

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Προπτυχιακό Πρόγραμμα Σπουδών στην Πληροφορική



Πτυχιακή Εργασία

**Υλοποίηση ηλεκτρονικού συστήματος κράτησης και αγοράς εισιτηρίων με χρήση της
πλατφόρμας Android Studio**

Σέρτζο Κοκαβέση

Επιβλέπων: Θεόδωρος Ανδρόνικος

7 Σεπτεμβρίου 2018

Επιβλέπων

Θεόδωρος Ανδρόνικος, Μόνιμος Επίκουρος Καθηγητής ,
Τμήματος Πληροφορικής του Ιονίου Πανεπιστημίου

Τριμελής Επιτροπή

Θεόδωρος Ανδρόνικος, Μόνιμος Επίκουρος Καθηγητής ,
Τμήματος Πληροφορικής του Ιονίου Πανεπιστημίου

Δημήτριος Τσουμάκος, Αναπληρωτής Καθηγητής,
Τμήματος Πληροφορικής του Ιονίου Πανεπιστημίου

Μιχαήλ Στεφανιδάκης, Επίκουρος Καθηγητής,
Τμήματος Πληροφορικής του Ιονίου Πανεπιστημίου

Η αφιέρωση της συγκεκριμένης Πτυχιακής Εργασίας στην οικογένεια μου,
αποτελεί το ελάχιστο ευχαριστώ, απέναντι στην
ανιδιοτελή υποστήριξη τους, καθ' όλη τη διάρκεια των
σπουδών μου.

Περίληψη

Σκοπός της πτυχιακής εργασίας αποτελεί η υλοποίηση μιας εφαρμογής για smartphones, τα οποία θα τρέχουν το λογισμικό android. Στην παρούσα εργασία γίνεται ανάλυση του θεωρητικού επιπέδου της εφαρμογής και του πρακτικού. Η εφαρμογή που θα φτάσει στα χέρια του τελικού χρήστη, θα του επιτρέψει να αναζητήσει δρομολόγια υπεραστικών λεωφορείων από και προς την πόλη της Κέρκυρας, καθώς και να κάνει κράτηση της θέσης που επιθυμεί στο λεωφορείο.

Στα κεφάλαια που ακολουθούν θα γίνει μια προσπάθεια λεπτομερείς ανάλυσης όλης της μεθοδολογίας που ακολουθήθηκε προκειμένου να υλοποιηθεί η εν λόγω εφαρμογή. Θα αναλυθεί επίσης λεπτομερώς και το λειτουργικό κομμάτι της εφαρμογής, δηλαδή ο κώδικας της εφαρμογής, ο οποίος είναι υπεύθυνος για τα UI(User Interface) και τις λειτουργίες της εφαρμογής.

Αποτελεί μια φιλική ως προς την χρήση της εφαρμογή, η οποία έχει ως σκοπό να δώσει στον χρήστη την δυνατότητα να κάνει κρατήσεις δρομολόγια λεωφορείων.

Πρόλογος και Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω την οικογένεια μου και τους φίλους μου για την στήριξη και για την ατελείωτη υπομονή και συμπαράσταση που μου έδωσαν κατά την διάρκεια των σπουδών μου. Επίσης θα ήθελα να ευχαριστήσω και τον καθηγητή μου, κύριο Θεόδωρο Ανδρόνικο που μου εμπιστεύτηκε την εκπόνηση της συγκεκριμένης πτυχιακής, και για όλη την πολύτιμη βοήθεια και την καθοδήγηση του. Σε αυτό το σημείο αξίζει να αναφερθώ και στον καθηγητή μου Σπυρίδων Σιούτα, που χωρίς αυτόν και την πολύτιμη βοήθεια του δεν θα είχα καταφέρει να ολοκληρώσω την συγκεκριμένη εργασία. Τέλος θα ήθελα να ευχαριστήσω και το τμήμα Πληροφορικής του Ιονίου Πανεπιστημίου και όλους τους καθηγητές μου που μου έδωσαν τις σωστές βάσεις προκειμένου να καταλήξω σε αυτό το αποτέλεσμα.

Περιεχόμενα

Περίληψη	4
Πρόλογος και Ευχαριστίες.....	5
Κατάλογος Πινάκων	7
Κεφάλαιο 1ο.....	9
1. Εισαγωγή	9
1.1.Στόχος της πτυχιακής εργασίας	9
1.2. Διάρθρωση της πτυχιακής εργασίας.....	9
Κεφάλαιο 2ο.....	10
2. Το Λογισμικό android	10
2.1. Εισαγωγή στο λειτουργικό σύστημα android	10
2.2. Ιστορική Αναδρομή.....	11
2.3. Η αρχιτεκτονική του Android	12
2.4. Η ανατομία μιας εφαρμογής Android.....	15
2.5. Ο κύκλος ζωής μιας Activity	19
2.6. Οι εκδόσεις του Android.....	21
Κεφάλαιο 3ο.....	31
3. Android Studio	31
1.1. Εγκατάσταση Λογισμικού.....	31
3.2 Διαδικασία Αποσφαλμάτωσης (Debugging)	38
3.3 Δημοσίευση Εφαρμογής	39
3.4 Android System Development Kit	39
3.5 Android Virtual Device Manager (AVD).....	40
Κεφάλαιο 4ο.....	42
4. Ανάλυση εφαρμογής	42
4.1 Ανάλυση απαιτήσεων για την δημιουργία της εφαρμογής.....	42
4.2 Ανάλυση της εφαρμογής με διαγράμματα UML.....	42
Κεφάλαιο 5ο.....	47
5. Παρουσίαση Corfu m-ticket	47
5.1 Περιγραφή εφαρμογής.....	47
5.2 Παρουσίαση εφαρμογής	48
Κεφάλαιο 6ο.....	60
6. Δομή της Εφαρμογής	60
Κεφάλαιο 7ο.....	65
7. Υλοποίηση Εφαρμογής	65
7.1 Περιγραφή Υλοποίησης.....	65
7.2 Πηγαίος Κώδικας-Κωδικοποίηση	65
Κεφάλαιο 8ο.....	111
8. Σύνοψη.....	111
8.1 Συμπεράσματα	111
8.2 Μελλοντικές Επεκτάσεις	111
Κεφάλαιο 9ο.....	112
9.Βιβλιογραφία	112

Κατάλογος Εικόνων

Εικόνα 1: Στατιστικό για το ποσοστό κάλυψης της αγοράς από android συστήματα (http://www.statista.com/statistics/272307/market-share-forecast-for-smartphone-operatingsystems/)	10
Εικόνα 2: Το πρώτο κινητό που υποστήριξε Android	11
Εικόνα 3: Αρχιτεκτονικό διάγραμμα του Android	12
Εικόνα 4: Το επίπεδο του πυρήνα Linux	12
Εικόνα 5: Το επίπεδο των βασικών βιβλιοθηκών του συστήματος	13
Εικόνα 6 : Το επίπεδο του Android Runtime.....	14
Εικόνα 7: Το επίπεδο Application Framework.....	14
Εικόνα 8 : Το επίπεδο των εφαρμογών του Android.....	15
Εικόνα 9: Η σύνταξη μιας activity σε Android Studio	15
Εικόνα 10 : Η σύνταξη μιας service σε Android Studio	16
Εικόνα 11: Η σύνταξη του Content Provider σε Android Studio	16
Εικόνα 12 : Η σύνταξη του Broadcast Receiver Component σε Android Studio	16
Εικόνα 13: Παράδειγμα για fragment	17
Εικόνα 14 : Παρουσίαση των Resources	18
Εικόνα 15: Κύκλος ζωής Android	20
Εικόνα 16 : Όλες οι εκδόσεις του Android.....	21
Εικόνα 17 : Όλες οι εκδόσεις του λειτουργικού Android.....	21
Εικόνα 18 : Android Cupcake.....	22
Εικόνα 19 : Android Donut	23
Εικόνα 20 : Android Éclair	23
Εικόνα 21: Android Froyo	24
Εικόνα 22: Android Έκδοση Gingerbread	24
Εικόνα 23: Android Έκδοση Honeycomb.....	25
Εικόνα 24: Android Έκδοση Ice Cream Sandwich.....	25
Εικόνα 25: Android Έκδοση Jelly Bean	26
Εικόνα 26 : Android Έκδοση KitKat	27
Εικόνα 27: Android Έκδοση Lollipop.....	28
Εικόνα 28 : Android Έκδοση Marshmallow	28
Εικόνα 29 : Android Έκδοση Nougat.....	29
Εικόνα 30 : Android Έκδοση Oreo	30
Εικόνα 31: Διαδικασία εγκατάστασης του Android Studio(http://www.tutorialspoint.com/android/android_studio.htm)	32
Εικόνα 32 : Διαδικασία επιλογής του αρχείου java	32
Εικόνα 33 : Επιλογή του αρχείου java	33
Εικόνα 34 : Επιλογή παραμέτρων.....	33
Εικόνα 35 : Επιλογή διαδρομής εγκατάστασης.....	34
Εικόνα 36: Ρύθμιση Προσομοιωτή.....	34
Εικόνα 37: Εγκατάσταση Android Studio	35
Εικόνα 38 : Αρχικό περιβάλλον πλατφόρμας.....	35
Εικόνα 39 : Δημιουργία εφαρμογής.....	36
Εικόνα 40 : Επιλογή των παραμέτρων της εφαρμογής	36
Εικόνα 41 : Επιλογή της οθόνης της εφαρμογής	37
Εικόνα 42: Περιβάλλον εργασίας πλατφόρμας	37
Εικόνα 43: Logcat	38
Εικόνα 44 : Android Device Monitor(DDMS)	38
Εικόνα 45 : Διαχείριση Εικονικών Συσκευών	39
Εικόνα 46 : Android Virtual Device Manager	40
Εικόνα 47: Android Virtual Device Manager Επιλογή API.....	40
Εικόνα 48 : Προσθήκη εικονικής συσκευής	41
Εικόνα 49 : Εικονική συσκευή προσομοίωσης	41
Εικόνα 50 : StarUML – Πρόγραμμα υλοποίησης UML.....	42
Εικόνα 51: Διάγραμμα Περιπτώσεων Χρήσης	43
Εικόνα 52: Αναπαράσταση σε screenshot, η εφαρμογή Corfu m-ticket την στιγμή που ο χρήστης επιλέγει εισιτήριο	44
Εικόνα 53 : Αναπαράσταση σε screenshot του Corfu m-ticket	45
Εικόνα 54 : Το μοντέλο της πρωτοτυποποίησης.....	46
Εικόνα 55 : Logo εφαρμογής.....	48

Εικόνα 56 : Εικονίδιο της Εφαρμογής.....	49
Εικόνα 57: Welcome Screens της Εφαρμογής	49
Εικόνα 58 : Οθόνη Register.....	50
Εικόνα 59 : Οθόνη Login.....	50
Εικόνα 60 : Αλλαγή Κωδικού Πρόσβασης.....	51
Εικόνα 61: Αναζήτηση Εισιτηρίου	51
Εικόνα 62 : Αναζήτηση Δρομολογίου και Επιλογή Θέσης.....	52
Εικόνα 63 : Επισκόπηση εισιτηρίου	52
Εικόνα 64 : Επιλογή τρόπου πληρωμής	53
Εικόνα 65 : Πληρωμή μέσω Paypal.....	54
Εικόνα 66 : Πληρωμή μέσω Κάρτα.....	54
Εικόνα 67: Μενού της Εφαρμογής.....	55
Εικόνα 68 : Η επιλογή της Ακύρωσης Εισιτηρίου	56
Εικόνα 69 : Ιστορικό Κρατήσεων.....	56
Εικόνα 70 : Όροι και Προϋποθέσεις	57
Εικόνα 71 : Επικοινωνία.....	57
Εικόνα 72 : Πληροφορίες	58
Εικόνα 73 : Προώθηση Εφαρμογής.....	58
Εικόνα 74 : Google Playstore	59
Εικόνα 75 : Φάκελος Res.....	60
Εικόνα 76 : Φάκελος Res/Drawable	61
Εικόνα 77 : Φάκελος Values.....	62
Εικόνα 78 : Φάκελος Res/Layout.....	62
Εικόνα 79 : Φάκελος Java	63
Εικόνα 80 : Φάκελος Manifest.....	63
Εικόνα 81 : Φάκελος Assets	64
Εικόνα 82 : Κώδικας του WelcomeScreen.java.....	66
Εικόνα 83: Values και Layout.....	66
Εικόνα 84 : Welcome_screen_1.xml	67
Εικόνα 85 : Welcome_screen_2.xml	69
Εικόνα 86 : LoginActivity.java	70
Εικόνα 87 : Layout_register_pin.xml	73
Εικόνα 88 : Layout_reset_password.xml	75
Εικόνα 89 : Layout_forgot_password.xml	75
Εικόνα 90 : activity_main_drawer.xml	76
Εικόνα 91: MainActivity.java	77
Εικόνα 92 : build.gradle	78
Εικόνα 93: build.gradle	78
Εικόνα 94 : SQLiteDatabase.java	79
Εικόνα 95 : Entinties.....	80
Εικόνα 96 : AppContext.java	81
Εικόνα 97 : App.java	81
Εικόνα 98 : BaseActivity.java	82
Εικόνα 99 : Αναζήτηση Δρομολογίου	83
Εικόνα 100 : Δημιουργία Λογαριασμού Paypal.....	84
Εικόνα 101: MainActivity.java	88
Εικόνα 102 : SelectBusActivity.java	89
Εικόνα 103 : SeatSelectionActivity.java	91
Εικόνα 104 : SeatAdapter.java	92
Εικόνα 105 : SeatAHolder.java.....	92
Εικόνα 106 : BookingActivity.java	95
Εικόνα 107 : HistoryBookingActivity.java.....	96
Εικόνα 108 : HistoryHolder.java	97
Εικόνα 109 : CancelTicketActivity.java.....	98
Εικόνα 110 : CancelTicketAdapter.java	99
Εικόνα 111 : CancelTicketHolder.java	100
Εικόνα 112 : AboutActivity.java	100
Εικόνα 113 : ConditionsActivity.java.....	101
Εικόνα 114 : SQLiteDatabase.java	102
Εικόνα 115 : Activity_main.xml.....	104
Εικόνα 116 : Layout_select_bus.xml	106
Εικόνα 117 : Activity_seat_selection_screen.xml	107
Εικόνα 118 : activity_booking.xml	108
Εικόνα 119 : androidmanifest.xml.....	110
Εικόνα 120 : Build.gradle	110

1. Εισαγωγή

1.1. Στόχος της πτυχιακής εργασίας

Η Ελλάδα είναι μια χώρα που φημίζεται για τα νησιά της και για τον τουρισμό της. Σπουδάζοντας λοιπόν σε ένα νησί που τα δρομολόγια είναι περιορισμένα και είναι ιδιαιτέρως δύσκολο να βρεις εισιτήρια σε περιόδους με αυξημένη κινητικότητα, και έχοντας νιώσει αυτό το άγχος και την αγωνία σχετικά με την κράτηση ενός εισιτηρίου, προσπάθησα στην συγκεκριμένη πτυχιακή εργασία να δημιουργήσω ένα σύστημα το οποίο θα προσπαθήσει να λύσει το πρόβλημα αυτό. Με λίγα λόγια γίνεται προσπάθεια ανάπτυξης μιας εφαρμογής αναζήτησης και κράτησης εισιτηρίων για υπεραστικά ταξίδια από και προς την πόλη της Κέρκυρας. Η εφαρμογή θα παρουσιάζει όλες τις απευθείας διαδρομές. Ενδεικτικό είναι ότι τα τελευταία χρόνια παρατηρείται ότι όλο και περισσότεροι πελάτες τείνουν στο να κάνουν τις κρατήσεις τους διαδικτυακά και ο λόγος είναι ότι εξοικονομούν χρόνο, ορισμένες φορές χρήμα (καθώς είναι σύνηθες το φαινόμενο μιας έκπτωσης ενός δρομολογίου, ιδιαίτερα για αεροπορικές κρατήσεις) και γλιτώνουν από την ταλαιπωρία της ουράς. Το σύστημα που αναπτύχθηκε για αυτή την πτυχιακή αποτελεί μία εφαρμογή για τους πελάτες που είναι κάτοχοι smartphone (λειτουργικό σύστημα android). Γίνεται παρακάτω αναλυτική παρουσίαση της εφαρμογής.

1.2. Διάρθρωση της πτυχιακής εργασίας

Η παρούσα πτυχιακή εργασία αποτελείται από εννέα κεφάλαια.

Στο πρώτο κεφάλαιο γίνεται μία εισαγωγή και παρουσιάζεται ο στόχος και η διάρθρωση της πτυχιακής εργασίας.

Στο δεύτερο κεφάλαιο γίνεται μία παρουσίαση του Android. Γίνεται μία ιστορική αναδρομή των εκδόσεων του λογισμικού Android, και παρουσιάζεται η αρχιτεκτονική και η ανατομία του.

Στο τρίτο κεφάλαιο γίνεται αναφορά στην πλατφόρμα Android Studio, που χρησιμοποιήσαμε για την υλοποίηση της εφαρμογής. Θα παρουσιαστούν στο κεφάλαιο αυτό τα βήματα εγκατάστασης της πλατφόρμας και τα σημαντικότερα χαρακτηριστικά της.

Στο τέταρτο κεφάλαιο πραγματοποιείται η ανάλυση της εφαρμογής, με χρήση διαγραμμάτων UML.

Το επόμενο κεφάλαιο είναι το πέμπτο, και σε αυτό το κεφάλαιο γίνεται η παρουσίαση της εφαρμογής που δημιουργήσαμε μέσα από εικόνες-screenshots.

Στο έκτο κεφάλαιο συναντάμε την δομή της εφαρμογής. Δηλαδή βλέπουμε τα αρχεία που χρησιμοποιήσαμε, πως τα διαχωρίσαμε και τι σημαίνουν αυτά τα αρχεία.

Στο έβδομο κεφάλαιο συναντάμε την υλοποίηση της εφαρμογής, δηλαδή τον πηγαίο κώδικα.

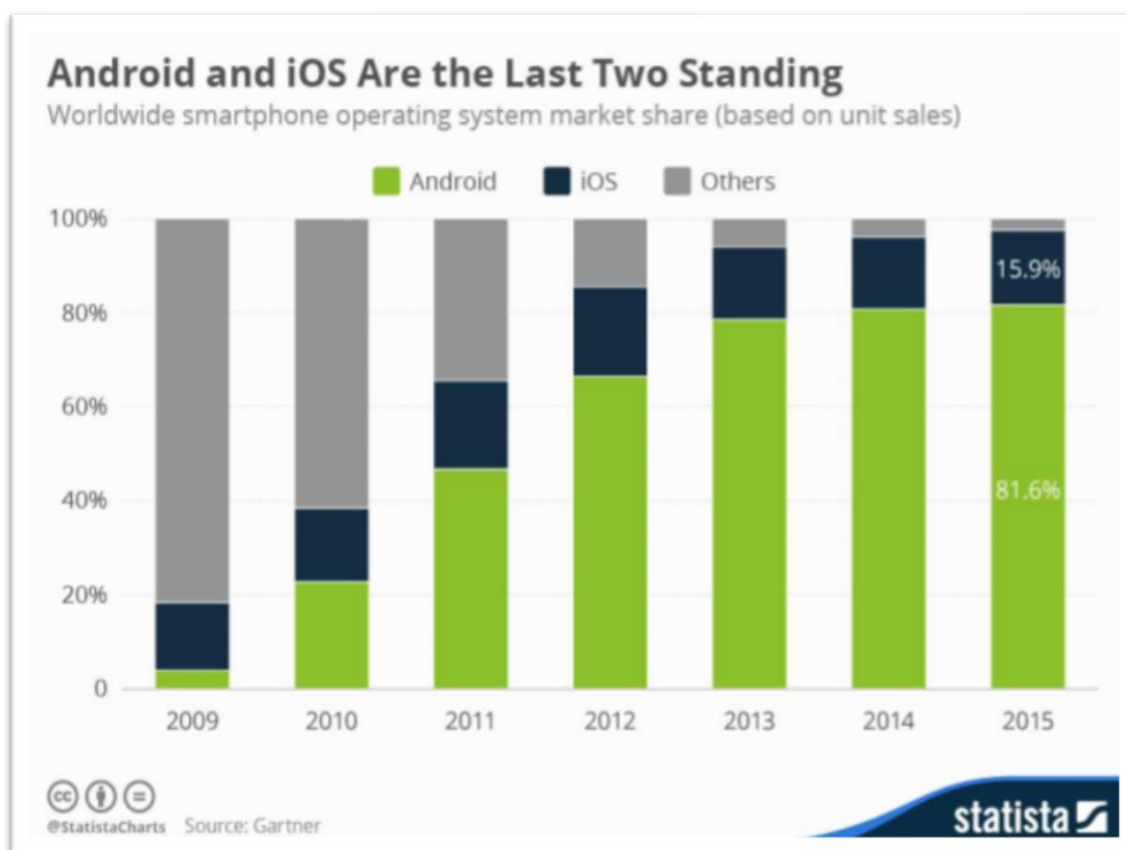
Στην συνέχεια συναντάμε το όγδοο κεφάλαιο στο οποίο καταλήγουμε στα συμπεράσματα της πτυχιακής εργασίας και προτείνουμε μελλοντικές επεκτάσεις στο application μας.

Τέλος στο κεφάλαιο εννιά συναντάμε την βιβλιογραφία της πτυχιακής εργασίας μας.

2. Το Λογισμικό android

2.1. Εισαγωγή στο λειτουργικό σύστημα android

Το android αποτελεί ένα λειτουργικό σύστημα το οποίο είναι σχεδιασμένο για κινητές συσκευές, οι οποίες τρέχουν τον πυρήνα Kernel του λειτουργικού συστήματος Linux και επιπλέον δίνει την δυνατότητα στους προγραμματιστές να συνθέσουν κώδικα με τη χρήση της γλώσσας προγραμματισμού Java. Σήμερα οι περισσότερες εταιρίες κινητής τηλεφωνίας χρησιμοποιούν το λειτουργικό σύστημα android. Αρκεί να κοιτάξει κάποιος την παρακάτω εικόνα προκειμένου να γίνει αντιληπτό τι ποσοστό της αγοράς καλύπτει το λογισμικό android και ποσό ραγδαία αυξάνεται.



Εικόνα 1: Στατιστικό για το ποσοστό κάλυψης της αγοράς από android συστήματα
(<http://www.statista.com/statistics/272307/market-share-forecast-for-smartphone-operatingsystems/>)

2.2. Ιστορική Αναδρομή

Το android δημιουργήθηκε το 2003 στο Palo Alto της Καλιφόρνια, από τους Andy Rubin, Rich Miner, Nick Sears και Chris White. Οι πρώτες προθέσεις της εταιρείας ήταν η ανάπτυξη ενός λειτουργικού συστήματος για ψηφιακές φωτογραφικές μηχανές ωστόσο η ιδέα αυτή εγκαταλείφθηκε γρήγορα καθώς διαπίστωσε ότι η αγορά των φωτογραφικών μηχανών δεν ήταν αρκετά μεγάλη για τους στόχους της. Έτσι λίγους μήνες αργότερα η εταιρεία έστρεψε της προσπάθειες της στην δημιουργία ενός λειτουργικού συστήματος το οποίο θα είχε ως στόχο τον εντοπισμό και την αναγνώριση της τοποθεσίας του ιδιοκτήτη οποιαδήποτε στιγμή το είχε ανάγκη. Στόχος της εταιρείας ήταν η δημιουργία ενός λειτουργικού που θα ανταγωνιζόταν την Microsoft Windows Mobile. Ο Rubin αντιμετώπιζε μια δυσκολία ως προς την προσέλκυση επενδυτών μέχρι την στιγμή που εμφανίστηκε η Google η οποία επιθυμούσε να την χρησιμοποιήσει ως κύρια μηχανή αναζήτησης για το T-mobile Sidesick. Έτσι το 2005 πραγματοποιείται μία συνάντηση μεταξύ του Rubin και του Larry Page(ήταν ένας από τους δύο ιδρυτές της Google). Στη συγκεκριμένη συνάντηση έγινε η παρουσίαση του Android με την μορφή ενός παγκόσμιου ανοιχτού λειτουργικού συστήματος για έξυπνα κινητά τηλέφωνα. Αυτή η παρουσίαση ολοκληρώθηκε επιτυχώς, και αυτό είχε ως αποτέλεσμα η Google να πειστεί και να εξαγοράσει το Android. Μετά την επίτευξη της συμφωνίας, η ομάδα της Google με επικεφαλής τον Rubin ανέπτυξε μία πλατφόρμα κινητών συσκευών που τροφοδοτείται από τον πυρήνα του Linux. Η Google προώθησε την συγκεκριμένη πλατφόρμα στους αμέσους συνεργάτες της και στους κατασκευαστές, με την υπόσχεση ότι θα δημιουργήσουν ένα ευέλικτο, αναβαθμισμένο και ανταγωνιστικό σύστημα για τα δεδομένα της εποχής. Το λειτουργικό σύστημα είχε δημιουργηθεί για συσκευές οι οποίες θα υποστηρίζαν κουμπιά, ωστόσο το 2008 , τόσο η Nokia , όσο και η Blackberry ανακοίνωσαν κινητές συσκευές που υποστήριζαν touch πράγμα που άλλαξε τα σχέδια. Το πρώτο smartphone της αγοράς που υποστήριζε android ήταν το T-Mobile G1.

