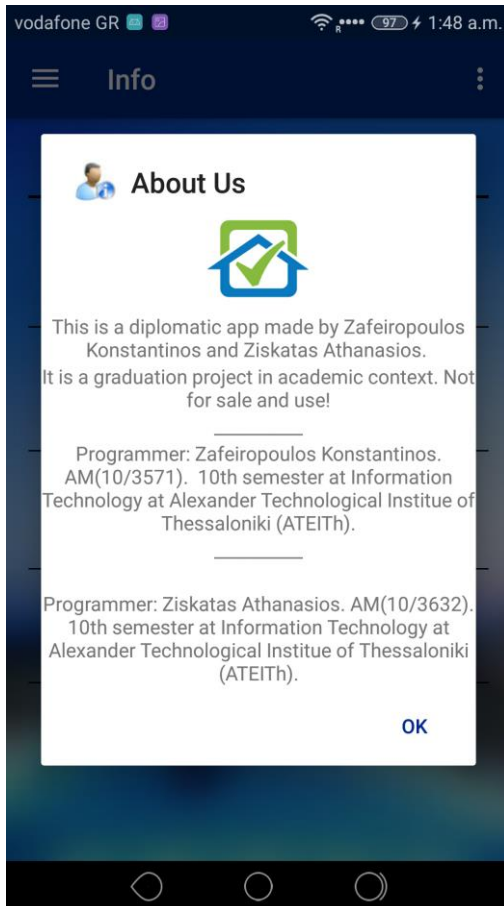
- About Us
- Owner Guide
- Contact Us
- Report A Problem



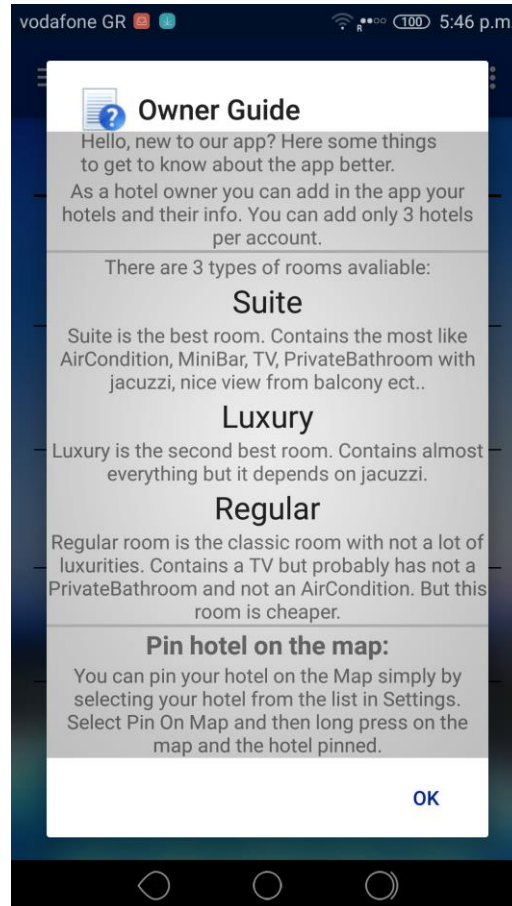
Εικόνα 6.17. Οθόνη «Information» (Owner)

Η πρώτη επιλογή, "About Us", εμφανίζει ένα Alert Dialog, το οποίο περιέχει πληροφορίες για τους δημιουργούς της εφαρμογής. (Εικόνα 6.20).

Η επόμενη επιλογή, "Owner Guide" (Εικόνα 6.21), περιγράφει μέσω ενός Alert Dialog τις επιλογές και τους περιορισμούς που συναντάει ο χρήστης ως ιδιοκτήτης. Εδώ περιγράφονται: το περιεχόμενο της κάθε κατηγορίας δωματίων, οι διαφορές τους καθώς και το γεγονός ότι ο χρήστης δεν έχει την δυνατότητα να ξεπεράσει την δήλωση πέραν των τριών ξενοδοχείων. Επίσης, εξηγεί στο χρήστη τον τρόπο να "καρφισώνει" το ξενοδοχείο στο χάρτη.



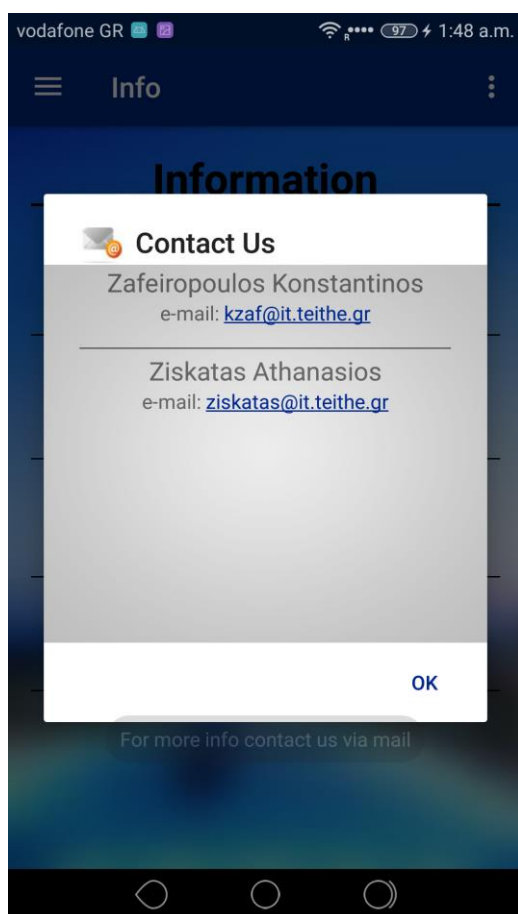
Εικόνα 6.20. Οθόνη «About Us»



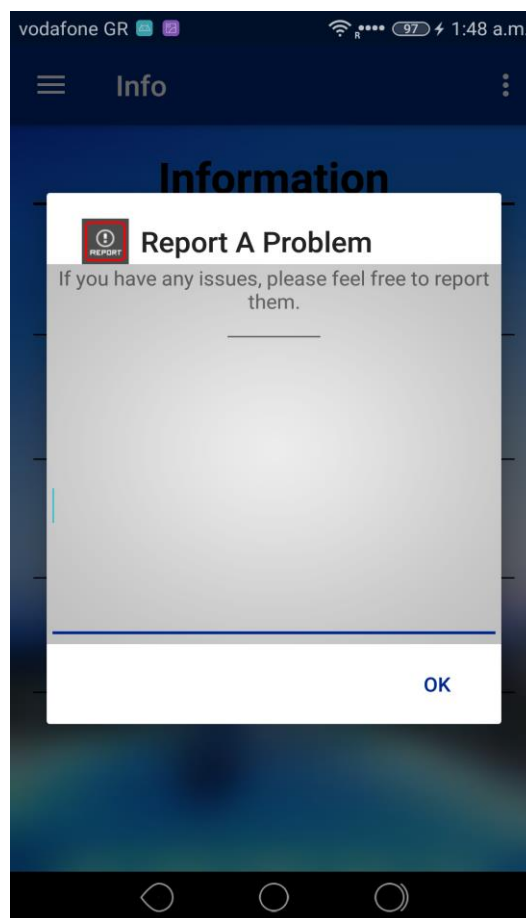
Εικόνα 6.21. Οθόνη «Owner Guide»

Οι επόμενες δύο επιλογές είναι “Contact Us” και “Report A Problem”. (Εικόνες 6.22 και 6.23). Επιλέγοντας “Contact Us”, εμφανίζεται ένα Alert Dialog με τα στοιχεία επικοινωνίας. Πατώντας σε ένα email από τα δύο (kzaf@it.theithe.gr, ziskatas@it.teithe.gr) ο χρήστης έχει τη δυνατότητα να επικοινωνήσει με τους προγραμματιστές.

Με την επιλογή “Report A Problem” ανοίγει ένα νέο Alert Dialog περιέχοντας μία φόρμα στην οποία ο χρήστης μπορεί να δηλώσει οποιοδήποτε παράπονο, σφάλμα ή οποιοδήποτε άλλο ζήτημα. Το μήνυμα αυτό θα προωθηθεί με email στους προγραμματιστές.



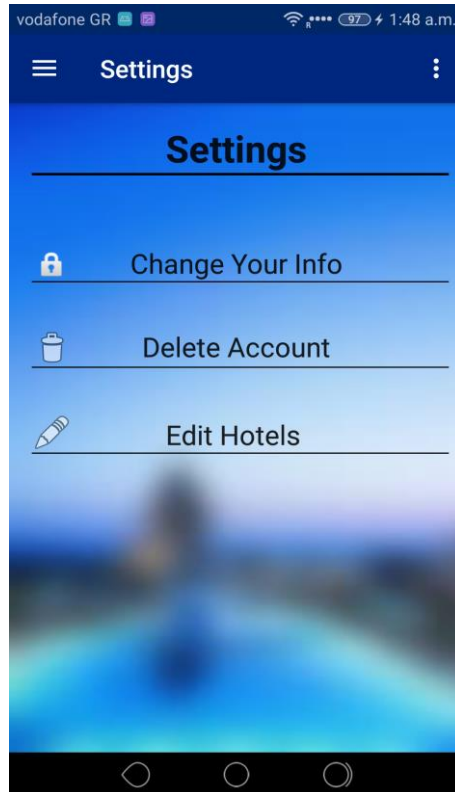
Εικόνα 6.22. Οθόνη «Contact us»



Εικόνα 6.23. Οθόνη «Report a problem»

Στην Εικόνα 6.24 βλέπουμε την οθόνη με τις ρυθμίσεις. Υπάρχουν τρεις επιλογές σε αυτήν την οθόνη:

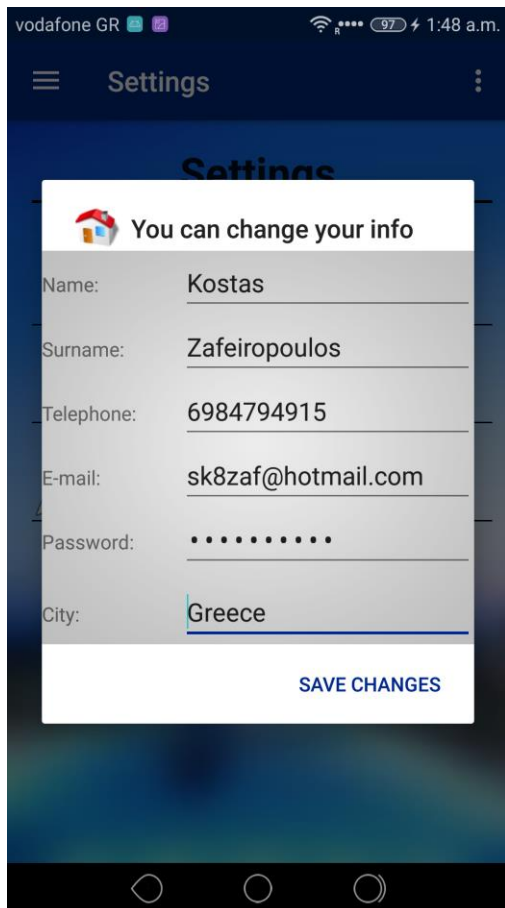
- Change Your Info
- Delete Account
- Edit Hotels



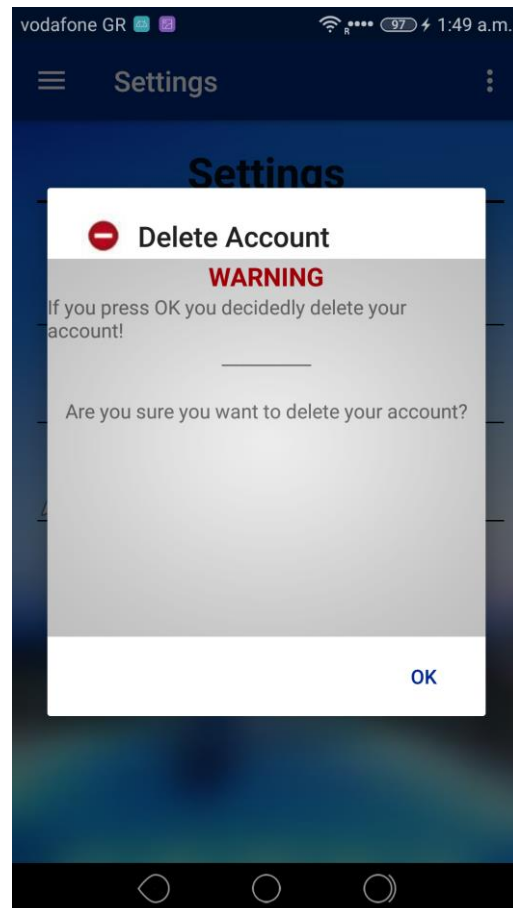
Εικόνα 6.18. Οθόνη ρυθμίσεων

Επιλέγοντας “Change Your Info”, παρέχεται η δυνατότητα στο χρήστη να αλλάξει τις πληροφορίες του λογαριασμού του, όπως όνομα, επώνυμο, τηλέφωνο κλπ. (Εικόνα 6.25). Πατώντας save changes γίνεται UPDATE μέσω SQL στη βάση δεδομένων με τα νέα στοιχεία του χρήστη.

Η δεύτερη επιλογή, “Delete Account” διαγράφει το λογαριασμό τελείως μαζί με τα ξενοδοχεία του συγκεκριμένου χρήστη. Το Alert Dialog (Εικόνα 6.26) προειδοποιεί τον χρήστη ότι η διαγραφή θα είναι οριστική. Σε συνέχεια της πράξης αυτής, η εγγραφή του χρήστη μαζί με τα ξενοδοχεία και τις αποθηκευμένες συντεταγμένες τους στη βάση δεδομένων καταργείται μέσω μιας εντολής DELETE στην SQL. Θα δούμε πιο αναλυτικά παρακάτω τον κώδικα PHP και τις σχετικές εντολές SQL που βρίσκονται στο server μαζί με τη βάση δεδομένων.



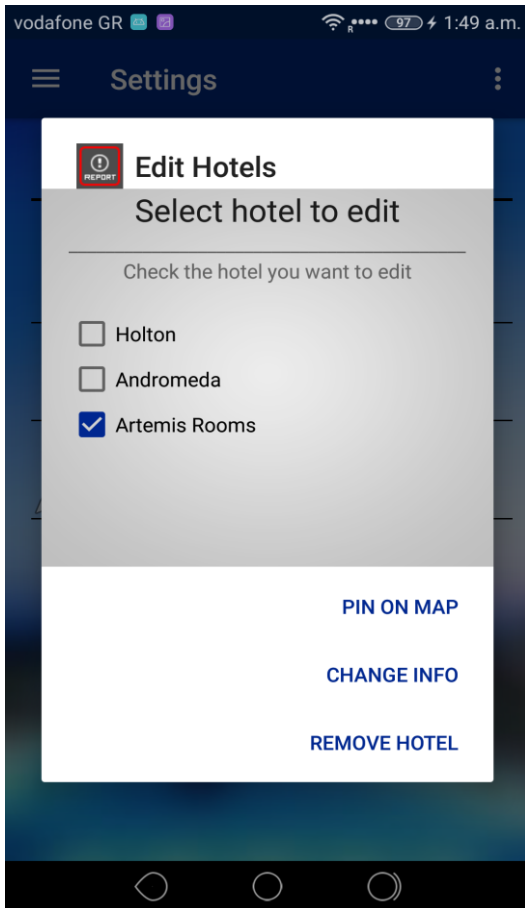
Εικόνα 6.19. Οθόνη κατάργησης λογαριασμού



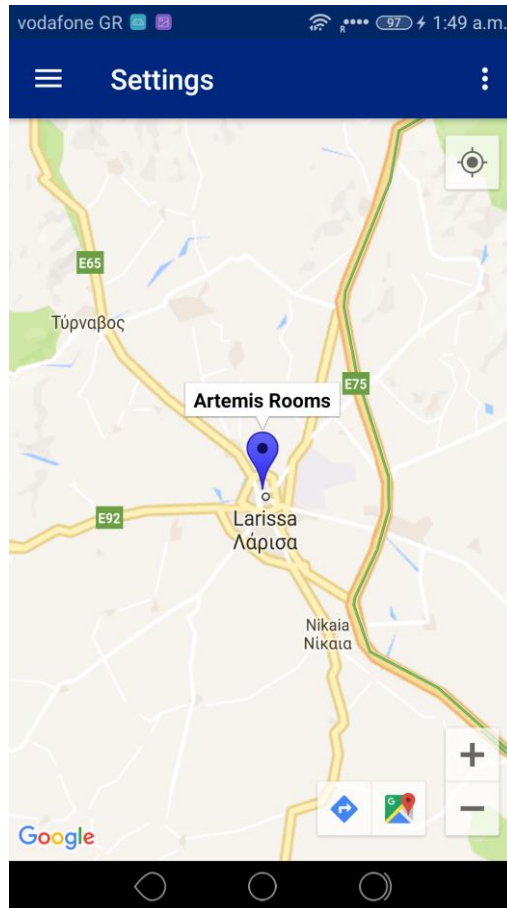
Εικόνα 6.20. Οθόνη διαγραφής λογαριασμού

Παρόμοια διαδικασία ακολουθείται και στην τρίτη επιλογή, "Edit Hotels". Εδώ, δίνεται η δυνατότητα στο χρήστη να αλλάξει τις πληροφορίες του κάθε ξενοδοχείου που του ανήκει. Για παράδειγμα, στην Εικόνα 6.27 βλέπουμε ένα Alert Dialog με τα ξενοδοχεία του χρήστη με τον οποίο είμαστε συνδεδεμένοι. Οι επιλογές είναι αρκετές. Ο χρήστης έχει την δυνατότητα να αλλάξει τα στοιχεία του επιλεγμένου ξενοδοχείου, να διαγράψει το ξενοδοχείο, ή να επιλέξει την τοποθεσία του στον παγκόσμιο χάρτη.

Στην Εικόνα 6.27 βλέπουμε να είναι επιλεγμένο το ξενοδοχείο με όνομα "Artemis Rooms". Σε περίπτωση επιλογής δεύτερου check box βγαίνει μήνυμα Toast: "Please select ONLY ONE hotel at a time" και καμία περαιτέρω ενέργεια δεν είναι δυνατή.



Εικόνα 6.21. Dialog με τα ξενοδοχεία του χρήστη ως επιλογή για περαιτέρω ενέργειες.

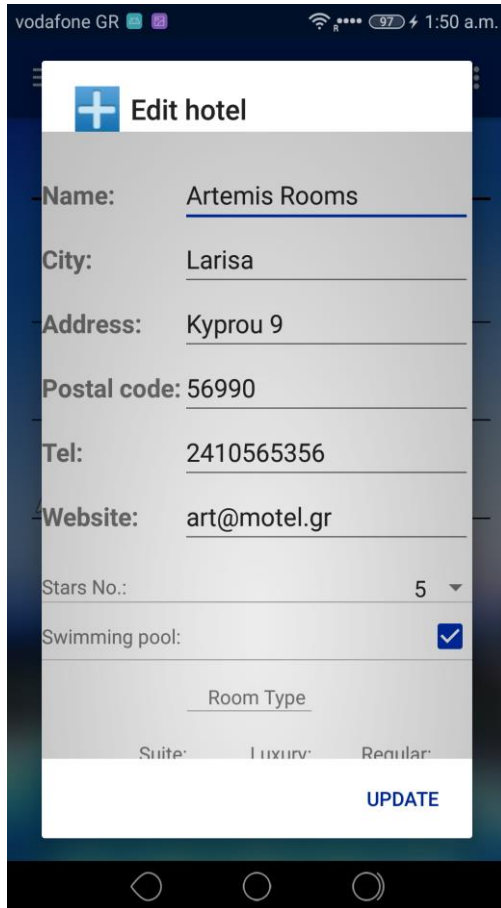


Εικόνα 6.22. Το ξενοδοχείο όπως φαίνεται στον χάρτη στο σημείο που το «καρφιτσώνει» ο χρήστης.

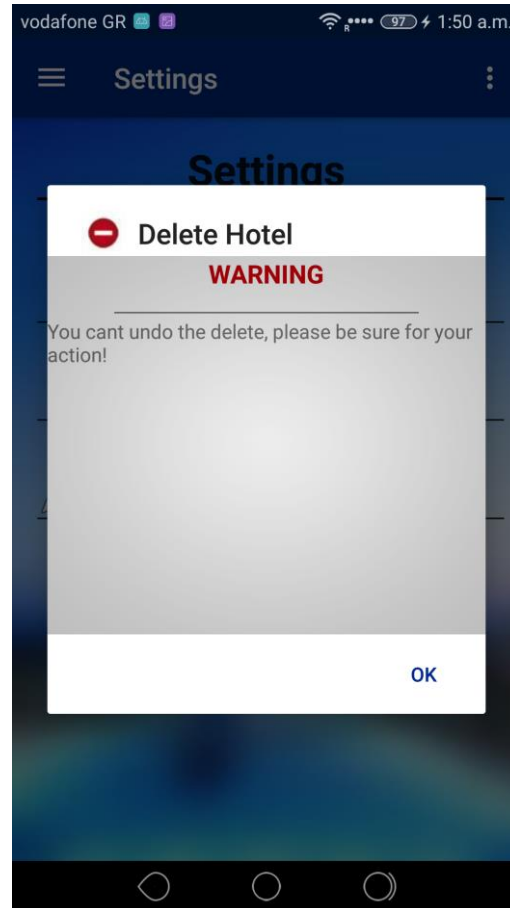
Εφόσον ο χρήστης επιλέξει ένα ξενοδοχείο, όπως φαίνεται στο παράδειγμα, έχει τη δυνατότητα να χρησιμοποιήσει μία από τις τρεις επιλογές.