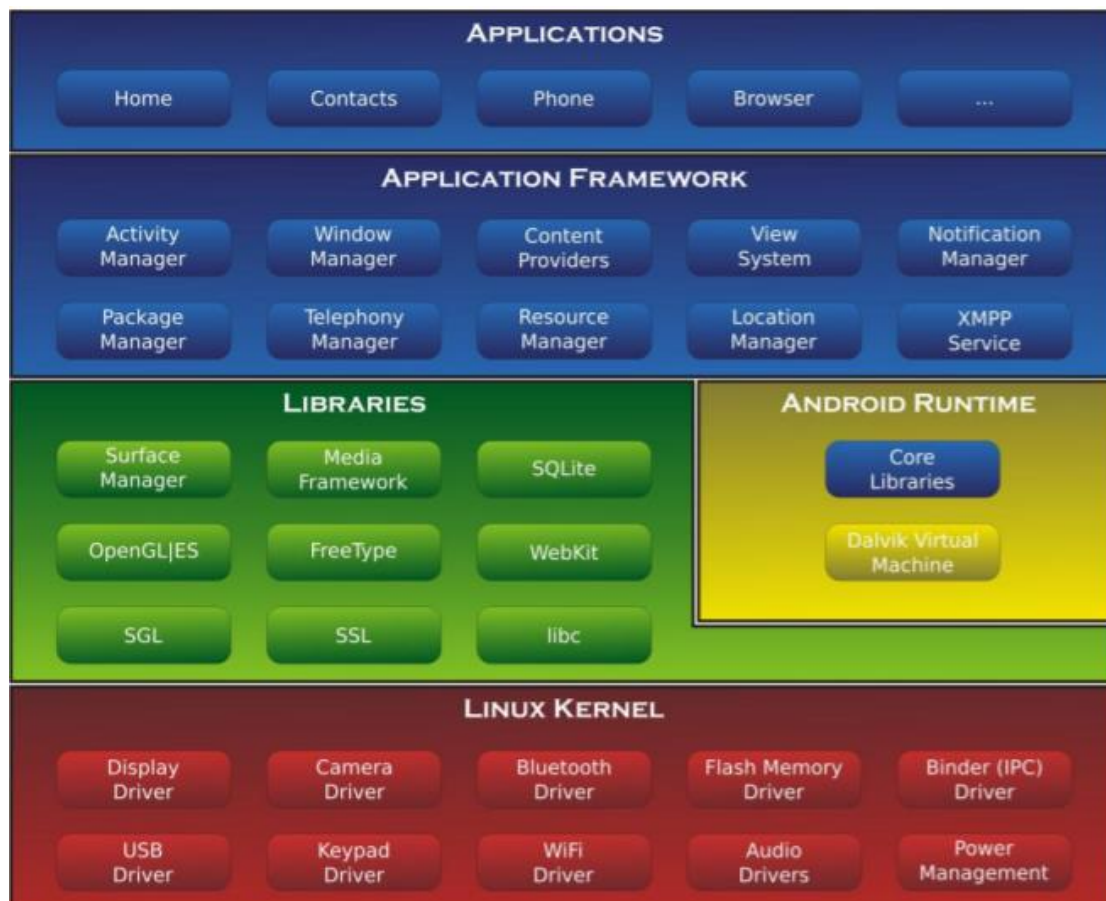


Εικόνα 2: Το πρώτο κινητό που υποστήριζε Android

Το φθινόπωρο του 2007, ανακοινώνεται η δημιουργία ενός συνεταιρισμού εταιρειών με το όνομα Open Handset Alliance. Ο συγκεκριμένος συνεταιρισμός αποτελούταν από πολλούς εκπροσώπους τόσο της Google όσο και της Texas Instruments, της Intel, της TMobile, της Sprint Nextel, της HTC, της Samsung, της Qualcomm (κατασκευάστρια εταιρία των επεξεργαστών) και άλλους με σκοπό να μετατρέψουν το Android σε μια πλατφόρμα ανοιχτού κώδικα που θα έχει ενσωματωμένο το λογισμικό Linux και σιγά σιγά να κυκλοφορήσουν στην αγορά τα πρώτα Smartphones με Android. Με αυτό τον τρόπο κατάφερε το android να αποτελέσει το κυρίαρχο λειτουργικό αυτή την στιγμή στην αγορά.

2.3. Η αρχιτεκτονική του Android

Το Android αποτελείται από ορισμένες συνιστώσες λογισμικού που συνθέτουν ένα ολοκληρωμένο και ενιαίο σύστημα. Το συγκεκριμένο σύστημα παρέχει όλα τα μέσα που απαιτούνται για τη χρήση εφαρμογών χωρίζοντάς το σε 4 επίπεδα και 5 ομάδες συνιστωσών. Όπως φαίνεται και στην εικόνα 3 τα επίπεδα είναι τα εξής : Linux Kernel, Libraries, Android Runtime, Application Framework και Applications. Παρακάτω περιγράφονται αναλυτικά από κάτω προς τα πάνω.



Εικόνα 3: Αρχιτεκτονικό διάγραμμα του Android

- **Linux Kernel**

Το Android όπως προαναφέραμε και παραπάνω έχει χτιστεί πάνω στο Linux Kernel και αυτό είναι που συναντάμε στο κατώτερο επίπεδο της λογικής στοίβας του Android. Το Android βασίζεται στον πυρήνα αυτόν για βασικές λειτουργίες της συσκευής όπως για παράδειγμα η διαχείριση των drivers. Η διαχείριση των διεργασιών και η διαχείριση της μνήμης της συσκευής καθώς και η διαχείριση των διεπαφών δικτύου που διαθέτει η συσκευή, όπως για παράδειγμα Wifi, Bluetooth (βλ. εικόνα 4).



Εικόνα 4: Το επίπεδο του πυρήνα Linux

- **Libraries- Επίπεδο βασικών Βιβλιοθηκών**

Οι βασικές βιβλιοθήκες του συστήματος είναι αυτές που παρέχουν τις βασικές λειτουργικότητες όπως αυτή των γραφικών ή του ήχου. Οι βιβλιοθήκες που βρίσκονται σε αυτό το επίπεδο είναι υλοποιημένες σε C/C++ και χρησιμοποιούνται από διάφορα components του συστήματος. Αυτές οι δυνατότητες διατίθενται στους developers μέσα από το Android Application Framework. Μερικές από τις βασικές βιβλιοθήκες είναι οι παρακάτω:

- **System C library:** είναι μια BSD-παραγόμενη εφαρμογή της standard C system library (libc), συντονισμένη για embedded Linux-based συσκευές.
- **Media Libraries:** βασισμένες στο OpenCore ^y PacketVideo. Οι βιβλιοθήκες υποστηρίζουν playback και recording σε πολλά δημοφιλή audio και video formats, όπως MPEG4, H.264, MP3, AAC, AMR, JPG και PNG.
- **Surface Manager:** συνθετικός window manager παρόμοιος με αυτόν των Windows Vista ή του Compi3, αλλά πολύ πιο απλό. Οι εντολές σχεδιασμού πηγαίνουν εκτός οθόνης bitmap που μετέπειτα συνδυάζονται με άλλα bitmaps για να σχηματίσουν την οθόνη που βλέπει ο χρήστης. Αυτό επιτρέπει στο σύστημα να δημιουργήσει πολλά εφέ όπως διάφανα παράθυρα και εντυπωσιακά transitions.
- **LibWebCore:** ένας μοντέρνος browser που χρησιμοποιεί την WebKit βιβλιοθήκη για την γρήγορη προβολή HTML περιεχομένου. Η ίδια μηχανή χρησιμοποιείται και στο Google Chrome browser, στον Safari της Apple και στα Nokia S60. • SGL: μηχανή για 2D γραφικά.
- **3D libraries:** βασίζονται στα OpenGL ES 1.0 APIs. Οι βιβλιοθήκες χρησιμοποιούν 3D υλικό αν η συσκευή το διαθέτει αλλιώς έναν γρήγορο software rasterizer.
- **SQLite:** μια δυναμική και ελαφριά βάση δεδομένων, που χρησιμοποιείται από εφαρμογές για μόνιμη αποθήκευση δεδομένων, παρόμοια με αυτή που χρησιμοποιείται στον Firefox και στο iPhone.



Εικόνα 5: Το επίπεδο των βασικών βιβλιοθηκών του συστήματος

- **Android Runtime**

Στο δεύτερο επίπεδο αλλά σε διαφορετική συνιστώσα αυτή την φορά, συναντάμε το Android Runtime. Όπως φαίνεται και στην παρακάτω εικόνα αποτελείται από τα εξής :

- Βασικές βιβλιοθήκες για τη διεπαφή των εφαρμογών Java με το περιβάλλον της συσκευής στην οποία εκτελούνται.
- Τη Dalvik Virtual Machine η οποία είναι υπεύθυνη για την δημιουργία των εκτελέσιμων αρχείων των εφαρμογών προκειμένου να τα τρέξει το λειτουργικό σύστημα. Η Dalvik είναι μια εικονική μηχανή (VM) που σχεδιάστηκε και γράφτηκε από τον Dan Bornstein στη Google. Ο κώδικας της εφαρμογής μεταγλωττίζεται σε εντολές μηχανής που ονομάζονται bytecodes και στη συνέχεια εκτελούνται από την VM στο κινητό τηλέφωνο

Κάθε εφαρμογή είναι γραμμένη σε γλώσσα προγραμματισμού Java, γλώσσα την οποία δεν αντιλαμβάνεται το λειτουργικό σύστημα, γι' αυτό τον λόγο Dalvik δημιουργεί εκτελέσιμα αρχεία τα οποία εκτελούνται από το λειτουργικό σύστημα.



Εικόνα 6 : Το επίπεδο του Android Runtime

- **Application Framework**

Το συγκεκριμένο επίπεδο είναι υπεύθυνο για την παροχή πρόσβασης στους προγραμματιστές για τα frameworks APIs που χρησιμοποιούνται από τις εφαρμογές του πυρήνα. Με απλά λόγια σε αυτό το επίπεδο συμπεριλαμβάνονται τα δομικά στοιχεία που μπορούν να χρησιμοποιηθούν για την δημιουργία εφαρμογών. Ορισμένες από τις βασικές λειτουργίες που περιλαμβάνονται στο Application Framework είναι τα εξής:

- **View System:** Επιτρέπει τη χρήση λιστών, πλαισίων, κουμπιών, κ.α.
- **Content Providers:** Επιτρέπει στις εφαρμογές την πρόσβαση σε δεδομένα άλλων εφαρμογών.
- **Resource Manager:** Δίνει πρόσβαση σε αρχεία σχετικά με τη διάταξη των στοιχείων του γραφικού περιβάλλοντος.
- **Notification Manager:** Είναι υπεύθυνο για την διαχείριση των μηνυμάτων των εφαρμογών που εμφανίζονται στην status bar. Παραδείγματος χάριν τα εισερχόμενα μηνύματα που εμφανίζονται από διαδικτυακές εφαρμογές, όπως το facebook ή το Instagram.
- **Activity Manager:** Διαχειρίζεται τον κύκλο ζωής των εφαρμογών και παρέχει την δυνατότητα μετάβασης σε προγενέστερες καταστάσεις.



Εικόνα 7: Το επίπεδο Application Framework

- **Applications**

Τελειώνοντας την περιγραφή της στοίβας φτάνουμε στο ανώτερο και τελευταίο επίπεδο όπου βρίσκονται οι εφαρμογές του Android. Στο συγκεκριμένο επίπεδο βρίσκονται οι εφαρμογές που αντιλαμβάνεται ο χρήστης, πρόκειται δηλαδή για εφαρμογές όπως email, SMS , το ημερολόγιο , οι χάρτες, ο περιηγητής κ.α. Προεγκατεστημένες εφαρμογές ως επί το πλείστον, οι οποίες είναι προγραμματισμένες σε Java.



Εικόνα 8 : Το επίπεδο των εφαρμογών του Android

2.4. Η ανατομία μιας εφαρμογής Android

Η υλοποίηση εφαρμογών στην πλατφόρμα Android είναι αρκετά εύκολη και μπορεί να γίνει ακόμα και από αρχάριους, χρησιμοποιώντας ανοιχτά λογισμικά όπως το eclipse και το Android Studio. Το μόνο που απαιτείται για την ανάπτυξη εφαρμογών android είναι μία στοιχειώδης γνώση προγραμματισμού, προκειμένου ο προγραμματιστής να μπορέσει να αντιληφθεί την λογική ανάπτυξης τους. Όταν φτάσει στο σημείο που θα κατανοήσει την λογική αυτή , τότε δεν θα απέχει πολύ από την υλοποίησή της εφαρμογής. Υπάρχουν τέσσερα είδη συστατικών μερών μιας Android εφαρμογής. Κάθε είδος εξυπηρετεί έναν συγκεκριμένο σκοπό και έχει συγκεκριμένο κύκλο ζωής που ορίζει πότε το συστατικό θα δημιουργηθεί και πότε θα καταστραφεί. Τα τέσσερα components είναι :

- Activities
- Services
- Content Providers
- Broadcast receivers

Παρακάτω αναλύονται τα συγκεκριμένα συστατικά των εφαρμογών.

- **Activities**

Κάθε activity αντιπροσωπεύει μια οθόνη με ένα UI (User Interface) που εκτελεί διάφορες ενέργειες. Για παράδειγμα μία εφαρμογή διαχείρισης μηνυμάτων, μπορεί να έχει μία activity που να δημιουργεί μηνύματα, μία που θα τα διαβάζεις και μία που θα τα διαχειρίζεσαι. Όλες αυτές οι activities συνεργάζονται για να δώσουν στον χρήστη την τελική εφαρμογή. Παρ'όλο όμως που συνεργάζονται , κάθε μία activity είναι ανεξάρτητη από τις υπόλοιπες. Μία δραστηριότητα (activity) υλοποιείται ως μία υποκατηγορία των δραστηριοτήτων (activities) και συντάσσεται ως εξής:

```
public class MainActivity extends Activity {  
  
}
```

Εικόνα 9: Η σύνταξη μιας activity σε Android Studio

- **Services**

Πρόκειται για υπηρεσίες που εκτελούνται στο background μιας εφαρμογής προκειμένου να διεκπεραιώσει μεγάλης χρονικής διάρκειας διαδικασίες. Για παράδειγμα μια υπηρεσία μπορεί να παίζει μουσική στο background ενώ ο χρήστης περιηγείται στο διαδίκτυο ή συντάσσει ένα μήνυμα, χωρίς ουσιαστικά να εμποδίζει την οποιαδήποτε αλληλεπίδραση του χρήστη με οποιαδήποτε εφαρμογή. Η υπηρεσία αυτή υλοποιείται ως υποκατηγορία της υπηρεσίας και συντάσσεται ως εξής:

```
public class MyService extends Service {  
}
```

Εικόνα 10 : Η σύνταξη μιας service σε Android Studio

- **Content Providers**

Είναι υπεύθυνος για την διαχείριση ενός συνόλου από διαμοιραζόμενα δεδομένα της εφαρμογής. Με αυτόν τρόπο τα δεδομένα μπορούν να αποθηκευτούν στην SQLite βάση δεδομένων ή σε οποιοδήποτε χώρο αποθήκευσης έχει πρόσβαση η εφαρμογή. Μέσω του content provider, άλλες εφαρμογές μπορούν να ζητήσουν άδεια προκειμένου να τροποποιήσουν τα δεδομένα. Για παράδειγμα το λειτουργικό σύστημα προσφέρει έναν content provider που διαχειρίζεται της πληροφορίες των επαφών. Συνεπώς οποιαδήποτε εφαρμογή μπορεί με την κατάλληλη άδεια να ζητήσει από τον content provider να διαβάσει ή να χρησιμοποιήσει πληροφορία της επαφής. Η σύνταξη ορίζεται ως εξής :

```
public class MyContentProvider extends ContentProvider {  
    public void onCreate(){  
    }  
}
```

Εικόνα 11: Η σύνταξη του Content Provider σε Android Studio

- **Broadcast Receivers Components**

Πρόκειται για ένα component που ανταποκρίνεται στις ανακοινώσεις των μεταδόσεων όλου του συστήματος. Πολλά από τα broadcasts προέρχονται από το σύστημα, για παράδειγμα μία ανακοίνωση ότι η μπαταρία είναι χαμηλή. Παρόλο που τα broadcasts δεν έχουν UI(User Interface) μπορούν να δημιουργήσουν μια ειδοποίηση στο status bar, προκειμένου να ενημερώσουν τον χρήστη όταν ένα broadcast συμβεί. Η σύνταξη του ορίζεται ως εξής:

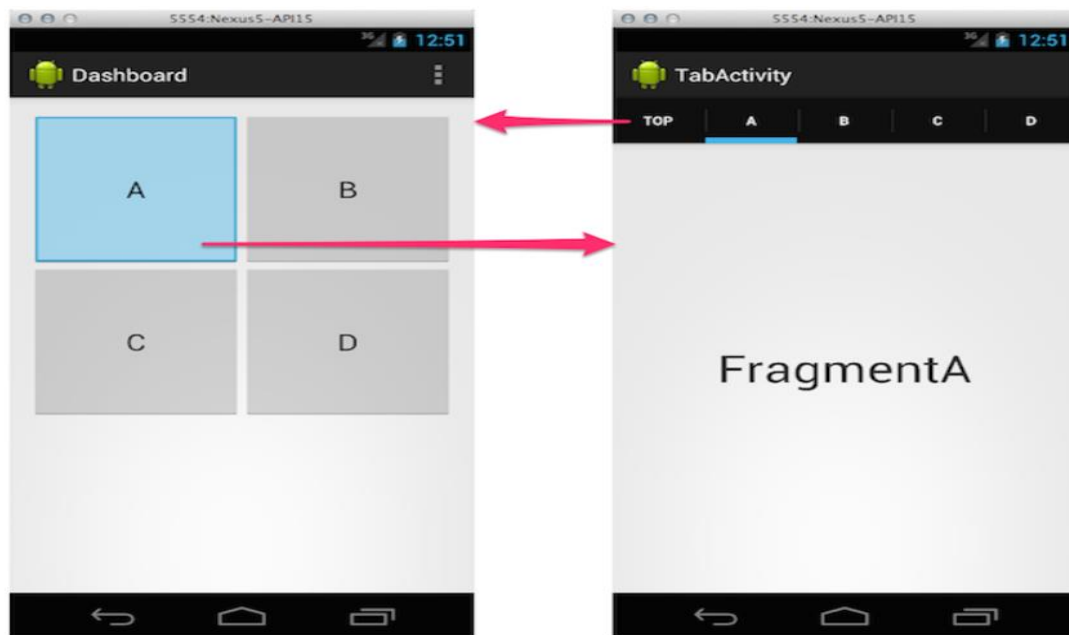
```
public class MyReceiver extends BroadcastReceiver {  
    public void onReceive(context,intent){  
    }  
}
```

Εικόνα 12 : Η σύνταξη του Broadcast Receiver Component σε Android Studio

Υπάρχουν ωστόσο και ορισμένα πρόσθετα συστατικά τα οποία θα χρησιμοποιηθούν στην κατασκευή μιας εφαρμογής Android. Αυτά τα συστατικά παρουσιάζονται παρακάτω:

- **Fragments**

Ουσιαστικά αντιπροσωπεύει ένα τμήμα της διεπαφής του χρήστη σε μία δραστηριότητα.



Εικόνα 13: Παράδειγμα για fragment

- **Views**

Στοιχεία User Interface που έχουν συνταχθεί στην οθόνη συμπεριλαμβανομένων των πλήκτρων κ.τ.λ.

- **Layout**

Διάταξη των στοιχείων στην οθόνη και εμφάνιση των views ανάλογα με το layout. Πρόκειται ουσιαστικά για την σκηνική διάταξη, για το πως θα φαίνεται το UI και είναι γραμμένο σε XML γλώσσα.

- **Intents**

Τα Intents είναι από τα πιο σημαντικά συστατικά της πλατφόρμας και περιγράφουν ένα μήνυμα συστήματος. Τα μηνύματα αυτά παράγονται και κυκλοφορούν στο σύστημα καθ' όλη τη χρονική περίοδο κατά την οποία η συσκευή βρίσκεται σε λειτουργία και ενημερώνουν τις εφαρμογές για κάθε συμβάν. Για παράδειγμα ένα τέτοιο συμβάν θα μπορούσε να είναι η ενεργοποίηση του GPS, των δεδομένων, του Wi-Fi ή του Bluetooth. Με λίγα λόγια είναι μηνύματα και συστατικά μαζί.

- **Manifest**

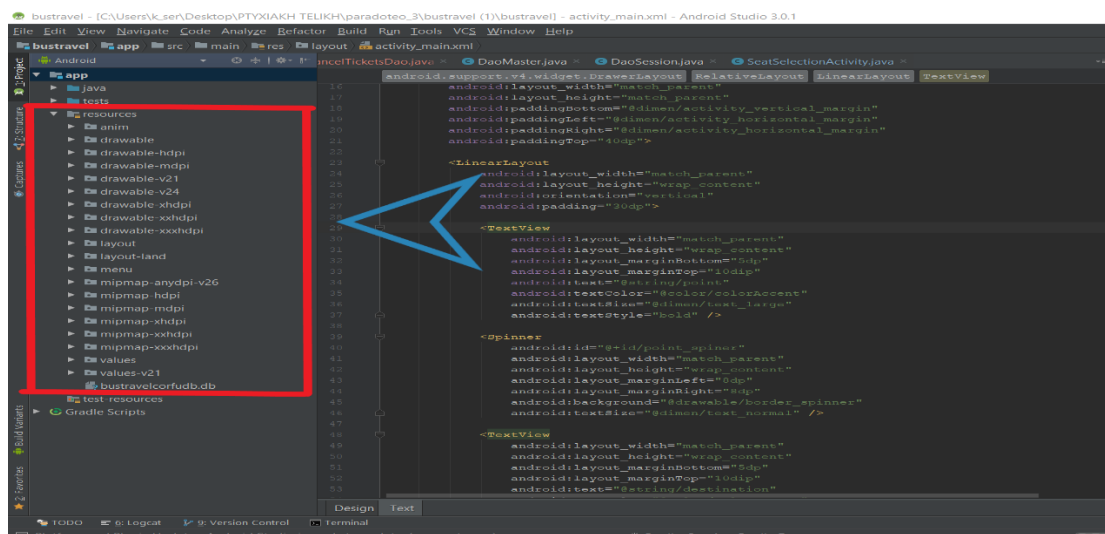
Κάθε Android project περιλαμβάνει ένα αρχείο manifest, το AndroidManifest.xml, που αποθηκεύεται στην κορυφή της ιεραρχίας των αρχείων στο project. Το manifest ορίζει τη δομή και τα μεταδεδομένα της εφαρμογής, τα συστατικά της και τις απαιτήσεις της. Περιλαμβάνει κόμβους για κάθε μια από τις Activities, Services, Content Providers και Broadcast Receivers που αποτελούν μια εφαρμογή και χρησιμοποιώντας τα Intent Filters και τα Permissions καθορίζει πως θα αλληλεπιδρούν μεταξύ τους και με άλλες εφαρμογές. Το manifest μπορεί επίσης να καθορίσει τα μεταδεδομένα της εφαρμογής (όπως τα icons, τον αριθμό της έκδοσης ή το theme) και επιπρόσθετους κόμβους στο top-level που ορίζουν τα απαιτούμενα permissions, τα unit tests ενώ ορίζονται και τις απαιτήσεις ως προς το υλικό, την οθόνη και την πλατφόρμα. Με λίγα λόγια το αρχείο manifest προκειμένου να ορίσει τα συστατικά των εφαρμογών κάνει τα εξής:

- Ορίζει το ελάχιστο API που απαιτείται από την εφαρμογή
- Προσδιορίζει κάθε άδεια του χρήστη που απαιτεί η εφαρμογή. Για παράδειγμα πρόσβαση στο διαδίκτυο ή ενεργοποίηση του Bluetooth
- Ορίζει τα χαρακτηριστικά που απαιτούνται από την εφαρμογή (κάμερα, Bluetooth, GPS κ.α.)
- Δηλώνει τις API βιβλιοθήκες με τις οποίες η εφαρμογή χρειάζεται να συνδεθεί.

Πρόκειται ουσιαστικά για το αρχείο ρυθμίσεων της εφαρμογής.

- **Resources**

Μία εφαρμογή δεν είναι μόνο κώδικας αλλά και διάφορα εξωτερικά στοιχεία όπως drawable εικόνες , strings , διάφοροι ήχοι-audio και οτιδήποτε άλλο σχετικό με την εμφάνιση της. Για παράδειγμα ο προγραμματιστής πρέπει να ορίσει animations, menus, styles, χρώματα και την εμφάνιση της activity με XML αρχεία. Χρησιμοποιώντας application resources είναι εύκολο να αναβαθμιστούν πολλά χαρακτηριστικά μιας εφαρμογής χωρίς να τροποποιηθεί ο κώδικας. Όλα τα resources γίνονται compile μέσα στο application και αποθηκεύονται στον φάκελο res μέσα στο project. Υπάρχουν ξεχωριστοί φάκελοι για τις εικόνες, τα strings και τα xml.



Εικόνα 14 : Παρουσίαση των Resources

2.5. Ο κύκλος ζωής μιας Activity

Οι activities στο σύστημα, διαχειρίζονται υπό τη μορφή στοίβας. Όταν μια activity ξεκινήσει, τότε τοποθετείται στην κορυφή της στοίβας και γίνεται η τρέχουσα activity και η προηγούμενη activity παραμένει στην από κάτω θέση και έρχεται στο προσκήνιο μόνο αν η καινούρια activity τελειώσει. Μια activity μπορεί ουσιαστικά να βρεθεί σε τέσσερις καταστάσεις:

- Αν μια διεργασία βρίσκεται στο προσκήνιο, εμφανίζεται δηλαδή στην οθόνη (είναι στην κορυφή της στοίβας), τότε βρίσκεται στην κατάσταση active ή running.
- Αν μια activity βρίσκεται στο background της εφαρμογής αλλά είναι ακόμα ορατή (δηλαδή μια νέα activity που δεν είναι full-sized είναι focused πάνω από την πρώτη activity), τότε βρίσκεται στην κατάσταση paused. Σε αυτήν την κατάσταση η activity είναι ζωντανή, αλλά μπορεί το σύστημα να την κλείσει σε περίπτωση χαμηλής μνήμης.
- Αν μια activity είναι πλήρως κρυμμένη από άλλη activity, τότε βρίσκεται στην κατάσταση stopped. Διατηρεί όλη την κατάσταση και τις πληροφορίες, ωστόσο δεν είναι πλέον ορατή στον χρήστη, έτσι το παράθυρό της είναι κρυμμένο και συνήθως κλείνει από το σύστημα όταν η υπάρχει ανάγκη για μνήμη.
- Αν κάποια διεργασία είναι στην κατάσταση paused ή stopped, το σύστημα μπορεί να την απομακρύνει από την μνήμη, είτε ζητώντας άδεια για να το κάνει είτε από μόνο του. Όταν ξαναεμφανιστεί στο χρήστη, πρέπει να εκκινήσει πάλι και να αποκαταστήσει την προηγούμενη της κατάσταση.

Το διάγραμμα που ακολουθεί δείχνει τα σημαντικά μονοπάτια κατάστασης μιας Activity. Τα τετραγωνισμένα παραλληλόγραμμα δείχνουν τις callback μεθόδους που μπορούν να εφαρμοστούν για τις διάφορες διεργασίες όταν μια Activity κινείται μεταξύ διαφόρων καταστάσεων. Τα χρωματιστά στρογγυλεμένα παραλληλόγραμμα δείχνουν τις κύριες καταστάσεις στις οποίες μπορεί να βρεθεί μια Activity.