Στην Εικόνα 6.28 φαίνεται η οθόνη μετά την επιλογή pin on map, στην οποία ο χρήστης έχει τη δυνατότητα να «καρφιτσώσει» το επιλεγμένο ξενοδοχείο στο χάρτη με παρατεταμένο πάτημα στο σημείο που θέλει. Τότε γίνεται μία εγγραφή στη βάση δεδομένων με τις συγκεκριμένες συντεταγμένες και έπειτα το ξενοδοχείο είναι διαθέσιμο στο χάρτη να το δουν όλοι οι χρήστες οποιαδήποτε στιγμή.

Άλλη μία επιλογή είναι η “Change Info”. Όπως στην περίπτωση του ιδιοκτήτη, έτσι και εδώ, ο χρήστης έχει τη δυνατότητα να αλλάξει τις πληροφορίες του επιλεγμένου ξενοδοχείου. Στην Εικόνα 6.29 φαίνεται το Alert Dialog με τις



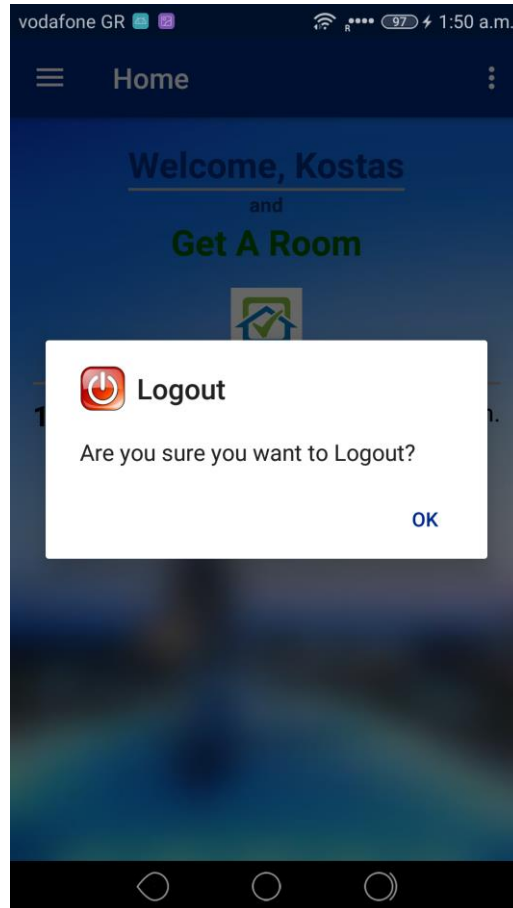
Εικόνα 6.23. Alert Dialog με τις πληροφορίες του επιλεγμένου ξενοδοχείου και την δυνατότητα αλλαγής τους



Εικόνα 6.24. Alert Dialog με την οθόνη ενημέρωσης για οριστική διαγραφή του ξενοδοχείου

πληροφορίες του ξενοδοχείου. Εκεί ο χρήστης μπορεί να αλλάξει οποιαδήποτε πληροφορία και όταν επιλέξει Update γίνεται μέσω SQL ένα UPDATE στη βάση δεδομένων στην οποία ανανεώνονται τα στοιχεία του ξενοδοχείου.

Στις ρυθμίσεις των ξενοδοχείων, υπάρχει και η επιλογή της διαγραφής ενός ξενοδοχείου οριστικά. Όπως και προηγουμένως (Εικόνα 6.26), έτσι και στην Εικόνα 6.30 έχουμε το μήνυμα ενημέρωσης για την οριστική διαγραφή του ξενοδοχείου. Πατώντας OK το ξενοδοχείο διαγράφεται μαζί με τις συντεταγμένες, αν έχουν οριστεί στο χάρτη. Και πάλι η διαδικασία είναι να εκτελεσθεί μία εντολή DELETE στη βάση δεδομένων.



Εικόνα 6.25. Alert Dialog για αποσύνδεση του χρήστη

Τέλος, αναφερόμενοι ξανά στη λίστα με τις επιλογές του Drawer (Εικόνα 6.10) και επιλέγοντας την τελευταία επιλογή Logout (Εικόνα 6.31), δείχνει στον χρήστη ένα Alert Dialog ενημερώνοντάς τον ότι θα γίνει αποσύνδεση του λογαριασμού του και θα επιστρέψει στην αρχική οθόνη. (Εικόνα 6.3).

6.2.2 Σύνδεση ως Traveler

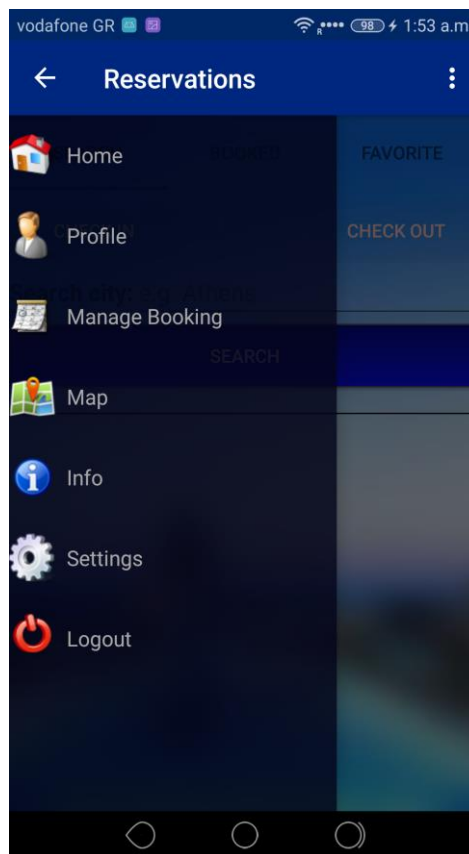
Ολοκληρώνοντας την περιγραφή του περιβάλλοντος του ιδιοκτήτη (Owner), περνάμε στην περιγραφή του περιβάλλοντος του ταξιδιώτη (Traveler). Το γενικό

layout είναι το ίδιο με αυτό του ιδιοκτήτη. αλλά εδώ υπάρχουν κάποιες επιλογές που φαίνονται ίδιες έχοντας μικροδιαφορές, αλλά όμως διαφέρουν στην βασική τους λειτουργία. Παρακάτω θα δούμε αναλυτικά τις επιλογές ενός συνδεδεμένου χρήστη ως ταξιδιώτη.

Επιλέγοντας στην Εικόνα 6.3 ο χρήστης το TRAVELER, οδηγείται στην οθόνη σύνδεσης ως traveler. (Εικόνα 6.4). Θεωρώντας πως έχει ήδη λογαριασμό, συνδέεται στην εφαρμογή. Η αρχική εικόνα που θα δει θα είναι παρόμοια με την αρχική (Home) που βλέπει κανείς συνδεδεμένος ως ιδιοκτήτης. (Εικόνα 6.9).

Το μενού επιλογών του drawer του ταξιδιώτη φαίνεται στην Εικόνα 6.32.

Διαπιστώνουμε ότι οι επιλογές είναι σχεδόν οι ίδιες με αυτές του ιδιοκτήτη, έχοντας



Εικόνα 6.26. Μενού επιλογών «Traveler»
(Navigation Drawer)

διαφορές σε μερικά σημεία. Πρέπει να τονισθεί σε αυτό το σημείο, πως κάποια Fragment μπορεί να είναι τα ίδια και για τις δύο επιλογές χρηστών, όπως για παράδειγμα το Home, με τη διαφορά απλά το Text-Field το οποίο θα περιέχει το όνομα του χρήστη όταν συνδεθεί στο λογαριασμό του.

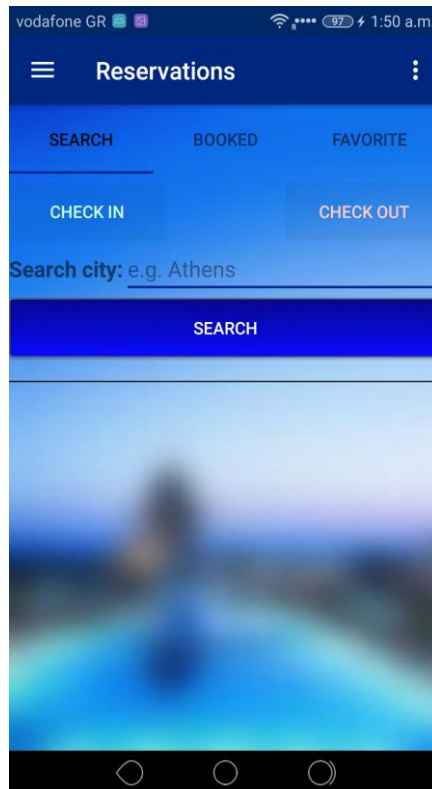
Οι επιλογές που έχει ο χρήστης, όταν συνδέεται ως “Traveler” στο menu του είναι:

- Home

- Profile
- Manage Booking
- Map
- Info
- Settings
- Logout

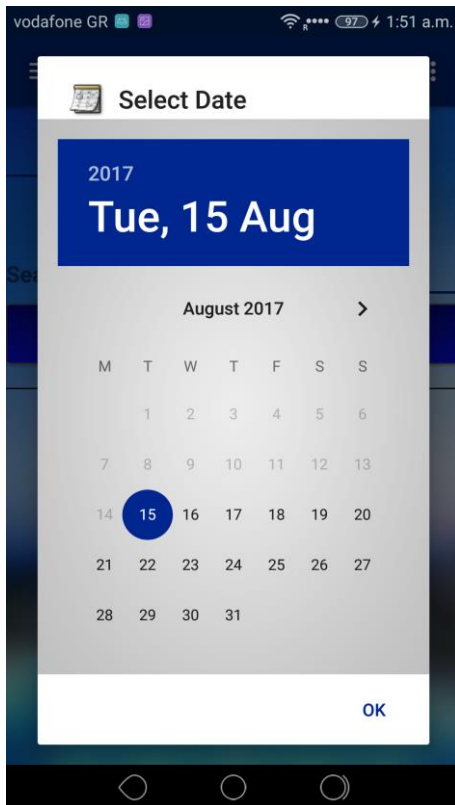
Οι επιλογές Home και Profile είναι οι ίδιες με αυτές του ιδιοκτήτη, δηλαδή, όπως προαναφέρθηκε δεν χρησιμοποιούν διαφορετικό Fragment αλλά χρησιμοποιούν το ίδιο με τη διαφορά ότι ενημερώνουν τα Text-field όπως όνομα, επώνυμο, email και τα λοιπά σύμφωνα με τα στοιχεία του κάθε συνδεδεμένου χρήστη. Οι πληροφορίες αυτές παρέχονται μέσω σύνδεσης της εφαρμογής στην βάση δεδομένων σε επιλεγμένο server. Ο τρόπος αυτός περιγράφεται παρακάτω.

Στην Εικόνα 6.33 φαίνεται η οθόνη που βλέπει ο χρήστης όταν επιλέξει Manage Booking. Υπάρχουν κι εδώ τρία tabs με επιλογές στο πάνω μέρος της οθόνης. Αρχικά εμφανίζεται το tab του search, εκεί όπου ο χρήστης έχει την δυνατότητα να αναζητήσει ξενοδοχείο βασιζόμενος στις ημερομηνίες check in – check out και την πόλη διαμονής. Υπάρχει ένα πεδίο συμπλήρωσης για την πόλη και δύο κουμπιά στο πάνω μέρος με το check in και το check out. Εάν ο χρήστης δεν ορίσει κάποιο από αυτά, η αναζήτηση δεν είναι δυνατή και εμφανίζεται κατάλληλο μήνυμα (Toast) ενημέρωσης. Όταν οριστούν όλα τα πεδία, τότε πατώντας το κουμπί search, στο κάτω μέρος της οθόνης εμφανίζεται μία λίστα με τα διαθέσιμα ξενοδοχεία της συγκεκριμένης πόλης. (Εικόνα 6.37).

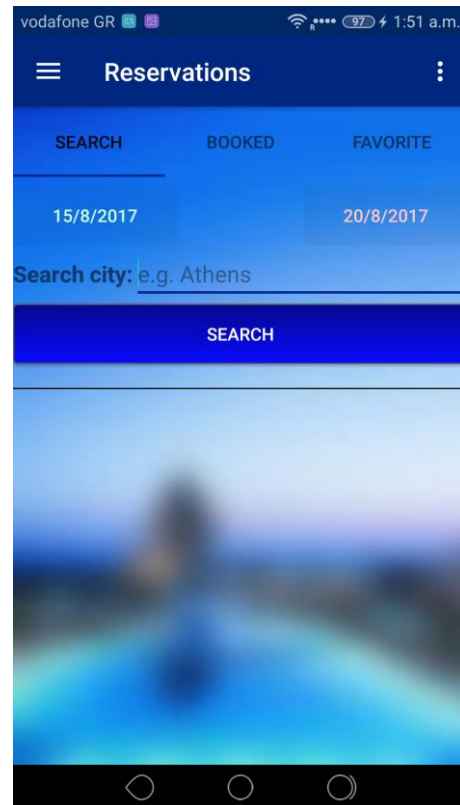


Εικόνα 6.27. Οθόνη «Manage Booking»
(Traveler)

Όταν ο χρήστης πατήσει το κουμπί check in, θα εμφανιστεί ένα Alert Dialog με το ημερολόγιο το οποίο ξεκινάει από την ημερομηνία δήλωσης της κράτησης του δωματίου, αποφεύγοντας έτσι το ενδεχόμενο επιλογής κράτησης σε προηγούμενη ημερομηνία (Εικόνα 6.34). Το ίδιο θα συμβεί και στην επιλογή του κουμπιού check out, ώστε εκεί να ορίσει ο χρήστης την ημερομηνία που θέλει να αποχωρήσει. Σε περίπτωση που ορίσει ημερομηνία μικρότερη από την ημερομηνία check in, θα υπάρξει μήνυμα αποτροπής και θα του ζητήσει να ξαναθέσει σωστές ημερομηνίες. Εφόσον αυτό γίνει, οι τιμές θα οριστούν στα κουμπιά, όπως φαίνεται στην Εικόνα 6.35. (Οι ημερομηνίες που αναγράφονται είναι παραδειγματικές).

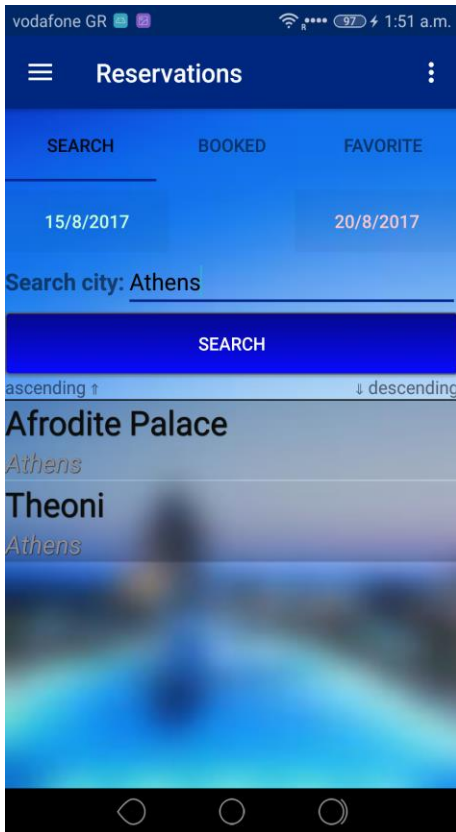


Εικόνα 6.28. Επιλογή ημερομηνίας
(Alert Dialog)

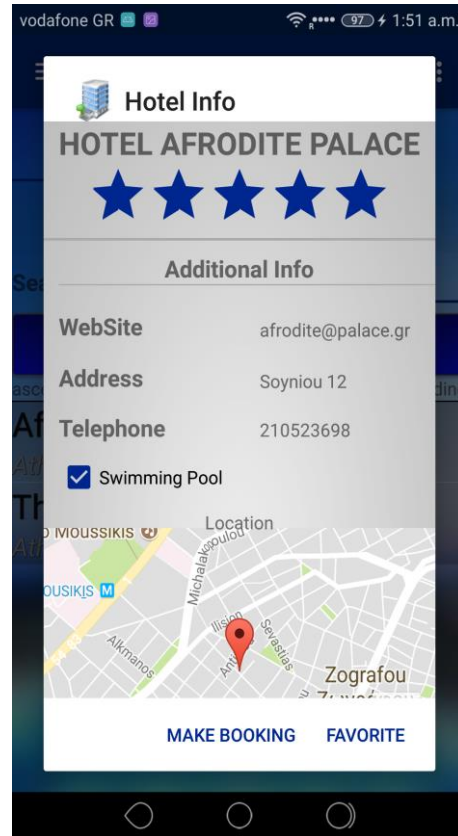


Εικόνα 6.29. Οθόνη έπειτα από επιλογή
ημερομηνιών check in και check out

Πρέπει κατόπιν να οριστεί η πόλη στην οποία θέλει ο χρήστης να ταξιδέψει ώστε να εμφανιστούν τα αποτελέσματα. Για παράδειγμα, χρησιμοποιώντας την Αθήνα (Εικόνα 6.36) και πατώντας το κουμπί αναζήτησης εμφανίζεται από κάτω μία λίστα με τα διαθέσιμα ξενοδοχεία. Τα αποτελέσματα αυτά γίνονται μέσω μιας αναζήτησης στην βάση δεδομένων. Εδώ, ο χρήστης έχει την επιλογή να διαλέξει το ξενοδοχείο που θέλει, απλά πατώντας πάνω του, ώστε να εμφανιστεί ένα νέο Alert Dialog (Εικόνα 6.37) με τις πληροφορίες του επιλεγμένου ξενοδοχείου. Στο παράδειγμα επιλέχθηκε το ξενοδοχείο Afrodite Palace στην Αθήνα. Όπως φαίνεται το Dialog περιέχει όλες τις πληροφορίες γι' αυτό, όπως: το νούμερο των αστεριών, την ιστοσελίδα, την διεύθυνση, το τηλέφωνο, αν έχει η όχι πισίνα (checkbox) και επιπλέον σε ποιο σημείο βρίσκεται στον παγκόσμιο χάρτη.



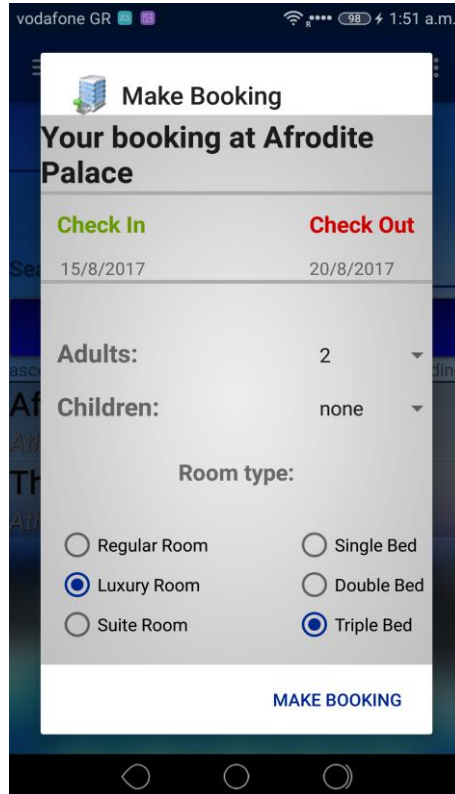
Εικόνα 6.36. Οθόνη με τα αποτελέσματα έπειτα από χρήση της αναζήτησης.



Εικόνα 6.37. Alert Dialog με τις πληροφορίες του επιλεγμένου ξενοδοχείου (Traveler).

Ο χρήστης μπορεί να επιλέξει αν θέλει να κάνει κράτηση στο ξενοδοχείο αυτό, πατώντας make booking ή μπορεί να το αποθηκεύσει στα αγαπημένα του πατώντας favorite. Σε οποιαδήποτε περίπτωση μπορεί να δει τις καταστάσεις κρατήσεων επιλέγοντας τα άλλα tabs, τα οποία θα εξηγηθούν παρακάτω.

Επιλέγοντας make booking, προτρέπεται ο χρήστης να ορίσει βασικές πληροφορίες της κράτησης του μέσω ενός νέου Alert Dialog παραθύρου (Εικόνα 6.38). Όπως φαίνεται, η ημερομηνίες κράτησης είναι ήδη ορισμένες. Στην κράτηση θα οριστεί ο αριθμός των ατόμων και θα γίνει, ο χωρισμός σε ενήλικες και παιδιά. Το εύρος τιμών για τον αριθμό ενηλίκων είναι ένας έως πέντε, ενώ για τα παιδιά είναι από μηδέν έως πέντε.



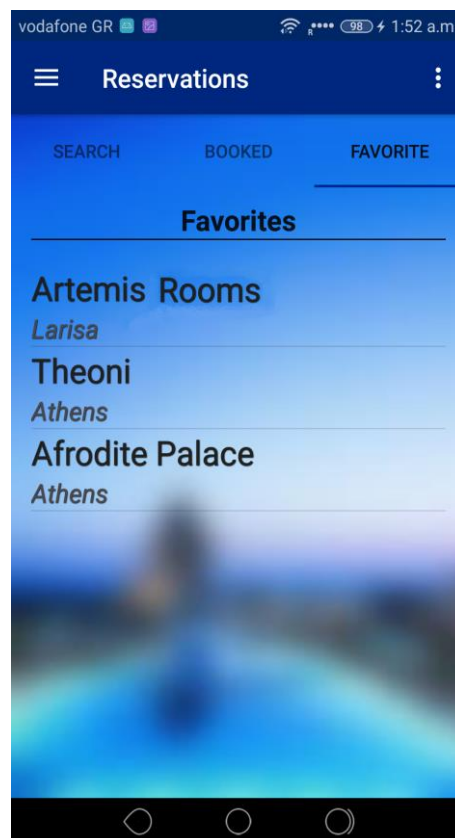
Εικόνα 6.30. Alert Dialog για συμπλήρωση στοιχείων κράτησης (Traveler)

Επίσης, ο χρήστης έχει να επιλέξει ανάμεσα στις τρεις κατηγορίες δωματίων (Regular, Luxury και Suite) καθώς και το νούμερο των κρεβατιών (Single, Double και Triple). Επιλέγοντας make booking γίνεται η κράτηση την οποία έπειτα ο ιδιοκτήτης του ξενοδοχείου μπορεί να δει (Εικόνα 6.12).