2.6. Οι εκδόσεις του Android

Η πρώτη έκδοση του λειτουργικού συστήματος Android, ήταν η έκδοση Android beta η οποία κυκλοφόρησε το Νοέμβριο του 2007. Ένα χρόνο αργότερα σχεδόν κυκλοφορεί και η πρώτη εμπορική έκδοση, η Android 1.0. Από τότε μέχρι σήμερα γίνονται συνεχείς ενημερώσεις στο λειτουργικό σύστημα Android με αποτέλεσμα να έχουν κυκλοφορήσει πολλές εκδόσεις, τις οποίες θα παρουσιάσουμε παρακάτω.



Εικόνα 16 : Όλες οι εκδόσεις του Android.

Από τον Απρίλιο του 2009, οι εκδόσεις του Android έχουν θέμα από την ζαχαροπλαστική στην κωδική ονομασία τους, και κυκλοφόρησαν σε αλφαβητική σειρά, εξαιρουμένων των εκδόσεων 1.0 και 1.1, που δεν τέθηκαν υπό συγκεκριμένα κωδικά ονόματα. Παρακάτω βλέπουμε έναν πίνακα για να κατανοήσουμε την εξέλιξη πιο αναλυτικά.

Κωδικό όνομα	Νούμερο έκδοσης	Ημερομηνία αρχικής κυκλοφορίας	Επίπεδο API
N/A	1.0	23 Σεπτεμβρίου 2008	1
	1.1	9 Φεβρουάριου 2009	2
Cupcake	1.5	27 Απριλίου 2009	3
Donut	1.6	15 Σεπτεμβρίου 2009	4
Eclair	2.0 – 2.1	26 Οκτωβρίου 2009	5–7
Froyo	2.2 – 2.2.3	20 Μαΐου 2010	8
Gingerbread	2.3 – 2.3.7	6 Δεκεμβρίου 2010	9–10
Honeycomb	3.0 – 3.2.6	22 Φεβρουάριου 2011	11–13
Ice Cream Sandwich	4.0 – 4.0.4	18 Οκτωβρίου 2011	14–15
Jelly Bean	4.1 – 4.3.1	9 Ιουλίου 2012	16–18
KitKat	4.4 – 4.4.4	31 Οκτωβρίου 2013	19–20
Lollipop	5.0 – 5.1.1	12 Νοεμβρίου 2014	21–22
Marshmallow	6.0 – 6.0.1	5 Οκτωβρίου 2015	23
Nougat	7.0 - 7.1.1	22 Αυγούστου 2016	24
Oreo	8.0	21 Αυγούστου 2017	26

Εικόνα 17 : Όλες οι εκδόσεις του λειτουργικού Android.

2.6.1 Android 1.0(API Level 1 Android Version 1.0)

Το Android σχεδιάστηκε αρχικά ως μια διεπαφή αφής για κάμερες. Η πρώτη εμπορικά διαθέσιμη έκδοση για κινητά τηλέφωνα κυκλοφόρησε τον Σεπτέμβριο του 2008 από την Google στο HTC Dream handset. Τα ονόματα με "γεύση καραμελών" δεν είχαν ακόμα εισαχθεί, όποτε αυτή η έκδοση ονομάστηκε τυπικά ως Android 1.0 όπως βλέπουμε και στην εικόνα 16.

Τα χαρακτηριστικά του ήταν τα βασικά τα όποια ορισμένα εξακολουθούν να υπάρχουν και σήμερα.

- Υπήρχε ένα απλό πρόγραμμα περιήγησης για το διαδίκτυο
- Παρείχε υποστήριξη για Gmail
- Ημερολόγιο Google
- Χάρτες Google
- YouTube
- Καθώς και οι εφαρμογές Messenger και Google Talk

2.6.2 Android 1.1(API Level 2 Android Version 1.1)

Η έκδοση Android 1.1 κυκλοφόρησε στις 9 Φεβρουάριου 2009. Έλυσε αρκετά σφάλματα της πρώτης έκδοσης και πρόσθεσε νέα χαρακτηριστικά:

- Λεπτομερή σχόλια για επιχειρήσεις στους χάρτες.
- Δυνατότητα αποθήκευση συνημμένων αρχείων.
- Υποστήριξη για μαρκίζα σε διατάξεις συστήματος.

2.6.3 Android Cupcake (API Level 3 Android Version 1.5)

Λίγους μήνες αργότερα ανακοινώθηκε η νέα βελτιωμένη έκδοση 1.5. η οποία ήταν η πρώτη που πήρε την κωδική ονομασία γλυκού και βασίστηκε στον Linux Kernel 2.6.27.

Το android cupcake έφερε τα εξής χαρακτηριστικά:

- Υποστήριξη widget
- Δυνατότητα αυτόματης περιστροφής οθόνης
- Δυνατότητα αγαπημένων με εικόνα στις επαφές
- Υποστήριξη εγγραφής και αναπαραγωγής βίντεο σε μορφή 3GP και MPEG4
- Αυτόματη υποστήριξη προτύπου Bluetooth A2DP
- Προστέθηκε κινούμενη εικόνα κατά την εκκίνηση
- Δυνατότητα ανεβάσματος βίντεο στο YouTube
- Νέο γραφικό περιβάλλον- menu
- Υποστήριξη έξυπνων πληκτρολογίων με πρόβλεψη και αυτόματο ορθογραφικό έλεγχο



Εικόνα 18 : Android Cupcake

2.6.4 Android Donat (API Level 4 Android Version 1.6)

Το Σεπτέμβριο του 2009 έρχεται ο διάδοχος του Cupcake, με το όνομα Donut. Όπως παρατηρούμε οι ενημερώσεις παίρνουν το όνομά τους από διάφορα γλυκά εδέσματα με αλφαβητική σειρά. Το Donut μας έδωσε μια βελτιωμένη λειτουργικότητα αναζήτησης, πιο γρήγορη εμπειρία και υποστήριξη για οθόνες 480x800 pixel. Επίσης ορισμένα χαρακτηριστικά που προστέθηκαν είναι :

- Υποστήριξη πολλαπλών αρχείων ταυτόχρονα
- Ανανεωμένη φωνητική αναζήτηση
- Βελτιωμένο Android Market Place με την ικανότητα ο χρήστης να βλέπει και στιγμιότυπα εφαρμογών
- Μηχανή μετατροπής κειμένου σε ομιλία
- Εικονικό πληκτρολόγιο αφής
- Stereo Bluetooth
- Αντιγραφή/ Επικόλληση και αναζήτηση στον browser
- Άμεση μεταμόρφωση στο YouTube



Εικόνα 19 : Android Donut

2.6.5 Android Éclair (API Level 5 & 6 & 7 Android Versions 2.0 & 2.1)

Τον Οκτώβριο του 2009 έκανε την εμφάνιση του μία νέα έκδοση του Android με εξέλιξη Level στο 2.0. Μόλις τρεις μήνες αργότερα, τον Ιανουάριο παρουσιάστηκε η έκδοση 2.1. Οι αλλαγές στα χαρακτηριστικά παρουσιάζονται παρακάτω. Αξίζει να σημειωθεί ότι η 2.1 δεν είχε νέα χαρακτηριστικά πάρα διορθώσεις σε σφάλματα.

- Πολλαπλούς λογαριασμούς χρήστη
- Υποστήριξη λήψης και αποστολής email
- Βελτιωμένη η εφαρμογή Google Maps 3.1.2
- Ανεπτυγμένη κάμερα με χρήση flash και ψηφιακό zoom
- Bluetooth 2.1
- Ανεπτυγμένο πληκτρολόγιο με προσαρμόσιμο λεξικό και προτάσεις από ονόματα επαφών
- Live wallpapers
- Ομιλία σε κείμενο (Speech to text) σε οποιοδήποτε πεδίο κειμένου



Εικόνα 20 : Android Éclair

2.6.6 Android Froyo (API Level 8 Android Versions 2.2 & 2.2.3)

Η επόμενη έκδοση ονομάστηκε Froyo και παρουσιάστηκε στις 20 Μαΐου του 2010. Η έκδοση αυτή περιλάμβανε ορισμένες βελτιστοποιήσεις κυρίως στα παρακάτω:

- Βελτιωμένη μνήμη
- Βελτιωμένη ταχύτητα
- Βελτιωμένη απόδοση
- Ενσωμάτωση μηχανισμού JavaScript στο Chrome v8
- Πληκτρολόγιο σε πολλές γλώσσες
- Ανανέωση Android Market
- Ανεπτυγμένη υποστήριξη Microsoft Exchange: Πολιτικές Ασφαλείας, Global Διευθύνσεις, συγχρονισμός ημερολογίου, απομακρυσμένη διαγραφή και καθαρισμός
- Δυνατότητα μετατροπής κινητής συσκευής σε Wi-Fi hotspot
- Υποστήριξη σύνδεσης USB
- Υποστήριξη Bluetooth για φωνητική πληκτρολόγηση και διαμοιρασμό επαφών(contact sharing)
- Υποστήριξη του Flash σε προγράμματα πλοήγησης



Εικόνα 21: Android Froyo

2.6.7 Android Gingerbread (API Level 9 & 10 Android Versions 2.3 & 2.3.7)

Το Android 2.3 γνωστό και ως Gingerbread παρουσιάστηκε τον Δεκέμβριο του 2010(με την ενημέρωσή του να έρχεται τον Φεβρουάριο του 2011). Στην έκδοση 2.3.7 υπάρχουν οι παρακάτω αλλαγές:

- Αλλαγή στην διεπαφή χρήστη , μετατράπηκε σε απλούστερη και ταχύτερη
- Υποστήριξη οθονών μεγαλύτερου μεγέθους και ανάλυσης
- Καλύτερη διαχείριση ισχύος(κυρίως μπαταρία)
- Δυνατότητα αντιγραφής/επικόλλησης σε όλο το σύστημα και όχι μόνο στην εφαρμογή
- Υποστήριξη NFC (Near Field Communication) , το οποίο είναι ένα πρότυπο ασύρματων επικοινωνιών μικρής απόστασης , που επιτρέπει την επικοινωνία ανάμεσα σε δύο συσκευές
- Συντόμευση Manage Applications όπου μέσα στο μενού Options της αρχικής οθόνης επιτρέπει στους χρήστες να προβάλλουν όλες τις εφαρμογές που εκτελούνται.
- Ύπαρξη πολλαπλών καμερών



Εικόνα 22: Android Έκδοση Gingerbread

2.6.8 Android Honeycomb (API Level 11 & 12 & 13 Android Versions 3.0 & 3.1-3.2.6)

Η έκδοση Honeycomb παρουσιάστηκε στις 22 Φεβρουαρίου του 2011 έχοντας ως διαφορά από τις προηγούμενες εκδόσεις το ότι προοριζόταν αποκλειστικά για tablets. Οι αλλαγές που έγιναν στην έκδοση αυτή έχουν να κάνουν κυρίως με τη βελτίωση της υποστήριξης των συσκευών αυτών. Κάποια λοιπόν από τα επιπλέον χαρακτηριστικά που προστέθηκαν ήταν τα εξής:

- Υποστήριξη γραφικών 3D
- Ιδιωτική πλοήγηση - Private Browsing
- Δυνατότητα Video Chat με την χρήση της υπηρεσίας Google Talk
- Bluetooth Tethering
- Υποστήριξη σύνδεσης περιφερειακών συσκευών όπως πληκτρολόγια και Gamepads
- Παραμετροποιήσιμα Widgets
- Δυνατότητα προβολής εικόνων από το άλμπουμ σε πλήρη οθόνη
- Διορθώσεις σφαλμάτων και σταθερότητας συστήματος

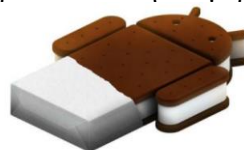


Εικόνα 23: Android Έκδοση Honeycomb

2.6.9 Android Ice-cream Sandwich (API Level 14 & 15 Android Versions 4.0-4.0.4)

Η έκδοση Ice Cream Sandwich παρουσιάστηκε στις 19 Οκτωβρίου του 2011. Για άλλη φορά έχει βελτιωθεί η ταχύτητα και η απόδοση του συστήματος. Πλέον στη διεπαφή χρήστη, η οποία διαφέρει για ακόμη μια φορά από τις προηγούμενες εκδόσεις, υπάρχουν εικονικά πλήκτρα τα οποία παίρνουν τη θέση των φυσικών ή αφής που υπήρχαν μέχρι στιγμής. Οι αλλαγές που έφερε είναι πολλές όπως θα δούμε παρακάτω:

- Δημιουργία ενός φακέλου με εφαρμογές, τραβώντας απλά την μία εφαρμογή πάνω στην άλλη
- Επανασχεδιασμός του Gmail
- Υποστήριξη μέχρι και 16 καρτέλες στο Google Chrome
- Ξεκλείδωμα οθόνης με αναγνώριση προσώπου
- Διορθώσεις σφαλμάτων πληκτρολογίου
- Μετρητής δεδομένων κινητής τηλεφωνίας
- Δυνατότητα τερματισμού λειτουργίας για εφαρμογές που εξαντλούν μεγάλη ισχύ
- Προσθετές λειτουργίες στην κάμερα
- Δυνατότητα αναγνώρισης προσώπου στην κάμερα και βελτίωση του άλμπουμ



**Android SDK
4.0.3 (Ice Cream Sandwich)**

Εικόνα 24: Android Έκδοση Ice Cream Sandwich

2.6.10 Android Jellybean (API Level 16 & 17 & 18 Android Versions 4.1 & 4.2 & 4.3)

Η έκδοση 4.1 ανακοινώθηκε στις 27 Ιουνίου του 2012 και ονομάστηκε Jelly Bean. Αποτελεί για πολλούς χρήστες, ίσως, η καλύτερη έκδοση του Android μέχρι σήμερα. Λίγους μήνες αργότερα, και συγκεκριμένα στις 29 Οκτωβρίου ακολούθησε η έκδοση 4.2, με την 4.3 να ανακοινώνεται στις 24 Ιουλίου του 2013. Τα κυριότερα χαρακτηριστικά και στις τρεις εκδόσεις ήταν η καλύτερη επίδοση που παρουσίασαν οι συσκευές στην ταχύτητα και στην λειτουργικότητα. Ορισμένα χαρακτηριστικά του είναι τα εξής:

- Βελτιωμένες λειτουργίες στις λήψεις φωτογραφιών
- Υποστήριξη τεχνολογίας Virtual Surround στον ήχο
- Ξεκλείδωμα με αναγνώριση προσώπου
- Βελτίωση απόδοσης της GPU και επίδοσης γραφικών παιχνιδιού
- Ευέλικτες ειδοποιήσεις
- Υποστήριξη διαφορετικών χρηστών
- Ενσωμάτωση υπηρεσίας ασφαλείας δεδομένων
- Βελτίωση στα widgets
- Ασύρματη μετάδοση εικόνας
- Προσδιορισμός θέσης μέσω Wi-Fi
- Ασύρματη μετάδοση ήχου
- Έξυπνο Bluetooth για επικοινωνία με έξυπνα περιφερειακά
- Βελτίωση NFC
- Υποστήριξη 4K
- Μετατροπή του Market Place σε Google Play Store
- Λειτουργίες προσβασιμότητας όπως (τριπλό χτύπημα οθόνης για μεγέθυνση, zoom με δυο δάκτυλα, φωνητική πλοήγηση για τυφλούς)
- Δυνατότητα εισαγωγής μπάρας γρήγορων ρυθμίσεων στο πάνελ ειδοποιήσεων για πιο εύκολη πρόσβαση στο Wi-Fi
- Δυνατότητα λειτουργίας Daydream

Η έκδοση Jelly Bean εφαρμόστηκε για πρώτη φορά στο tablet Google Nexus 7 ενώ η έκδοση Android 4.2 πρωτοεμφανίστηκε στα Nexus 4 και Nexus 10.



Android 4.1 Jelly Bean

Εικόνα 25: Android Έκδοση Jelly Bean

2.6.11 Android KitKat (API Level 19 & 20 Android Version 4.4)

Η έκδοση KitKat ανακοινώθηκε στις 3 Σεπτέμβρη του 2013. Το KitKat έκανε το ντεμπούτο του στο Nexus 5 της Google και έχει υποστεί βελτίωση ώστε να τρέχει σε μεγαλύτερο εύρος συσκευών από τις προηγούμενες εκδόσεις του Android. Συγκεκριμένα υπάρχει μία συνολική βελτίωση της απόδοσης των προεγκατεστημένων εφαρμογών αλλά και των εφαρμογών της Google, ενώ η κατάργηση διεργασιών που δεν έκαναν ουσιαστικά τίποτα, έχουν μειώσει το συνολικό αποτύπωμα του λειτουργικού συστήματος στη RAM. Αποτέλεσμα όλων αυτών είναι το Android να τρέχει καλύτερα σε συσκευές με 512MB RAM.

Κάποια από τα επί μέρους χαρακτηριστικά που συναντάμε είναι:

- Πληρωμές μέσω NFC
- Μεγαλύτερη Ασφάλεια
- Βελτίωση της μπαταρίας
- Προσθήκης βηματομετρητή
- Βελτίωση του Bluetooth
- Βελτίωση της διαχείρισης του Cloud
- Βελτίωση περιήγησης του περιηγητή διαδικτύου
- Εισαγωγή print screen στην οθόνη της συσκευής
- Νέα λειτουργία immersive mode όπου οι εφαρμογές μπορούν να κρύβουν την πλοήγηση και την μπάρα κατάστασης
- Ευκολότερη εκτύπωση με το Google Cloud Print
- Hangouts



Εικόνα 26 : Android Έκδοση KitKat

2.6.12 Android Lollipop (API Level 21 & 22 Android Versions 5.0-5.1.1)

Η έκδοση Lollipop ανακοινώθηκε στις 25 Ιουνίου του 2014 και έγινε διαθέσιμη στις 12 Νοεμβρίου του 2014 για συσκευές που τρέχουν το λογισμικό Android που διατίθενται απευθείας από την Google, συμπεριλαμβανομένου και των Nexus και Google Play συσκευές. Η έκδοση Lollipop ως βασική αλλαγή θα παρουσιάσει ένα επανασχεδιασμένο περιβάλλον εργασίας χρήστη χτισμένο γύρω από μία διαδραστική σχεδιαστική γλώσσα που αποκαλείται «material design». Επιπλέον, η Google έκανε εσωτερικές αλλαγές στην πλατφόρμα, με το Android Runtime (ART) να αντικαθιστά και επίσημα το Dalvik για βελτιωμένη απόδοση των εφαρμογών, καθώς και με τις αλλαγές που αποσκοπούν στη βελτίωση και τη βελτιστοποίηση της χρήσης της μπαταρίας.

Η νέα έκδοση στη συνέχεια του Lollipop 5.1 φέρνει σημαντικές αλλαγές όπως:

- Silent mode
- Βελτίωση στη σταθερότητα του συστήματος
- Καλύτερη διαχείριση της RAM
- Διορθώσεις σε κάποια apps τα οποία κλείνουν ξαφνικά
- Διορθώσεις σε προβλήματα με τη σύνδεση σε ασύρματα δίκτυα
- Βελτιώσεις στην λειτουργία του Okay Google,
- Βελτιώσεις στα Notifications



Android 5.0, Lollipop

Εικόνα 27: Android Έκδοση Lollipop

2.6.12 Android Marshmallow (API Level 23 Android Versions 6.0-6.0.1)

Η έκδοση που διαδέχτηκε το Lollipop ήταν το Marshmallow το οποίο ανακοινώθηκε στις 28 Μαΐου του 2015 και οι ενημερώσεις αφορούσαν κυρίως τα κινητά Nexus 5 και Nexus 6, το tablet Nexus 9 και την συσκευή Nexus Player. Έγινε διαθέσιμη στην αγορά στις 5 Οκτωβρίου του 2015 και περιλάμβανε μεταξύ άλλων :

- Ανάγνωση δακτυλικών αποτυπωμάτων
- Υποστήριξη USB type-C
- Λειτουργία οθόνης 4K
- Android Pay
- Παρακολούθηση Συγχρονισμού Συσκευής
- Καλύτερες επιδόσεις σε λειτουργίες κειμένου

Το Marshmallow επικεντρώνεται κυρίως στη βελτίωση της συνολικής εμπειρίας του χρήστη σε σύγκριση με τον προκάτοχο του. Εισάγει μια νέα αρχιτεκτονική στα δικαιώματα των εφαρμογών. Οι άδειες σε εφαρμογές - όπως για χρήση του μικροφώνου, της κάμερας, της περιοχής, του τηλεφώνου, των μηνυμάτων, του ημερολογίου που είχαν προηγουμένως σοβαρά κενά ασφαλείας στις προηγούμενες συσκευές.



**Android 6.0
Marshmallow**

Εικόνα 28 : Android Έκδοση Marshmallow

2.6.12 Android Nougat (API Level 24 & 25 Android Versions 7.0-7.1.2)

Η έκδοση Nougat ανακοινώθηκε στις 9 Μαρτίου του 2016 και έγινε διαθέσιμη στην αγορά στις 22 Αυγούστου του 2016. Λίγους μήνες αργότερα βγήκε και η ενημέρωση του, η 7.1 η οποία είχε ορισμένες βελτιώσεις στο λειτουργικό σύστημα και την πλατφόρμα ανάπτυξης του. Οι σημαντικότερες αλλαγές που έφερε η συγκεκριμένη έκδοση είναι τα παρακάτω:

- Εισαγωγή κινούμενων σχεδίων και emoji
- Δυνατότητα ταυτόχρονης λειτουργίας εφαρμογών στην ίδια οθόνη
- Γρήγορη εναλλαγή εφαρμογών
- Εισαγωγή ενός API το οποίο προσφέρει καλύτερα γραφικά στα παιχνίδια (Vulkan)
- Daydream
- Άμεσες απαντήσεις στις ειδοποιήσεις των εφαρμογών , κυρίως messenger
- Εξοικονόμηση δεδομένων
- Έλεγχος ειδοποιήσεων
- Ο έλεγχος και η εγκατάσταση των ενημερώσεων γίνεται στο παρασκήνιο
- Έλεγχος μεγέθους προβολής, δίνει την δυνατότητα να κάνουμε τα γράμματα ή τα αντικείμενα μεγαλύτερα στην οθόνη
- Κρυπτογράφηση ανά αρχείο
- Βελτίωση στον χρόνο εκκίνησης μιας συσκευής και με μεγαλύτερη ασφάλεια



Εικόνα 29 : Android Έκδοση Nougat

2.6.13 Android O (API Level 26 Android Version 8.0)

Τον Μάρτιο του 2017 η Google παρουσίασε την έκδοση Android Oreo και παρουσίασε ορισμένα χαρακτηριστικά που θα διαθέτει η έκδοση αυτή.

- Βελτίωση στην διάρκεια ζωής της μπαταρίας
- Smart Text Selection το οποίο θα βελτιώσει την διαδικασία copy & paste
- Εισαγωγή feature με όνομα Google Play Protect το οποίο θα εμφανιστεί στο Play Store και θα είναι υπεύθυνο για να σκανάρει τις εφαρμογές για κακόβουλες συμπεριφορές
- Εισαγωγή manual scan προκειμένου να σκανάρει ο χρήστης οποιαδήποτε στιγμή επιθυμεί την συσκευή του
- Εισαγωγή του Find my Device το οποίο θα δίνει την δυνατότητα να βλέπουμε την διεύθυνση που βρίσκεται το κινητό μας
- Autofill API προκειμένου να γίνονται ευκολά login
- Snooze-able για τις ειδοποιήσεις
- Το Picture-in-Picture Mode θα κάνει αυτό που γίνεται με το YouTube app όταν πατάμε ένα video να παίζει και στη συνέχεια θέλουμε να πάμε πίσω στο OS. Το video θα συνεχίσει να παίζει, ενώ ο χρήστης θα κάνει άλλα πράγματα στο κινητό.
- Όταν έρχεται κάποια ειδοποίηση για συγκεκριμένο app, αυτό θα γίνεται εμφανές στο εικονίδιο του.
- Βελτίωση του boot time, το οποίο είναι δύο φορές ταχύτερο σε σχέση με τις προηγούμενες εκδόσεις.



Εικόνα 30 : Android Έκδοση Oreo

3. Android Studio

Το Android Studio αποτελεί ένα ολοκληρωμένο προγραμματιστικό περιβάλλον για ανάπτυξη εφαρμογών. Ανακοινώθηκε στις 16 Μαΐου το 2013 σε συνέδριο της Google και έκτοτε χρησιμοποιείτε αποκλειστικά για εφαρμογές android. Η τελευταία του έκδοση ήταν η έκδοση Oreo 8.0. Το Android Studio είναι ένα ελεύθερο λογισμικό με την άδεια Apache License 2.0. Είναι βασισμένο στο λογισμικό της JetBrains IntelliJ IDEA, και αντικατέστησε το Eclipse Android Development Tools (ADT) ως το κύριο IDE της Google για ανάπτυξη εφαρμογών Android.

1.1. Εγκατάσταση Λογισμικού

Το Android Studio όπως αναφέραμε και παραπάνω αποτελεί το βασικό εργαλείο ανάπτυξης εφαρμογών Android και διατίθεται δωρεάν από την ιστοσελίδα της Google και συγκεκριμένα μπορούμε να το βρούμε στην συγκεκριμένη διεύθυνση :

- <https://developer.android.com/studio/index.html>

Οποιοσδήποτε χρήστης μπορεί να κατεβάσει την πλατφόρμα και να την εγκαταστήσει εύκολα και απλά. Παρακάτω θα παρουσιάσουμε τα βήματα εγκατάστασης της πλατφόρμας. Ως ελάχιστη προϋπόθεση για την εγκατάσταση της πλατφόρμας αποτελεί η ύπαρξη στον Η/Υ μας μία από τα παρακάτω λειτουργικά συστήματα ή κάποιο νεότερο από αυτά.

- Microsoft® Windows® 8/7/Vista/2003 (32 or 64-bit).
- Mac® OS X® 10.8.5 ή πάνω από 10.9 (Mavericks)
- GNOME ή KDE desktop

Έπειτα προκειμένου να δουλέψει η πλατφόρμα είναι απαραίτητο να εγκατασταθεί και το πακέτο Java, το οποίο διατίθεται και αυτό δωρεάν από της σελίδα της Oracle.