Επιστρέφοντας στην Εικόνα 6.36, η δεύτερη (favorite) επιλογή είναι να προσθέσει ο χρήστης το επιλεγμένο ξενοδοχείο στην λίστα με τα αγαπημένα. Στην Εικόνα 6.35 φαίνεται πως υπάρχουν δύο ακόμα tabs, το ένα (bookings) καταγράφει τις καταστάσεις με τις κρατήσεις που έκανε ο συγκεκριμένος χρήστης ενώ το άλλο (favorites) καταγράφει την λίστα με τα αγαπημένα του ξενοδοχεία. Οι οθόνες με αυτές τις πληροφορίες φαίνονται στις Εικόνες 6.39 και 6.40.

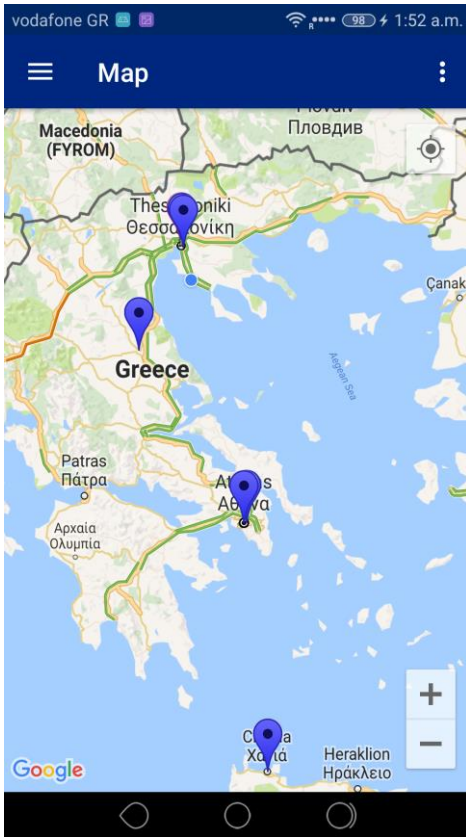


Εικόνα 6.31. Λίστα με τις κρατήσεις του χρήστη

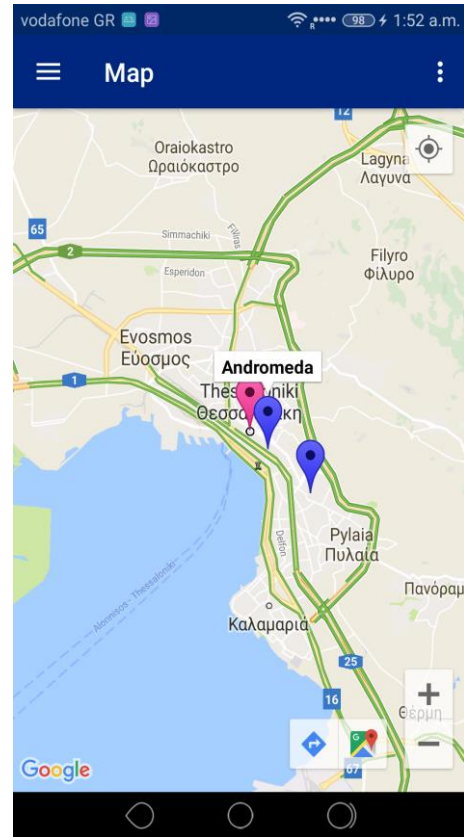


Εικόνα 6.32. Λίστα με τα αγαπημένα ξενοδοχεία του χρήστη

Στην Εικόνα 6.39 φαίνεται στην λίστα το όνομα του ξενοδοχείου και από κάτω οι ημερομηνίες κράτησης, ενώ στην Εικόνα 6.40 φαίνονται το όνομα του ξενοδοχείου και η πόλη.



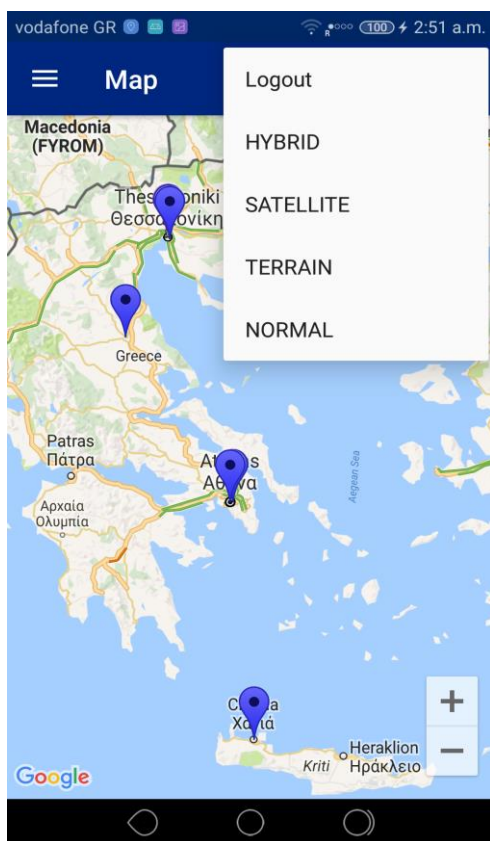
Εικόνα 6.33. Οθόνη Map με γενική εικόνα της Ελλάδας και τα ξενοδοχεία



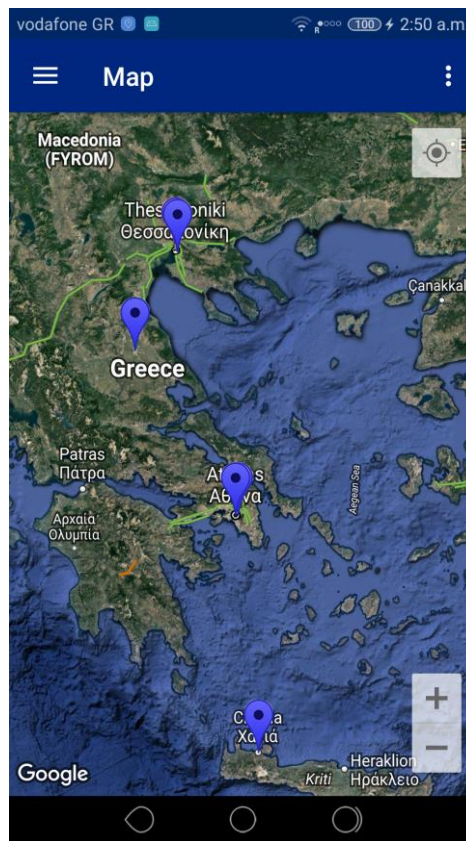
Εικόνα 6.34. Οθόνη Map με την Θεσσαλονίκη και τα ξενοδοχεία της

Περνώντας από τις κρατήσεις και πηγαίνοντας στην επόμενη επιλογή (Map) από την λίστα του μενού, ο ταξιδιώτης έχει την δυνατότητα να δει τα διαθέσιμα ξενοδοχεία στον χάρτη, εφόσον ο ιδιοκτήτης τα έχει «καρφίσει». Στην Εικόνα 6.41 φαίνεται μία γενική εικόνα όλης της Ελλάδας με τα ξενοδοχεία που μέχρι στιγμής έχουν οριστεί, ενώ στην Εικόνα 6.41 φαίνεται συγκεκριμένα η Θεσσαλονίκη και τα ξενοδοχεία της, έχοντας επιλεγμένο το ξενοδοχείο Andromeda.

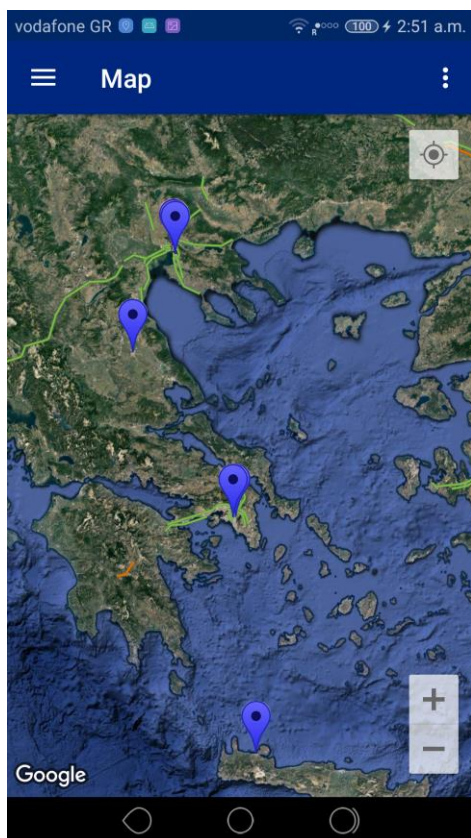
Επιλέγοντας το μενού πάνω δεξιά (τρεις τελείες) (Εικόνα 6.43) ο ταξιδιώτης μπορεί να επιλέξει ανάμεσα σε τέσσερις διαφορετικές μορφές χάρτη, Hybrid, Satellite, Terrain και Normal η οποία είναι και η προκαθορισμένη μορφή. Στις Εικόνες 6.44, 6.45 και 6.46 φαίνεται η μορφή του χάρτη αντίστοιχα σε Normal, Hybrid, Satellite και Terrain



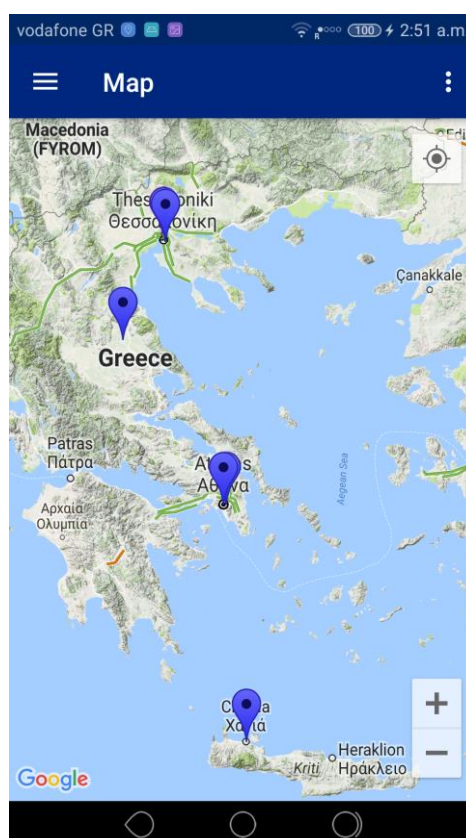
Εικόνα 6.43. Μορφή χάρτη ως Normal μαζί με Μενού επιλογών μορφής χάρτη



Εικόνα 6.44. Μορφή χάρτη ως Hybrid



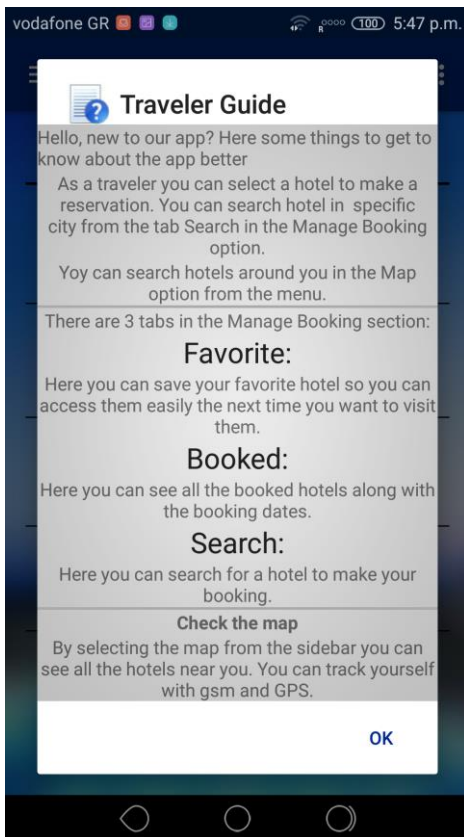
Εικόνα 6.35. Μορφή χάρτη ως Satellite



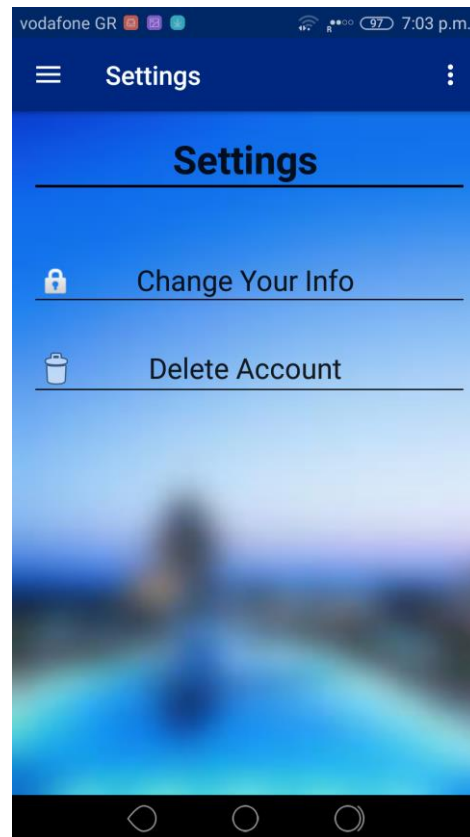
Εικόνα 6.36. Μορφή χάρτη ως Terrain

Οι επόμενες επιλογές από το μενού (Navigation Drawer) του ταξιδιώτη είναι οι ίδιες με αυτές του ιδιοκτήτη (Εικόνα 32), Info, Settings και Logout.

Η επιλογή Info είναι η ίδια όπως στην Εικόνα 6.19, έχοντας τις ίδιες επιλογές όπως στην περίπτωση του ιδιοκτήτη αλλά με μόνη τη διαφορά πως εδώ αντί για Owner Guide έχει Traveler Guide, το οποίο έχει και διαφορετική επεξήγηση. (Εικόνα 6.47)



Εικόνα 6.37. Alert Dialog Traveler Guide



Εικόνα 6.38. Οθόνη «Setting» (Traveler)

Στην Εικόνα 6.48 φαίνεται η οθόνη ρυθμίσεων του ταξιδιώτη. Οι επιλογές είναι ίδιες με αυτές του ιδιοκτήτη, εκτός από την τρίτη επιλογή (Edit Hotels), επειδή αυτή δεν χρειάζεται. Οι Change Your Info και Delete Account επιλογές είναι ακριβώς οι ίδιες με του ιδιοκτήτη.

Η τελευταία επιλογή του μενού Drawer είναι και εδώ Logout και έχει ακριβώς την ίδια λειτουργία με του ιδιοκτήτη.

6.3 Βάση δεδομένων και σύνδεση με τον Server

Όπως περιεγράφηκε στην ανάλυση της εφαρμογής, χρειάζεται να γίνει δημιουργία λογαριασμού για να μπορεί κανείς να χρησιμοποιήσει τις δυνατότητες

της. Ο λογαριασμός του κάθε χρήστη ξεχωριστά, αποθηκεύεται σε μία βάση δεδομένων η οποία βρίσκεται σε κάποιον server online ώστε να μπορεί ο κάθε χρήστης να έχει πρόσβαση στα στοιχεία του χρησιμοποιώντας τον κωδικό και το email για να συνδεθεί.

Για την σωστή λειτουργία της εφαρμογής, για την διαχείριση των στοιχείων του κάθε χρήστη ήταν απαραίτητη η χρήση μίας «τοπικής» βάσης δεδομένων, δηλαδή μια βάση η οποία θα υπάρχει μέσα στην συσκευή που θα χρησιμοποιεί ο χρήστης ώστε να συλλέγει τα δεδομένα του λογαριασμού του και να μπορεί να τα χρησιμοποιεί σε διάφορες λειτουργίες της. Αυτό γίνεται χρησιμοποιώντας μία μικρή βάση SQLite σε τοπικό επίπεδο, και η λειτουργία της είναι να εγγράφει κάθε φορά που συνδέεται κάποιος στον λογαριασμό του, τα στοιχεία του, από την βάση δεδομένων του server στην τοπική βάση. Αυτό βοηθάει ώστε να μην υπάρχουν άπειρες αιτήσεις στον server κάθε φορά που η εφαρμογή ήθελε να κάνει χρήση κάποιο από τα στοιχεία του χρήστη, αλλά να κάνει μία απλή αναφορά στην τοπική βάση όπου θα περιέχει τα απαραίτητα δεδομένα. Κάθε φορά που ο χρήστης αποσυνδέεται η τοπική βάση (SQLite) διαγράφεται (DROP) με σκοπό να εγγραφεί ξανά στην επόμενη σύνδεση με τα νέα στοιχεία. Για να επιτευχθεί αυτό, πρέπει να υπάρχει μία σύνδεση της εφαρμογής με την απομακρυσμένη βάση δεδομένων. Αυτό γίνεται χρησιμοποιώντας την βιβλιοθήκη volley. Η Volley είναι μια βιβλιοθήκη που κάνει τη δικτύωση για εφαρμογές Android ευκολότερη και το πιο σημαντικό, πιο γρήγορα! Διαχειρίζεται την επεξεργασία και την προσωρινή αποθήκευση των αιτημάτων δικτύου και εξοικονομεί πολύτιμο χρόνο από τους προγραμματιστές να γράψουν ξανά και ξανά τον ίδιο κωδικό κλήσης της cache μνήμης δικτύου.

Πλεονεκτήματα της χρήσης Volley:

- Προγραμματίζει αυτόματα όλα τα αιτήματα δικτύου. Αυτό σημαίνει ότι η Volley θα φροντίζει για όλα τα αιτήματα δικτύου που εκτελεί η εφαρμογή σας για την απόκτηση απόκρισης από τον ιστό.

- Παρέχει ισχυρό API αιτήματος ακύρωσης. Αυτό σημαίνει ότι μπορεί να ακυρώσει κανείς ένα μόνο αίτημα ή μπορεί να ορίσει μπλοκ ή πεδία των αιτημάτων για ακύρωση.
- Παρέχει ισχυρές δυνατότητες προσαρμογής.
- Παρέχει εργαλεία εντοπισμού σφαλμάτων και ανίχνευσης

Υπάρχουν δύο κύριες κατηγορίες Volley:

1. Request queue
2. Request

Request queue: Είναι η ηθελημένη αποστολή αιτημάτων (requests) στο δίκτυο, όταν αυτό είναι επιθυμητό. Συνήθως η Request queue δημιουργείται νωρίς, κατά την εκκίνηση, την διατηρούμε σε ετοιμότητα και την χρησιμοποιούμε ως Singleton.

Request: Περιέχει όλες τις απαραίτητες λεπτομέρειες για την πραγματοποίηση κλήσεων Web API. Για παράδειγμα: ποια μέθοδος χρήσης (GET ή POST), αίτημα δεδομένων για πέρασμα, ακροατής απόκρισης, ακροατής σφάλματος.

6.3 Εργαλεία που χρησιμοποιήθηκαν

Για την περάτωση της εφαρμογής Android, χρησιμοποιήθηκαν διάφορα προγράμματα και frameworks. Ο προγραμματισμός έγινε σε λειτουργικό σύστημα Windows 10. Αρχικά η επεξεργασία του κώδικα Android έγινε σε περιβάλλον Eclipse έκδοση Luna δημιουργώντας τον αρχικό σχεδιασμό κάποιον layouts και την αρχική ιδέα για την βασική εναλλαγή των πρώτων activities. Στην συνέχεια και για περισσότερες δυνατότητες αλλά και επειδή η δημιουργία Android εφαρμογών από το Eclipse πλέον δεν υποστηριζόταν, ο προγραμματισμός συνεχίστηκε σε περιβάλλον Android Studio, έκδοση 2.3.3.

Βασική προγραμματιστική γλώσσα για εφαρμογές Android και αυτή η οποία χρησιμοποιεί το περιβάλλον Android Studio είναι η Java, αλλά πλέον η δυνατότητα δημιουργίας τέτοιων εφαρμογών είναι εφικτή και από άλλες γλώσσες προγραμματισμού και frameworks, όπως για παράδειγμα είναι το περιβάλλον Microsoft Visual Studio με την χρήση του framework Xamarin, χρησιμοποιώντας γλώσσα C#. Σε κάθε περίπτωση η λογική είναι ίδια, μία αντικειμενοστραφής γλώσσα προγραμματισμού για back end και η χρήση XML για την δημιουργία της εμφάνισης δηλαδή το front end.

Για την δημιουργία της βάσης δεδομένων χρησιμοποιήθηκε ένας διαδικτυακός free host όπου φιλοξενεί την βάση καθώς και τα script που χρησιμοποιεί η εφαρμογή. Η δημιουργία της βάσης δεδομένων έγινε χρησιμοποιώντας εντολές SQL ενώ τα script για την επικοινωνία της εφαρμογής με την βάση δεδομένων αλλά και για τα απαραίτητα request έγιναν χρησιμοποιώντας γλώσσα PHP. Χρησιμοποιήθηκε διαδικτυακό περιβάλλον phpMyAdmin για την επεξεργασία της βάσης δεδομένων και ένας διαδικτυακός επεξεργαστής για τα PHP scripts.