- <https://www.oracle.com/technetwork/java/javase/downloads/index.html>

Το πακέτο Java που θα κάνουμε λήψη θα πρέπει να πληροί τα παρακάτω χαρακτηριστικά :

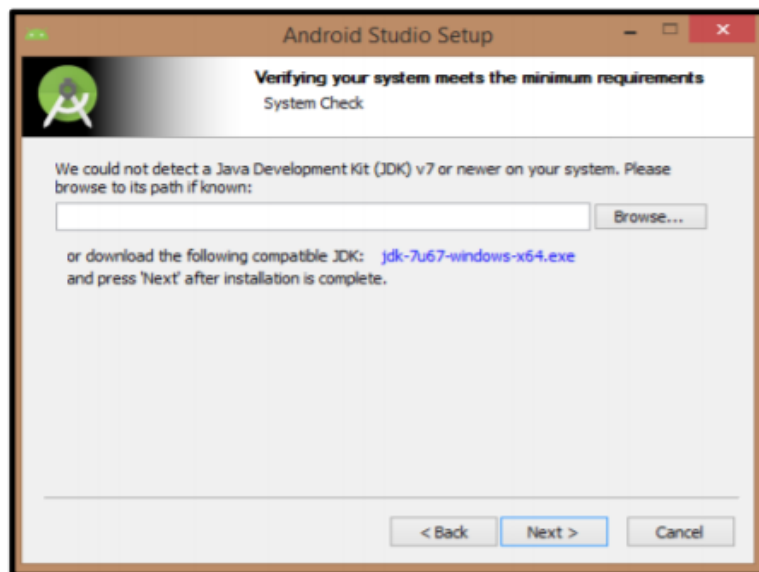
- Java JDK5 or later version
- Java Runtime Environment (JRE) 6
- Android Studio

Αρχικά κάνουμε λήψη το αρχείο Android studio.exe από την σελίδα του Android Studio στον προσωπικό μας υπολογιστή και πατάμε κλικ στο κουμπί εγκατάσταση.



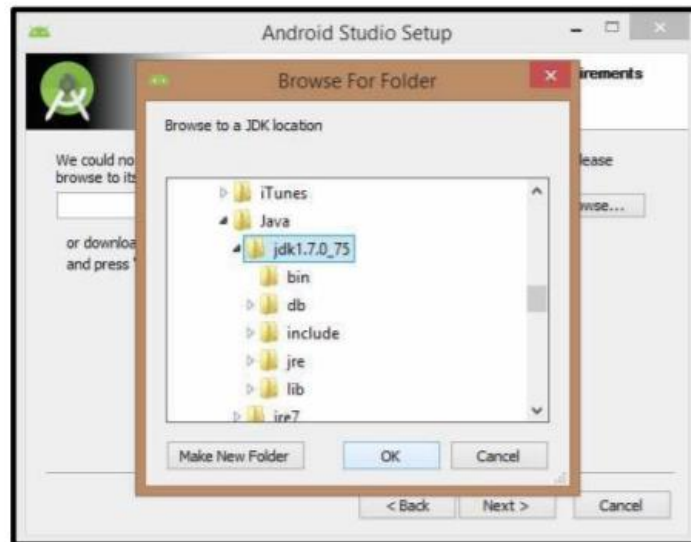
Εικόνα 31: Διαδικασία εγκατάστασης του Android Studio(http://www.tutorialspoint.com/android/android_studio.htm)

Στην συνέχεια από την παραπάνω εικόνα πατάμε το Next (Επόμενο) προκειμένου να οδηγηθούμε στο παρακάτω παράθυρο.



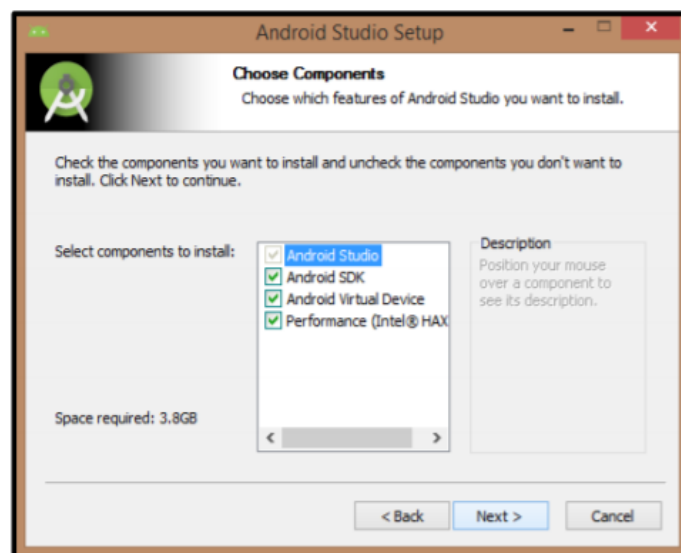
Εικόνα 32 : Διαδικασία επιλογής του αρχείου java

Στην παραπάνω περίπτωση πρέπει ο χρήστης να επιλέξει την έκδοση java που εγκατέστησε, σε περίπτωση που δεν έχει κάνει λήψη κάποια έκδοση, θα πρέπει να την κάνει λήψη προκειμένου να συνεχίσει παρακάτω. Επιλέγοντας το κουμπί της αναζήτησης (Browser) βρίσκουμε το αρχείο Jdk που έχουμε κάνει λήψη από την oracle και πατάμε το κουμπί ok (βλ. εικόνα 33).



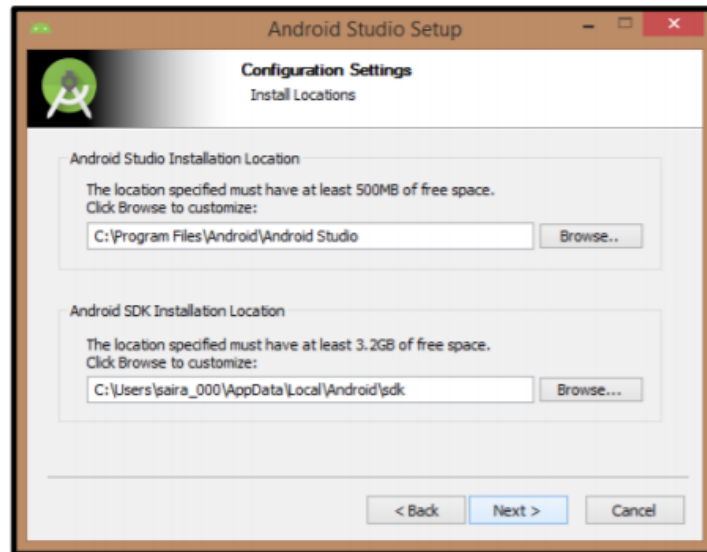
Εικόνα 33 : Επιλογή του αρχείου java

Στην συνέχεια οδηγούμαστε στην παρακάτω εικόνα και πατάμε το κουμπί Next(Επόμενο) .



Εικόνα 34 : Επιλογή παραμέτρων

Στην παραπάνω περίπτωση επιλέγουμε καθορίζουμε τα χαρακτηριστικά που θα έχει η πλατφόρμα μας. Αφού επιλέξουμε τα χαρακτηριστικά αυτά επιλέγουμε το κουμπί Next (Επόμενο) και μεταφερόμαστε στο παράθυρο επιλογής της διαδρομής εγκατάστασης.



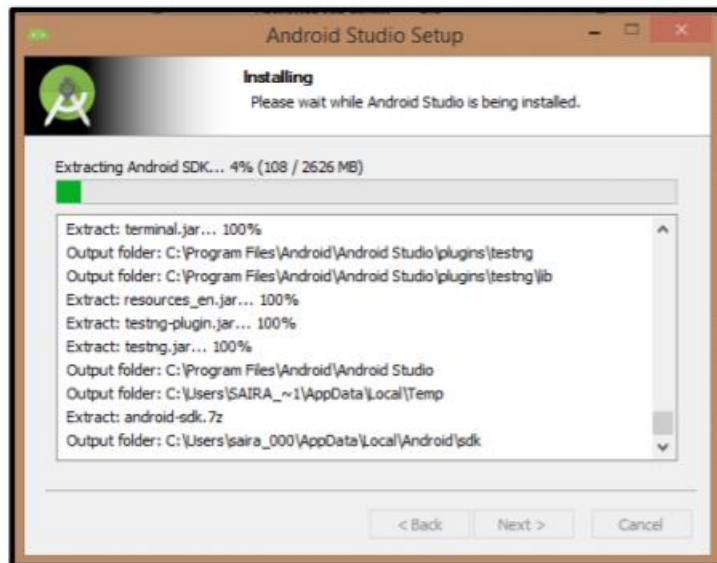
Εικόνα 35 : Επιλογή διαδρομής εγκατάστασης

Στην συνέχεια ρυθμίζουμε τον προσομοιωτή στο οποίο θα πραγματοποιηθούν οι δοκιμές και θα τρέξουν οι εφαρμογές που θα υλοποιήσουμε.



Εικόνα 36: Ρύθμιση Προσομοιωτή

Στην εικόνα 36 γίνεται μία ρύθμιση της μνήμης RAM, η οποία μπορεί να αυξηθεί ή να μειωθεί ανάλογα με τις απαιτήσεις του χρήστη. Αφού γίνει η επιλογή της RAM πατάμε το κουμπί Next (Επόμενο) και μεταφερόμαστε στο παράθυρο που γίνεται η εγκατάσταση της πλατφόρμας.



Εικόνα 37: Εγκατάσταση Android Studio

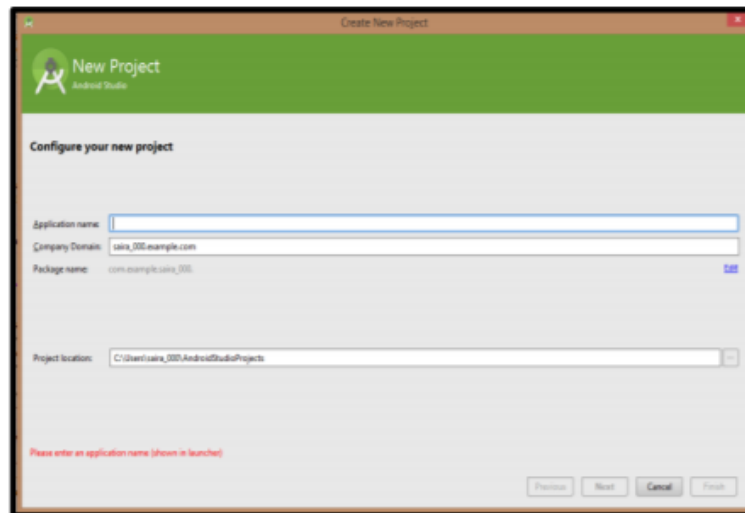
Στην παραπάνω εικόνα πραγματοποιείται η εγκατάσταση της πλατφόρμας. Μόλις ολοκληρωθεί η εγκατάσταση πατάμε το κουμπί Next (Επόμενο). Στην συνέχεια ανοίγει η πλατφόρμα και μεταφερόμαστε στο αρχικό περιβάλλον της πλατφόρμας.



Εικόνα 38 : Αρχικό περιβάλλον πλατφόρμας

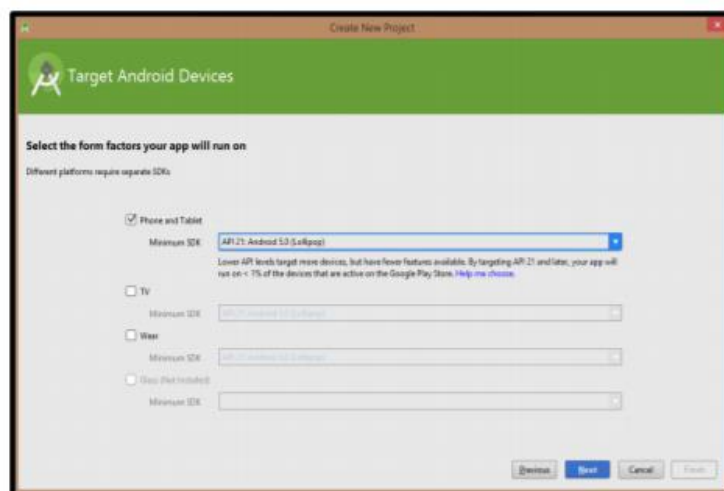
Η εγκατάσταση μόλις έχει ολοκληρωθεί με επιτυχία και ο χρήστης είναι έτοιμος να χρησιμοποιήσει την πλατφόρμα προκειμένου να δημιουργήσει τις δικές του εφαρμογές. Παρακάτω γίνεται μία μίνι παρουσίαση των βημάτων που χρειάζονται για να γίνει ένα νέο Project.

Αρχικά επιλέγουμε από το αρχικό περιβάλλον της πλατφόρμας την επιλογή Start a new Android Project(Δημιουργία νέου Android Project).



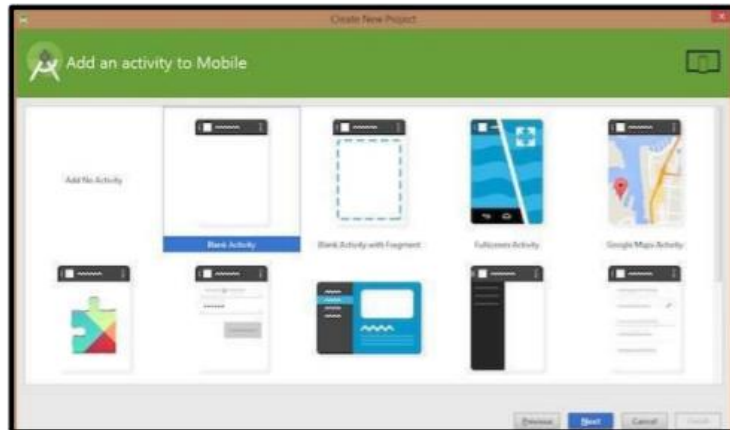
Εικόνα 39 : Δημιουργία εφαρμογής

Στο παράθυρο αυτό δίνουμε ένα όνομα στην εφαρμογή μας. Για παράδειγμα η εφαρμογή που υλοποιήσαμε στα πλαίσια της πτυχιακής αυτής πήρε το όνομα Corfu m-ticket. Στο Company Domain συμπληρώνουμε το όνομα της εταιρείας αν υπάρχει. Και τέλος στο Project Location βάζουμε την διαδρομή της εφαρμογής που θα αποθηκευτεί πχ Desktop (Επιφάνεια Εργασίας).



Εικόνα 40 : Επιλογή των παραμέτρων της εφαρμογής

Στο παράθυρο της εικόνας 40 επιλέγουμε εάν η εφαρμογή που δημιουργήσαμε προορίζεται για έξυπνο κινητό τηλέφωνο (Smartphone) για Tablet για Έξυπνη τηλεόραση ή για έξυπνο ψηφιακό ρολόι. Επίσης έχουμε την δυνατότητα να επιλέξουμε και την έκδοση του Android στο οποίο θα τρέξει η εφαρμογή. Επιλέγουμε το Next(Επόμενο)

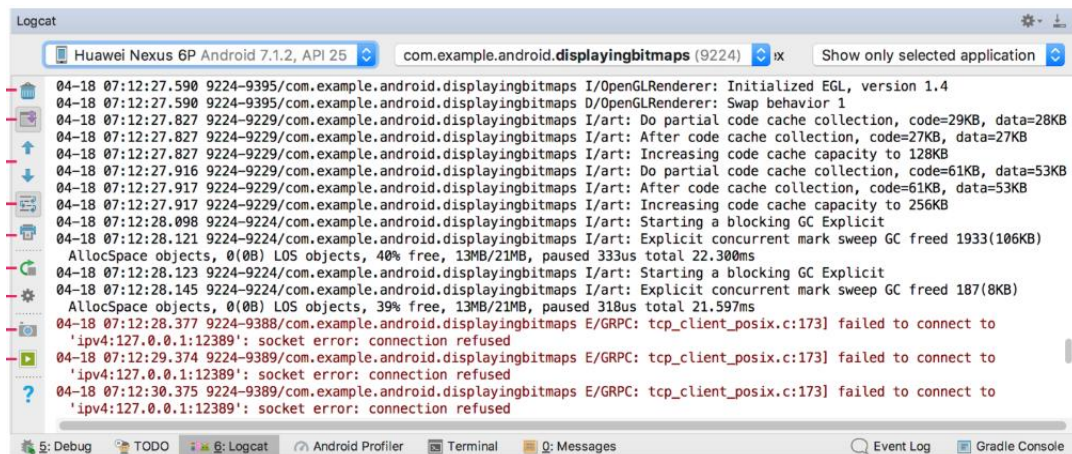


Αφού επιλέξουμε και την οθόνης της εφαρμογής και πατήσουμε Next (Επόμενο) , φτάνουμε στο τελικό περιβάλλον της εργασίας όπου από αυτό το σημείο ξεκινάει ο προγραμματισμός και η υλοποίηση της εφαρμογής. (Εικόνα 42).

3.2 Διαδικασία Αποσφαλμάτωσης (Debugging)

Κάθε περιβάλλον προγραμματισμού περιλαμβάνει ορισμένα εργαλεία τα οποία είναι υπεύθυνα για τον έλεγχο και την ανεύρεση σφαλμάτων στις εφαρμογές. Ο προγραμματιστής είναι αυτός που επιλέγει τον τρόπο υλοποίησης της εφαρμογής, αλλά το λειτουργικό σύστημα καθορίζει τους πόρους που θα χρησιμοποιηθούν.

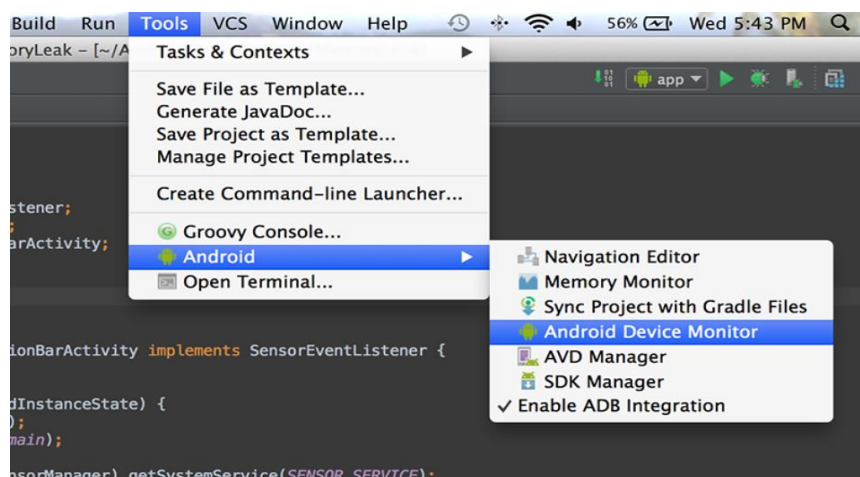
Στην διαδικασία Debugging χρησιμοποιείται το Logcat, το οποίο είναι ένας καταγραφέας για οτιδήποτε συμβαίνει στην εφαρμογή. Είναι υπεύθυνος για την ενημέρωση του προγραμματιστή για οποιοδήποτε σφάλμα προκύψει.



Εικόνα 43: Logcat

Επίσης ένα άλλο σημαντικό εργαλείο αποσφαλμάτωσης που χρησιμοποιούν οι προγραμματιστές είναι το DDMS. Πρόκειται για ένα εργαλείο το οποίο δίνει την δυνατότητα να ελεγχθεί η εφαρμογή σε εικονικές συσκευές που δεν έχουν πρόσβαση σε ζωντανά δεδομένα. Για να καταλάβουμε καλύτερα πως λειτουργεί το DDMS ας δούμε το παρακάτω παράδειγμα :

Υλοποιούμε μια εφαρμογή ανταλλαγής μηνυμάτων κειμένου και στατικών εικόνων, πριν την δοκιμάσουμε σε κανονική συσκευή, μπορούμε να στείλουμε fake notifications και μηνύματα σε μία εικονική συσκευή με στόχο της βελτίωση της εφαρμογής μέσα από εικονικές δοκιμές.



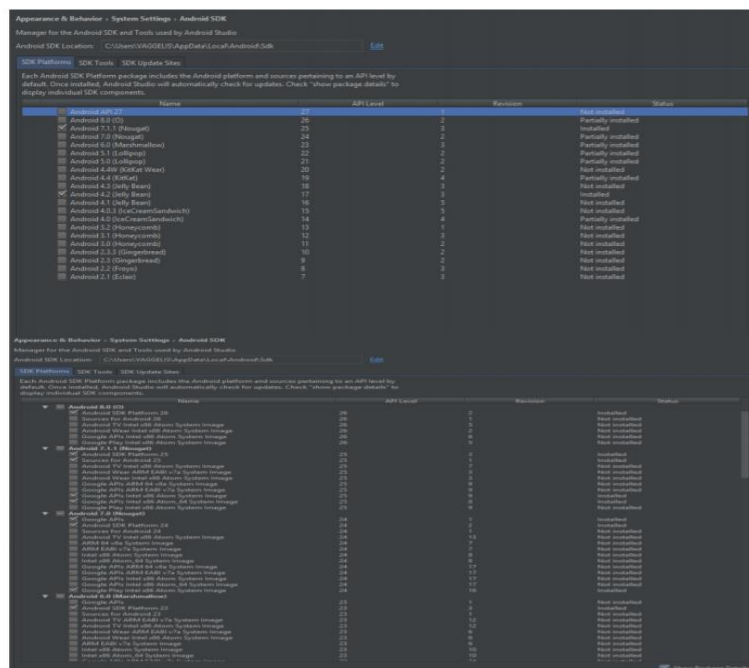
Εικόνα 44 : Android Device Monitor(DDMS)

3.3 Δημοσίευση Εφαρμογής

Με την ολοκλήρωση της ανάπτυξης της εφαρμογής , κάνουμε διάφορες δοκιμές τόσο στον προσομοιωτή όσο και στην Android συσκευή μας προκειμένου να ελεγχθεί η λειτουργία της. Αφού ολοκληρωθούν αυτές οι δοκιμές, το επόμενο βήμα είναι η δημοσίευση της εφαρμογής στο Google Play προκειμένου να διανεύουμε την εφαρμογή στο παγκόσμιο ιστό. Για να δημοσιευτεί μία εφαρμογή στο Google Play χρειαζόμαστε το αρχείο .apk του προγράμματος μας, το οποίο θα μας το δώσει ο compiler εφόσον περάσει επιτυχώς την διαδικασία της αποσφαλμάτωσης. Προκειμένου να δημοσιευτεί μία εφαρμογή στο Playstore είναι απαραίτητο να γίνει εγγραφή του χρήστη σαν προγραμματιστής και στην συνέχεια δίνοντας μία περιγραφή της εφαρμογής μας, συμπεριλαμβάνοντας και ορισμένα screenshots η εφαρμογή ελέγχεται από την Google και αν περάσει τον έλεγχο δημοσιεύεται. Η εφαρμογή μπορεί να διατεθεί με τιμή πώλησης ή δωρεάν. Επίσης, ένα άλλο μέσο διάθεσης είναι η προσωπική ή εταιρική ιστοσελίδα του προγραμματιστή. Αυτή η επιλογή όμως κρύβει μειονεκτήματα καθώς δεν θα υπάρχει έλεγχος για ενημερώσεις της εφαρμογής σε αντίθεση με τα ψηφιακά καταστήματα όπου υπάρχει αυτόματος μηχανισμός.

3.4 Android System Development Kit

Για να δημοσιευτεί μια εφαρμογή και να είναι διαθέσιμη στο ευρύ κοινό θα πρέπει να χρησιμοποιηθούν ορισμένες βιβλιοθήκες και συγκεκριμένα εργαλεία από τον προγραμματιστή που είναι διαθέσιμα μέσω του Android SDK. Επίσης μέσα από το Android Studio θα μπορέσει ο προγραμματιστής να προβεί σε διαθέσιμες ενημερώσεις ανάλογα με τις εκδόσεις του λειτουργικού συστήματος. Μπορεί να κατεβάσει τα απαραίτητα αυτά εργαλεία μέσα από τον SDK Manager που είναι μία επέκταση του προγραμματιστικού περιβάλλοντος του Android Studio. Ο τρόπος με τον οποίο είναι διαθέσιμα τα πακέτα ανάλογα με τα API's παρουσιάζεται στην παρακάτω εικόνα.

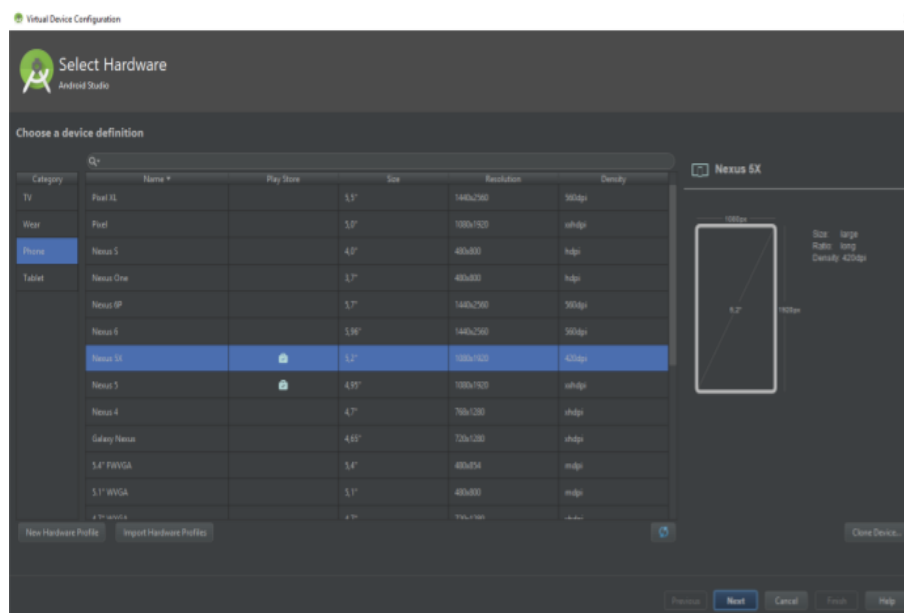


Εικόνα 45 : Διαχείριση Εικονικών Συσκευών

3.5 Android Virtual Device Manager (AVD)

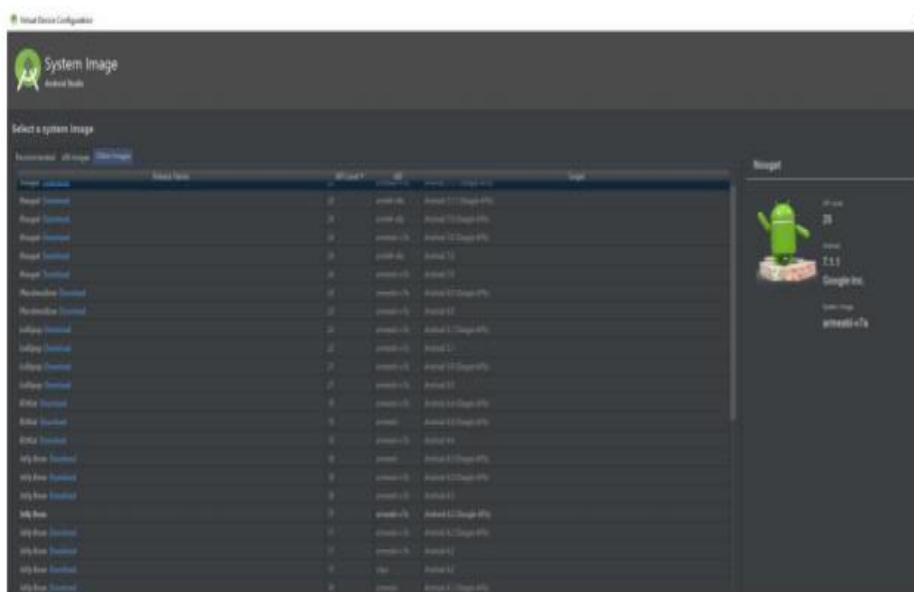
Επίσης προκειμένου να δημιουργήσουμε μια εικονική συσκευή για την δοκιμή της εφαρμογής, θα πρέπει πρώτα να κατεβάσουμε ορισμένα αρχεία που είναι υπεύθυνα για την λειτουργία της. Την συγκεκριμένη δουλειά την αναλαμβάνει το Android Virtual Device Manager με το κατάλληλο SDK και έκδοση λειτουργικού και με τον κατάλληλο επεξεργαστή. Παρακάτω βλέπουμε τον τρόπο λειτουργίας του AVD:

- Αρχικά επιλέγουμε από την λίστα την συσκευή που επιθυμούμε για να δοκιμάσουμε την εφαρμογή

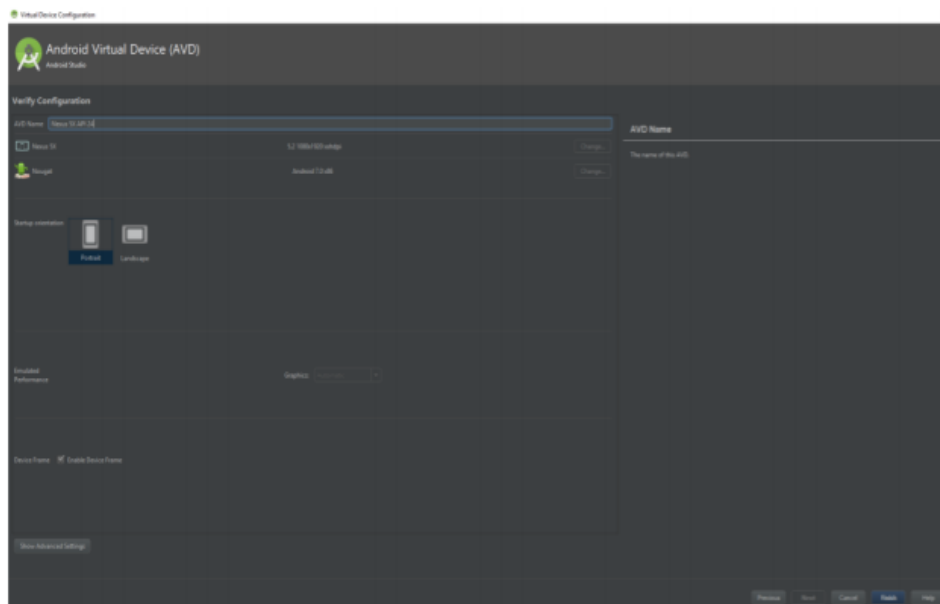


Εικόνα 46 : Android Virtual Device Manager

- Στην συνέχεια επιλέγουμε το SDK και την επιθυμητή έκδοση που επιθυμούμε να τρέχει η εικονική μας συσκευή.

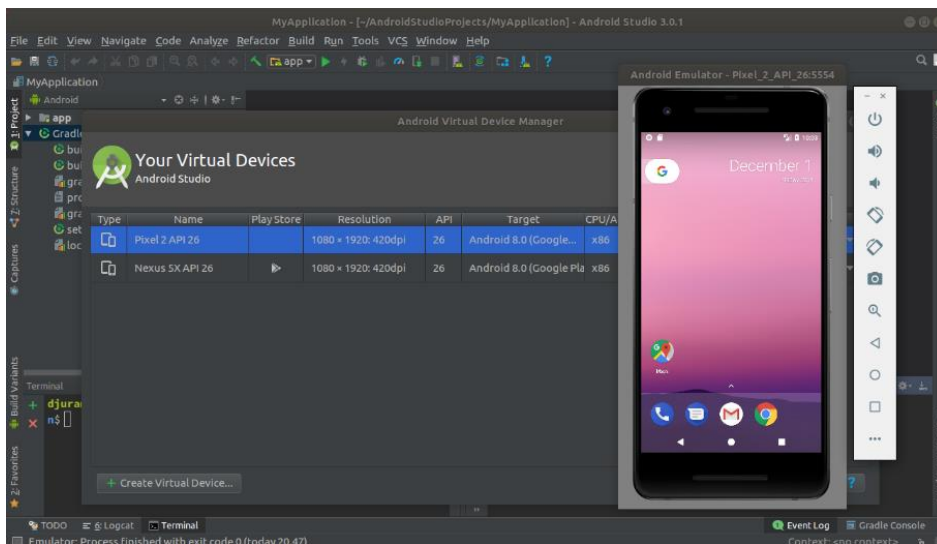


Εικόνα 47: Android Virtual Device Manager Επιλογή API



Εικόνα 48 : Προσθήκη εικονικής συσκευής

- Με αυτό τον τρόπο δημιουργείται μία εικονική συσκευή όπως η παρακάτω της εικόνας 49.



Εικόνα 49 : Εικονική συσκευή προσομοίωσης

4. Ανάλυση εφαρμογής

4.1 Ανάλυση απαιτήσεων για την δημιουργία της εφαρμογής

Για να δημιουργήσεις μία επιτυχημένη εφαρμογή, το βασικότερο στοιχείο είναι η ιδέα πάνω στην οποία θα την χτίσεις. Για να υλοποιηθεί στην πράξη αυτή η ιδέα θα πρέπει να ικανοποιούνται ορισμένοι σημαντικοί παράγοντες. Ορισμένοι από αυτούς τους παράγοντες είναι οι εξής:

- Οι δυνατότητες του προγραμματιστή που θα υλοποιήσει την αρχική ιδέα
- Το περιβάλλον στο οποίο θα υλοποιηθεί η ιδέα
- Και τα εργαλεία με τα οποία θα υλοποιηθεί

4.2 Ανάλυση της εφαρμογής με διαγράμματα UML

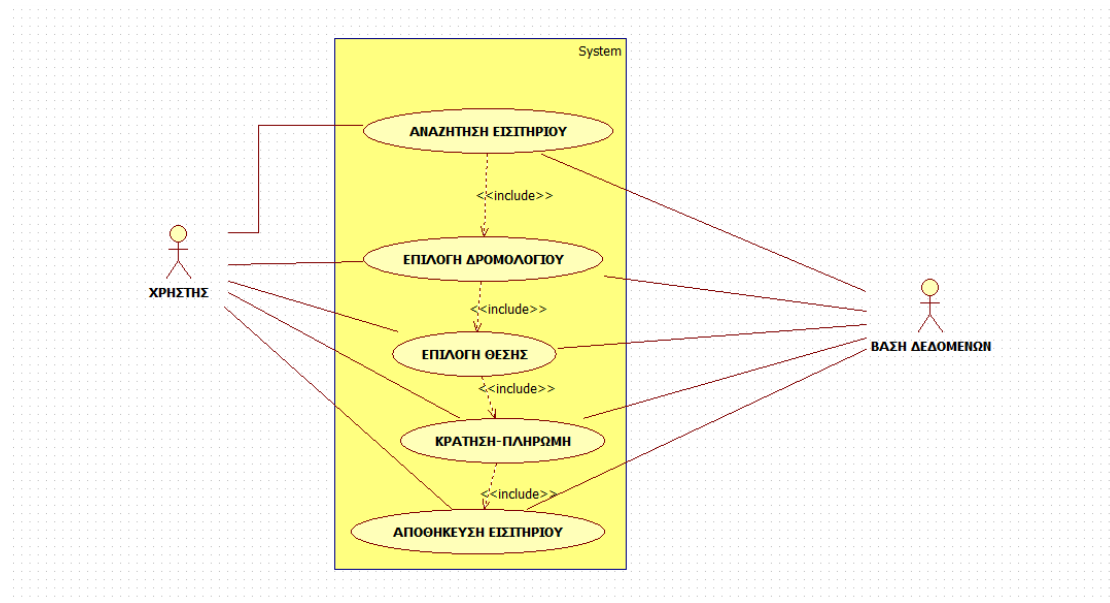
Ένα από τα βασικά και σημαντικά θέματα στην κατασκευή ενός πληροφοριακού συστήματος είναι η κατασκευή του ορθού συστήματος, δηλαδή του συστήματος που ικανοποιεί τις απαιτήσεις των χρηστών και των άλλων συμμετεχόντων. Πολλά συστήματα πληροφορικής αποτυγχάνουν να ικανοποιήσουν αυτόν τον φαινομενικά στοιχειώδη στόχο: δεν ικανοποιούν τις προδιαγραφές του χρήστη. Εδώ έρχονται να δώσουν τη λύση οι γλώσσες προδιαγραφών λογισμικού, οι οποίες σε συνδυασμό με τις μεθοδολογίες ανάπτυξης αποτελούν τα δομικά συστατικά για την σωστή ανάπτυξη συστημάτων λογισμικού. Εφόσον λοιπόν μιλάμε για αντικειμενοστραφή προγραμματισμό έχουμε σχεδιάσει -αρχικά στο χαρτί κι έπειτα στον υπολογιστή- το μοντέλο και τα σενάρια λειτουργίας της εφαρμογής μας, προκειμένου να προχωρήσουμε αργότερα στην υλοποίηση. Για τη δημιουργία των διαγραμμάτων UML έγινε χρήση του λογισμικού StarUML το οποίο είναι ένα πρόγραμμα ανοιχτού κώδικα εξ ολοκλήρου κατασκευασμένο σε Java.



Εικόνα 50 : StarUML – Πρόγραμμα υλοποίησης UML

4.2.1 Διάγραμμα περιπτώσεων χρήσης

Το μοντέλο περιπτώσεων χρήσης δίνει έμφαση στη λειτουργικότητα ενός συστήματος όπως αυτή είναι ορατή από τους εξωτερικούς χρήστες του. Μία περίπτωση χρήσης διαμερίζει τη λειτουργικότητα ενός συστήματος σε συναλλαγές (περιπτώσεις χρήσης) που έχουν νόημα για τους χρήστες συστήματος. Για να έχουμε λοιπόν μία πρώτη εικόνα της διαδικασίας χρήσης της εφαρμογής ας δούμε το παρακάτω διάγραμμα :

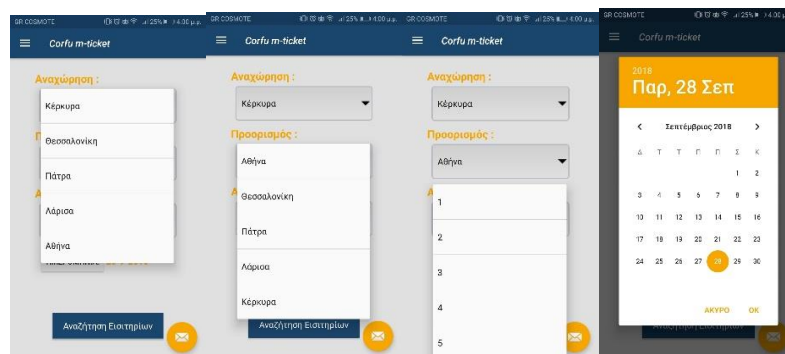


Εικόνα 51: Διάγραμμα Περιπτώσεων Χρήσης

4.2.2 Διάγραμμα περιπτώσεων χρήσης(Σενάριο)

Με βάση το παραπάνω διάγραμμα θα προσπαθήσουμε να αναλύσουμε τι ακριβώς θα κάνει η εφαρμογή και θα καταγράψουμε το σενάριο χρήσης της.

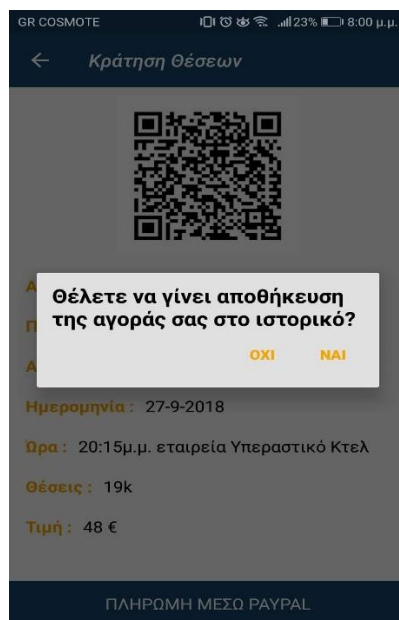
1. Ο χρήστης επιλέγει πόλη-σημείο αναχώρησης από το πτυσσόμενο μενού που υπάρχει στην αρχική οθόνη. Στην συνέχεια επιλέγει με τον ίδιο τρόπο και πόλη προορισμού, αριθμό εισιτηρίων και ημερομηνία και πατάει αναζήτηση εισιτηρίων προκειμένου να μεταβεί στην επιλογή του δρομολογίου, όπως φαίνεται στις εικόνες παρακάτω .



Εικόνα 52: Αναπαράσταση σε screenshot, η εφαρμογή Corfu m-ticket την στιγμή που ο χρήστης επιλέγει εισιτήριο

2. Όταν ο χρήστης επιλέξει και το δρομολόγιο με το οποίο επιθυμεί να ταξιδέψει και πατήσει το κουμπί επιλογή θέσης τότε θα μεταβεί σε μία τρίτη οθόνη προκειμένου να επιλέξει την θέση στην οποία επιθυμεί να ταξιδέψει.
3. Σε αυτή την οθόνη επιλογής λεωφορείου-δρομολογίου δίνεται η επιλογή του να πάει ο χρήστης πίσω
4. Η επόμενη οθόνη είναι η επιλογή θέσης. Εδώ ο χρήστης δεν έχει την δυνατότητα να επιλέξει περισσότερες θέσεις από αυτές που είχε επιλέξει στην αρχική οθόνη. Σε περίπτωση που επιλέξει μία θέση γίνεται η θέση αυτή κίτρινη από άσπρη. Αν επιλέξει παραπάνω βγαίνει ανάλογο μήνυμα λάθους-προειδοποίησης και πηγαίνει στην προηγούμενη οθόνη αυτόματα προκειμένου να επιλέξει ξανά τον αριθμό των θέσεων που επιθυμεί για κράτηση. Αφού ολοκληρώσει πατάει το κουμπί Επισκόπηση, το οποίο σε μεταβιβάζει στην επόμενη οθόνη.
5. Και αυτή η οθόνη δίνει την επιλογή στον χρήστη να επιστρέψει στην αρχική οθόνη.
6. Στην οθόνη επισκόπησης , η εφαρμογή δείχνει στον χρήστη τις πληροφορίες σχετικά με το δρομολόγιο του. Αν δεν έχει πραγματοποιηθεί κάποιο λάθος ο χρήστης μπορεί να προχωρήσει μέσω του κουμπιού πληρωμή μέσω Paypal.
7. Σε αυτή την οθόνη μπορεί ο χρήστης είτε να πληρώσει μέσω Paypal είτε μέσω Κάρτας. Και στις δύο περιπτώσεις πρέπει να εισάγει τα στοιχεία του προκειμένου να συνδεθεί στις υπηρεσίες αυτές. Η πληρωμή λειτουργεί μόνο με την χρήση διαδικτύου, χωρίς διαδίκτυο δεν μπορεί να πραγματοποιηθεί συναλλαγή. Σε περίπτωση που ο χρήστης επιθυμεί να επιστρέψει στην προηγούμενη σελίδα , πατάει το κουμπί επιστροφής στο κινητό του τηλέφωνο.

8. Από την στιγμή που συνδεθεί επιτυχώς στο λογαριασμό Paypal αρκεί να πατήσει Pay και η πληρωμή θα ολοκληρωθεί.
9. Σε περίπτωση επιτυχής αγοράς του εισιτηρίου εμφανίζεται το παρακάτω μήνυμα της οθόνης.



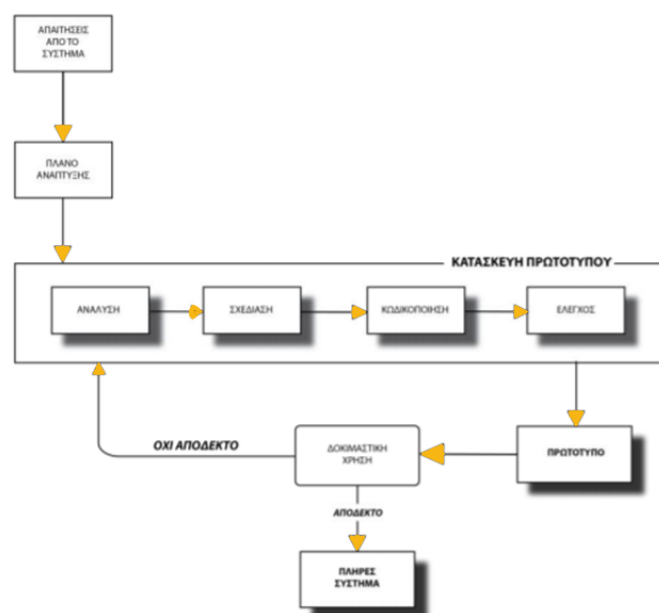
Εικόνα 53 : Αναπαράσταση σε screenshot του Corfu m-ticket

10. Στην περίπτωση που αποτύχει η αγορά τότε εμφανίζεται μήνυμα αποτυχίας.
11. Με την ολοκλήρωση της αγοράς ο χρήστης επανέρχεται στην αρχική οθόνη όπου από εκείνο το σημείο μπορεί να περιηγηθεί στο μενού και να δει το ιστορικό των κρατήσεων του, να ακυρώσει ένα εισιτήριο κ.α.

Η συγκεκριμένη εφαρμογή λειτουργεί αποτελεσματικά χάρις τα δεδομένα που λαμβάνει από μία βάση δεδομένων που είναι εγκατεστημένη τοπικά στο κινητό μας.

4.2.3 Μοντέλο Πρωτοτυποποίησης

Το μοντέλο πρωτοτυποποίησης υλοποιεί την ανάπτυξη του λογισμικού σε τμήματα, τα οποία ονομάζονται πρωτότυπα. Οι διαδικασίες ανάπτυξης επαναλαμβάνονται για ένα τμήμα του συστήματος κάθε φορά και γι' αυτό τον λόγο το μοντέλο αυτό χαρακτηρίζεται ως επαναληπτικό μοντέλο. Κάθε τμήμα του συστήματος περιλαμβάνει τις βασικές λειτουργίες που προορίζονται για εκτέλεση από το Λογισμικό και τίθενται σε δοκιμασία από τον πελάτη. Από τις δοκιμασίες αυτές συλλέγονται σημαντικές πληροφορίες και γίνονται παρατηρήσεις. Η διαδικασία κατασκευής του πρωτοτύπου επαναλαμβάνεται μέχρις ότου το πρωτότυπο να ικανοποιήσει τις απαιτήσεις που έχει θέσει ο πελάτης. Από το σημείο αυτό και μετά μπορούν να προστεθούν και οι υπόλοιπες λειτουργίες, ώστε το λειτουργικό να φτάσει στο σημείο να ολοκληρωθεί. Παρακάτω απεικονίζεται ένα διάγραμμα πρωτοτυποποίησης και τα στάδια του.



Εικόνα 54 : Το μοντέλο της πρωτοτυποποίησης

Ένα σημαντικό πλεονέκτημα του μοντέλου αυτού είναι η δυνατότητα απόκτησης άποψης για την εφαρμογή του Λογισμικού πολύ νωρίτερα από ότι στο μοντέλο του καταρράκτη. Αυτό σημαίνει ότι, το μοντέλο αυτό μπορεί να γλιτώσει την ανάπτυξη από καθυστερήσεις και λάθη. Κάθε κατασκευή πρωτοτύπου μπορεί θεωρηθεί ως ένα μικρό έργο Λογισμικού, το οποίο κατασκευάζεται με διαδικασίες που μπορούν να ακολουθούν άλλα μοντέλα κύκλου ζωής, όπως αυτό του καταρράκτη.

Με βάση λοιπόν τις παραπάνω παρατηρήσεις το μοντέλο αυτό χρησιμοποιείται στην ανάπτυξη των εφαρμογών λογισμικού για τις απαιτήσεις από τις οποίες δεν υπάρχει βεβαιότητα στην αρχή της ανάπτυξης, οπότε δεν μπορούν να συμφωνηθούν και να παγιωθούν αυτές. Τέτοιες είναι εφαρμογές που κατασκευάζονται για πρώτη φορά και είναι στενά εξαρτημένες με τον πελάτη, χωρίς να υπάρχει αποδεκτό προηγούμενο παράδειγμα. Για αυτούς τους λόγους το μοντέλο αυτό προτιμήθηκε για την ανάπτυξη της εφαρμογής μας.

5. Παρουσίαση Corfu m-ticket

Σε αυτό το κεφάλαιο θα γίνει η παρουσίαση της εφαρμογής που υλοποιήσαμε. Θα ξεκινήσουμε από μία γενική παρουσίαση της εφαρμογής και παρακάτω θα παρουσιάσουμε λεπτομερώς τα χαρακτηριστικά της.

5.1 Περιγραφή εφαρμογής

Η εφαρμογή που κατασκευάσαμε δίνει την δυνατότητα στο χρήστη που επιθυμεί να ταξιδέψει με τα υπεραστικά ΚΤΕΛ Κέρκυρας να αγοράσει διαδικτυακά εισιτήρια μέσω κινητού τηλεφώνου.

Το app «Corfu m-ticket» επιτρέπει στον κάτοχο του κινητού να αγοράσει όσα εισιτήρια επιθυμεί χωρίς κάποιο περιορισμό, αρκεί να έχει διαθέσιμα δρομολόγια.

Η διαδικασία έχει ως εξής:

- Εγγραφή-Σύνδεση στην εφαρμογή
- Αναζήτηση Εισιτηρίου
- Επιλογή Θέσης
- Επισκόπηση Εισιτηρίου
- Πληρωμή μέσω PAYPAL
- Αγορά Εισιτηρίου
- Ολοκλήρωση

1. Παράλληλα η εφαρμογή δίνει την δυνατότητα στους επιβάτες να ακυρώσουν οποιαδήποτε στιγμή το εισιτήριο τους, επιστρέφοντας και τα χρήματα τους πίσω. Το μόνο που έχουν να κάνουν είναι να επιλέξουν από το μενού την επιλογή Ακύρωση Εισιτηρίου.

2. Επίσης δίνει την δυνατότητα στον κάθε χρήστη ξεχωριστά να δει το ιστορικό εισιτηρίων που έχει αγοράσει.

3. Από την στιγμή που θα ολοκληρώνεται επιτυχημένα η πληρωμή μέσω Paypal, τα εισιτήρια θα θεωρούνται επικυρωμένα και θα μπορούν οι επιβάτες να εισέρχονται στο λεωφορείο με την επίδειξη του barcode του εισιτηρίου.

4. Ο χρήστης θα έχει την δυνατότητα να αγοράσει εισιτήρια για περισσότερους επιβάτες στο πλαίσιο μιας ομαδικής μετακίνησης. Ωστόσο όλα τα άτομα, για τα οποία έχουν εκδοθεί εισιτήρια ως μέρος μίας ενιαίας συναλλαγής της εφαρμογής, θα πρέπει να μετακινηθούν μαζί με τον κάτοχο του «κινητού» εισιτηρίου.

5. Σε περίπτωση ελέγχου, ο χρήστης επιδεικνύει το εισιτήριο στον ελεγκτή, ανοίγοντας την εφαρμογή και επιλέγοντας το κατάλληλο εισιτήριο από το ιστορικό.

6. Τα εισιτήρια δεν χρειάζονται τις υπηρεσίες κινητού τηλεφώνου (σύνδεση δεδομένων) για να εμφανιστούν.

Συνεπώς η εφαρμογή μας χαρακτηρίζεται από :

- Εύκολη κράτηση για κάποιο δρομολόγιο από την έξυπνη συσκευή μας
- Εύκολη αγορά εισιτηρίων από την έξυπνη συσκευή μας
- Ασφαλείς συναλλαγές με την χρήση προπληρωμένης κάρτας
- Ταχύτητα συναλλαγών: το μόνο που απαιτείται είναι τα στοιχεία εισόδου στο σύστημα της προπληρωμένης κάρτας σας
- Προσφέρεται δωρεάν
- Χωρίς πρόσθετες χρεώσεις

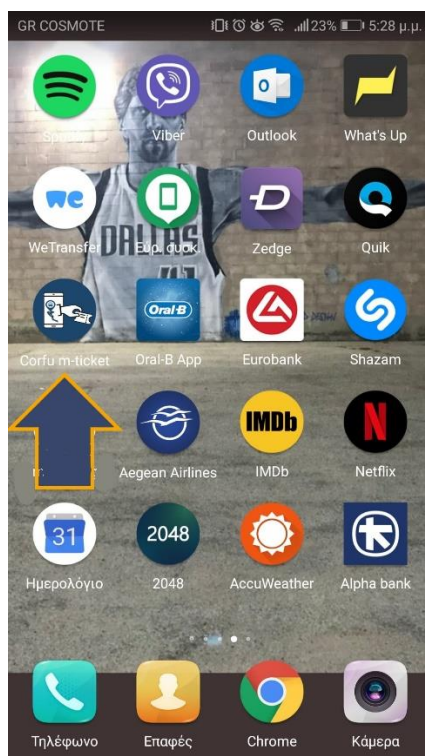
5.2 Παρουσίαση εφαρμογής

Σε αυτή την ενότητα θα παρουσιάσουμε μέσω screenshots την εφαρμογή μας. Το logo του app «Corfu m-ticket» είναι το παρακάτω :



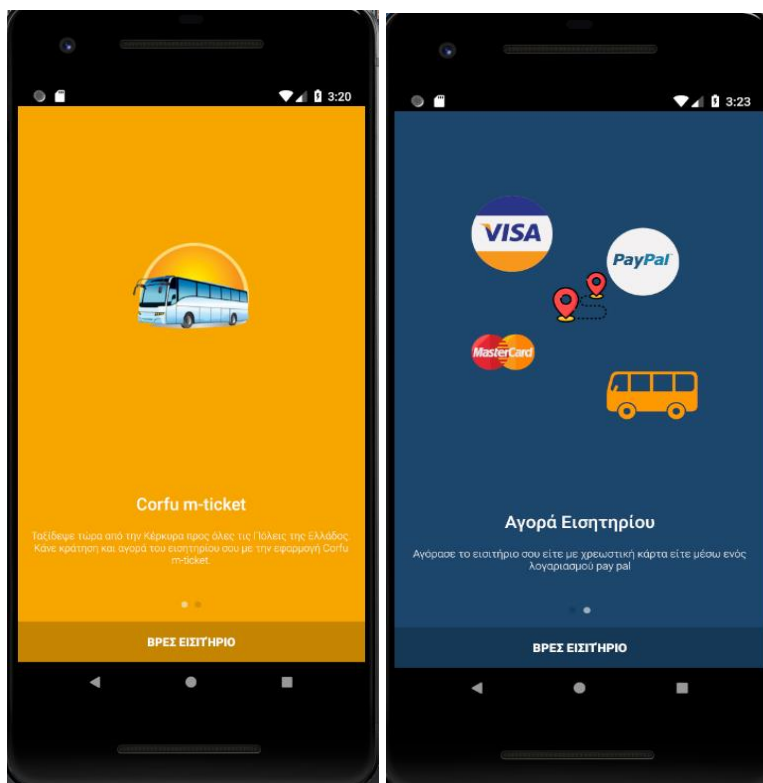
Εικόνα 55 : Logo εφαρμογής

Το logo της εφαρμογής δημιουργήθηκε με την χρήση του εργαλείου Adobe Photoshop. Πήραμε τέσσερις διαφορετικές εικόνες και τις ενώσαμε προκειμένου να δημιουργήσαμε ένα λογότυπο που θα εκφράζει την εφαρμογή μας.