Επίλογος

Στο κεφάλαιο αυτό αναλύθηκαν τα χαρακτηριστικά της παρούσας εφαρμογής που αφορά τον ιδιοκτήτη ξενοδοχείου/ων αλλά και ως προς τον ταξιδιώτη. Έγινε ανάλυση βασικών λειτουργιών της και εξηγήθηκε πώς είναι δυνατόν να αναπτυχθεί η εφαρμογή αυτή σε πιο λειτουργικό επίπεδο. Στο επόμενο κεφάλαιο αναφέρονται οι ατέλειες της εφαρμογής αλλά παράλληλα τονίζονται οι μελλοντικές βελτιώσεις που μπορούν να γίνουν σε αυτήν.

Κεφάλαιο 7 Συμπεράσματα

Πρόλογος

Το κεφάλαιο αυτό αφενός παρουσιάζει τα διάφορα προβλήματα που παρουσιάστηκαν κατά τον σχεδιασμό και την ανάπτυξη αυτής της εφαρμογής, και αφετέρου τις παρατηρήσεις και τις επισημάνσεις για τη μελλοντική βελτίωσή της

7.1 Εφαρμογή και περιβάλλον Android

Η εγγραφή του κώδικα από τον προγραμματιστή μπορεί να ξεκινήσει σε σύντομο χρονικό διάστημα καθώς η google (η εταιρεία που είναι υπεύθυνη για το android) φρόντισε μέσω του Android Studio και του documentation της να βοηθήσει τον προγραμματιστή ώστε να έχει όποιες πληροφορίες χρειάζεται.

Στα αρνητικά, το documentation που παρέχεται δίνει μία βασική εικόνα χωρίς να καθοδηγεί τον προγραμματιστή σε σύνθετα θέματα αναγκάζοντας τον να απευθυνθεί σε διάφορα forum και ιστοσελίδες με ερωταπαντήσεις πάνω σε συγκεκριμένα θέματα (π.χ. stackoverflow, thenewboston). Επιπλέον, οι χάρτες της Google (Google maps) είναι μέρος του google play services δυσχεραίνοντας έτσι την εισαγωγή τους στην εφαρμογή.

Ακόμα, ένα εμπόδιο που αντιμετωπίστηκε ήταν, όταν σταμάτησε η υποστήριξη του eclipse, με αποτέλεσμα την μεταπήδηση στο Android Studio δημιουργώντας προβλήματα λόγω ασυμβατότητας των δύο πλατφορμών ανάπτυξης κώδικα. Βασική τους διαφορά είναι πως το Android Studio κάνει χρήση του αρχείου Gradle σε αντίθεση με το eclipse.

Σε αντίθεση με τις πιο πολλές εμπορικές εφαρμογές η εφαρμογή αυτή δίνει τη δυνατότητα στον χρήστη να μπορεί να δημιουργήσει λογαριασμό και σαν ιδιοκτήτης ξενοδοχείων αλλά και σαν ταξιδιώτης από την πρώτη σελίδα της εφαρμογής, δηλαδή η εφαρμογή έχει δύο διαφορετικές λειτουργίες οι οποίες μπορούν να λειτουργήσουν ανεξάρτητα η μία από την άλλη. Μπορεί να γίνει χρήση της από έναν χρήστη χρησιμοποιώντας μόνο την μία επιλογή, παραδείγματος χάριν σαν ιδιοκτήτης ξενοδοχείων για να παρακολουθεί και να διαχειρίζεται τις κρατήσεις στα ξενοδοχεία του. Γι' αυτό και απαιτούνται διαφορετικές εγγραφές για κάθε ρόλο.

7.2 Προτάσεις για μελλοντική έρευνα/Μελλοντική επέκταση

Η εφαρμογή είναι πλήρως λειτουργική και έτοιμη προς χρήση, με περιορισμένες δυνατότητες στα πλαίσια μίας πτυχιακής εργασίας με σκοπό να παρουσιάσει την γνώση και την δυνατότητα δημιουργίας ενός πληροφοριακού συστήματος. Κάποιες από τους πιθανές μελλοντικές επεκτάσεις παρουσιάζονται παρακάτω.

- Βασικό στοιχείο είναι πως η εφαρμογή δεν πληροί την δήλωση δωματίων σε κάθε ξενοδοχείο, την εμφάνιση πληρότητας και την χρήση τους. Υπάρχει εικονικά χωρίς να έχει υλοποιηθεί λειτουργικά.
- Ένα άλλο στοιχείο είναι η συμβατότητα της εφαρμογής με άλλες εκδόσεις. Οι παραλλαγές που θα χρειαζόταν ο κώδικας ήταν αρκετές με αποτέλεσμα η ανάπτυξη να γίνει στοχεύοντας κυρίως στην έκδοση 4.4 καθώς αποτελεί την πλέον διαδεδομένη και ήταν η βάση του backwards compatibility.
- Πολύ βασικές επεκτάσεις είναι το φιλτράρισμα στα αποτελέσματα της αναζήτησης, ή, η ταξινόμησή τους ανάλογα με κάποια επιλογή (π.χ. τιμή, αριθμό αστεριών, βαθμολογία κτλ.).
- Μία λειτουργία που θα μπορούσε να προστεθεί είναι η σύγκριση των ξενοδοχείων σε ένα παράθυρο της εφαρμογής από τον χρήστη ώστε να μπορεί να επιλέξει σωστότερα.
- Ακόμα μία λειτουργία που δεν κατέστη δυνατόν να υλοποιηθεί είναι η άμεση πληρωμή μέσω e-banking στο περιβάλλον της εφαρμογής. Οι περιορισμοί που βάζουν οι τράπεζες για να δώσουν πρόσβαση σε τέτοιου είδους συναλλαγές, καθώς και τα νομικά θέματα, ήταν αρκετό ώστε να τεθεί υπό υλοποίηση σε περίπτωση που το πρόγραμμα βγει για εμπορική χρήση.
- Θα μπορούσε η είσοδος και η εγγραφή του χρήστη να γίνεται με τα social media ένεκα που αυτά αποτελούν σημαντικό κομμάτι της καθημερινότητας.

- Δεν είναι δυνατόν από τον ταξιδιώτη να κάνει ακύρωση της κράτησης καθώς και να αφαιρέσει ένα ξενοδοχείο από τα αγαπημένα μέσα από την εφαρμογή, αλλά μόνο αν γίνει διαγραφή από την βάση δεδομένων.

Επίλογος

Παρουσιάστηκαν οι δυσκολίες και οι περιορισμοί που υπήρξαν κατά την διάρκεια υλοποίησης της συγκεκριμένης εφαρμογής. Επίσης επισημάνθηκαν τα κενά που παρουσιάζει η εφαρμογή, η οποία παρότι είναι λειτουργική, θα μπορούσε να ικανοποιεί κάποιες ακόμα λειτουργίες οι οποίες θα την έκαναν ακόμη καλύτερη.

Βιβλιογραφία

Ελληνική βιβλιογραφία

- [1] Larry Ullman (2009), Εισαγωγή στις PHP 6 & MySQL 6 ΜΕ ΕΙΚΟΝΕΣ. Εκδόσεις Κλειδάριθμος.
- [2] Κοκκινίδης Χρήστος , «Σχεδίαση και ανάπτυξη διαδικτυακής εφαρμογής με τη χρήση της PHP.» , Πτυχιακή Εργασία , Ιούνιος 2004.
- [3] Σαλαμπάσης, Μ. (2008), Εισαγωγή στον προγραμματισμό διαδικτυακών εφαρμογών. Εκδόσεις ΙΔΙΩΤΙΚΗ.

Ξένη βιβλιογραφία

- [4] Raghu Ramakrishnan, Johannes Gehrke (2003), Database Management Systems edition 3. Εκδόσεις McGraw-Hill.
- [5] Bill Phillips and Brian Hardy (2013), Android Programming: The Big Nerd Ranch Guide, Big Nerd Ranch, Inc. Atlanta.
- [6] Keramopoulos Euclid., Dervos Dimitris A., Deligiannis Ignatios, Evangelidis George, Pouyioutas Phillipos. Critical Analysis of Designing a Graphical Query Language. Conference: 4th annual International Conference on Education and New Learning Technologies, At Barcelona, Spain. July 2012.

Διαδικτυακές πηγές

- [7] <http://developer.android.com/index.html> Android Developers. android.com
- [8] http://www.pcmag.com/encyclopedia_term/0,2542,t=Smartphone&i=51537,00.asp#fbid=PsmZ Definitions of Smartphone. Pc Magazine Encyclopedia
- [9] https://en.wikipedia.org/wiki/Mobile_operating_system Mobile Operating System - Wikipedia
- [10] <https://www.w3schools.com/php/default.asp> PHP Tutorials
- [11] <https://stackoverflow.com/> Stack Over Flow forum

Παράρτημα Α΄: Κώδικας Android

```
package com.example.hotelreservation.DataBase;

public class AppConfig {
    // Server user load and login urls
    public static String URL_LOGIN_OWNER = "http://ptyxiaki2016.eu.pn/loginowner.php";
    public static String URL_LOGIN_TRAVELER =
"http://ptyxiaki2016.eu.pn/logintraveler.php";
    public static String URL_LOAD_HOTELS = "http://ptyxiaki2016.eu.pn/loadhotels.php";
    public static String URL_LOAD_COORDINATES =
"http://ptyxiaki2016.eu.pn/loadcoordinates.php";
    public static String URL_LOAD_RESULT_HOTELS =
"http://ptyxiaki2016.eu.pn/resulthotels.php";
    public static String URL_LOAD_BOOKING_COORDS =
"http://ptyxiaki2016.eu.pn/bookingcoords.php";
    public static String URL_MAKE_BOOKING = "http://ptyxiaki2016.eu.pn/makebooking.php";
    public static String URL_LOAD_FAVORITES =
"http://ptyxiaki2016.eu.pn/loadfavorites.php";
    public static String URL_LOAD_OWNER_RESERVATIONS =
"http://ptyxiaki2016.eu.pn/loadreservations.php";
    public static String URL_LOAD_TRAVELER_MYBOOKINGS =
"http://ptyxiaki2016.eu.pn/loadmybookings.php";
    public static String URL_LOAD_TRAVELER =
"http://ptyxiaki2016.eu.pn/gettravelerbyid.php";

    // Server user register urls
    public static String URL_REGISTER_OWNER =
"http://ptyxiaki2016.eu.pn/registerowner.php";
    public static String URL_REGISTER_TRAVELER =
"http://ptyxiaki2016.eu.pn/registertraveler.php";
    public static String URL_REGISTER_HOTEL =
"http://ptyxiaki2016.eu.pn/registerhotel.php";
    public static String URL_REGISTER_COORDINATES =
"http://ptyxiaki2016.eu.pn/registercoordinates.php";
    public static String URL_REGISTER_FAVORITES =
"http://ptyxiaki2016.eu.pn/registerfavorites.php";

    // Server delete urls
    public static String URL_DELETE_OWNER = "http://ptyxiaki2016.eu.pn/deleteowner.php";
    public static String URL_DELETE_TRAVELER =
"http://ptyxiaki2016.eu.pn/deletetraveler.php";
    public static String URL_DELETE_HOTEL = "http://ptyxiaki2016.eu.pn/deletehotel.php";

    // Server update urls
    public static String URL_UPDATE_OWNER = "http://ptyxiaki2016.eu.pn/updateowner.php";
    public static String URL_UPDATE_TRAVELER =
"http://ptyxiaki2016.eu.pn/updatetraveler.php";
    public static String URL_UPDATE_HOTEL = "http://ptyxiaki2016.eu.pn/updatehotel.php";
}
```

Κώδικας 1. AppConfig.java


```
package com.example.hotelreservation.DataBase;

import android.app.Application;
import android.text.TextUtils;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.toolbox.Volley;

public class AppController extends Application {

    public static final String TAG = AppController.class.getSimpleName();

    private RequestQueue mRequestQueue;

    private static AppController mInstance;

    @Override
    public void onCreate() {
        super.onCreate();
        mInstance = this;
    }

    public static synchronized AppController getInstance() {
        return mInstance;
    }

    public RequestQueue getRequestQueue() {
        if (mRequestQueue == null) {
            mRequestQueue = Volley.newRequestQueue(getApplicationContext());
        }

        return mRequestQueue;
    }

    public <T> void addToRequestQueue(Request<T> req, String tag) {
        req.setTag(TextUtils.isEmpty(tag) ? TAG : tag);
        getRequestQueue().add(req);
    }

    public <T> void addToRequestQueue(Request<T> req) {
        req.setTag(TAG);
        getRequestQueue().add(req);
    }

    public void cancelPendingRequests(Object tag) {
        if (mRequestQueue != null) {
            mRequestQueue.cancelAll(tag);
        }
    }
}
```

Κώδικας 2. AppController.java

```
package com.example.hotelreseration.DataBase;

import android.content.Context;
import android.content.SharedPreferences;
import android.content.SharedPreferences.Editor;
import android.util.Log;

public class SessionManager {
    // LogCat tag
    private static String TAG = SessionManager.class.getSimpleName();

    // Shared Preferences
    SharedPreferences pref;

    Editor editor;
    Context _context;

    // Shared pref mode
    int PRIVATE_MODE = 0;

    // Shared preferences file name
    private static final String PREF_NAME = "GetARoom";

    private static final String KEY_IS_LOGGEDIN = "isLoggedIn";

    public SessionManager(Context context) {
        this._context = context;
        pref = _context.getSharedPreferences(PREF_NAME, PRIVATE_MODE);
        editor = pref.edit();
    }

    public void setLogin(boolean isLoggedIn) {

        editor.putBoolean(KEY_IS_LOGGEDIN, isLoggedIn);

        // commit changes
        editor.commit();

        Log.d(TAG, "User login session modified!");
    }

    public boolean isLoggedIn(){
        return pref.getBoolean(KEY_IS_LOGGEDIN, false);
    }
}
```

Κώδικας 3. SessionManager.java

```

package com.example.hotelreservation.DataBase;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

import java.util.HashMap;

public class SQLiteHandler extends SQLiteOpenHelper {

    private static final String TAG = SQLiteHandler.class.getSimpleName();

    // All Static variables
    // Database Version
    private static final int DATABASE_VERSION = 1;

    // Database Name
    private static final String DATABASE_NAME = "user_db";

    // Login table name
    private static final String TABLE_USER = "user";

    // Login Table Columns names
    private static final String KEY_ID = "ida";
    private static final String KEY_UID = "id";
    private static final String KEY_NAME = "name";
    private static final String KEY_SURNAME = "surname";
    private static final String KEY_COUNTRY = "country";
    private static final String KEY_EMAIL = "email";
    private static final String KEY_PASSWORD = "password";
    private static final String KEY_TELEPHONE = "telephone";

    public SQLiteHandler(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    // Creating Tables
    @Override
    public void onCreate(SQLiteDatabase db) {
        String CREATE_LOGIN_TABLE = "CREATE TABLE " + TABLE_USER + "(" + KEY_ID + " "
INTEGER PRIMARY KEY," + KEY_UID + " TEXT,"
        + KEY_NAME + " TEXT," + KEY_SURNAME + " TEXT," + KEY_COUNTRY + "
TEXT,"
        + KEY_EMAIL + " TEXT," + KEY_PASSWORD + " TEXT,"
        + KEY_TELEPHONE + " TEXT" + ")";
        db.execSQL(CREATE_LOGIN_TABLE);

        Log.d(TAG, "Database tables created");
    }

    //Delete the DB
    // public void dropDB(SQLiteDatabase db){
    //     db.execSQL("DROP TABLE IF EXISTS " + TABLE_USER);
    // }

    // Upgrading database
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // Drop older table if existed
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_USER);

        // Create tables again
        onCreate(db);
    }
}

```

Κώδικας 4. SQLiteHandler.java

```

package com.example.hotelreseration.NavigationDrawer;

import java.util.ArrayList;
import java.util.HashMap;

import com.example.hotelreseration.R;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.SimpleAdapter;
import android.widget.TextView;

import static com.example.hotelreseration.NavigationDrawer.MainActivity.dboFKey;

public class Fragment_Booked extends Fragment {
    static ListView myBookingslistView;
    static SimpleAdapter adapter;
    static ArrayList<HashMap<String,String>> records = new ArrayList<>(); //Dhmiourgw
    HashMap gia na mporw na valw polla String na fainontai se kathe stoixeio ths listas

    static TextView nobookings;

    @Override
    public View onCreateView(
        LayoutInflater inflater,
        ViewGroup container,
        Bundle savedInstanceState){
        View view = inflater.inflate(R.layout.fragment_traveler_booked, container,false);

        nobookings=(TextView) view.findViewById(R.id.nobookingtxt);

        myBookingslistView = (ListView) view.findViewById(R.id.myBookingsList);
        adapter = new SimpleAdapter(getActivity(), records, R.layout.mytextView, new
String[] {"hotelname","dates"}, new int[] {R.id.tv,R.id.sub});
        myBookingslistView.setAdapter(adapter);// Assign adapter to ListView

        ((MainActivity) getActivity()).loadMyBookings(dboFKey);

        if(myBookingslistView.getCount()==0){
            nobookings.setVisibility(View.VISIBLE);
        }else{
            nobookings.setVisibility(View.GONE);
        }

        return view;
    }
}

```

Κώδικας 5. Fragment_Booked.java


```
package com.example.hotelreseration.NavigationDrawer;

import java.text.DateFormat;
import java.util.Date;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentTransaction;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import com.example.hotelreseration.LoginActivity;
import com.example.hotelreseration.R;

public class Fragment_home extends Fragment {
    TextView date;
    TextView UserWelcomeLabel;

    public View onCreateView(
        LayoutInflater inflater,
        ViewGroup container,
        Bundle savedInstanceState){
        View rootView = inflater.inflate(R.layout.fragment_home, container, false);

        UserWelcomeLabel=(TextView) rootView.findViewById(R.id.user_Label);
        UserWelcomeLabel.setText("Welcome, "+MainActivity.dbname);
        if(LoginActivity.flagkzaf){
            UserWelcomeLabel.setText("Welcome, Kzaf");
        }

        String currentDateString = DateFormat.getDateInstance().format(new Date());

        date=(TextView) rootView.findViewById(R.id.DateTextView);
        date.setText(currentDateString);

        return rootView;
    }
}
```

Κώδικας 7. Fragment_home.java

```

package com.example.hotelreseration.NavigationDrawer;

import java.util.ArrayList;
import java.util.HashMap;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.Button;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.TextView;
import android.widget.Toast;
import com.example.hotelreseration.R;

public class Fragment_hotels extends Fragment {

    static ListView listView;
    SimpleAdapter adapter;
    ArrayList<HashMap<String,String>> records = new ArrayList<>(); //Dhmiourgw HashMap gia
na mporw na valw polla String na fainontai se kathe stoixeio ths listas

    static TextView txt;

    @Override
    public View onCreateView( LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState){

        View view = inflater.inflate(R.layout.fragment_hotels, container,false);
        listView = (ListView) view.findViewById(R.id.list);
        txt=(TextView) view.findViewById(R.id.nobookingtxt);
        MainActivity.flagpinhotel = false;

        adapter = new SimpleAdapter(getActivity(), records, R.layout.mytextview,
            new String[] {"name","city"}, new int[] {R.id.tv,R.id.sub});

        // Assign adapter to ListView
        listView.setAdapter(adapter);

        if(listView.getCount()==0){
            txt.setVisibility(View.VISIBLE); //show/hide the "add a new hotel" label
        }else{
            txt.setVisibility(View.GONE);
        }

        // BUTTON
        Button btnAddHotel = (Button) view.findViewById(R.id.addbutton);
        btnAddHotel.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                txt.setVisibility(View.GONE);
                switch (v.getId()) {
                    case R.id.addbutton:
                        if (listView.getCount()<=2){
                            ((MainActivity) getActivity()).alerthotel();
                        }
                        else{
                            Toast.makeText(getActivity(),"You can add up to 3 hotels! Please read the
info",Toast.LENGTH_LONG).show();
                        }

                        break;
                }
            }
        })
    }
}

```

Κώδικας 8. Fragment_hotels.java

```
package com.example.hotelreseration.NavigationDrawer;

import com.example.hotelreseration.R;
import com.example.hotelreseration.SelectUserActivity;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

public class Fragment_info extends Fragment {

    public View onCreateView(
        LayoutInflater inflater,
        ViewGroup container,
        Bundle savedInstanceState){
        View rootView = inflater.inflate(R.layout.fragment_info, container, false);

        final TextView OwnerGuide = (TextView)rootView.findViewById(R.id.guide);
        if (SelectUserActivity.flagOwner){
            OwnerGuide.setText("Owner Guide");
        }else{
            OwnerGuide.setText("Traveler Guide");
        }
        //OwnerGuide = (TextView)rootView.findViewById(R.id.guide);
        OwnerGuide.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                ((MainActivity) getActivity()).OwnerGuide();
            }
        });
        final TextView AboutUs = (TextView)rootView.findViewById(R.id.changeinfo);
        AboutUs.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                ((MainActivity) getActivity()).AboutUs();
            }
        });
        final TextView ContactUs = (TextView)rootView.findViewById(R.id.contactus);
        ContactUs.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                ((MainActivity) getActivity()).ContactUs();
            }
        });
        final TextView ReportProblem =
        (TextView)rootView.findViewById(R.id.reportaproblem);
        ReportProblem.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                ((MainActivity) getActivity()).ReportProblem();
            }
        });
        return rootView;
    }
}
```