Εικόνα 56 : Εικονίδιο της Εφαρμογής

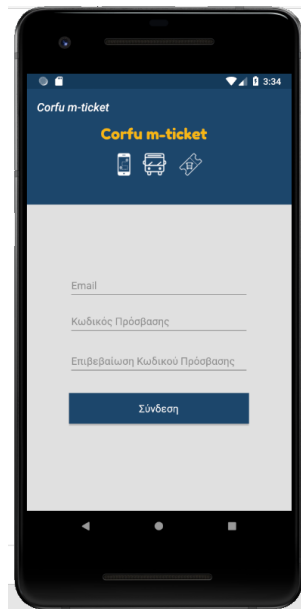
Με το άνοιγμα της εφαρμογής **Corfu m-ticket** οι πρώτες οθόνες που αντικρίζει ο χρήστης είναι τα Welcome Screens, τα οποία διαδέχονται το ένα το άλλο με εφέ και δίνουν στον χρήστη μία σύντομη περιγραφή της εφαρμογής.



Εικόνα 57: Welcome Screens της Εφαρμογής

Μετά τα Welcome Screens και αφού πατήσει ο χρήστης το κουμπί «Βρες Εισιτήριο», η εφαρμογή σε μεταφέρει στην οθόνη Register. Στην συγκεκριμένη οθόνη ο χρήστης κάνει εγγραφή στην εφαρμογή, συμπληρώνοντας το e-mail του και κατοχυρώνοντας έναν προσωπικό κωδικό ασφαλείας.

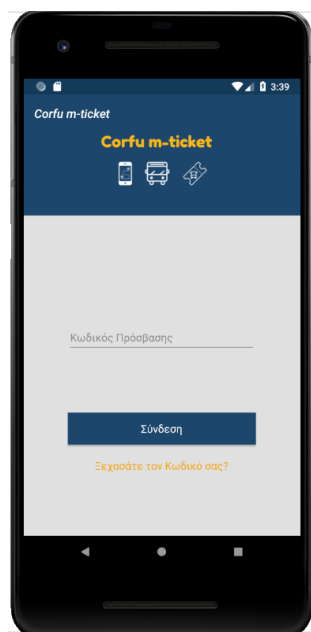
Προκειμένου να διασφαλιστούν όλα τα δεδομένα του χρήστη, χρησιμοποιήσαμε τον αλγόριθμο κρυπτογράφησης Cipher στην βάση δεδομένων μας.



Εικόνα 58 : Οθόνη Register

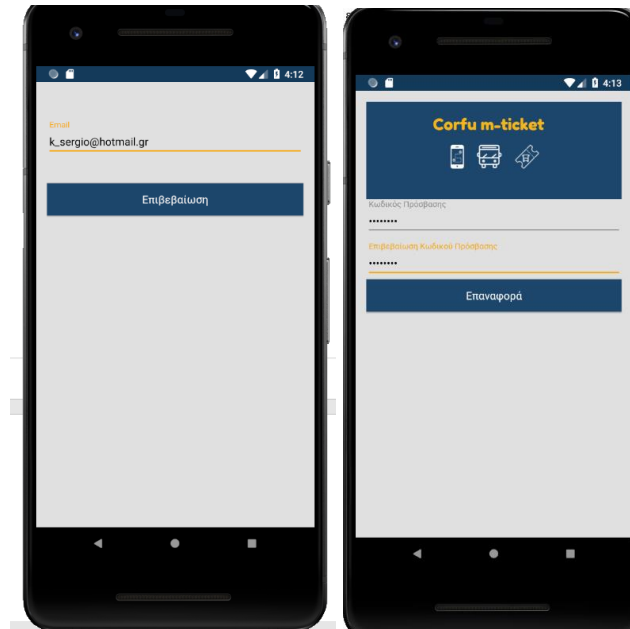
Εφόσον ο Χρήστης δημιουργήσει τον κωδικό ασφαλείας η εφαρμογή θα κλειδώνει όταν βρίσκεται στο background και όταν γίνεται kill.

Όταν ο χρήστης θα ξαναμπαίνει στην εφαρμογή τότε θα εμφανίζεται η οθόνη που θα ζητάει τον κωδικό πρόσβασης που έχει δημιουργήσει ο χρήστης.



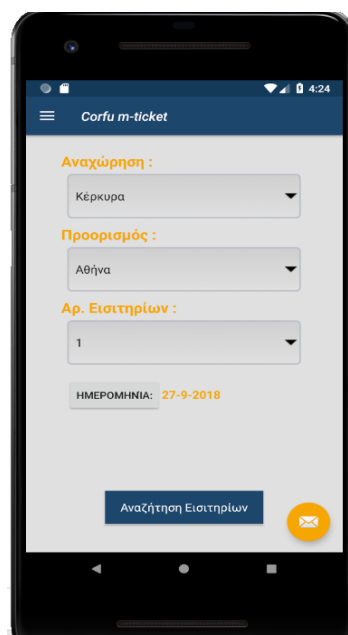
Εικόνα 59 : Οθόνη Login

Σε περίπτωση που ο χρήστης ξεχάσει τον προσωπικό του κωδικό, τότε θα πρέπει να πατήσει στο κουμπί «Ξεχάσατε τον Κωδικό σας?», Αμέσως η εφαρμογή σας πετάει σε μία οθόνη όπου απαιτείται η εισαγωγή του E-mail του χρήστη και μετά μεταφέρεται ο χρήστης σε νέα οθόνη και πληκτρολογεί τον καινούριο του κωδικό.



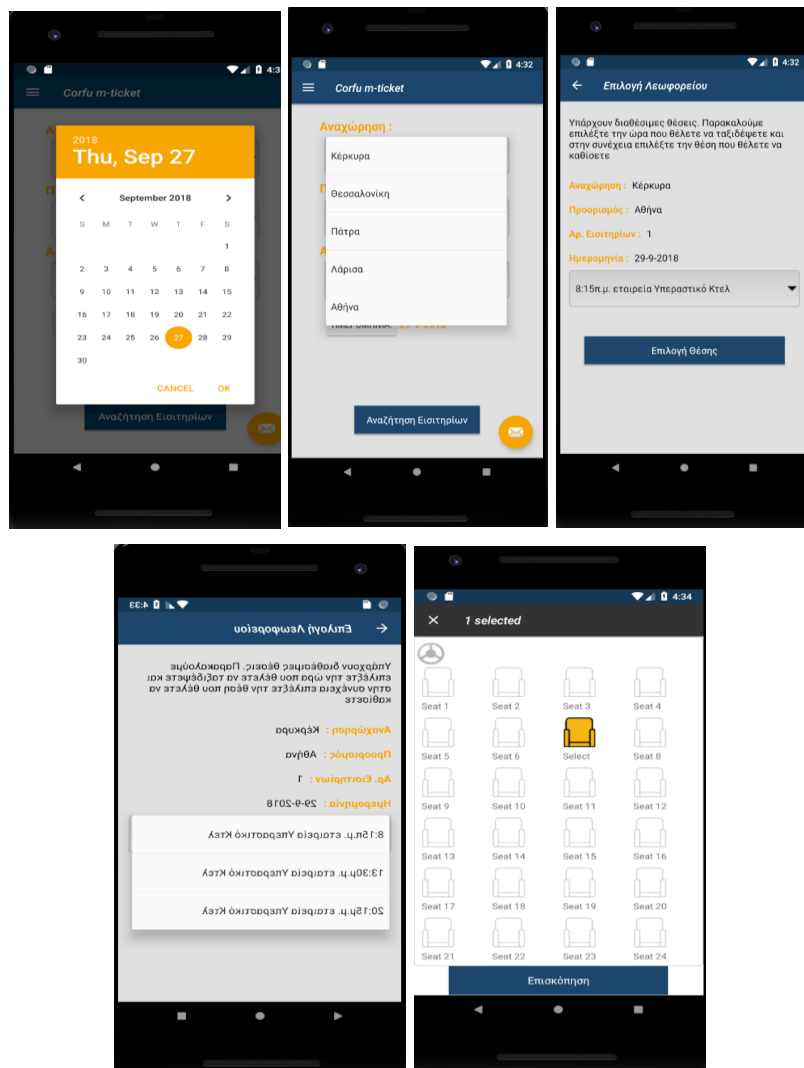
Εικόνα 60 : Αλλαγή Κωδικού Πρόσβασης

Όταν ο χρήστης συνδεθεί με επιτυχία στην εφαρμογή εμφανίζεται η αρχική οθόνη, στην οποία μπορεί να επιλέξει σημείο αναχώρησης και προορισμό, αριθμό εισιτηρίων και ημερομηνία ταξιδιού.



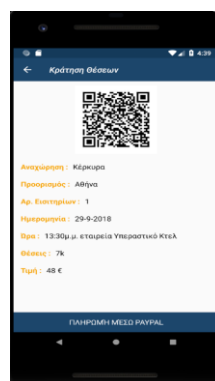
Εικόνα 61: Αναζήτηση Εισιτηρίων

Στην συνέχεια πατώντας του κουμπί Αναζήτηση Εισιτηρίων, η εφαρμογή δίνει την επιλογή στον χρήστη να επιλέξει το δρομολόγιο με το οποίο θέλει να ταξιδέψει, και στην πορεία να επιλέξει και την θέση στην οποία επιθυμεί να ταξιδέψει.



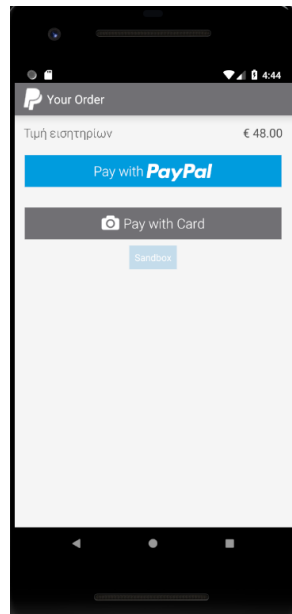
Εικόνα 62 : Αναζήτηση Δρομολογίου και Επιλογή Θέσης

Αφού επιλέξει και την θέση ή τις θέσεις που επιθυμεί ο χρήστης προκειμένου να ταξιδέψει πατάει το κουμπί επισκόπηση. Η εφαρμογή τον μεταφέρει σε μία οθόνη επισκόπησης και παρουσιάζει όλα τα στοιχεία του ταξιδιού.



Εικόνα 63 : Επισκόπηση εισιτηρίου

Αν ο χρήστης συμφωνεί με την επισκόπηση του εισιτηρίου προχωράει στην αγορά του, μέσω Paypal.



Εικόνα 64 : Επιλογή τρόπου πληρωμής

Ο χρήστης καλείται να συνδεθεί στο λογαριασμό του στην υπηρεσία Paypal και αφού συνδεθεί μπορεί να πραγματοποιήσει την αγορά του εισιτηρίου. Αν η αγορά πραγματοποιηθεί με επιτυχία τότε ο χρήστης ενημερώνεται με μήνυμα ότι η αγορά ήταν επιτυχής και μεταφέρεται στην αρχική οθόνη.

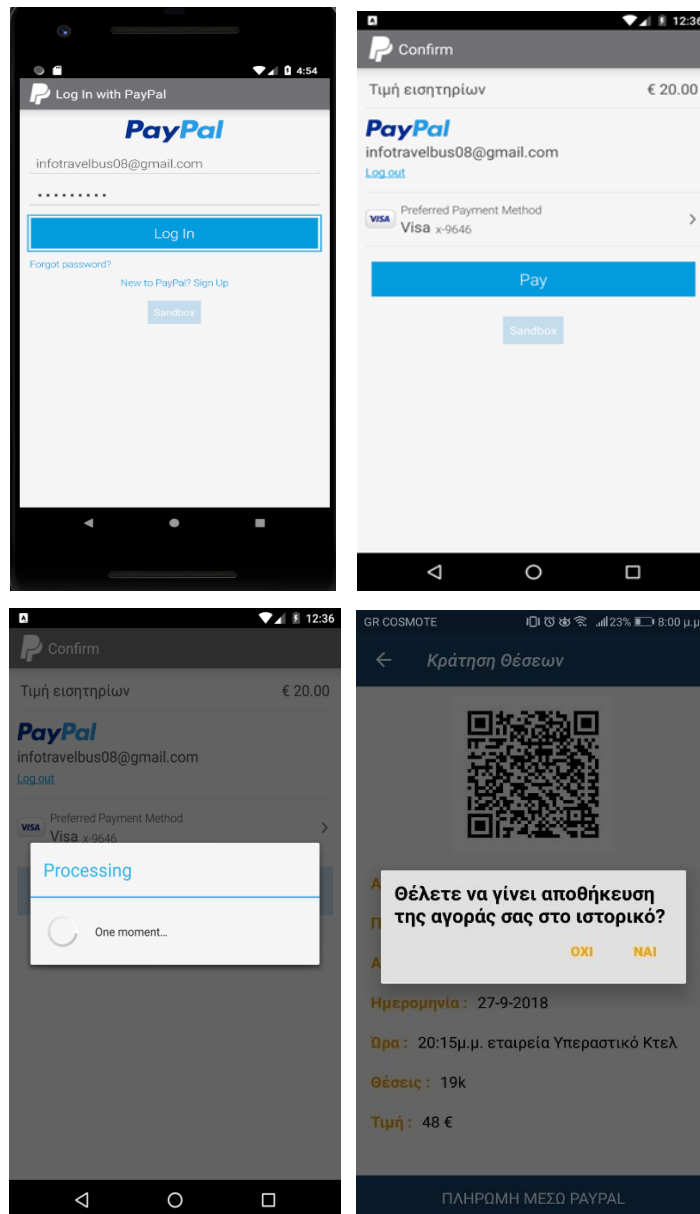
Για τις ανάγκες της εφαρμογής να γίνεται αγορά με Paypal δημιουργήθηκε ένας λογαριασμός Gmail και ένας tester λογαριασμός Paypal.

Στοιχεία Gmail account:

- Username : infotravelbus08@gmail.com
- Password : busCoTravel

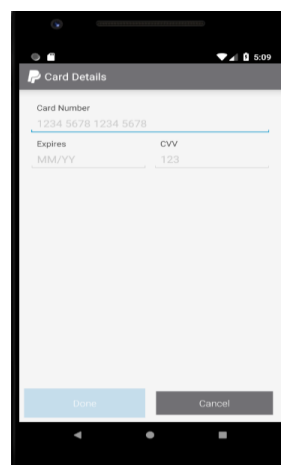
Στοιχεία Pay pal tester account:

- Username: infotravelbus08@gmail.com
- Password : 828f51ho?



Εικόνα 65 : Πληρωμή μέσω Paypal

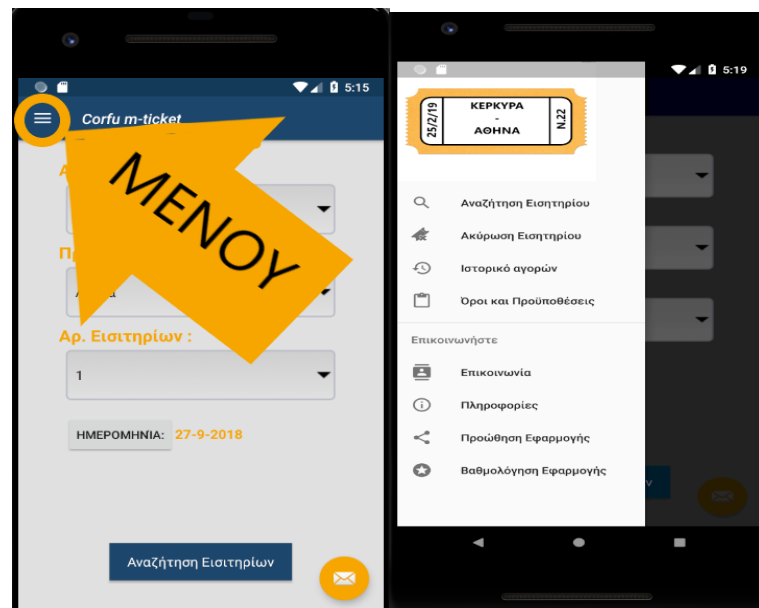
Η εφαρμογή δίνει την δυνατότητα επίσης η πληρωμή να γίνει με πιστωτική κάρτα.



Εικόνα 66 : Πληρωμή μέσω Κάρτα

Με αυτό τον τρόπο ολοκληρώσαμε μία επιτυχημένη αγορά εισιτηρίου. Προκειμένου να δούμε ή να ακυρώσουμε το εισιτήριο αυτό πρέπει να πάμε στο μενού της εφαρμογής. Το μενού της εφαρμογής περιλαμβάνει τα εξής:

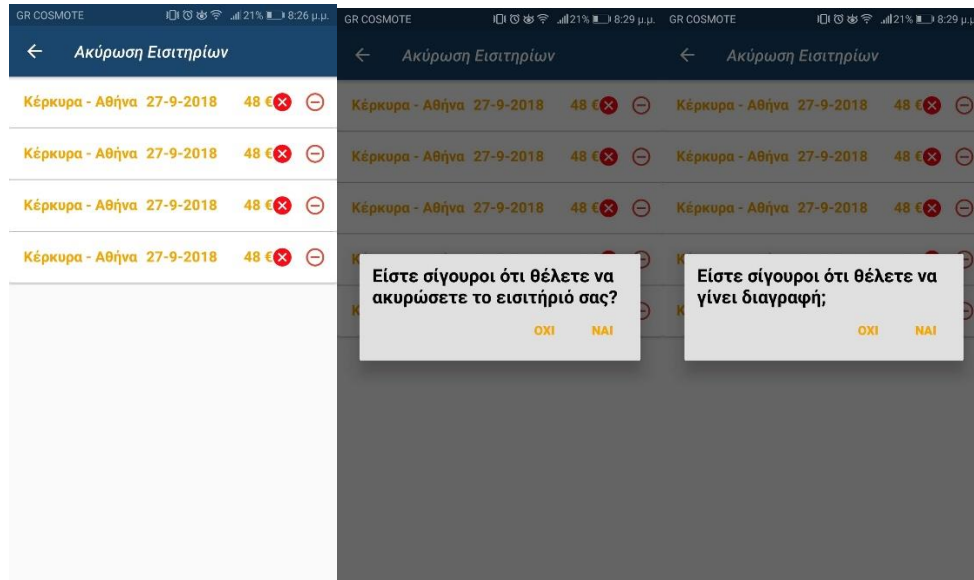
- Αναζήτηση Εισιτηρίου
- Ακύρωση Εισιτηρίου
- Ιστορικό Αγορών
- Όροι και Προϋποθέσεις
- Επικοινωνία
- Πληροφορίες
- Προώθηση Εφαρμογής
- Βαθμολογία Εφαρμογής



Εικόνα 67: Μενού της Εφαρμογής

Την επιλογή της αναζήτησης δρομολογίου την είδαμε ήδη καθώς αποτελεί την αρχική οθόνη της εφαρμογής.

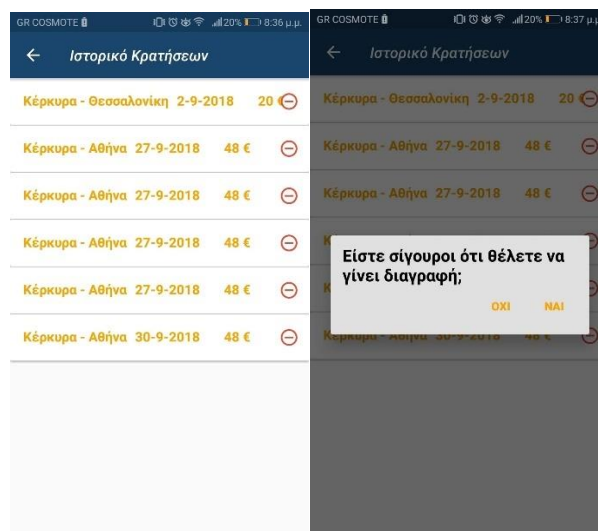
Παρακάτω θα δούμε την επιλογή Ακύρωση Εισιτηρίου. Όπως αναφέρει και η ονομασία της δίνει στον χρήστη την δυνατότητα να ακυρώσει ένα εισιτήριο και επιστρέφει τα χρήματα του χρήστη πίσω.



Εικόνα 68 : Η επιλογή της Ακύρωσης Εισιτηρίου

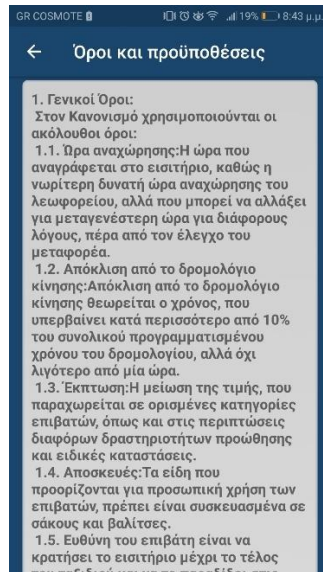
Να σημειωθεί ότι για να ακυρωθεί ένα εισιτήριο θα πρέπει να η ημερομηνία ακύρωσης να απέχει μία μέρα τουλάχιστον από την ημέρα αναχώρησης. Σε περίπτωση που η αναχώρηση είναι λιγότερο από ένα 24ώρο δεν μπορεί να πραγματοποιηθεί η ακύρωση.

Η επόμενη επιλογή του μενού είναι το ιστορικό Κρατήσεων. Σε αυτή την επιλογή δίνεται η δυνατότητα στον χρήστη να δει όλα τα εισιτήρια που έχει αγοράσει κατά καιρούς.



Εικόνα 69 : Ιστορικό Κρατήσεων

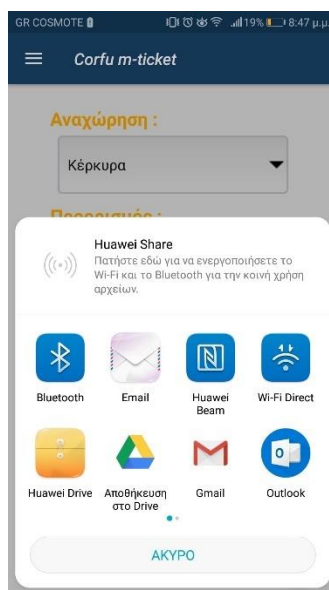
Στην συνέχεια έχουμε την επιλογή Όροι και Προϋποθέσεις, η οποία περιγράφει τους όρους της εφαρμογής μας. Τους όρους τους έχουμε πάρει από το επίσημο site των Υπεραστικών Κτελ Κέρκυρας(<https://greenbuses.gr/el/>).



Εικόνα 70 : Όροι και Προϋποθέσεις

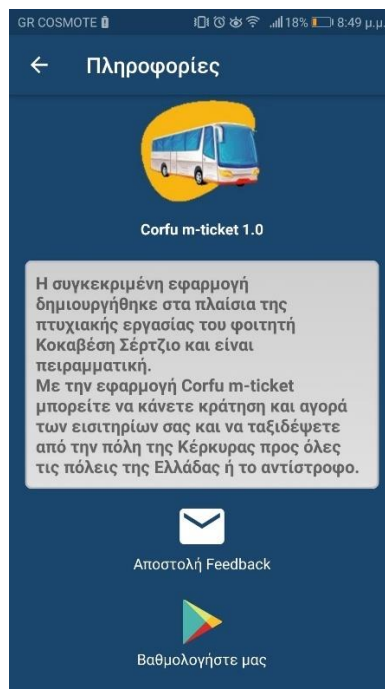
Έπειτα συναντάμε την επιλογή της Επικοινωνίας όπου ο χρήστης μπορεί να επικοινωνήσει μαζί μας με μία από τις παρακάτω επιλογές :

- E-mail
- G-mail
- Outlook
- Viber
- Messenger



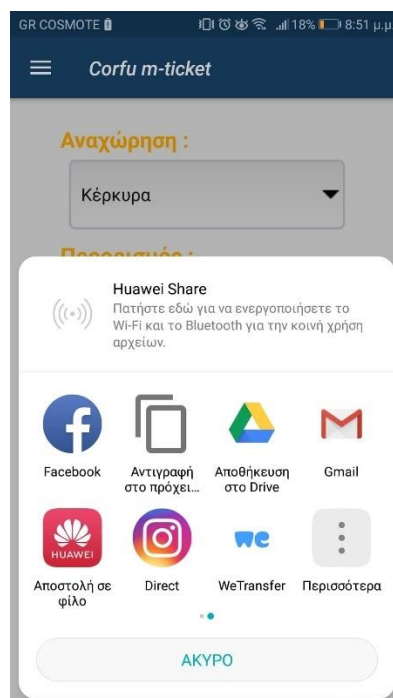
Εικόνα 71 : Επικοινωνία

Σαν επόμενη επιλογή στο μενού συναντάμε τις Πληροφορίες της εφαρμογής.



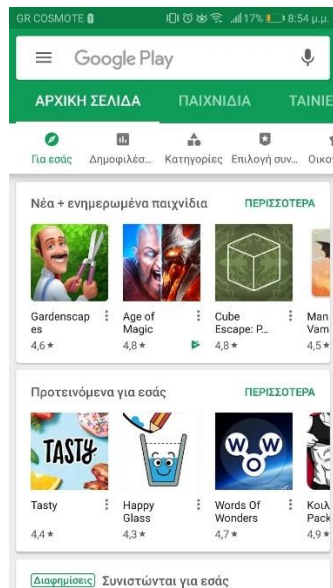
Εικόνα 72 : Πληροφορίες

Επόμενη επιλογή είναι η προώθηση της εφαρμογής και μπορεί να γίνει διαδικτυακά με μία από τις παρακάτω επιλογές της εικόνας.



Εικόνα 73 : Προώθηση Εφαρμογής

Και τέλος συναντάμε την επιλογή Βαθμολόγηση Εφαρμογής η οποία σε μεταφέρει στο Google Playstore προκειμένου μελλοντικά όταν ανέβει η εφαρμογή να βαθμολογηθεί.



Εικόνα 74 : Google Playstore

Αυτή ήταν μία απλή παρουσίαση της εφαρμογής Corfu m-ticket. Στο επόμενο κεφάλαιο θα παρουσιάσουμε την δομή της εφαρμογής.

6. Δομή της Εφαρμογής

- **Δομή**

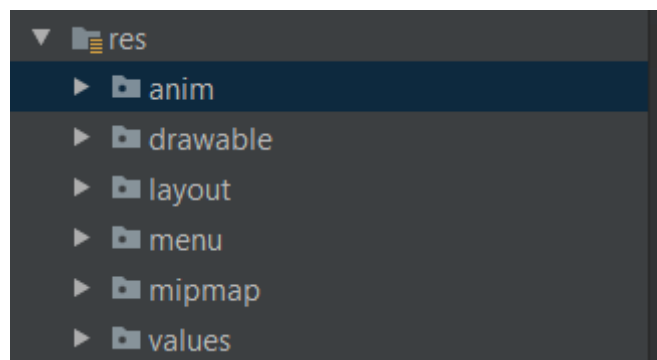
Το Corfu m-ticket έχει γραφτεί σε Java και Xml. Αποτελείτε από 77 κλάσεις (συμπεριλαμβανομένου και της βιβλιοθήκης 'com.stephentuso:welcome:1.4.1' που χρησιμοποιήθηκε για τα welcome screens και περιείχε 32 κλάσεις). Περιλαμβάνει επίσης και μερικά xml αρχεία. Για την ακρίβεια περιλαμβάνει 9 αρχεία που είναι υπεύθυνα για τον ορισμό των τιμών και τον πόρων. Τα αρχεία αυτά είναι τα εξής:

1. **attrs.xml**
2. **bools.xml**
3. **colors.xml**
4. **dimens.xml**
5. **drawables.xml**
6. **ic_launcher_background.xml**
7. **public.xml**
8. **strings.xml**
9. **styles.xml**

Περιλαμβάνει επίσης ένα αρχείο Manifest, 40 αρχεία για την υλοποίηση των layouts των οθονών της εφαρμογής και 2 που αφορούν το μενού της εφαρμογής. Ακόμα περιλαμβάνονται 44 πόροι εικόνων και δεν χρησιμοποιήσαμε κάποιον πόρο για ήχο καθώς δεν θεωρήθηκε απαραίτητο.

- **Φάκελος Res**

Ο φάκελος res (resources) περιέχει όλους τους πόρους της εφαρμογής. Οι πόροι αυτοί μπορεί να είναι xml αρχεία, εικόνες, ήχοι, animation και string τιμές. Για την κατηγοριοποίησή των πόρων αυτών υπάρχει ένας μεταγλωττιστής, ο resource compiler, ο οποίος τα κατηγοριοποιεί με βάση τον τύπο τους. Οι εικόνες για παράδειγμα εντάσσονται στον φάκελο drawable, οι διατάξεις που ορίζουν την δομή των διεπαφών της εφαρμογής στον φάκελο layout και ούτω καθεξής. Υπάρχει η δυνατότητα να δημιουργήσουμε όσους υποφακέλους χρειαζόμαστε για την καλύτερη οργάνωση της εφαρμογής μας



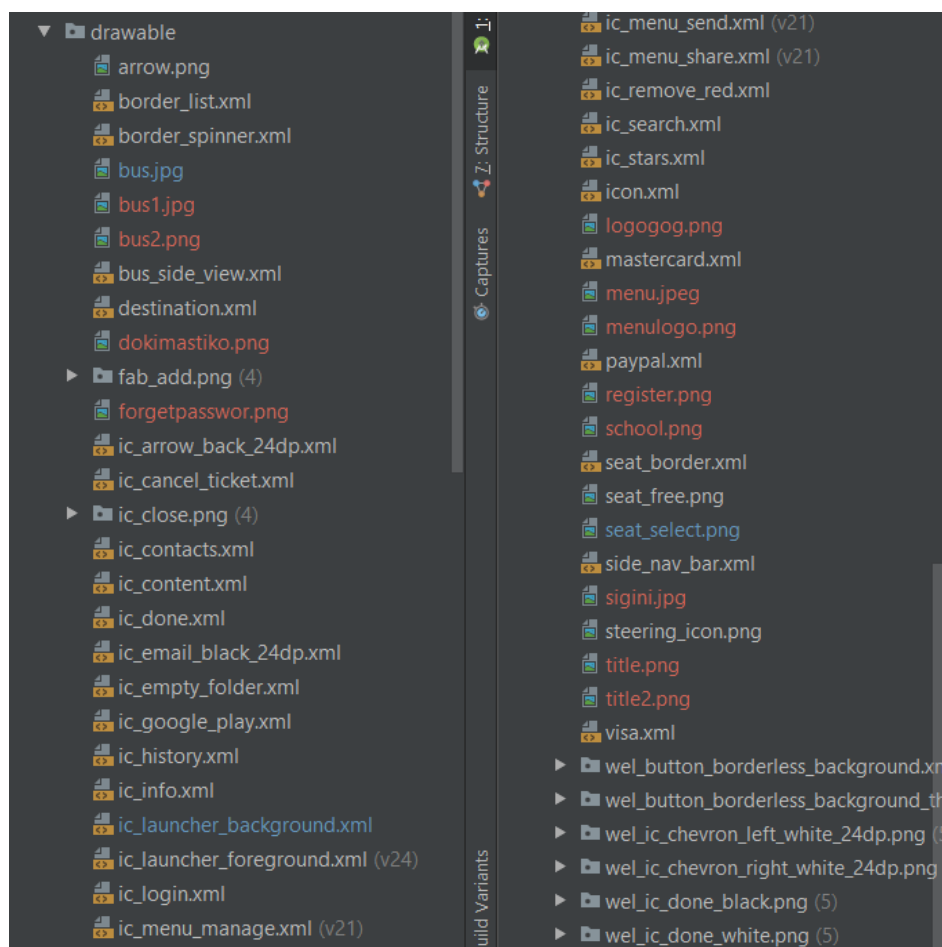
Εικόνα 75 : Φάκελος Res

• Υπόφάκελος Res/Drawable

Στον φάκελο Drawable βρίσκονται κυρίως τα αρχεία που αναπαριστούν στατικές εικόνες, κινούμενες εικόνες και σχήματα. Το Android Studio υποστηρίζει εικόνες τύπου PNG, JPG, GIF. Επίσης υποστηρίζει XML αρχεία τα οποία αναπαριστούν και ορίζουν σχήματα, χρώματα, διαγράμματα, σχέδια κ.α.

Με την δημιουργία ενός Project το Android Studio δημιουργεί αυτόματα τέσσερις υποφάκελους drawable οι οποίοι κατηγοριοποιούνται στις εξής κατηγορίες:

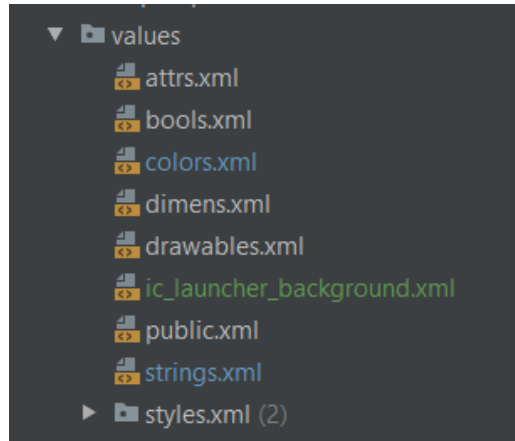
- ldpi (120dpi) – 36x36px
- mdpi (160dpi) – 48x48px
- hdpi (240dpi) – 72x72px
- xhdpi (320dpi) – 96x96px



Εικόνα 76 : Φάκελος Res/Drawable

- **Διάφοροι Τύποι Πόρων (Values)**

Ο συγκεκριμένος υποφάκελος είναι υπεύθυνος για την οργάνωση των διαφορετικών πόρων. Στην εφαρμογή Corfu m-ticket περιέχονται οι συγκεκριμένοι πόροι που βλέπουμε στην εικόνα 74.

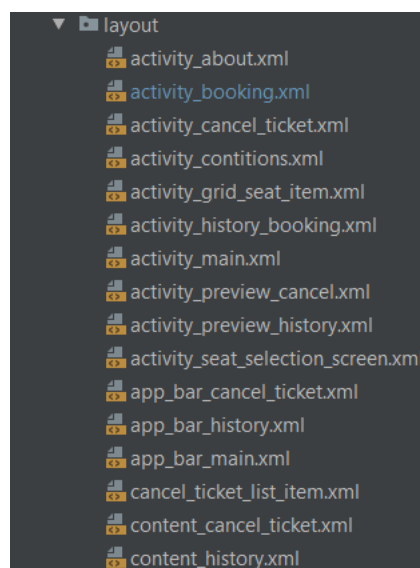


Εικόνα 77 : Φάκελος Values

- **Υποφάκελος Res/Layout**

Η χρήση αρχείων σε μορφή xml είναι ο ευκολότερος τρόπος απεικόνισης γραφικών στοιχείων σε εφαρμογές. Πρόκειται για έναν πολύ εύκολο τρόπο δημιουργίας γραφικού περιβάλλοντος. Ο πηγαίος κώδικας που είναι υπεύθυνος για τις λειτουργίες της εφαρμογής διαχωρίζεται από τα αρχεία που είναι υπεύθυνα για την εμφάνιση της, με αποτέλεσμα ο σχεδιασμός και η αλλαγή της εμφάνισης της να γίνεται πιο εύκολα. Και όλη αυτή η αλλαγή μπορεί να πραγματοποιηθεί χωρίς να τροποποιήσουμε τον πηγαίο κώδικα.

Ο σχεδιασμός της εφαρμογής μπορεί να πραγματοποιηθεί είτε με drag and drop, είτε γράφοντας στο χέρι τον κώδικα εμφάνισης της εφαρμογής ακολουθώντας της φιλοσοφία της διάταξης των αντικειμένων όπως αυτή ορίζεται στην γλώσσα Html. Όλα αυτά τα αρχεία που είναι υπεύθυνα για την εμφάνιση της εφαρμογής μας αποθηκεύονται στον υποφάκελο res/layout.



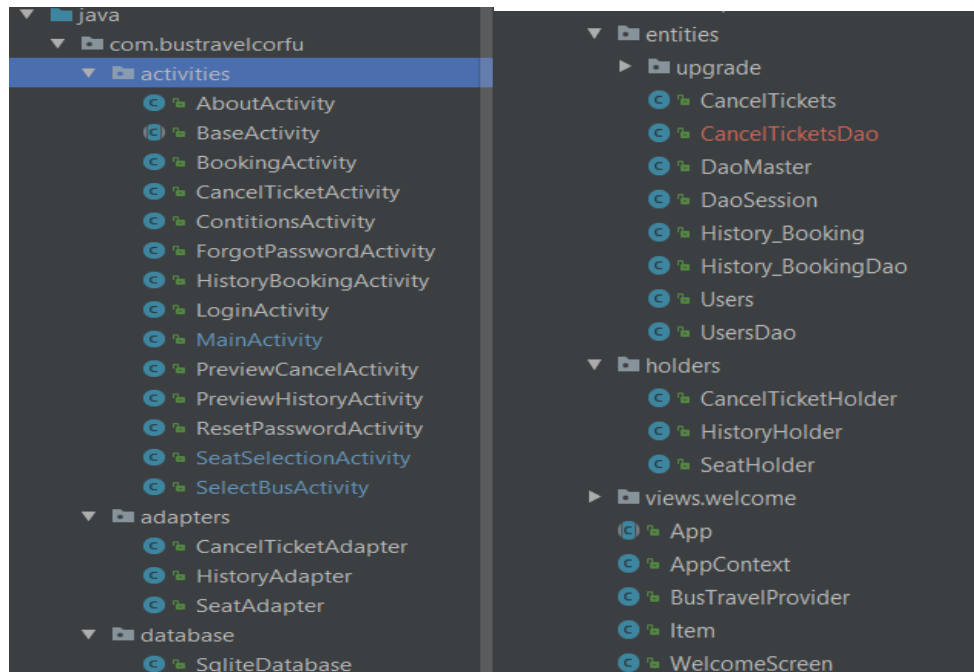
Εικόνα 78 : Φάκελος Res/Layout

- **Φάκελος Java**

Αυτός αποτελεί αν όχι τον σημαντικότερο, έναν από τους σημαντικότερους φακέλους της εφαρμογής. Εδώ περιέχεται όλος ο πηγαίος κώδικας της εφαρμογής. Βρίσκονται όλες οι κλάσεις, οι οποίες διαχωρίζονται σε φακέλους ανάλογα με την λειτουργία τους. Η ταξινόμηση σε φακέλους πραγματοποιείται από τον προγραμματιστή για δικιά του διευκόλυνση.

Κάθε αρχείο Java είναι υπεύθυνο για την λειτουργία ενός Layout.

Ο πηγαίος κώδικας της εφαρμογής Corfu m-ticket είναι ο παρακάτω:

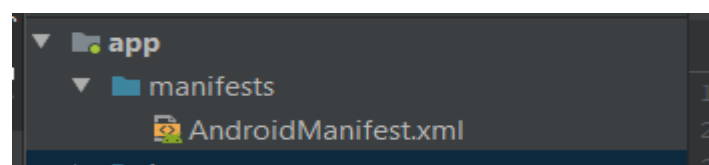


Εικόνα 79 : Φάκελος Java

- **Φάκελος Manifest**

Το manifest είναι το αρχείο στο οποίο ο προγραμματιστής ορίζει της απαιτήσεις της εφαρμογής, δηλώνει τα δικαιώματα της και καθορίζει τις εκδόσεις τις οποίες θα μπορέσουν να το τρέξουν. Είναι φανερό ότι αποτελεί ένα από τα σημαντικότερα αρχεία της εφαρμογής. Για να γίνει αντιληπτό πόσο σημαντικό είναι αυτό το αρχείο ας πούμε απλά ότι εδώ δηλώνονται τα δικαιώματα πρόσβασης που θα έχει η κάθε εφαρμογή στα δεδομένα του χρήστη(επαφές, μηνύματα, τοποθεσία, camera κ.α.).

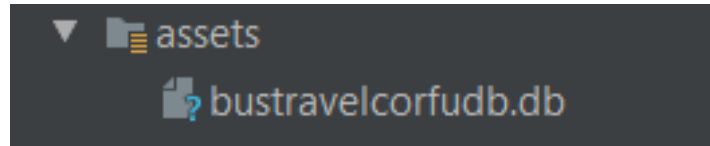
Η κάθε εφαρμογή προκειμένου να γίνει εκτελέσιμη είναι υποχρεωμένη να διαθέτει το συγκεκριμένο αρχείο. Εδώ, δηλώνονται επίσης οι δραστηριότητάς (Activities) μαζί με τις ενέργειες τους. Με την δημιουργία κάθε νέου Project, δηλώνεται αυτόματα η Activity της κεντρικής οθόνης (Main Screen).



Εικόνα 80 : Φάκελος Manifest

- **Φάκελος Assets**

Εδώ συναντάμε οποιοδήποτε πρόσθετο αρχείο θέλουμε να προσθέσουμε στην εφαρμογή. Συνήθως συναντάμε αρχεία που είναι γραμμένα σε javascript/php ή μία βάση δεδομένων.



Εικόνα 81 : Φάκελος Assets

- **Φάκελος Bin**

Τέλος εδώ βρίσκονται τα αρχεία που παράγονται μετά την μεταγλώττιση του πηγαίου κώδικα. Συγκεκριμένα το τελικό αρχείο .apk το οποίο χρησιμοποιείται για την εγκατάσταση της εφαρμογής στις συσκευές. Αξίζει να σημειωθεί ότι το αρχείο αυτό από μόνο του δεν μπορεί να δημοσιευτεί στο Google Play, γιατί απαιτείται να φέρει ψηφιακή υπογραφή.

7. Υλοποίηση Εφαρμογής

Σε αυτό το κεφάλαιο θα ασχοληθούμε κυρίως με το κεντρικό κομμάτι της εφαρμογής, με τον τρόπο δηλαδή που έχει δομηθεί η εφαρμογή καθώς και με την κωδικοποίηση της ως προς το UI και την λειτουργικότητα της. Θα δούμε μέσα από τον κώδικα όλες τις λειτουργίες που προσφέρει η εφαρμογή προκειμένου ο χρήστης να αγοράσει επιτυχώς ένα εισιτήριο για ένα δρομολόγιο της επιλογής του.

7.1 Περιγραφή Υλοποίησης

Προτού προχωρήσουμε στην υλοποίηση της εφαρμογής Corfu m-ticket, στα προηγούμενα κεφάλαια αναλύσαμε τις προδιαγραφές και τις απαιτήσεις της εφαρμογής. Με βάση το μοντέλο προτυποποίησης προσπαθήσαμε να δημιουργήσουμε ένα πρότυπο της εφαρμογής μας. Ακολούθησε η ανάλυση των απαιτήσεων της εφαρμογής με την χρήση διαγράμματος περιπτώσεων χρήσης. Εφόσον αναλύθηκαν σωστά οι απαιτήσεις της εφαρμογής, το επόμενο βήμα είναι η υλοποίηση του project, προγραμματίζοντάς το. Στον πυρήνα της εφαρμογής θα δούμε κώδικες σε γλώσσα προγραμματισμού Java, που θα είναι υπεύθυνοι για την λειτουργικότητα της εφαρμογής. Επίσης θα έρθουμε σε επαφή και με κώδικα γραμμένο σε γλώσσα xml, ο οποίος είναι υπεύθυνος για την μορφοποίηση και την εμφάνιση της εφαρμογής.

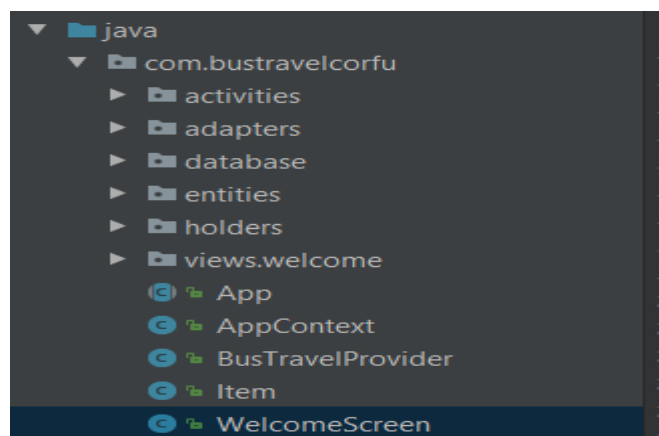
7.2 Πηγαίος Κώδικας-Κωδικοποίηση

7.2.1 Welcome Screen

Αρχικά για την δημιουργία των Welcome Screens έγινε ενσωμάτωση της βιβλιοθήκης stephentuso στην εφαρμογή μας.

Η βιβλιοθήκη περιέχει τα εξής αρχεία:

- WelcomeScreen.java



Εικόνα 78: WelcomeScreen.java

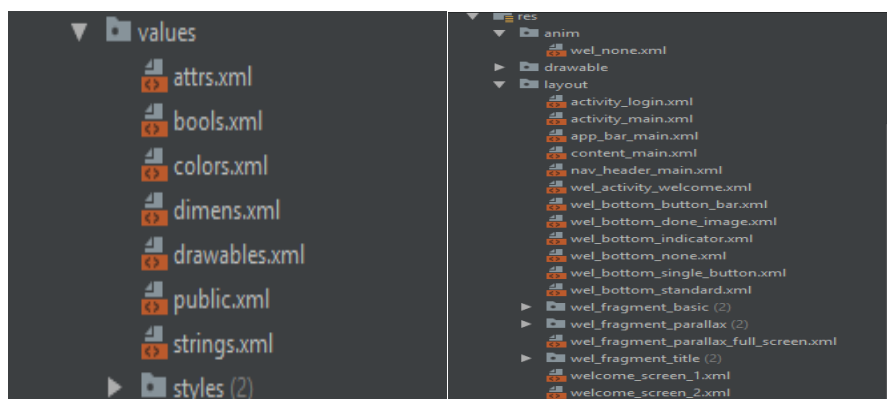
```

1 package com.bustravelcorfu;
2
3 import android.graphics.Typeface;
4 import android.os.Bundle;
5 import android.view.View;
6 import android.widget.Button;
7
8 import com.bustravelcorfu.views.welcome.ParallaxPage;
9 import com.bustravelcorfu.views.welcome.WelcomeActivity;
10 import com.bustravelcorfu.views.welcome.WelcomeConfiguration;
11
12
13 public class WelcomeScreen extends WelcomeActivity implements View.OnClickListener {
14
15     public static final String WELCOME_SCREEN_ACTIVE = "welcome_screen_active";
16     public static final int REQUEST_WELCOME_RESULT_START = 1;
17
18     @Override
19     protected WelcomeConfiguration configuration() {
20
21         return new WelcomeConfiguration.Builder(context, this)
22             .canSkip(false)
23             .backButtonNavigatesPages(false)
24             .bottomLayout(WelcomeConfiguration.BottomLayout.BUTTON_BAR_SINGLE)
25             .page(new ParallaxPage(R.layout.welcome_screen_1, "Corfu m-ticket", "Ταξίδεψε τώρα από την Κέρκυρα προς όλες τις πόλεις της ...").background(R.color.welcome_screen_1_bg))
26             .page(new ParallaxPage(R.layout.welcome_screen_2, "Αγορά Εισιτηρίων", "Αγόρασε το εισιτήριό σου είτε με κερματοκιβώτιο είτε μ...").background(R.color.welcome_screen_2_bg))
27             .build();
28     }
29
30     @Override
31     protected void onCreate(Bundle savedInstanceState) {
32         super.onCreate(savedInstanceState);
33
34         Button dologin = (Button) findViewById(R.id.wel_button_bar_first);
35         if (dologin != null) {
36             dologin.setText("Είσεξ εισιτήριο");
37             dologin.setOnClickListener(this);
38             dologin.setTag(REQUEST_WELCOME_RESULT_START);
39             dologin.setTypeface(Typeface.create(dologin.getTypeface(), Typeface.BOLD));
40         }
41
42         /* Button dodemo = (Button) findViewById(R.id.wel_button_bar_second);
43         if (dodemo != null) {
44             dodemo.setText(getResources().getString(R.string.welcome_button_2));
45             dodemo.setOnClickListener(this);
46             dodemo.setTag(REQUEST_WELCOME_RESULT_DEMO);
47             dodemo.setTypeface(Typeface.create(dodemo.getTypeface(), Typeface.BOLD));
48         } */
49     }
50
51     @Override
52     public void onClick(View v) {
53         setResult((Integer) v.getTag());
54         finish();
55     }
56
57 }
58

```

Εικόνα 82 : Κώδικας του WelcomeScreen.java

- Όλα τα αρχεία μέσα στον φάκελο welcome που βρίσκεται στον φάκελο views
- Το αρχείο wel_none.xml του φακέλου anim που βρίσκεται στον φάκελο resources
- Τα εικονίδια που χρειαζόμαστε βρίσκονται στον φάκελο drawable
- Τα χρώματα, τα κείμενα και τα μεγέθη των γραμματοσειρών που χρησιμοποιούμε βρίσκονται στον φάκελο values, στα αρχεία colors.xml, strings.xml, attrs.xml, dims.xml, styles.xml αντίστοιχα.
- Η σχεδίαση των οθονών βρίσκεται στον φάκελο Layout

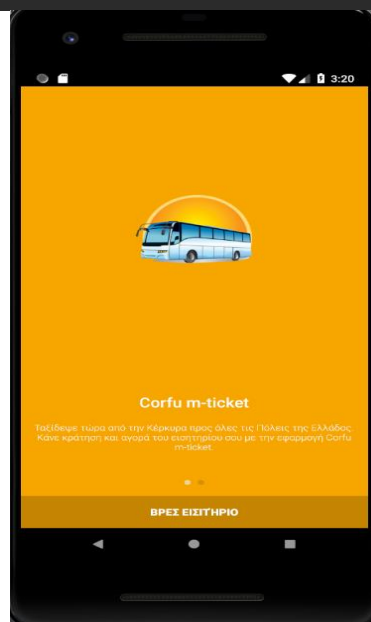


Εικόνα 83: Values και Layout

7.2.2 Σχεδίαση Οθονών Welcome

- Welcome_screen_1.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:app="http://schemas.android.com/apk/res-auto"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   android:clipChildren="false">
7
8   <android.support.v7.widget.AppCompatImageView
9     android:id="@+id/imageLogo"
10    android:layout_width="300dp"
11    android:layout_height="140dp"
12    app:srcCompat="@drawable/bus2"
13    android:layout_centerVertical="true"
14    android:layout_centerHorizontal="true"/>
15
16   <android.support.v7.widget.AppCompatImageView
17     android:layout_width="60dp"
18     android:layout_height="60dp"
19     android:id="@+id/imageView2"
20     android:layout_centerVertical="true"
21     android:layout_centerHorizontal="true"/>
22
23   <android.support.v7.widget.AppCompatImageView
24     android:layout_width="70dp"
25     android:layout_height="70dp"
26     android:id="@+id/imageView5"
27     android:layout_below="@+id/imageView2"/>
28
29 </RelativeLayout>
```



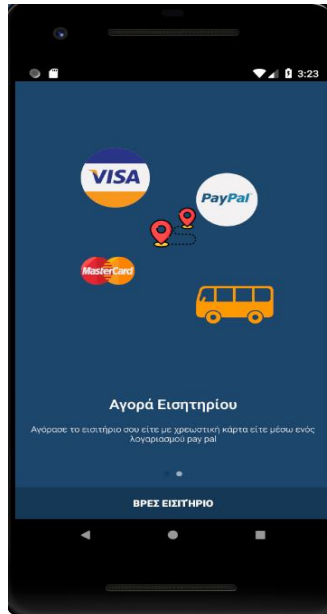
Εικόνα 84 : Welcome_screen_1.xml

- Welcome_screen_2.xml

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      android:clipChildren="false">
8
9      <android.support.v7.widget.AppCompatImageView
10         android:layout_width="60dp"
11         android:layout_height="60dp"
12         app:srcCompat="@drawable/destination"
13         android:id="@+id/imageView2"
14         android:layout_centerVertical="true"
15         android:layout_centerHorizontal="true"/>
16
17         <android.support.v7.widget.AppCompatImageView
18             android:layout_width="70dp"
19             android:layout_height="70dp"
20             app:srcCompat="@drawable/mastercard"
21             android:id="@+id/imageView5"
22             android:layout_below="@+id/imageView2"
23             android:layout_alignLeft="@+id/imageView3"
24             android:layout_alignStart="@+id/imageView3"/>
25
26         <android.support.v7.widget.AppCompatImageView
27             android:layout_width="80dp"
28             android:layout_height="80dp"
29             app:srcCompat="@drawable/paypal"
30             android:id="@+id/imageView"
31             android:layout_alignTop="@+id/imageView3"
32             android:layout_alignLeft="@+id/imageView4"
33             android:layout_alignStart="@+id/imageView4"
34             android:layout_marginTop="38dp"/>
35
36         <android.support.v7.widget.AppCompatImageView
37             android:layout_width="90dp"
38             android:layout_height="90dp"
39             app:srcCompat="@drawable/visa"
40             android:id="@+id/imageView3"
41             android:layout_above="@+id/imageView2"
42             android:layout_toLeftOf="@+id/imageView2"
43             android:layout_toStartOf="@+id/imageView2"
44             tools:ignore="VectorDrawableCompat" />
45
46         <android.support.v7.widget.AppCompatImageView
47             android:layout_width="100dp"
48             android:layout_height="100dp"
49             app:srcCompat="@drawable/bus_side_view"
50             android:id="@+id/imageView4"
51             android:layout_marginTop="36dp"
52             android:layout_alignTop="@+id/imageView5"
53             android:layout_toRightOf="@+id/imageView2"
54             android:layout_toEndOf="@+id/imageView2"/>
55
56  </RelativeLayout>

```



Εικόνα 85 : Welcome_screen_2.xml

7.2.3 LoginActivity.java

Στο LoginActivity.java χρησιμοποιώντας το εργαλείο SharedPreferences ορίζουμε μέσα στην μέθοδο onResume του android ότι το WelcomeScreen θα εμφανιστεί μόνο την πρώτη φορά που θα γίνει εγκατάσταση της εφαρμογής.