Κώδικας 9. Fragment_info.java


```

package com.example.hotelreseration.NavigationDrawer;

import android.Manifest;
import android.content.Context;
import android.content.pm.PackageManager;
import android.location.Criteria;
import android.location.Location;
import android.location.LocationManager;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v4.app.ActivityCompat;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentTransaction;
import android.support.v4.content.ContextCompat;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Toast;

import com.example.hotelreseration.R;
import com.example.hotelreseration.SelectUserActivity;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.GoogleMap.OnMapClickListener;
import com.google.android.gms.maps.GoogleMap.OnMapLongClickListener;
import com.google.android.gms.maps.MapView;
import com.google.android.gms.maps.MapsInitializer;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.CameraPosition;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;

public class Fragment_maps extends android.support.v4.app.Fragment implements
OnMapLongClickListener, OnMapClickListener, OnMapReadyCallback {

    MapView mMapView;
    private static GoogleMap googleMap;

    public String latlong;
    String latitude;
    String longitude;

    @Override
    public void onMapReady(GoogleMap googleMap) {

        this.googleMap = googleMap;

        // Detect location
        if(checkPermission())
            googleMap.setMyLocationEnabled(true);
        else askPermission();

        googleMap.getUiSettings().setMyLocationButtonEnabled(true);

        // Turns traffic layer on
        googleMap.setTrafficEnabled(true);

        // Enables indoor maps
        googleMap.setIndoorEnabled(true);
    }
}

```

Κώδικας 10. Fragment_maps.java

```

package com.example.hotelreseration.NavigationDrawer;

import com.example.hotelreseration.R;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.TextView;

import java.util.ArrayList;
import java.util.HashMap;

import static com.example.hotelreseration.NavigationDrawer.MainActivity.dboFKey;

public class Fragment_Past extends Fragment {
    static ArrayList<HashMap<String,String>> records = new ArrayList<>();
    static ListView PastlistView;
    static SimpleAdapter adapter;
    static TextView txt;

    @Override
    public View onCreateView(
        LayoutInflater inflater,
        ViewGroup container,
        Bundle savedInstanceState){
        View view = inflater.inflate(R.layout.fragment_owner_past, container,false);

        PastlistView = (ListView) view.findViewById(R.id.pastResList);
        txt=(TextView) view.findViewById(R.id.nopasttxt);

        adapter = new SimpleAdapter(getActivity(), records, R.layout.mytextview, new
String[] {"HotelandTravelertitle", "dates"},
        new int[] {R.id.tv,R.id.sub});

        PastlistView.setAdapter(adapter);// Assign adapter to ListView

        ((MainActivity) getActivity()).loadOwnerReservations(dboFKey);
        Fragment_Past.adapter.notifyDataSetChanged();

        if(PastlistView.getCount()==0){
            txt.setVisibility(View.VISIBLE);
        }else{
            txt.setVisibility(View.GONE);
        }

        return view;
    }
}

```

Κώδικας 11. Fragment_Past.java

```
package com.example.hotelrereservation.NavigationDrawer;

import com.example.hotelrereservation.R;
import com.example.hotelrereservation.SelectUserActivity;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentTabHost;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class Fragment_reservations extends Fragment {

    private FragmentTabHost mTabHost;

    //Mandatory Constructor
    public Fragment_reservations() {
    }

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {

        View rootView = inflater.inflate(R.layout.fragment_reservations,container, false);

        mTabHost = (FragmentTabHost)rootView.findViewById(android.R.id.tabhost);
        mTabHost.setup(getActivity(), getChildFragmentManager(), R.id.realtabcontent);

        if(SelectUserActivity.flagOwner){
            mTabHost.addTab(mTabHost.newTabSpec("fragmentb").setIndicator("Running"),
                Fragment_Running.class, null);
            mTabHost.addTab(mTabHost.newTabSpec("fragmentd").setIndicator("Upcoming"),
                Fragment_Upcoming.class, null);
            mTabHost.addTab(mTabHost.newTabSpec("fragmentc").setIndicator("Past"),
                Fragment_Past.class, null);
        }else{
            mTabHost.addTab(mTabHost.newTabSpec("fragmentb").setIndicator("Search"),
                Fragment_Search.class, null);
            mTabHost.addTab(mTabHost.newTabSpec("fragmentc").setIndicator("Booked"),
                Fragment_Booked.class, null);
            mTabHost.addTab(mTabHost.newTabSpec("fragmentd").setIndicator("Favorite"),
                Fragment_Favorite.class, null);
        }

        return rootView;
    }
}
```

Κώδικας 12. Fragment_reservations.java

```

package com.example.hotelreseration.NavigationDrawer;

//import android.R;
import java.util.ArrayList;
import java.util.HashMap;

import com.example.hotelreseration.R;

import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.TextView;

import static com.example.hotelreseration.NavigationDrawer.MainActivity.dboFKey;

public class Fragment_Running extends Fragment {

    static ArrayList<HashMap<String,String>> records = new ArrayList<>();
    static ListView RunninglistView;
    static SimpleAdapter adapter;
    static TextView txt;

    @Override
    public View onCreateView(
        LayoutInflater inflater,
        ViewGroup container,
        Bundle savedInstanceState){
        View view = inflater.inflate(R.layout.fragment_owner_running, container,false);

        RunninglistView = (ListView) view.findViewById(R.id.runningResList);
        txt=(TextView) view.findViewById(R.id.nobookingtxt);

        adapter = new SimpleAdapter(getActivity(), records, R.layout.mytextview, new String[]
{"HotelandTravelertitle","dates"},
        new int[] {R.id.tv,R.id.sub});

        RunninglistView.setAdapter(adapter);// Assign adapter to ListView

        ((MainActivity) getActivity()).loadOwnerReservations(dboFKey);
//        Fragment_Running.adapter.notifyDataSetChanged();

        if(RunninglistView.getCount()==0){
            txt.setVisibility(View.VISIBLE);
        }else{
            txt.setVisibility(View.GONE);
        }

        return view;
    }
}

```

Κώδικας 13. Fragment_Running.java

```

package com.example.hotelreseration.NavigationDrawer;

import android.content.Context;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.inputmethod.InputMethodManager;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.TextView;
import android.widget.Toast;

import com.example.hotelreseration.R;

import java.util.ArrayList;
import java.util.HashMap;

public class Fragment_Search extends Fragment {

    static ListView listView;
    static SimpleAdapter adapter;
    static ArrayList<HashMap<String,String>> records = new ArrayList<>(); //Dhmiourgw
HashMap gia na mporw na valw polla String na fainontai se kathe stoixeio ths listas

//    public boolean flag=false;
    public boolean checkflag = false;

    @Override
    public View onCreateView(
        LayoutInflater inflater,
        ViewGroup container,
        Bundle savedInstanceState){
        final View rootView = inflater.inflate(R.layout.fragment_traveler_search,
container, false);

        listView = (ListView) rootView.findViewById(R.id.searchList);
        adapter = new SimpleAdapter(getActivity(), records,
R.layout.traveler_hotel_search_list,
        new String[] {"name","city"}, new int[] {R.id.tv,R.id.sub});
        // Assign adapter to ListView
        listView.setAdapter(adapter);
//
//        if (!flag){
//            final TextView Ascending = (TextView) rootView.findViewById(R.id.ascending);
//            final TextView Descending = (TextView)
rootView.findViewById(R.id.descending);
//
//            Ascending.setVisibility(View.GONE);
//            Descending.setVisibility(View.GONE);
//        }

        final Button Search = (Button)rootView.findViewById(R.id.search);
        Search.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {

                //the next two rows of code hides the keyboard on button click
                InputMethodManager inputManager = (InputMethodManager)
getActivity().getSystemService(Context.INPUT_METHOD_SERVICE);

inputManager.hideSoftInputFromWindow(getActivity().getCurrentFocus().getWindowToken(),

```

Κώδικας 14. Fragment_Search.java

```

package com.example.hotelreseration.NavigationDrawer;

import com.example.hotelreseration.R;
import com.example.hotelreseration.SelectUserActivity;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

public class Fragment_settings extends Fragment {
    @Override
    public View onCreateView(
        LayoutInflater inflater,
        ViewGroup container,
        Bundle savedInstanceState){
        View rootView = inflater.inflate(R.layout.fragment_settings, container, false);

        final TextView EditHotels = (TextView)rootView.findViewById(R.id.edithotels);
        final ImageView HotelImage=(ImageView)rootView.findViewById(R.id.hotelImage);
        View View = (View)rootView.findViewById(R.id.View03);
        if(!SelectUserActivity.flagOwner){
            EditHotels.setVisibility(rootView.INVISIBLE);
            HotelImage.setVisibility(rootView.INVISIBLE);
            View.setVisibility(rootView.INVISIBLE);
        }else{
            EditHotels.setOnClickListener(new View.OnClickListener() {
                public void onClick(View v) {
                    if(Fragment_hotels.listView.getCount()==0){
                        Toast.makeText(getActivity(),"No hotels added
yet!",Toast.LENGTH_SHORT).show();
                    }
                    else{
                        ((MainActivity) getActivity()).EditHotels();
                    }
                }
            });
        }

        final TextView DeleteAcc = (TextView)rootView.findViewById(R.id.deleteaccount);
        DeleteAcc.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                ((MainActivity) getActivity()).DeleteAccount();
            }
        });

        final TextView ChangeInfo = (TextView)rootView.findViewById(R.id.changeinfo);
        ChangeInfo.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                ((MainActivity) getActivity()).ChangeInfo();
            }
        });

        return rootView;
    }
}

```

Κώδικας 15. Fragment_settings.java

```

package com.example.hotelreseration.NavigationDrawer;

import java.util.ArrayList;
import java.util.HashMap;

import com.example.hotelreseration.R;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.SimpleAdapter;
import android.widget.TextView;

import static com.example.hotelreseration.NavigationDrawer.MainActivity.dboFKey;

public class Fragment_Upcoming extends Fragment {

    static ArrayList<HashMap<String,String>> records = new ArrayList<>();
    static ListView UpcomminglistView;
    static SimpleAdapter adapter;
    static TextView txt;

    @Override
    public View onCreateView(
        LayoutInflater inflater,
        ViewGroup container,
        Bundle savedInstanceState){
        View view = inflater.inflate(R.layout.fragment_owner_upcoming, container,false);

        UpcomminglistView = (ListView) view.findViewById(R.id.upcomingResList);
        txt=(TextView) view.findViewById(R.id.nobookingtxt);

        adapter = new SimpleAdapter(getActivity(), records, R.layout.mytextview, new
String[] { "HotelandTravelertitle", "dates"},
        new int[] { R.id.tv, R.id.sub});

        UpcomminglistView.setAdapter(adapter);// Assign adapter to ListView

        ((MainActivity) getActivity()).loadOwnerReservations(dboFKey);
        Fragment_Upcoming.adapter.notifyDataSetChanged(); //Notify the adapter for the
update

        if(UpcomminglistView.getCount()==0){
            txt.setVisibility(View.VISIBLE);
        }else{
            txt.setVisibility(View.GONE);
        }

        return view;
    }
}

```

Κώδικας 16. Fragment_Upcoming.java

```

package com.example.hotelreseration.NavigationDrawer;

import android.content.Intent;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.graphics.drawable.BitmapDrawable;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.EditText;
import android.widget.ImageView;
import com.example.hotelreseration.LoginActivity;
import com.example.hotelreseration.R;

public class Fragment_user extends Fragment {

    private Uri mSelectedImageUri;

    private static final int SELECT_PICTURE = 1;
    private String selectedImagePath=null;
    private ImageView img;

    @Override
    public View onCreateView(
        LayoutInflater inflater,
        ViewGroup container,
        Bundle savedInstanceState){
        View rootView = inflater.inflate(R.layout.fragment_user, container, false);

        EditText ChangeName=(EditText)rootView.findViewById(R.id.ChangeName);
        EditText ChangeSurname=(EditText)rootView.findViewById(R.id.ChangeSurname);
        EditText ChangeTelephone=(EditText)rootView.findViewById(R.id.ChangeTelephone);
        EditText ChangeMail=(EditText)rootView.findViewById(R.id.ChangeMail);
        EditText ChangeCountry=(EditText)rootView.findViewById(R.id.ChangeCountry);

        ChangeName.setText(new StringBuilder().append(MainActivity.dbname));
        ChangeSurname.setText(new StringBuilder().append(MainActivity.dbsurname));
        ChangeTelephone.setText(new StringBuilder().append(MainActivity.dbtelephone));
        ChangeMail.setText(new StringBuilder().append(MainActivity.dbmail));
        ChangeCountry.setText(new StringBuilder().append(MainActivity.dbcountry));

        if(LoginActivity.flagkzaf){
            ChangeName.setText(new StringBuilder().append("Konstantinos"));
            ChangeSurname.setText(new StringBuilder().append("Zafeiropoulos"));
            ChangeTelephone.setText(new StringBuilder().append("+30 6984794915"));
            ChangeMail.setText(new StringBuilder().append("kzaf@it.teithe.gr"));
            ChangeCountry.setText(new StringBuilder().append("Greece"));
        }

        //Apo edw kai katw einai gia tin eikona pou tha epilegei o xrhsths na emfanizetai
        img = (ImageView)rootView.findViewById(R.id.userImage);
        if (mSelectedImageUri != null) {
            img.setImageURI(mSelectedImageUri);
        }

        img.setOnClickListener(new OnClickListener() {
            public void onClick(View arg0) {
                Intent intent = new Intent();
                intent.setType("image/*");
                intent.setAction(Intent.ACTION_GET_CONTENT);
                startActivityForResult(Intent.createChooser(intent,"Select

```

Κώδικας 17. Fragment_user.java


```

package com.example.hotelreseration.NavigationDrawer;

import android.annotation.SuppressLint;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.res.Configuration;
import android.net.Uri;
import android.os.Bundle;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarActivity;
import android.support.v7.app.ActionBarDrawerToggle;

import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;

import android.util.Log;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.RadioGroup;
import android.widget.RatingBar;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.example.hotelreseration.DataBase.AppConfig;
import com.example.hotelreseration.DataBase.AppController;
import com.example.hotelreseration.DataBase.SQLiteHandler;
import com.example.hotelreseration.LoginActivity;
import com.example.hotelreseration.R;
import com.example.hotelreseration.SelectUserActivity;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.MapView;
import com.google.android.gms.maps.MapsInitializer;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.Map;

import static android.view.View.GONE;
import static com.example.hotelreseration.DataBase.AppConfig.URL_DELETE_HOTEL;
import static com.example.hotelreseration.DataBase.AppConfig.URL_LOAD_BOOKING_COORDS;
import static com.example.hotelreseration.DataBase.AppConfig.URL_LOAD_COORDINATES;
import static com.example.hotelreseration.DataBase.AppConfig.URL_LOAD_FAVORITES;

```

Κώδικας 18. MainActivity.java

```

package com.example.hotelreseration.NavigationDrawer;

import com.example.hotelreseration.R;
import android.content.Context;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import android.widget.TextView;

public class MenuListAdapter extends BaseAdapter{

    // Fields -----
    private Context mContext;
    private String[] titles;
    private int[] icons;
    private LayoutInflater inflater;

    // Constructor -----
    public MenuListAdapter(
        Context context,
        String[] titles,
        int[] icons){
        mContext = context;
        this.titles = titles;
        this.icons = icons;
        inflater =
(LayoutInflater)mcontext.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    }

    // Accessors -----
    @Override
    public int getCount(){
        return titles.length;
    }
    @Override
    public Object getItem(int position){
        return titles[position];
    }
    @Override
    public long getItemId(int position){
        return position;
    }

    // Methods -----
    public View getView(int position, View convertView, ViewGroup parent){

        ViewHolder viewHolder;

        // Only inflate the view if convertView is null
        if (convertView == null){
            viewHolder = new ViewHolder();
            if(inflater!=null)
            {
                convertView = inflater.inflate(R.layout.drawr_list_item, parent, false);
                viewHolder.txtTitle = (TextView)convertView.findViewById(R.id.title);
                viewHolder.imgIcon = (ImageView)convertView.findViewById(R.id.icon);

                // This is the first time this view has been inflated,
                // so store the view holder in its tag fields
                convertView.setTag(viewHolder);
            }
            else
            {

```

Κώδικας 19. MenuListAdapter.java

```

package com.example.hotelreseration;

//import android.R;
//import android.R;
import android.content.Intent;
import android.graphics.Typeface;
import android.net.Uri;
import android.os.Bundle;
import android.support.v7.app.ActionBarActivity;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.TextView;

public class ForgotPassActivity extends ActionBarActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_forgot_pass);

        //disable the ActionBar
        android.support.v7.app.ActionBar AB=getSupportActionBar();
        AB.hide();

        //me aytes tis entoles allazw thn grammatoseira enos Text View
        TextView fpt = (TextView) findViewById(R.id.forgotpasstext);
        TextView dw = (TextView) findViewById(R.id.dontworry);
        TextView luk = (TextView) findViewById(R.id.LetUsKnow);
        Typeface fptface = Typeface.createFromAsset(getAssets(),"fonts/KaushanScript-
Regular.ttf");
        fpt.setTypeface(fptface);
        Typeface dwface = Typeface.createFromAsset(getAssets(),"fonts/KaushanScript-
Regular.ttf");
        dw.setTypeface(fptface);
        Typeface lukface = Typeface.createFromAsset(getAssets(),"fonts/KaushanScript-
Regular.ttf");
        luk.setTypeface(fptface);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.forgot_pass, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();

        return id == R.id.action_settings || super.onOptionsItemSelected(item);
    }

    //Edw orizw ta mail sta opoia tha steilei email o xrhsths se periptwsh pou ksexasei to
password tou
    String subject="Forgot Password!";
    String message="I forgot my account password at HotelReservation app, please give me a
new one. My account name is: ";
    public void sentMail(View v){
        Intent intent = new Intent(Intent.ACTION_SENDTO,
Uri.fromParts("mailto","kzaf@it.teithe.gr,ziskatas@it.teithe.gr", null));
        intent.putExtra(Intent.EXTRA_SUBJECT, subject);
        intent.putExtra(Intent.EXTRA_TEXT, message);
        startActivity(Intent.createChooser(intent, "Choose an Email client :"));
    }
}

```

Κώδικας 20. CreateAccountActivity.java

```

package com.example.hotelreseration;

//import android.R;
//import android.R;
import android.content.Intent;
import android.graphics.Typeface;
import android.net.Uri;
import android.os.Bundle;
import android.support.v7.app.ActionBarActivity;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.TextView;

public class ForgotPassActivity extends ActionBarActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_forgot_pass);

        //disable the ActionBar
        android.support.v7.app.ActionBar AB=getSupportActionBar();
        AB.hide();

        //me aytes tis entoles allazw thn grammatoseira enos Text View
        TextView fpt = (TextView) findViewById(R.id.forgotpasstext);
        TextView dw = (TextView) findViewById(R.id.dontworry);
        TextView luk = (TextView) findViewById(R.id.LetUsKnow);
        Typeface fptface = Typeface.createFromAsset(getAssets(),"fonts/KaushanScript-
Regular.ttf");
        fpt.setTypeface(fptface);
        Typeface dwface = Typeface.createFromAsset(getAssets(),"fonts/KaushanScript-
Regular.ttf");
        dw.setTypeface(fptface);
        Typeface lukface = Typeface.createFromAsset(getAssets(),"fonts/KaushanScript-
Regular.ttf");
        luk.setTypeface(fptface);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.forgot_pass, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();

        return id == R.id.action_settings || super.onOptionsItemSelected(item);
    }

    //Edw orizw ta mail sta opoia tha steilei email o xrhsths se periptwsh pou ksexasei to
password tou
    String subject="Forgot Password!";
    String message="I forgot my account password at HotelReservation app, please give me a
new one. My account name is: ";
    public void sentMail(View v){
        Intent intent = new Intent(Intent.ACTION_SENDTO,
Uri.fromParts("mailto","kzaf@it.teithe.gr,ziskatas@it.teithe.gr", null));
        intent.putExtra(Intent.EXTRA_SUBJECT, subject);
        intent.putExtra(Intent.EXTRA_TEXT, message);
        startActivity(Intent.createChooser(intent, "Choose an Email client :"));
    }
}

```

Κώδικας 21. ForgotPassActivity.java

```

package com.example.hotelreseration;

import android.app.ProgressDialog;
import android.content.Intent;
import android.graphics.Typeface;
import android.os.Bundle;
import android.support.v7.app.ActionBarActivity;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.example.hotelreseration.DataBase.AppConfig;
import com.example.hotelreseration.DataBase.AppController;
import com.example.hotelreseration.DataBase.SQLiteHandler;
import com.example.hotelreseration.DataBase.SessionManager;
import com.example.hotelreseration.NavigationDrawer.MainActivity;

import org.json.JSONException;
import org.json.JSONObject;

import java.lang.reflect.Method;
import java.util.HashMap;
import java.util.Map;

public class LoginActivity extends ActionBarActivity {

    public static String mail;
    public static String pass;
    public static boolean flagkzaf=false;
    private static final String TAG = CreateAccountActivity.class.getSimpleName();
    private ProgressDialog pDialog;
    private SessionManager session;
    private SQLiteHandler db;
    public String URL_LOGIN;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        //disable the ActionBar
        android.support.v7.app.ActionBar AB=getSupportActionBar();
        AB.hide();

        //Get Text Values
        final Button mButton= (Button)findViewById(R.id.SignInbutton);
        final EditText Email= (EditText)findViewById(R.id.EmaileditText);
        final EditText Pass= (EditText)findViewById(R.id.PasswordeditText);

        // Progress dialog
        pDialog = new ProgressDialog(this);
        pDialog.setCancelable(false);

        // SQLite database handler
        db = new SQLiteHandler(getApplicationContext());

        // Session manager
    }
}

```

Κώδικας 22. LoginActivity.java

```

package com.example.hotelreseration;

import android.content.Intent;
import android.graphics.Typeface;
import android.os.Bundle;
import android.support.v7.app.ActionBarActivity;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.example.hotelreseration.DataBase.SQLiteHandler;

public class SelectUserActivity extends ActionBarActivity {

    private SQLiteHandler db;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_select_user);
        //disable the ActionBar
        android.support.v7.app.ActionBar AB=getSupportActionBar();
        AB.hide();

        //mhdenismos tw n flag
        SelectUserActivity.flagOwner=false;
        LoginActivity.flagkzaf=false;

        // SQLite database handler
        db = new SQLiteHandler(getApplicationContext());
        db.deleteUsers();

        //Orizw to label sto loginactivity na leei Log in as Owner otan epilegw Owner
        //opote to startActivity event energopoiείται apo edw.
        //An den epileksw Owner kai epileksw Traveler, to flagOwner den ginetai true, opote
to startActivity
        //energopoiείται kanonika apo th methodo 'changeToLoginScreen' pou vrisketai parakatw
        final Button ownerButton= (Button)findViewById(R.id.Owner);
        ownerButton.setOnClickListener(
            new View.OnClickListener()
            {
                public void onClick(View arg0)
                {
                    flagOwner=true;
                    startActivity(new Intent(SelectUserActivity.this,
LoginActivity.class));
                }
            });
        //me aytes tis entoles allazw thn grammatoseira enos Text View
        TextView sr = (TextView) findViewById(R.id.SelectRole);
        Typeface srface = Typeface.createFromAsset(getAssets(),"fonts/black_jack.ttf");
        sr.setTypeface(srface);

        TextView welcome = (TextView) findViewById(R.id.Welcome);
        Typeface welcomeface = Typeface.createFromAsset(getAssets(),"fonts/KaushanScript-
Regular.ttf");
        welcome.setTypeface(welcomeface);

        //orizw to flag gia to listview me ta hotel na einai false
    }

    /**

```

Κώδικας 23. SelectUserActivity.java

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/LinearLayout2"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="top|center"
    android:orientation="vertical">

    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="82dp"
        android:layout_height="76dp"
        android:src="@drawable/getaroom" />

    <TextView
        android:id="@+id/textView8"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal|center|top"
        android:gravity="center"
        android:text="This is a diplomatic app made by Zafeiropoulos Konstantinos and
Ziskatas Athanasios. " />

    <TextView
        android:id="@+id/date"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:gravity="center"
        android:text="It is a graduation project in academic context. Not for sale and
use!" />

    <TextView
        android:id="@+id/TextView02"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="_____ "
        android:textAppearance="?android:attr/textAppearanceSmall" />

    <TextView
        android:id="@+id/deleteaccount"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="Programmer: Zafeiropoulos Konstantinos. AM(10/3571). 10th semester
at Information Technology at Alexander Technological Institue of Thessaloniki (ATEITH)." />

    <TextView
        android:id="@+id/textView9"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="_____ "
        android:textAppearance="?android:attr/textAppearanceSmall" />

    <TextView
        android:id="@+id/nobookingtxt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <TextView
        android:id="@+id/TextView03"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="Programmer: Ziskatas Athanasios. AM(10/3632). 10th semester at

```

Κώδικας 24. aboutus.xml

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/blurbackground"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.hotelreseration.CreateAccountActivity" >

    <LinearLayout
        android:id="@+id/linearLayout1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/LoginAs"
        android:orientation="vertical" >

        <EditText
            android:id="@+id/Country"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="Name"
            android:textColorHint="#c1bdbd"
            android:inputType="textCapSentences"
            android:textColor="#FFFFFF" >

            <requestFocus />
        </EditText>

        <EditText
            android:id="@+id/ChangeSurname"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:inputType="textCapSentences"
            android:hint="Surname"
            android:textColorHint="#c1bdbd"
            android:textColor="#FFFFFF" />

        <EditText
            android:id="@+id/ChangeMail"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:inputType="textEmailAddress"
            android:hint="E-mail"
            android:textColorHint="#c1bdbd"
            android:textColor="#FFFFFF" />

        <EditText
            android:id="@+id/InputPass"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:inputType="numberPassword"
            android:hint="Password"
            android:textColorHint="#c1bdbd"
            android:textColor="#FFFFFF" />

        <EditText
            android:id="@+id/ReEnterPass"
            android:layout_width="match_parent"

```

Κώδικας 25. activity_create_account.xml


```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/blurbackground"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.hotelreseration.ForgotPassActivity"
    tools:ignore="TextFields" >

    <TextView
        android:id="@+id/forgotpasstext"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="20dp"
        android:text="@string/forgot_password_"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textColor="#FFFFFF"
        android:textSize="@dimen/abc_text_size_title_material_toolbar" />

    <TextView
        android:id="@+id/dontworry"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/forgotpasstext"
        android:layout_centerHorizontal="true"
        android:text="Dont Worry!"
        android:textAppearance="?android:attr/textAppearanceSmall"
        android:textColor="#FFFFFF" />

    <TextView
        android:id="@+id/LetUsKnow"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/dontworry"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="45dp"
        android:text="Let us know!"
        android:textAppearance="?android:attr/textAppearanceSmall"
        android:textColor="#FFFFFF" />

    <Button
        android:id="@+id/ok"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/LetUsKnow"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="72dp"
        android:onClick="sentMail"
        android:background="@drawable/custom_button"
        android:textColor="#fff"
        android:text="Sent an e-mail" />

</RelativeLayout>

```

Κώδικας 26. acticity_forgot_pass.xml

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/blurbackground"
    android:gravity="top"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.hotelreseration.LoginActivity" >

    <TextView
        android:id="@+id/NameTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@+id/EmailEditText"
        android:layout_alignParentLeft="true"
        android:layout_marginLeft="16dp"
        android:textColor="#FFFFFF"
        android:text="@string/email_" />

    <TextView
        android:id="@+id/PasswordtextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@+id/PasswordeditText"
        android:layout_alignLeft="@+id/NameTextView"
        android:textColor="#FFFFFF"
        android:text="@string/password_" />

    <Button
        android:id="@+id/SignInbutton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/custom_button"
        android:textColor="#fff"
        android:text="@string/sign_in"
        android:layout_below="@+id/PasswordtextView"
        android:layout_marginTop="63dp"
        android:layout_alignStart="@+id/CreateAccountbutton"
        android:layout_alignEnd="@+id/CreateAccountbutton"/>

    <Button
        android:id="@+id/CreateAccountbutton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/PasswordtextView"
        android:layout_alignRight="@+id/EmailEditText"
        android:layout_below="@+id/OrtextView"
        android:onClick="changeToCreateAccount"
        android:background="@drawable/custom_button"
        android:textColor="#fff"
        android:text="@string/create_account" />

    <TextView
        android:id="@+id/OrtextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/SignInbutton"
        android:layout_centerHorizontal="true"
        android:text="@string/or"
        android:textColor="#FFFFFF"
        android:textAppearance="?android:attr/textAppearanceSmall" />

    <EditText

```

Κώδικας 27. activity_login.xml

```
<android.support.v4.widget.DrawerLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:id="@+id/drawer_layout"
  android:layout_width="match_parent"
  android:layout_height="match_parent">

  <FrameLayout
    android:id="@+id/content_frame"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>
  <ListView
    android:id="@+id/listview_drawer"
    android:layout_width="250dp"
    android:layout_height="match_parent"
    android:layout_gravity="start"
    android:background="#c804051b"
    android:choiceMode="singleChoice"
    android:divider="@android:color/transparent"
    android:dividerHeight="1dp"/>

</android.support.v4.widget.DrawerLayout>

<!--Here are the settings for the navigation bar, like the color of the navigation bar,
width ect-->
```

Κώδικας 28. activity_main.xml

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/rolewall"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.hotelreseration.SelectUserActivity" >

    <!--android:fitsSystemWindows="true"--> <!--this is for the tranparent navigation bar-->

    <TextView
        android:id="@+id/Welcome"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="16dp"
        android:text="@string/welcome"
        android:textColor="#FFFFFF"
        android:textSize="@dimen/abc_action_bar_icon_vertical_padding_material" />

    <TextView
        android:id="@+id/SelectRole"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/Welcome"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="26dp"
        android:ellipsize="start"
        android:text="Select Role"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textColor="#FFFFFF"
        android:textSize="@dimen/abc_action_bar_progress_bar_size"
        android:textStyle="bold|italic" />

    <Button
        android:id="@+id/Owner"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/Traveler"
        android:layout_alignBottom="@+id/Traveler"
        android:layout_toLeftOf="@+id/Welcome"
        android:onClick="changeToLoginScreen"
        android:background="@drawable/custom_button"
        android:text="@string/owner"
        android:textColor="#fff" />

    <Button
        android:id="@+id/Traveler"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/SelectRole"
        android:layout_marginTop="72dp"
        android:layout_toRightOf="@+id/Welcome"
        android:background="@drawable/custom_button"
        android:onClick="changeToLoginScreen"
        android:text="Traveler"
        android:textColor="#fff" />

</RelativeLayout>

```

Κώδικας 29. activity_select_user.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/RelativeLayout1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="top|center"
    android:background="@drawable/simpleb"
    android:orientation="vertical" >

    <EditText
        android:id="@+id/editTextCountry"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/textView5"
        android:layout_alignBottom="@+id/textView5"
        android:layout_alignParentEnd="true"
        android:inputType="textCapSentences"
        android:ems="10"
        tools:ignore="RtlHardcoded,LabelFor" />

    <TextView
        android:id="@+id/todaysdate"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignRight="@+id/results"
        android:layout_marginTop="14dp"
        android:text="Name:"
        tools:ignore="HardcodedText,RtlHardcoded" />

    <EditText
        android:id="@+id/editTextName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:inputType="textCapSentences"
        android:ems="10"
        tools:ignore="RtlHardcoded,LabelFor" />

    <EditText
        android:id="@+id/editTextSurname"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/editTextName"
        android:inputType="textCapSentences"
        android:ems="10"
        tools:ignore="TextFields,RtlHardcoded,LabelFor" />

    <EditText
        android:id="@+id/editTextTelephone"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/editTextSurname"
        android:ems="10"
        tools:ignore="TextFields,RtlHardcoded,LabelFor" />

    <EditText
        android:id="@+id/editTextEmail"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/editTextTelephone"
        android:ems="10"

```

Κώδικας 30. changeinfo.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/LinearLayout2"
    android:background="@drawable/simpleb"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="top|center"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/nobookingtxt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="Zafeiropoulos Konstantinos"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <TextView
        android:id="@+id/edithoteltitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:autoLink="email"
        android:text="e-mail: kzaf@it.teithe.gr" />

    <TextView
        android:id="@+id/pickeddate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="_____ " />

    <TextView
        android:id="@+id/textView6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="Ziskatas Athanasios"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <TextView
        android:id="@+id/date"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:autoLink="email"
        android:text="e-mail: ziskatas@it.teithe.gr" />
</LinearLayout>
```

Κώδικας 31. contactus.xml

```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/scrollView1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:tools="http://schemas.android.com/tools">

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/simpleb"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:ignore="ScrollViewSize" >

    <DatePicker
        android:id="@+id/datePicker1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:headerYearTextAppearance="@array/pref_example_list_titles"
        tools:ignore="UnusedAttribute,RtlHardcoded" />

</RelativeLayout>

</ScrollView>
```

Κώδικας 32. datepicker.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/LinearLayout2"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="top|center"
    android:background="@drawable/simpleb"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/pinonmap"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="top|center"
        android:text="WARNING"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textColor="#9f0007"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/pickeddate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="If you press OK you decidedly delete your account!" />

    <TextView
        android:id="@+id/textView9"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="_____ "
        android:textAppearance="?android:attr/textAppearanceSmall" />

    <TextView
        android:id="@+id/nobookingtxt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <TextView
        android:id="@+id/textView8"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="Are you sure you want to delete your account?" />
</LinearLayout>
```

Κώδικας 33. deleteaccount.xml


```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/LinearLayout2"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="top|center"
    android:background="@drawable/simpleb"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/nobookingtxt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="WARNING"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textColor="#9f0007"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/pickeddate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="_____ " />

    <TextView
        android:id="@+id/edithoteltitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="You cant undo the delete, please be sure for your action!" />

    <TextView
        android:id="@+id/results"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
```

Κώδικας 34. deletehotel.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/LinearLayout1"
    style="?attr/dropDownListPreferredItemHeight"
    android:layout_width="match_parent"
    android:layout_height="fill_parent"
    android:background="@android:color/transparent"
    android:orientation="horizontal"
    android:weightSum="10" >

    <ImageView
        android:id="@+id/icon"
        android:layout_width="36dp"
        android:layout_height="fill_parent"
        android:layout_weight="0"
        android:adjustViewBounds="true"
        tools:ignore="ContentDescription" />

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="fill_parent"
        android:layout_weight="0"
        android:gravity="center_vertical|left"
        android:orientation="vertical"
        android:showDividers="beginning|middle|end"
        android:weightSum="10"
        tools:ignore="UnusedAttribute,RtlHardcoded" >

        <TextView
            android:id="@+id/textView"
            android:layout_width="wrap_content"
            android:layout_height="fill_parent"
            android:layout_weight="2"
            android:text=""
            tools:ignore="NestedWeights" />

        <TextView
            android:id="@+id/title"
            style="?attr/spinnerDropDownItemStyle"
            android:layout_width="wrap_content"
            android:layout_height="fill_parent"
            android:layout_gravity="center"
            android:layout_weight="0"
            android:ellipsize="end"
            android:singleLine="true"
            android:textColor="@android:color/darker_gray"
            android:textSize="16sp" />

        <TextView
            android:id="@+id/map"
            android:layout_width="wrap_content"
            android:layout_height="fill_parent"
            android:layout_weight="2"
            android:text="" />
    </LinearLayout>
</LinearLayout>

```

Κώδικας 35. drawr_list_item.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/RelativeLayout1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/simpleb"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/changeinfo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_gravity="center"
        android:text="Select hotel to edit"
        android:textAppearance="?android:attr/textAppearanceLarge"
        tools:ignore="HardcodedText" />

    <TextView
        android:id="@+id/pickeddate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/changeinfo"
        android:layout_centerHorizontal="true"
        android:layout_gravity="center"
        android:text="_____ " />

    <TextView
        android:id="@+id/results"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/edithoteltitle" />

    <TextView
        android:id="@+id/edithoteltitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/pickeddate"
        android:layout_centerHorizontal="true"
        android:text="Check the hotel you want to edit" />

    <CheckBox
        android:id="@+id/checkHotel1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/pickeddate"
        android:layout_below="@+id/results"
        android:text="Hotel1" />

    <CheckBox
        android:id="@+id/checkHotel2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/checkHotel1"
        android:layout_below="@+id/checkHotel1"
        android:text="Hotel2" />

    <CheckBox
        android:id="@+id/checkHotel3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/checkHotel2"
        android:layout_below="@+id/checkHotel2"

```

Κώδικας 36. edithotels.xml

```

<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/scrollView1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

    <RelativeLayout
        android:id="@+id/RelativeLayout1"
        android:layout_width="wrap_content"
        android:layout_height="710dp"
        android:background="@drawable/simpleb"
        android:columnCount="2"
        android:orientation="horizontal" >

        <Space
            android:id="@+id/Space1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_column="1"
            android:layout_gravity="fill_vertical"
            android:layout_row="5" />

        <Spinner
            android:id="@+id/hotelstars"
            android:layout_width="50dp"
            android:layout_height="20dp"
            android:layout_alignRight="@+id/View04"
            android:layout_alignTop="@+id/date"
            android:entries="@array/spinnerItems"
            android:spinnerMode="dialog" />

        <EditText
            android:id="@+id/hotelcity"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentRight="true"
            android:layout_below="@+id/hotelname"
            android:ems="10"
            android:inputType="textCapSentences"/>

        <EditText
            android:id="@+id/hoteladdress"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignLeft="@+id/hotelcity"
            android:layout_alignRight="@+id/hotelcity"
            android:layout_below="@+id/hotelcity"
            android:ems="10"
            android:inputType="textCapSentences"
            />

        <EditText
            android:id="@+id/hotelpostalcode"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignLeft="@+id/hoteladdress"
            android:layout_alignRight="@+id/hoteladdress"
            android:layout_below="@+id/hoteladdress"
            android:ems="10"
            android:inputType="textPostalAddress"/>

        <EditText
            android:id="@+id/hoteltelephone"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignLeft="@+id/hotelpostalcode"
            android:layout_alignParentRight="true"

```

Κώδικας 37. fragment_add_hotel.xml

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bckgrnd"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:theme="@style/AppBaseTheme"
    tools:context="com.example.hotelreseration.MainScreenActivity$PlaceholderFragment" >

    <TextView
        android:id="@+id/user_label"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:text="Welcome, User!"
        android:textColor="#042c83"
        android:textSize="25sp"
        android:textStyle="bold"
        tools:ignore="HardcodedText" />

    <View
        android:id="@+id/separator"
        android:layout_width="wrap_content"
        android:layout_height="2dp"
        android:layout_alignLeft="@+id/user_label"
        android:layout_alignRight="@+id/user_label"
        android:layout_below="@+id/user_label"
        android:background="@android:color/darker_gray"
        android:visibility="visible"
        tools:ignore="RtlHardcoded" />

    <TextView
        android:id="@+id/onoma"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/separator"
        android:layout_centerHorizontal="true"
        android:text="and"
        android:textStyle="bold"
        tools:ignore="HardcodedText" />

    <TextView
        android:id="@+id/TextView01"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/onoma"
        android:layout_centerHorizontal="true"
        android:text="Get A Room"
        android:textColor="#025900"
        android:textSize="25sp"
        android:textStyle="bold"
        tools:ignore="HardcodedText" />

    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:layout_below="@+id/TextView01"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="14dp"
        android:src="@drawable/getaroom"
        tools:ignore="ContentDescription" />

```

Κώδικας 38. fragment_home.xml

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/fragment_hotels"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bckgrnd"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.hotelreseration.NavigationDrawer.Fragment_hotels" >

    <ListView
        android:id="@+id/list"
        android:layout_width="match_parent"
        android:layout_height="310dp"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/BookedText"
        tools:ignore="RtlHardcoded" >

    </ListView>

    <TextView
        android:id="@+id/BookedText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:textSize="30sp"
        android:text="@string/my_hotels"
        android:textStyle="bold"
        android:shadowDx="-1"
        android:shadowDy="-1"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <View
        android:id="@+id/separator"
        android:layout_width="wrap_content"
        android:layout_height="1dp"
        android:layout_alignBottom="@+id/BookedText"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:background="#000000"
        android:visibility="visible"
        tools:ignore="RtlHardcoded" />

    <Button
        android:id="@+id/addbutton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/list"
        android:layout_alignParentBottom="true"
        android:layout_alignRight="@+id/list"
        android:background="@drawable/custom_button"
        android:onClick="addhotel"
        android:text="Add"
        android:textColor="#FFFFFF"
        tools:ignore="HardcodedText,RtlHardcoded" />

    <TextView
        android:id="@+id/nobookingtxt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/list"
        android:layout_centerHorizontal="true"

```

Κώδικας 39. fragment_hotels.xml

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:background="@drawable/bckgrnd"
    tools:context="com.example.hotelreservation.MainScreenActivity$PlaceholderFragment" >

    <View
        android:id="@+id/View01"
        android:layout_width="wrap_content"
        android:layout_height="1dp"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/changeinfo"
        android:background="#000000"
        android:visibility="visible"
        tools:ignore="RtlHardcoded" />

    <TextView
        android:id="@+id/user_label"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:text="Information"
        android:textColor="#000000"
        android:textSize="30sp"
        android:textStyle="bold"
        tools:ignore="HardcodedText" />

    <View
        android:id="@+id/View02"
        android:layout_width="wrap_content"
        android:layout_height="1dp"
        android:layout_alignLeft="@+id/View01"
        android:layout_below="@+id/guide"
        android:background="#000000"
        android:visibility="visible"
        tools:ignore="RtlHardcoded" />

    <View
        android:id="@+id/View06"
        android:layout_width="wrap_content"
        android:layout_height="1dp"
        android:layout_alignLeft="@+id/View04"
        android:layout_alignRight="@+id/View04"
        android:layout_below="@+id/reportaproblem"
        android:background="#000000"
        android:visibility="visible"
        tools:ignore="RtlHardcoded" />

    <View
        android:id="@+id/View05"
        android:layout_width="wrap_content"
        android:layout_height="1dp"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/contactus"
        android:background="#000000"
        android:visibility="visible"
        tools:ignore="RtlHardcoded" />

    <View
        android:id="@+id/View08"

```

Κώδικας 40. fragment_info.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.hotelreseration.MainScreenActivity$PlaceholderFragment" >

    <TextView
        android:id="@+id/section_label"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="19dp" />

</RelativeLayout>
```

Κώδικας 41. fragment_main_screen.xml


```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/simpleb">

    <ScrollView
        android:id="@+id/scrollView1"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <RelativeLayout
            xmlns:tools="http://schemas.android.com/tools"
            android:id="@+id/RelativeLayout1"
            android:layout_width="match_parent"
            android:layout_height="500dp"
            android:background="@drawable/simpleb"
            android:columnCount="2"
            android:orientation="horizontal">

            <TextView
                android:id="@+id/bookingtitle"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentTop="true"
                android:layout_centerHorizontal="true"
                android:layout_marginTop="19dp"
                android:text="Your booking at hotel"
                android:textAppearance="?android:attr/textAppearanceLarge"
                android:textSize="24sp"
                android:textStyle="bold"
                tools:ignore="HardcodedText"/>

            <View
                android:id="@+id/separator"
                android:layout_width="wrap_content"
                android:layout_height="3dp"
                android:layout_alignParentLeft="true"
                android:layout_below="@id/bookingtitle"
                android:background="@android:color/darker_gray"
                android:visibility="visible"
                tools:ignore="RtlHardcoded"/>

            <TextView
                android:id="@+id/textView3"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignBaseline="@+id/textView4"
                android:layout_alignBottom="@+id/textView4"
                android:layout_marginEnd="26dp"
                android:layout_toStartOf="@+id/textView17"
                android:text="Check In"
                android:textColor="@android:color/holo_green_dark"
                android:textSize="18sp"
                android:textStyle="bold"/>

            <TextView
                android:id="@+id/textView4"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@+id/separator"
                android:layout_marginStart="12dp"
                android:layout_marginTop="11dp"
                android:layout_toEndOf="@+id/textView17"
                android:text="Check Out"

```

Κώδικας 42. fragment_make_booking.xml

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

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <com.google.android.gms.maps.MapView
        android:id="@+id/map"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        class="com.google.android.gms.maps.SupportMapFragment"/>
</LinearLayout>
```

Κώδικας 43. fragment_maps.xml

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bckgrnd"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"

    tools:context="com.example.hotelreseration.NavigationDrawer.Fragment_hotels" >

    <ListView
        android:id="@+id/pastResList"
        android:layout_width="match_parent"
        android:layout_height="330dp"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        tools:ignore="RtlHardcoded">
    </ListView>

    <View
        android:id="@+id/separator"
        android:layout_width="wrap_content"
        android:layout_height="1dp"
        android:layout_alignBottom="@+id/pastRes"
        android:layout_alignLeft="@+id/searchList"
        android:background="#000000"
        android:visibility="visible"
        tools:ignore="RtlHardcoded" />

    <TextView
        android:id="@+id/nopasttxt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:text="No past reservations"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textColor="#9f0007"
        tools:ignore="HardcodedText"/>

    <TextView
        android:id="@+id/pastRes"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/pastResList"
        android:layout_centerHorizontal="true"
        android:text="Past Reservations"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textStyle="bold"
        tools:ignore="HardcodedText"/>

</RelativeLayout>

```

Κώδικας 44.fragment_owner_past.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bckgrnd"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.hotelreseration.NavigationDrawer.Fragment_hotels" >

    <ListView
        android:id="@+id/runningResList"
        android:layout_width="match_parent"
        android:layout_height="330dp"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        tools:ignore="RtlHardcoded">
    </ListView>

    <View
        android:id="@+id/separator"
        android:layout_width="wrap_content"
        android:layout_height="1dp"
        android:layout_alignBottom="@+id/onoma"
        android:layout_alignLeft="@+id/searchList"
        android:background="#000000"
        android:visibility="visible"
        tools:ignore="RtlHardcoded" />

    <TextView
        android:id="@+id/nobookingtxt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:text="No running reservations"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textColor="#9f0007"
        tools:ignore="HardcodedText"/>

    <TextView
        android:id="@+id/onoma"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Running Reservations"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textStyle="bold"
        tools:ignore="HardcodedText"
        android:layout_above="@+id/runningResList"
        android:layout_centerHorizontal="true"/>
</RelativeLayout>
```

Κώδικας 45. fragment_owner_running.xml

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bckgrnd"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.hotelreseration.NavigationDrawer.Fragment_hotels" >

    <ListView
        android:id="@+id/upcomingResList"
        android:layout_width="match_parent"
        android:layout_height="330dp"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        tools:ignore="RtlHardcoded">
    </ListView>

    <View
        android:id="@+id/separator"
        android:layout_width="wrap_content"
        android:layout_height="1dp"
        android:layout_alignBottom="@+id/onoma"
        android:layout_alignLeft="@+id/searchList"
        android:background="#000000"
        android:visibility="visible"
        tools:ignore="RtlHardcoded" />

    <TextView
        android:id="@+id/nobookingtxt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:text="No upcoming reservations"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textColor="#9f0007"/>

    <TextView
        android:id="@+id/onoma"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Upcoming Reservations"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textStyle="bold"
        android:layout_above="@+id/upcomingResList"
        android:layout_centerHorizontal="true"/>

</RelativeLayout>

```

Κώδικας 46. fragment_owner_upcoming.xml

```
<android.support.v4.app.FragmentTabHost
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@android:id/tabhost"
    android:layout_width="match_parent"
    android:textColor="#FFFFFF"
    android:background="@drawable/bckgrnd"
    android:layout_height="match_parent">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:textColor="#FFFFFF"
        android:layout_height="match_parent">

        <FrameLayout
            android:id="@+id/realtabcontent"
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:textColor="#FFFFFF"
            android:layout_weight="1"/>

    </LinearLayout>
</android.support.v4.app.FragmentTabHost>
```

Κώδικας 47. fragment_reservations.xml

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bckgrnd" >

    <TextView
        android:id="@+id/BookedText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="16dp"
        android:shadowDx="-1"
        android:shadowDy="-1"
        android:text="@string/Settings"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textSize="30sp"
        android:textStyle="bold" />

    <View
        android:id="@+id/separator"
        android:layout_width="330dp"
        android:layout_height="2dp"
        android:layout_below="@+id/BookedText"
        android:background="#000000"
        android:visibility="visible"
        tools:ignore="RtlHardcoded"
        android:layout_alignStart="@+id/imageView2" />

    <TextView
        android:id="@+id/changeinfo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="56dp"
        android:text="Change Your Info"
        android:textAppearance="?android:attr/textAppearanceLarge"
        tools:ignore="HardcodedText"
        android:layout_below="@+id/separator"
        android:layout_centerHorizontal="true" />

    <ImageView
        android:id="@+id/userImage"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="18dp"
        android:src="@android:drawable/ic_menu_delete"
        android:layout_alignTop="@+id/deleteaccount"
        android:layout_alignParentStart="true"
        android:layout_marginStart="26dp" />

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@android:drawable/ic_lock_idle_lock"
        android:layout_above="@+id/View01"
        android:layout_alignStart="@+id/userImage"
        android:layout_alignEnd="@+id/userImage" />

    <TextView
        android:id="@+id/deleteaccount"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="37dp"
        android:text="Delete Account"
    
```

Κώδικας 48. fragment_settings.xml

Κώδικας 49. fragment_traveler_booked.xml


```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/RelativeLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/blue"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/map"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="120dp"
        android:text="Completed"
        tools:ignore="HardcodedText" />

</RelativeLayout>
```

Κώδικας 50. fragment_traveler_completed.xml

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bckgrnd"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.hotelreseration.NavigationDrawer.Fragment_hotels" >

    <TextView
        android:id="@+id/nofavoriteslabel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:text="No favorite hotels has been added yet!"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textColor="#9f0007"/>

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Favorites"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textStyle="bold"
        android:layout_marginTop="50dp"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"/>

    <View
        android:id="@+id/view"
        android:layout_width="wrap_content"
        android:layout_height="1dp"
        android:layout_alignBottom="@+id/onoma"
        android:layout_alignLeft="@+id/searchList"
        android:background="#000000"
        android:visibility="visible"
        android:layout_below="@id/textView2"
        tools:ignore="RtlHardcoded"/>

    <ListView
        android:id="@+id/favoriteList"
        android:layout_width="match_parent"
        android:layout_height="330dp"
        tools:ignore="RtlHardcoded"
        android:layout_below="@+id/view"
        android:layout_alignParentStart="true"
        android:layout_marginTop="19dp">
    </ListView>
</RelativeLayout>

```

Κώδικας 51. fragment_traveler_favorite.xml

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/RelativeLayout1"
    android:layout_width="fill_parent"
    android:background="@drawable/bckgrnd"
    android:layout_height="fill_parent" >

    <ListView
        android:id="@+id/searchList"
        android:layout_width="300dp"
        android:layout_height="300dp"
        android:layout_alignParentStart="true"
        android:layout_alignParentBottom="true"
        android:layout_alignParentEnd="true">
    </ListView>

    <Button
        android:id="@+id/checkin"
        android:layout_width="130dp"
        android:layout_height="wrap_content"
        android:text="Check In"
        android:background = "#0a04051b"
        android:textColor="#cbffd8"
        android:radius="10dp"
        tools:ignore="HardcodedText,RtlHardcoded"
        android:layout_above="@+id/citysearch"
        android:layout_alignParentStart="true"/>

    <Button
        android:id="@+id/checkout"
        android:layout_width="130dp"
        android:layout_height="wrap_content"
        android:text="Check Out"
        tools:ignore="HardcodedText,RtlHardcoded"
        android:background = "#0a04051b"
        android:textColor="#ffc8c8"
        android:radius="10dp"
        android:layout_alignBaseline="@+id/checkin"
        android:layout_alignBottom="@+id/checkin"
        android:layout_alignParentEnd="true"/>

    <TextView
        android:id="@+id/onoma"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center_vertical"
        android:text="Search city:"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textStyle="bold"
        tools:ignore="HardcodedText,RtlHardcoded"
        android:layout_above="@+id/search"
        android:layout_alignStart="@+id/checkin"
        android:layout_marginBottom="12dp"/>

    <View
        android:id="@+id/View02"
        android:layout_width="wrap_content"
        android:layout_height="1dp"
        android:layout_below="@+id/search"
        android:background="#000000"
        android:visibility="visible"
        tools:ignore="RtlHardcoded"
        android:layout_alignParentEnd="true"
        android:layout_alignParentStart="true"/>

```

Κώδικας 52. fragment_traveler_search.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/RelativeLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/blue"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/map"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="120dp"
        android:text="Upcoming"
        tools:ignore="HardcodedText" />

</RelativeLayout>
```

Κώδικας 53. fragment_traveler_upcoming.xml

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bckgrnd"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:theme="@style/AppBaseTheme"
    tools:context="com.example.hotelreseration.MainScreenActivity$PlaceholderFragment" >

    <EditText
        android:id="@+id/ChangeSurname"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:clickable="false"
        android:cursorVisible="false"
        android:ems="10"
        android:focusable="false"
        android:focusableInTouchMode="false"
        android:gravity="center_vertical|center"
        android:hint="Surname"
        android:inputType="textPersonName"
        android:text="Epwnymo"
        android:textColor="#0e0e0e"
        android:textColorHint="#c1bdbl"
        android:textStyle="bold"
        android:textSize="23sp"
        tools:ignore="HardcodedText,RtlHardcoded"
        android:layout_alignBaseline="@+id/ChangeName"
        android:layout_alignBottom="@+id/ChangeName"
        android:layout_alignParentEnd="true" />

    <EditText
        android:id="@+id/ChangeName"
        style="@android:style/Widget.AutoCompleteTextView"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_below="@+id/userImage"
        android:clickable="false"
        android:cursorVisible="false"
        android:editable="false"
        android:ems="10"
        android:focusable="false"
        android:focusableInTouchMode="false"
        android:freezesText="true"
        android:gravity="center"
        android:hint="Name"
        android:inputType="textPersonName"
        android:text="Onoma"
        android:textColor="#0e0e0e"
        android:textColorHint="#c1bdbl"
        android:textSize="23sp"
        android:textStyle="bold"
        tools:ignore="Deprecated,HardcodedText" />

    <ImageView
        android:id="@+id/userImage"
        android:layout_width="90dp"
        android:layout_height="90dp"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"

```

Κώδικας 54. fragment_user.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/RelativeLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/simpleb"
    android:columnCount="2"
    android:orientation="horizontal"
    tools:context="${relativePackage}.${activityClass}" >

    <TextView
        android:id="@+id/pinonmap"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="31dp"
        android:clickable="false"
        android:onClick="PinOnMap"
        android:text="Pin Hotel On The Map?"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <ImageView
        android:id="@+id/userImage"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@+id/pinonmap"
        android:layout_alignParentLeft="true"
        android:layout_alignTop="@+id/pinonmap"
        android:layout_toLeftOf="@+id/pinonmap"
        android:src="@drawable/ic_maps_indicator_current_position" />

    <View
        android:id="@+id/separator"
        android:layout_width="wrap_content"
        android:layout_height="1dp"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/pinonmap"
        android:background="@android:color/darker_gray"
        android:visibility="visible"
        tools:ignore="RtlHardcoded" />

    <TextView
        android:id="@+id/todaysdate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/separator"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="30dp"
        android:text="Long press on the place you want to pin your hotel" />

</RelativeLayout>
```

Κώδικας 55. hotelinfo.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/RelativeLayout1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">

    <TextView
        android:id="@+id/tv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:shadowColor="#000000"
        android:shadowDx="1"
        android:shadowDy="1"
        android:shadowRadius="2"
        android:textColor="#202020"
        android:textSize="26sp"
        android:textStyle="normal" />

    <TextView
        android:id="@+id/sub"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/tv"
        android:shadowColor="#000000"
        android:shadowDx="1"
        android:shadowDy="1"
        android:shadowRadius="2"
        android:textColor="#4a494d"
        android:textSize="20sp"
        android:textStyle="italic" />

</RelativeLayout>
```

Κώδικας 56. mytextview.xml

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/scrollView1"
    android:background="@drawable/simpleb"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

    <LinearLayout
        android:id="@+id/LinearLayout2"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:gravity="center"
        android:orientation="vertical" >

        <TextView
            android:id="@+id/textView8"
            android:layout_width="262dp"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal|center|top"
            android:text="Hello, new to our app? Here some things to get to know about the
app better." />

        <TextView
            android:id="@+id/edithoteltitle"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:text="As a hotel owner you can add in the app your hotels and their
info. You can add only 3 hotels per account." />

        <View
            android:layout_width="match_parent"
            android:layout_height="2dp"
            android:background="@android:color/darker_gray"/>

        <TextView
            android:id="@+id/date"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="There are 3 types of rooms available:" />

        <TextView
            android:id="@+id/textView10"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Suite"
            android:textAppearance="?android:attr/textAppearanceLarge" />

        <TextView
            android:id="@+id/textView7"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:text="Suite is the best room. Contains the most like AirCondition,
MiniBar, TV, PrivateBathroom with jacuzzi, nice view from balcony ect.." />

        <TextView
            android:id="@+id/textView6"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Luxury"
            android:textAppearance="?android:attr/textAppearanceLarge" />

        <TextView
            android:id="@+id/textView11"

```

Κώδικας 57. ownerguide.xml


```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/LinearLayout2"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/simpleb"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/textView8"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal|center|top"
        android:gravity="center"
        android:text="If you have any issues, please feel free to report them." />

    <TextView
        android:id="@+id/textView9"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="_____ "
        android:textAppearance="?android:attr/textAppearanceSmall" />

    <TextView
        android:id="@+id/nobookingtxt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <EditText
        android:id="@+id/suite1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="0.55"
        android:ems="10"
        tools:ignore="InefficientWeight" >

        <requestFocus />
    </EditText>
</LinearLayout>
```

Κώδικας 58. reportproblem.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/simpleb">

    <ScrollView
        android:id="@+id/scrollView1"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <RelativeLayout
            android:id="@+id/RelativeLayout1"
            android:layout_width="wrap_content"
            android:layout_height="710dp"
            android:background="@drawable/simpleb"
            android:columnCount="2"
            android:orientation="horizontal">

            <TextView
                android:id="@+id/HotelInfoTitle"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentEnd="true"
                android:layout_alignParentStart="true"
                android:layout_weight="1"
                android:gravity="center"
                android:text="Hotel Name"
                android:textAlignment="center"
                android:textAllCaps="true"
                android:textSize="24sp"
                android:textStyle="bold"/>

            <View
                android:id="@+id/View01"
                android:layout_width="wrap_content"
                android:layout_height="2dp"

                android:layout_below="@+id/ratingBar"
                android:background="@android:color/darker_gray"
                android:visibility="visible"/>

            <View
                android:id="@+id/view13"
                android:layout_width="wrap_content"
                android:layout_height="1dp"

                android:layout_below="@+id/additionalInfoTextView"
                android:background="@android:color/darker_gray"
                android:visibility="visible"
                android:layout_alignEnd="@+id/mapView2"
                android:layout_alignStart="@+id/websiteTextView"/>

            <TextView
                android:id="@+id/additionalInfoTextView"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginTop="9dp"
                android:text="Additional Info"
                android:textAppearance="@style/TextAppearance.AppCompat.Caption"
                android:textSize="18sp"
                android:textStyle="bold"
                android:layout_below="@+id/View01"
                android:layout_centerHorizontal="true"/>

            <com.google.android.gms.maps.MapView

```

Κώδικας 59. traveler_hotel_registration.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/RelativeLayout1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:background="#4104051b">

    <TextView
        android:id="@+id/tv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:shadowColor="#000000"
        android:shadowDx="1"
        android:shadowDy="1"
        android:shadowRadius="2"
        android:textColor="#0d0d0d"
        android:textSize="26sp"
        android:textStyle="normal" />

    <TextView
        android:id="@+id/sub"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/tv"
        android:shadowColor="#000000"
        android:shadowDx="1"
        android:shadowDy="1"
        android:shadowRadius="2"
        android:textColor="#8e8c94"
        android:textSize="20sp"
        android:textStyle="italic" />

</RelativeLayout>
```

Κώδικας 60. `traveler_hotel_search_list.xml`

```

<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/scrollView1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

    <LinearLayout
        android:id="@+id/LinearLayout2"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:gravity="center"
        android:background="@drawable/simpleb"
        android:orientation="vertical" >

        <TextView
            android:id="@+id/textView14"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Hello, new to our app? Here some things to get to know about the app
better"/>

        <TextView
            android:id="@+id/edithoteltitle"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:text="As a traveler you can select a hotel to make a reservation. You can
search hotel in specific city from the tab Search in the Manage Booking option."/>

        <TextView
            android:id="@+id/reportaproblem"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:text="Yoy can search hotels around you in the Map option from the menu."/>

        <View
            android:layout_width="match_parent"
            android:layout_height="2dp"
            android:background="@android:color/darker_gray"/>

        <TextView
            android:id="@+id/date"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="There are 3 tabs in the Manage Booking section:" />

        <TextView
            android:id="@+id/textView6"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Favorite:"
            android:textAppearance="?android:attr/textAppearanceLarge" />

        <TextView
            android:id="@+id/textView7"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:text="Here you can save your favorite hotel so you can access them
easily the next time you want to visit them." />

        <TextView
            android:id="@+id/textView10"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

```

Κώδικας 61. travelerguide.xml

Παράρτημα Β΄: Κώδικας PHP

```
<?php
```

```
/**  
 * Database config variables  
 */  
define("DB_HOST", "fdb3.freehostingeu.com");  
define("DB_USER", "2043488_ptyxiaki");  
define("DB_PASSWORD", "pr0v0c@teur");  
define("DB_DATABASE", "2043488_ptyxiaki");  
?>
```

Κώδικας 62. Config.php

```
<?php
```

```
class DB_Connect {  
    private $conn;  
  
    // Connecting to database  
    public function connect() {  
        require_once 'Config.php';  
  
        // Connecting to mysql database  
        $this->conn = new mysqli(DB_HOST, DB_USER, DB_PASSWORD, DB_DATABASE);  
  
        // return database handler  
        return $this->conn;  
    }  
}  
  
?>
```

Κώδικας 63. DB_Connect.php

```
<?php
```

```
class DB_Functions {
```

```
    private $conn;
```

```
    // constructor
```

```
    function __construct() {
```

```
        require_once 'DB_Connect.php';
```

```
        // connecting to database
```

```
        $db = new Db_Connect();
```

```
        $this->conn = $db->connect();
```

```
    }
```

```
    // destructor
```

```
    function __destruct() {
```

```
    }
```

```
    /**
```

```
     * Storing new Owner
```

```
     * returns user details
```

```
     */
```

```
    public function storeOwner($name, $surname, $email, $password, $telephone, $country) {
```

```
        $hash = $this->hashSSHA($password);
```

```
        $encrypted_password = $hash["encrypted"]; // encrypted password
```

```
        $salt = $hash["salt"]; // salt
```

```
        $stmt = $this->conn->prepare("INSERT INTO owner (name, surname, country, email, password, salt, telephone) VALUES(?, ?, ?, ?, ?, ?, ?)");
```

```
        $stmt->bind_param("ssssss", $name, $surname, $country, $email, $encrypted_password, $salt, $telephone);
```

```
        $result = $stmt->execute();
```

```
Κώδικας 64. DB_Functions.php
```

```
<?php
//<form method="post">
require_once 'DB_Functions.php';
require_once 'DB_Connect.php';
$db = new DB_Functions();

// json response array
$response = array("error" => FALSE);

if (($_POST['hotelname'])) {

    // receiving the post params
    $hotelname = $_POST['hotelname'];

    $coords = $db->loadBookingCoordinates($hotelname);

    if ($coords != false) {

        foreach($coords as $coo){
            $response["error"] = FALSE;
            $response["coords"][] = array(
                "cooid" => $coo["cooid"],
                "hotelname"=> $coo["hotelname"],
                "oidFK" => $coo["oidFK"],
                "latitude" => $coo["latitude"],
                "longitude" => $coo["longitude"]
            );
        }
        echo json_encode($response);
    } else {
        // coordinates not found
    }
}

```

Κώδικας 65. bookingcoords.php

```
<?php
//<form method="post">
require_once 'DB_Functions.php';
require_once 'DB_Connect.php';
$db = new DB_Functions();

// json response array
$response = array("error" => FALSE);

if (isset($_POST['hotelname'])) {

    // receiving the post params
    $hotelname = $_POST['hotelname'];

    // get the hotel by name
    $hotel = $db->deleteHotelandCoords($hotelname);

    if ($hotel != false) {
        // user is found
        $response["error"] = FALSE;
        $response["error_msg"] = "Hotel has been successfully deleted and unpinned";

        echo json_encode($response);
    } else {
        $hotel = $db->deleteHotel($hotelname);
        $response["error"] = TRUE;
        $response["error_msg"] = "Hotel has been successfully deleted";
        echo json_encode($response);
    }
} else {
    // required post params are missing
}

```

Κώδικας 66. deletehotel.php


```
<?php
//<form method="post">
require_once 'DB_Functions.php';
require_once 'DB_Connect.php';
$db = new DB_Functions();

// json response array
$response = array("error" => FALSE);

if (isset($_POST['email'])) {

    // receiving the post params
    $email = $_POST['email'];

    // get the user by email and password
    $user = $db->deleteOwnerandHotels($email);

    if ($user != false) {
        // user is found
        $response["error"] = FALSE;
        $response["error_msg"] = "User has been successfully deleted";

        echo json_encode($response);
    } else {

        $user = $db->deleteOwner($email);

        // user is not found with the credentials
        $response["error"] = TRUE;
        $response["error_msg"] = "User has been deleted";
        echo json_encode($response);
    }
}

Κώδικας 67. deleteowner.php
```

```
<?php
//<form method="post">
require_once 'DB_Functions.php';
require_once 'DB_Connect.php';
$db = new DB_Functions();

if (isset($_POST['email'])) {

    // receiving the post params
    $email = $_POST['email'];

    // get the user by email and password
    $user = $db->deleteTraveler($email);

    if ($user != false) {
        // user is found
        $response["error"] = FALSE;
        $response["error_msg"] = "User has been successfully deleted";

        echo json_encode($response);
    } else {
        // user is not found with the credentials
        $response["error"] = TRUE;
        $response["error_msg"] = "An unexpected error occurred. Please try again!";
        echo json_encode($response);
    }
} else {
    // required post params are missing
    $response["error"] = TRUE;
    $response["error_msg"] = "Required parameters email or password are missing!";
}

```

Κώδικας 68. deletetraveler.php

```
<?php
//<form method="post">
require_once 'DB_Functions.php';
require_once 'DB_Connect.php';
$db = new DB_Functions();

// json response array
$response = array("error" => FALSE);

if (isset($_POST['tid'])) {

    // receiving the post params
    $tid = $_POST['tid'];
    $password = $_POST['password'];

    // get the user by email and password
    $user = $db->getTravelerbyId($tid);

    if ($user != false) {
        // user is found
        $response["error"] = FALSE;
        $response["user"]["id"] = $user["tid"];
        $response["user"]["name"] = $user["name"];
        $response["user"]["surname"] = $user["surname"];
        $response["user"]["country"] = $user["country"];
        $response["user"]["email"] = $user["email"];
        $response["user"]["password"] = $user["password"];
        $response["user"]["salt"] = $user["salt"];
        $response["user"]["telephone"] = $user["telephone"];
        echo json_encode($response);
    } else {
Κώδικας 69. gettravelerbyid.php
```

```
<?php
//<form method="post">
require_once 'DB_Functions.php';
require_once 'DB_Connect.php';
$db = new DB_Functions();

// json response array
$response = array("error" => FALSE);

$coords = $db->loadCoordinates();

if ($coords != false) {

foreach($coords as $coo){
    $response["error"] = FALSE;
    $response["coords"][] = array(
        "cooid" => $coo["cooid"],
        "hotelname"=> $coo["hotelname"],
        "oidFK" => $coo["oidFK"],
        "latitude" => $coo["latitude"],
        "longitude" => $coo["longitude"]
    );
}
    echo json_encode($response);
} else {
    // coordinates not found
    $response["error"] = TRUE;
    $response["error_msg"] = "There are no hotels around :(";
    echo json_encode($response);
}
?>
```

Κώδικας 70. loadcoordinates.php

```
<?php
//<form method="post">
require_once 'DB_Functions.php';
require_once 'DB_Connect.php';
$db = new DB_Functions();

// json response array
$response = array("error" => FALSE);

if (isset($_POST['oidFK'])) {

    // receiving the post params
    $oidFK = $_POST['oidFK'];

    // get the hotel by oidFK
    $hotel = $db->getFavoriteHotelbyUserID($oidFK);

    if ($hotel != false) {
        // hotel(s) has been found

        foreach($hotel as $h){
            $response["error"] = FALSE;
            $response["hotel"][] = array(
                "fid" => $h["fid"],
                "name"=> $h["hotelname"],
                "city" => $h["hotelcity"],
                "userID" => $h["userID"]
            );
        }
    }
}
```

Κώδικας 71. loadfavorites.php

```
<?php
//<form method="post">
require_once 'DB_Functions.php';
require_once 'DB_Connect.php';
$db = new DB_Functions();

// json response array
$response = array("error" => FALSE);

if (isset($_POST['oidFK'])) {

    // receiving the post params
    $oidFK = $_POST['oidFK'];

    // get the hotel by oidFK
    $hotel = $db->getHotelbyOidFK($oidFK);

    if ($hotel != false) {
        // hotel(s) has been found

        foreach($hotel as $h){
            $response["error"] = FALSE;
            $response["hotel"][] = array(
                "hid" => $h["hid"],
                "name"=> $h["hotelname"],
                "city" => $h["city"],
                "address" => $h["address"],
                "tk" => $h["tk"],
                "telephone" => $h["telephone"],
                "stars" => $h["stars"],
```

Κώδικας 72. loadhotels.php

```
<?php
//<form method="post">
require_once 'DB_Functions.php';
require_once 'DB_Connect.php';
$db = new DB_Functions();

// json response array
$response = array("error" => FALSE);

if (isset($_POST['oidFK'])) {

    // receiving the post params
    $oidFK = $_POST['oidFK'];

    // get the hotel by oidFK
    $hotel = $db->getmyBookingsbyUserID($oidFK);

    if ($hotel != false) {
        // hotel(s) has been found

        foreach($hotel as $h){
            $response["error"] = FALSE;
            $response["hotel"][] = array(
                "reservationID" => $h["reservationID"],
                "hotelname"=> $h["hotelname"],
                "checkin" => $h["checkin"],
                "checkout" => $h["checkout"],
                "tidFK" => $h["tidFK"]
            );
        }
    }
}
```

Κώδικας 73. loadmybookings.php

```
<?php
//<form method="post">
require_once 'DB_Functions.php';
require_once 'DB_Connect.php';
$db = new DB_Functions();

// json response array
$response = array("error" => FALSE);

if (isset($_POST['oidFK'])) {

    // receiving the post params
    $oidFK = $_POST['oidFK'];

    // get the hotel by oidFK
    $hotel = $db->getReservationsbyUserID($oidFK);

    if ($hotel != false) {
        // hotel(s) has been found

        foreach($hotel as $h){
            $response["error"] = FALSE;
            $response["hotel"][] = array(
                "reservationID" => $h["reservationID"],
                "hotelname"=> $h["hotelname"],
                "checkin" => $h["checkin"],
                "checkout" => $h["checkout"],
                "tidFK" => $h["tidFK"]
            );
        }
    }
}
```

Κώδικας 74. loadreservations.php


```
<?php
//<form method="post">
require_once 'DB_Functions.php';
require_once 'DB_Connect.php';
$db = new DB_Functions();

// json response array
$response = array("error" => FALSE);

if (isset($_POST['email']) && isset($_POST['password'])) {

    // receiving the post params
    $email = $_POST['email'];
    $password = $_POST['password'];

    // get the user by email and password
    $user = $db->getOwnerByEmailAndPassword($email, $password);

    if ($user != false) {
        // user is found
        $response["error"] = FALSE;
        $response["user"]["id"] = $user["oid"];
        $response["user"]["name"] = $user["name"];
        $response["user"]["surname"] = $user["surname"];
        $response["user"]["country"] = $user["country"];
        $response["user"]["email"] = $user["email"];
        $response["user"]["password"] = $user["password"];
        $response["user"]["salt"] = $user["salt"];
        $response["user"]["telephone"] = $user["telephone"];
        echo json_encode($response);
    } else {
```

Κώδικας 75. loginowner.php

```
<?php
//<form method="post">
require_once 'DB_Functions.php';
require_once 'DB_Connect.php';
$db = new DB_Functions();

// json response array
$response = array("error" => FALSE);

if (isset($_POST['email']) && isset($_POST['password'])) {

    // receiving the post params
    $email = $_POST['email'];
    $password = $_POST['password'];

    // get the user by email and password
    $user = $db->getTravelerByEmailAndPassword($email, $password);

    if ($user != false) {
        // user is found
        $response["error"] = FALSE;
        $response["user"]["id"] = $user["tid"];
        $response["user"]["name"] = $user["name"];
        $response["user"]["surname"] = $user["surname"];
        $response["user"]["country"] = $user["country"];
        $response["user"]["email"] = $user["email"];
        $response["user"]["password"] = $user["password"];
        $response["user"]["salt"] = $user["salt"];
        $response["user"]["telephone"] = $user["telephone"];
        echo json_encode($response);
    } else {
```

Κώδικας 76. logintraveler.php

```
<?php

require_once 'DB_Functions.php';
$db = new DB_Functions();

// json response array
$response = array("error" => FALSE);

if (($_POST['hotelname']) && ($_POST['checkin']) && ($_POST['checkout']) && ($_POST['tidFK']) &&
($_POST['adults']) && ($_POST['children']) && ($_POST['bedtype']) && ($_POST['roomtype']))
{
    // receiving the post params
    $hotelname = $_POST['hotelname'];
    $checkin = $_POST['checkin'];
    $checkout = $_POST['checkout'];
    $tidFK = $_POST['tidFK'];
    $adults = $_POST['adults'];
    $children = $_POST['children'];
    $bedtype = $_POST['bedtype'];
    $roomtype = $_POST['roomtype'];

    $reservation = $db->storeBooking($hotelname, $checkin, $checkout, $tidFK, $adults, $children,
    $bedtype, $roomtype);
    if ($reservation) {
        // reservation stored successfully
        $response["error"] = FALSE;
        $response["reservation"]["hotelname"] = $reservation["hotelname"];
        $response["reservation"]["checkin"] = $reservation["checkin"];
        $response["reservation"]["checkout"] = $reservation["checkout"];
        $response["reservation"]["tidFK"] = $reservation["tidFK"];
        $response["reservation"]["adults"] = $reservation["adults"];
        $response["reservation"]["children"] = $reservation["children"];
    }
}
```

Κώδικας 77. makebooking.php

```
<?php

require_once 'DB_Functions.php';
$db = new DB_Functions();

// json response array
$response = array("error" => FALSE);

if ( ($_POST['hotelname']) && ($_POST['oidFK']) && ($_POST['latitude']) && ($_POST['longitude']) )
{
    // receiving the post params
    $hotelname = $_POST['hotelname'];
    $oidFK = $_POST['oidFK'];
    $latitude = $_POST['latitude'];
    $longitude = $_POST['longitude'];

    // check if coords is already existed with the same address
    if ($db->checkPlace($hotelname)) {
        // coords already existed
        $response["error"] = TRUE;
        $response["error_msg"] = "There is already a Hotel named ". $hotelname;
        echo json_encode($response);
    } else {
        // create a new coords
        $coords = $db->storeCoordinates($hotelname, $oidFK, $latitude, $longitude);
        if ($coords) {
            // coords stored successfully
            $response["error"] = FALSE;
            $response["coords"]["id"] = $coords["cooid"];
            $response["coords"]["hotelname"] = $coords["hotelname"];
            $response["coords"]["oidFK"] = $coords["oidFK"];
        }
    }
}
```

Κώδικας 78. registercoordinates.php

```
<?php

require_once 'DB_Functions.php';
$db = new DB_Functions();

// json response array
$response = array("error" => FALSE);

if (($_POST['hotelname']) && ($_POST['city']) && ($_POST['userID']))
{
    // receiving the post params
    $hotelname = $_POST['hotelname'];
    $city = $_POST['city'];
    $userID = $_POST['userID'];

    // check if hotel is already existed with the same hotelname
    if ($db->isHotelFavorited($hotelname, $userID)) {
        // hotel already existed
        $response["error"] = TRUE;
        $response["error_msg"] = "Hotel is already labeled as favorite";
        echo json_encode($response);
    } else {
        // create a new hotel
        $hotel = $db->storeFavorite($hotelname, $city, $userID);
        if ($hotel) {
            // hotel stored successfully
            $response["error"] = FALSE;
            $response["hotel"]["id"] = $hotel["hid"];
            $response["hotel"]["name"] = $hotel["hotelname"];
            $response["hotel"]["city"] = $hotel["city"];
            $response["hotel"]["userID"] = $hotel["userID"];
        }
    }
}
```

Κώδικας 79. registerfavorites.php

```
<?php
```

```
require_once 'DB_Functions.php';
```

```
$db = new DB_Functions();
```

```
// json response array
```

```
$response = array("error" => FALSE);
```

```
if (($_POST['hotelname']) && ($_POST['city']) && ($_POST['address']) && ($_POST['tk']) &&
($_POST['telephone']) && ($_POST['stars']) && ($_POST['oidFK']) && ($_POST['website'])&&
($_POST['swimmingpool']) )
```

```
{
```

```
    // receiving the post params
```

```
    $hotelname = $_POST['hotelname'];
```

```
    $city = $_POST['city'];
```

```
    $address = $_POST['address'];
```

```
    $tk = $_POST['tk'];
```

```
    $telephone = $_POST['telephone'];
```

```
    $stars = $_POST['stars'];
```

```
    $oidFK = $_POST['oidFK'];
```

```
    $website = $_POST['website'];
```

```
    $swimmingpool = $_POST['swimmingpool'];
```

```
    // $latitude = $_POST['latitude'];
```

```
    // $longitude = $_POST['longitude'];
```

```
    // check if hotel is already existed with the same address
```

```
    if ($db->isHotelExisted($address)) {
```

```
        // hotel already existed
```

```
        $response["error"] = TRUE;
```

```
        $response["error_msg"] = "Hotel already exists with " . $address;
```

```
        echo json_encode($response);
```

```
    } else {
```

```
Κώδικας 80. registerhotel.php
```

```
<?php

require_once 'DB_Functions.php';
$db = new DB_Functions();

// json response array
$response = array("error" => FALSE);

if (($_POST['name']) && ($_POST['surname']) && ($_POST['email']) && ($_POST['password']) &&
($_POST['telephone'] && ($_POST['country'] ) )
{
    // receiving the post params
    $name = $_POST['name'];
    $surname = $_POST['surname'];
    $email = $_POST['email'];
    $password = $_POST['password'];
    $telephone = $_POST['telephone'];
    $country = $_POST['country'];

    // check if user is already existed with the same email
    if ($db->isOwnerExisted($email)) {
        // user already existed
        $response["error"] = TRUE;
        $response["error_msg"] = "User already exists with " . $email;
        echo json_encode($response);
    } else {
        // create a new user
        $user = $db->storeOwner($name, $surname, $email, $password, $telephone, $country);
        if ($user) {
            // user stored successfully
            $response["error"] = FALSE;
        }
    }
}

Κώδικας 81. registerowner.php
```

```
<?php

require_once 'DB_Functions.php';
$db = new DB_Functions();

// json response array
$response = array("error" => FALSE);

if (($_POST['name']) && ($_POST['surname']) && ($_POST['email']) && ($_POST['password']) &&
($_POST['telephone'] && ($_POST['country'] ) )
{
    // receiving the post params
    $name = $_POST['name'];
    $surname = $_POST['surname'];
    $email = $_POST['email'];
    $password = $_POST['password'];
    $telephone = $_POST['telephone'];
    $country = $_POST['country'];

    // check if user is already existed with the same email
    if ($db->isTravelerExisted($email)) {
        // user already existed
        $response["error"] = TRUE;
        $response["error_msg"] = "User already exists with " . $email;
        echo json_encode($response);
    } else {
        // create a new user
        $user = $db->storeTraveler($name, $surname, $email, $password, $telephone, $country);
        if ($user) {
            // user stored successfully
            $response["error"] = FALSE;
        }
    }
}

Κώδικας 82. registertraveler.php
```



```
<?php
//<form method="post">
require_once 'DB_Functions.php';
require_once 'DB_Connect.php';
$db = new DB_Functions();

// json response array
$response = array("error" => FALSE);

if (($_POST['city']) && ($_POST['checkin']) && ($_POST['checkout'])) {

    // receiving the post params
    $city = $_POST['city'];
    $checkin = $_POST['checkin'];
    $checkout = $_POST['checkout'];

    // get the hotel by oidFK
    $hotel = $db->getHotelResults($city, $checkin, $checkout);

    if ($hotel != false) {
        // hotel(s) has been found

        foreach($hotel as $h){
            $response["error"] = FALSE;
            $response["hotel"][] = array(
                "hid" => $h["hid"],
                "name"=> $h["hotelname"],
                "city" => $h["city"],
                "address" => $h["address"],
                "tk" => $h["tk"],
```

Κώδικας 83. resulthotels.php

```
<?php
//<form method="post">
require_once 'DB_Functions.php';
require_once 'DB_Connect.php';
$db = new DB_Functions();

// json response array
$response = array("error" => FALSE);

if (($_POST['city']) && ($_POST['checkin']) && ($_POST['checkout'])) {

    // receiving the post params
    $city = $_POST['city'];
    $checkin = $_POST['checkin'];
    $checkout = $_POST['checkout'];

    // get the hotel by oidFK
    $hotel = $db->getHotelResults($city, $checkin, $checkout);

    if ($hotel != false) {
        // hotel(s) has been found

        foreach($hotel as $h){
            $response["error"] = FALSE;
            $response["hotel"][] = array(
                "hid" => $h["hid"],
                "name"=> $h["hotelname"],
                "city" => $h["city"],
                "address" => $h["address"],
                "tk" => $h["tk"],
```

Κώδικας 84. resulthotels.php

```
<?php
//<form method="post">
require_once 'DB_Functions.php';
require_once 'DB_Connect.php';
$db = new DB_Functions();

// json response array
$response = array("error" => FALSE);

if (($_POST['dbhname']) && ($_POST['nhname']) && ($_POST['nhcity']) && ($_POST['nhaddress'])
&& ($_POST['nhpostalcode']) && ($_POST['nhtelephone']) && ($_POST['nhwebsite']) &&
($_POST['nhstars']) && ($_POST['nhswimmingpool'])) {

    // receiving the post params
    $dbhname = $_POST['dbhname'];
    $nhname = $_POST['nhname'];
    $nhcity = $_POST['nhcity'];
    $nhaddress = $_POST['nhaddress'];
    $nhpostalcode = $_POST['nhpostalcode'];
    $nhtelephone = $_POST['nhtelephone'];
    $nhwebsite = $_POST['nhwebsite'];
    $nhstars = $_POST['nhstars'];
    $nhswimmingpool = $_POST['nhswimmingpool'];

    // get the user by email and password
    $user = $db->updateHotel($dbhname, $nhname, $nhcity, $nhaddress, $nhpostalcode,
    $nhtelephone, $nhwebsite, $nhstars, $nhswimmingpool);

    if ($user != false) {
        // user is found
        $response["error"] = FALSE;
        $response["error_msg"] = "Hotel has been successfully updated";
    }
}
```

Κώδικας 85. updatehotel.php

```
<?php
//<form method="post">
require_once 'DB_Functions.php';
require_once 'DB_Connect.php';
$db = new DB_Functions();

// json response array
$response = array("error" => FALSE);

if (($_POST['dbmail']) && ($_POST['nname']) && ($_POST['nsurname']) && ($_POST['nemail']) &&
($_POST['npass']) && ($_POST['ntelephone'] && ($_POST['ncountry']))) {

    // receiving the post params
    $dbmail = $_POST['dbmail'];
    $nname = $_POST['nname'];
    $nsurname = $_POST['nsurname'];
    $nemail = $_POST['nemail'];
    $npassword = $_POST['npassword'];
    $ntelephone = $_POST['ntelephone'];
    $ncountry = $_POST['ncountry'];

    // get the user by email and password
    $user = $db->updateOwner($dbmail, $nname, $nsurname, $nemail, $npassword, $ntelephone,
    $ncountry);

    if ($user != false) {
        // user is found
        $response["error"] = FALSE;
        $response["error_msg"] = "User has been successfully updated";
        echo json_encode($response);
    } else {
        // user is not found with the credentials
    }
}

```

Κώδικας 86. updateowner.php

```
<?php
//<form method="post">
require_once 'DB_Functions.php';
require_once 'DB_Connect.php';
$db = new DB_Functions();

// json response array
$response = array("error" => FALSE);

if (($_POST['dbmail']) && ($_POST['nname']) && ($_POST['nsurname']) && ($_POST['nemail']) &&
($_POST['npass']) && ($_POST['ntelephone'] && ($_POST['ncountry']))) {

    // receiving the post params
    $dbmail = $_POST['dbmail'];
    $nname = $_POST['nname'];
    $nsurname = $_POST['nsurname'];
    $nemail = $_POST['nemail'];
    $npassword = $_POST['npassword'];
    $ntelephone = $_POST['ntelephone'];
    $ncountry = $_POST['ncountry'];

    // get the user by email and password
    $user = $db->updateTraveler($dbmail, $nname, $nsurname, $nemail, $npassword, $ntelephone,
$ncountry);

    if ($user != false) {
        // user is found
        $response["error"] = FALSE;
        $response["error_msg"] = "User has been successfully updated";

        echo json_encode($response);
    } else {
Κώδικας 87. updatetraveler.php
```