Όταν ο χρήστης συνδέεται για πρώτη φορά στην εφαρμογή εκτός από τον κωδικό, καλείται να βάλει και το email του ώστε σε περίπτωση που κάποια στιγμή ξεχάσει τον κωδικό του να μπορεί με το email του να επαναφέρει τον κωδικό.

```

1  package com.bustravelcorfu.activities;
2
3  import ...
4
27
28
29  public class LoginActivity extends BaseActivity {
30      TextView mPassword;
31      TextView mConfirmPass;
32      TextView mEmailUser;
33      List<Users> mUsersList;
34      Button mLoginPin;
35      Button mRegisterPin;
36      AppCompatTextView mForgotPas;
37      Toolbar mActionBarToolbar;
38      private String pas;
39      private ProgressBar spinner;
40      private SharedPreferences prefs;
41      private SharedPreferences.Editor editor;
42      WelcomeHelper mWelcomeScreen;
43
44      @Override
45      public void onCreate(Bundle savedInstanceState) {
46          super.onCreate(savedInstanceState);
47          // get shared preferences
48          prefs = getSharedPreferences("MasterPrefs", MODE_PRIVATE);
49          editor = prefs.edit();
50      }
51
52      public void loginPinUser() {
53          final String cPassword = mPassword.getText().toString();
54
55          if (cPassword.equals(pas)) {
56              spinner.setVisibility(View.GONE);
57              spinner.setVisibility(View.VISIBLE);
58              Intent detailsScreen = new Intent(getApplicationContext(), MainActivity.class);
59              startActivity(detailsScreen);
60          } else {
61              Toast.makeText(getApplicationContext(), "Wrong password try again", Toast.LENGTH_LONG).show();

```

```

62     }
63 }
64
65 public void registerPinUser() {
66     String cPassword = mPassword.getText().toString();
67     String cConfirmPass = mConfirmPass.getText().toString();
68     String cEmail = mEmailUser.getText().toString();
69
70     ContentValues values = new ContentValues();
71     if (cPassword.isEmpty() || cConfirmPass.isEmpty() || cEmail.isEmpty()) {
72         Toast.makeText(getApplicationContext(), "Wrong password. Password is empty", Toast.LENGTH_LONG).show();
73     } else if (cPassword.length() < 4) {
74         Toast.makeText(getApplicationContext(), "Password must be at least 4 digits", Toast.LENGTH_LONG).show();
75     } else if (!cPassword.equals(cConfirmPass)) {
76         Toast.makeText(getApplicationContext(), "Password is different with confirm Password", Toast.LENGTH_LONG).show();
77     } else {
78         values.put(BusTravelProvider.COLUMNV_USER_PASSWORD, cPassword);
79         values.put(BusTravelProvider.COLUMNV_USERNAME, cEmail);
80         getContentResolver().insert(BusTravelProvider.CONTENT_USERS_URI_INSERT, values);
81         Intent intent = new Intent(getApplicationContext(), MainActivity.class);
82         startActivity(intent);
83     }
84 }
85
86 public void initializeMainActivity() {
87     setContentView(R.layout.layout_register_pin);
88
89     RelativeLayout linearLayout = (RelativeLayout) findViewById(R.id.register_pin);
90     linearLayout.setBackgroundColor(getResources().getColor(R.color.colorBorder));
91     mActionBarToolbar = (android.support.v7.widget.Toolbar) findViewById(R.id.toolbar);
92     setSupportActionBar(mActionBarToolbar);
93
94     TextInputLayout confirmPassword = (TextInputLayout) findViewById(R.id.register_password_wrapper);
95     TextInputLayout email_register = (TextInputLayout) findViewById(R.id.register_email_wrapper);
96     TextView algorithmText = (TextView) findViewById(R.id.register_main);
97     TextView createPassword = (TextView) findViewById(R.id.create_password);
98     TextView securePassword = (TextView) findViewById(R.id.secure_password);
99
100
101     mUsersList = new ArrayList<>();
102
103     mPassword = (TextView) findViewById(R.id.password_register);
104     mConfirmPass = (TextView) findViewById(R.id.password_confirm);
105     mEmailUser = (TextView) findViewById(R.id.email_register);
106     mLoginPin = (Button) findViewById(R.id.login_pin);
107     mRegisterPin = (Button) findViewById(R.id.register);
108     spinner = (ProgressBar) findViewById(R.id.progressBar1);
109     mForgotPas = (AppCompatTextView) findViewById(R.id.forgotPassword);
110
111     String URL = "content://com.bustravelcorfu.provider/" + BusTravelProvider.USERS_PATH + "";
112     final Uri _users = Uri.parse(URL);
113     final Cursor cursor = getContentResolver().query(_users, projection, null, selection, null, selectionArgs, null, sortOrder, null);
114
115     if (cursor != null) {
116         while (cursor.moveToNext()) {
117             Users users = new Users();
118             users.setId(cursor.getLong( columnIndex 0));
119             users.setUsername(cursor.getString( columnIndex 1));
120             users.setPassword(cursor.getString( columnIndex 2));
121             users.setPinuser(cursor.getString( columnIndex 3));
122             mUsersList.add(users);
123         }
124     }
125
126     for (int i = 0; i < mUsersList.size(); i++) {
127         pas = mUsersList.get(i).getPassword();
128     }
129
130     if (mUsersList.size() > 0) {
131         confirmPassword.setVisibility(View.INVISIBLE);
132         algorithmText.setVisibility(View.INVISIBLE);
133         mRegisterPin.setVisibility(View.INVISIBLE);
134         createPassword.setVisibility(View.INVISIBLE);
135         mEmailUser.setVisibility(View.INVISIBLE);
136         email_register.setVisibility(View.INVISIBLE);
137         mLoginPin.setVisibility(View.VISIBLE);
138
139         mLoginPin.setVisibility(View.VISIBLE);
140         securePassword.setVisibility(View.VISIBLE);
141         mForgotPas.setVisibility(View.VISIBLE);
142
143     } else {
144         confirmPassword.setVisibility(View.VISIBLE);
145         algorithmText.setVisibility(View.VISIBLE);
146         mRegisterPin.setVisibility(View.VISIBLE);
147         createPassword.setVisibility(View.VISIBLE);
148         mEmailUser.setVisibility(View.VISIBLE);
149         email_register.setVisibility(View.VISIBLE);
150         mLoginPin.setVisibility(View.INVISIBLE);
151         securePassword.setVisibility(View.INVISIBLE);
152         mForgotPas.setVisibility(View.INVISIBLE);
153     }
154
155     mLoginPin.setOnClickListener((view) -> { loginPinUser(); });
156
157     mRegisterPin.setOnClickListener((view) -> { registerPinUser(); });
158
159     mForgotPas.setOnClickListener((w) -> {
160         Intent forgotPasActivity = new Intent(getApplicationContext(), ForgotPasswordActivity.class);
161         startActivity(forgotPasActivity);
162     });
163 }
164
165 @Override
166 public void onResume() {
167     super.onResume();
168     // first run preference
169     if (prefs.getBoolean( key: "firstrun", defValue: true)) { // get firstrun preference
170         mWelcomeScreen = new WelcomeHelper( activity: this, WelcomeScreen.class);
171         mWelcomeScreen.forceShow(); // force show welcome screen
172         editor.putBoolean("firstrun", false).apply(); // put firstrun preference
173     } else {
174         initializeMainActivity(); // initialize main activity
175     }
176 }
177
178 }

```

Εικόνα 86 : LoginActivity.java

7.2.4 Σχεδίαση Οθόνης Login

- Layout_register_pin.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   xmlns:app="http://schemas.android.com/apk/res-auto"
5   android:id="@+id/register_pin"
6   android:layout_width="match_parent"
7   android:layout_height="match_parent"
8   android:fitsSystemWindows="true"
9   tools:context=".activities.LoginActivity">
10   <android.support.v7.widget.Toolbar...>
11   <LinearLayout
12     android:layout_width="0px"
13     android:layout_height="0px"
14     android:focusable="true"
15     android:focusableInTouchMode="true">
16
17     <requestFocus/>
18   </LinearLayout>
19
20   <ProgressBar
21     android:id="@+id/progressBar1"
22     style="?android:attr/progressBarStyleLarge"
23     android:layout_width="wrap_content"
24     android:layout_height="wrap_content"
25     android:layout_centerHorizontal="true"
26     android:layout_centerVertical="true"
27     android:layout_marginBottom="16dp"
28     android:visibility="gone"/>
29
30   <RelativeLayout
31     android:id="@+id/register_title"
32     android:layout_width="match_parent"
33     android:layout_height="match_parent"
34     android:layout_above="@id/register_form">
35
36     <ImageView
37       android:id="@+id/imageView2"
38       android:layout_width="match_parent"
39       android:layout_height="match_parent"
40       android:layout_height="144dp"
41       android:layout_alignBottom="@id/secure_password"
42       android:background="@color/colorPrimary"
43       app:srcCompat="@drawable/title" />
44
45     <TextView
46       android:id="@+id/register_main"
47       android:layout_width="300dp"
48       android:layout_height="wrap_content"
49       android:layout_centerHorizontal="true"
50       android:layout_centerVertical="true"
51       android:gravity="center"
52       android:textColor="@color/colorBlack"
53       android:textStyle="bold"
54       android:textAlignment="center"
55       android:textSize="14sp"
56       android:visibility="invisible"
57     />
58
59     <TextView
60       android:id="@+id/create_password"
61       android:layout_width="wrap_content"
62       android:layout_height="wrap_content"
63       android:layout_alignTop="@id/secure_password"
64       android:layout_centerHorizontal="true"
65       android:textColor="@color/colorPrimary"
66       android:textSize="21sp"
67       android:visibility="invisible"
68       android:textStyle="bold"
69     />
70
71     <TextView
72       android:id="@+id/secure_password"
73       android:layout_width="wrap_content"
74       android:layout_height="wrap_content"
75       android:layout_below="@id/register_main"
76     />
77
78   </RelativeLayout>
79
80   <TextView
81     android:id="@+id/register_main"
82     android:layout_width="300dp"
83     android:layout_height="wrap_content"
84     android:layout_centerHorizontal="true"
85     android:layout_centerVertical="true"
86     android:gravity="center"
87     android:textColor="@color/colorBlack"
88     android:textStyle="bold"
89     android:textAlignment="center"
90     android:textSize="14sp"
91     android:visibility="invisible"
92   />
93
94   <TextView
95     android:id="@+id/create_password"
96     android:layout_width="wrap_content"
97     android:layout_height="wrap_content"
98     android:layout_alignTop="@id/secure_password"
99     android:layout_centerHorizontal="true"
100    android:textColor="@color/colorPrimary"
101    android:textSize="21sp"
102    android:visibility="invisible"
103    android:textStyle="bold"
104  />
105
106   <TextView
107     android:id="@+id/secure_password"
108     android:layout_width="wrap_content"
109     android:layout_height="wrap_content"
110     android:layout_below="@id/register_main"
111  />
112</RelativeLayout>
```

```

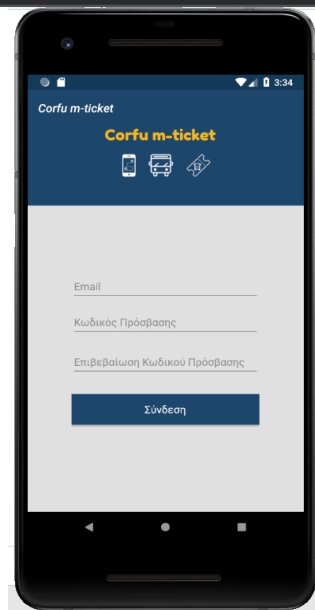
83         android:layout_below="@+id/register_main"
84         android:layout_centerHorizontal="true"
85         android:layout_marginTop="16dp"
86         android:text=""
87         android:textColor="@color/colorAccent"
88         android:textSize="22sp"
89         android:textStyle="bold"
90         android:visibility="visible" />
91     </RelativeLayout>
92
93     <RelativeLayout
94         android:id="@+id/register_form"
95         android:layout_width="280dp"
96         android:layout_height="wrap_content"
97         android:layout_alignParentBottom="true"
98         android:layout_centerHorizontal="true"
99         android:layout_marginBottom="94dp"
100        android:paddingTop="25dp">
101
102        <android.support.design.widget.TextInputLayout
103            android:id="@+id/register_email_wrapper"
104            android:layout_width="match_parent"
105            android:layout_height="56dp"
106            android:visibility="invisible">
107
108            <android.support.design.widget.TextInputEditText
109                android:id="@+id/email_register"
110                android:layout_width="match_parent"
111                android:layout_height="wrap_content"
112                android:hint="Email"
113                android:inputType="textEmailAddress"
114                android:maxLines="1"
115                android:textColor="@color/colorBlack"
116                android:textSize="16sp" />
117        </android.support.design.widget.TextInputLayout>
118
119        <android.support.design.widget.TextInputLayout
120            android:id="@+id/register_pas_wrapper"
121            android:layout_width="match_parent"
122            android:layout_height="56dp"
123            android:layout_below="@+id/register_email_wrapper">
124
125            <android.support.design.widget.TextInputEditText
126                android:id="@+id/password_register"
127                android:layout_width="match_parent"
128                android:layout_height="wrap_content"
129                android:hint="Κωδικός Πρόσβασης"
130                android:inputType="textPassword"
131                android:maxLines="1"
132                android:textColor="@color/colorBlack"
133                android:textSize="16sp" />
134        </android.support.design.widget.TextInputLayout>
135
136        <android.support.design.widget.TextInputLayout
137            android:id="@+id/register_password_wrapper"
138            android:layout_width="match_parent"
139            android:layout_height="56dp"
140            android:layout_below="@+id/register_pas_wrapper"
141            android:layout_marginTop="6dp"
142            android:visibility="invisible">
143
144            <android.support.design.widget.TextInputEditText
145                android:id="@+id/password_confirm"
146                android:layout_width="match_parent"
147                android:layout_height="wrap_content"
148                android:hint="Επβεβαίωση Κωδικού Πρόσβασης"
149                android:inputType="textPassword"
150                android:maxLines="1"
151                android:textColor="@color/colorBlack"
152                android:textSize="16sp" />
153        </android.support.design.widget.TextInputLayout>
154
155        <Button
156            android:id="@+id/register"
157            style="@style/Base.TextAppearance.AppCompat.Small"

```

```

159         android:layout_width="match_parent"
160         android:layout_height="wrap_content"
161         android:layout_alignParentStart="true"
162         android:layout_below="@+id/register_password_wrapper"
163         android:layout_marginTop="25dp"
164         android:background="@color/colorPrimary"
165         android:fontFamily="sans-serif-thin"
166         android:text="Σύνδεση"
167         android:textAllCaps="false"
168         android:textColor="@color/colorWhite"
169         android:textSize="16sp"
170         android:textStyle="bold"
171         android:visibility="visible" />
172
173     <android.support.v7.widget.AppCompatTextView
174         android:id="@+id/forgotPassword"
175         android:layout_width="match_parent"
176         android:layout_height="wrap_content"
177         android:layout_below="@+id/register"
178         android:layout_marginTop="20dp"
179         android:gravity="center"
180         android:text="Ξεχάσατε τον Κωδικό σας;"
181         android:textColor="@color/colorAccent"
182         android:textSize="16sp" />
183
184     <Button
185         android:id="@+id/login_pin"
186         style="@style/Base.TextAppearance.AppCompat.Small"
187         android:layout_width="match_parent"
188         android:layout_height="wrap_content"
189         android:layout_alignBaseline="@+id/register"
190         android:layout_alignBottom="@+id/register"
191         android:layout_alignParentStart="true"
192         android:background="@color/colorPrimary"
193         android:fontFamily="sans-serif-thin"
194         android:text="Σύνδεση"
195         android:textAllCaps="false"
196         android:textColor="@color/colorWhite"
197         android:textSize="16sp"
198         android:textStyle="bold"
199         android:visibility="invisible" />
200 </RelativeLayout>
201
202 </RelativeLayout>

```

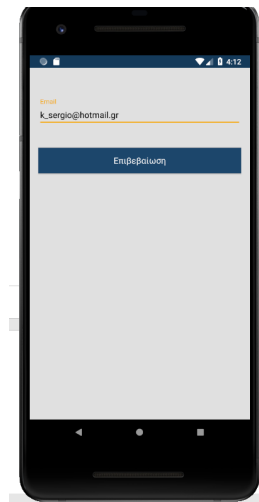


Εικόνα 87 : Layout_register_pin.xml

7.2.5 Σχεδίαση Οθόνης Forgot and Reset Password

- Layout_reset_password.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2
3
4
5 <LinearLayout
6     xmlns:android="http://schemas.android.com/apk/res/android"
7     android:id="@+id/resetPasswordLayout"
8     xmlns:app="http://schemas.android.com/apk/res-auto"
9     android:layout_width="match_parent"
10    android:layout_height="match_parent"
11    android:orientation="vertical"
12    android:background="@color/greyish"
13    android:paddingBottom="16dp"
14    android:paddingLeft="16dp"
15    android:paddingRight="16dp"
16    android:paddingTop="16dp">
17
18    <ImageView
19        android:id="@+id/imageView2"
20        android:layout_width="match_parent"
21        android:layout_height="144dp"
22        android:layout_alignBottom="@+id/secure_password"
23        android:background="@color/colorPrimary"
24        app:srcCompat="@drawable/title" />
25
26
27
28    <android.support.design.widget.TextInputLayout
29        android:id="@+id/reset_pas_wrapper"
30        android:layout_width="match_parent"
31        android:layout_height="56dp">
32
33        <android.support.design.widget.TextInputEditText
34            android:id="@+id/password_reset"
35            android:layout_width="match_parent"
36            android:layout_height="wrap_content"
37            android:hint="Κωδικός Πρόσβασης"
38            android:inputType="textPassword"
39            android:maxLines="1"
40            android:textColor="@color/colorBlack"
41            android:textSize="16sp"/>
42    </android.support.design.widget.TextInputLayout>
43
44
45    <android.support.design.widget.TextInputLayout
46        android:id="@+id/reset_password_wrapper"
47        android:layout_width="match_parent"
48        android:layout_height="56dp"
49        android:layout_marginTop="6dp">
50
51        <android.support.design.widget.TextInputEditText
52            android:id="@+id/password_reset_confirm"
53            android:layout_width="match_parent"
54            android:layout_height="wrap_content"
55            android:hint="Επιβεβαίωση Κωδικού Πρόσβασης"
56            android:inputType="textPassword"
57            android:maxLines="1"
58            android:textColor="@color/colorBlack"
59            android:textSize="16sp"/>
60    </android.support.design.widget.TextInputLayout>
61
62    <Button
63        android:id="@+id/buttonReset"
64        style="@style/Base.TextAppearance.AppCompat.Small"
65        android:layout_width="match_parent"
66        android:layout_height="wrap_content"
67        android:background="@color/colorPrimary"
68        android:fontFamily="sans-serif-thin"
69        android:text="Εναλλακτικά"
70        android:textAllCaps="false"
71        android:textColor="@color/colorWhite"
72        android:textSize="16sp"
73        android:textStyle="bold"/>
74 </LinearLayout>
75
```



Εικόνα 88 : Layout_reset_password.xml

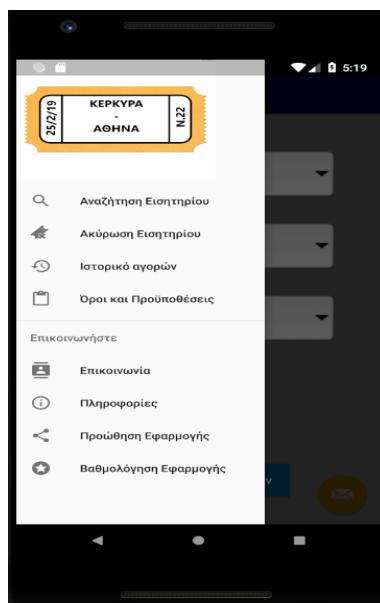
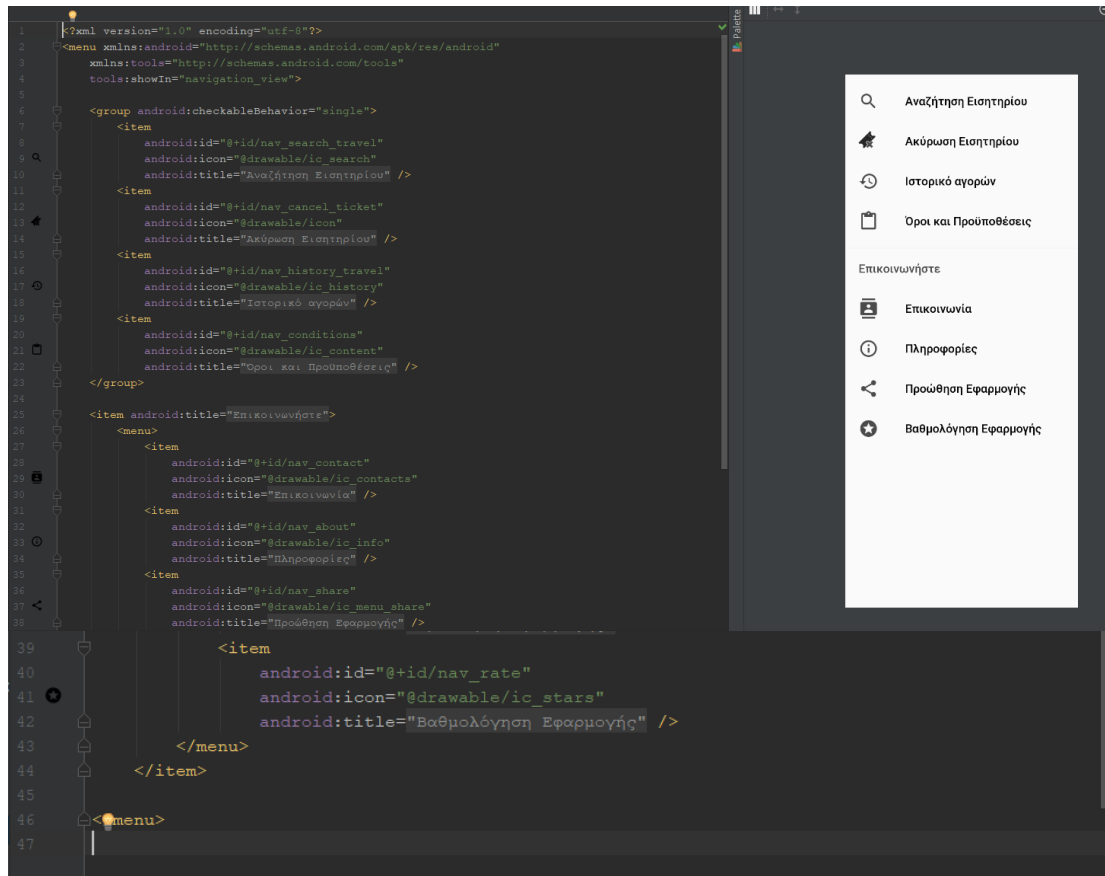
- Layout_forgot_password.xml

```
<< android.support.design.widget.TextInputLayout android.support.design.widget.TextInputEditText
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      android:id="@+id/forgotPasswordLayout"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      android:orientation="vertical"
8      android:background="@color/greyish"
9      android:paddingBottom="16dp"
10     android:paddingLeft="16dp"
11     android:paddingRight="16dp"
12     android:paddingTop="16dp">
13     <android.support.design.widget.TextInputLayout
14         android:id="@+id/textInputLayoutEmail"
15         android:layout_width="match_parent"
16         android:layout_height="wrap_content"
17         android:textColorHint="@color/colorBlack"
18         android:layout_marginTop="40dp" >
19
20         <android.support.design.widget.TextInputEditText
21             android:id="@+id/textEmailConfirm"
22             android:layout_width="match_parent"
23             android:layout_height="wrap_content"
24             android:textSize="16sp"
25             android:hint="Email"
26             android:inputType="text"
27             android:maxLines="1"
28             android:textColor="@color/colorBlack" />
29     </android.support.design.widget.TextInputLayout>
30
31     <Button
32         android:id="@+id/emailButtonConfirm"
33         style="@style/Base.TextAppearance.AppCompat.Small"
34         android:layout_width="match_parent"
35         android:layout_height="wrap_content"
36         android:layout_marginTop="40dp"
37         android:background="@color/colorPrimary"
38         android:fontFamily="sans-serif-thin"
39         android:text="Επιβεβαίωση"
40         android:textAllCaps="false"
41         android:textColor="@color/colorWhite"
42         android:textSize="16sp"
43         android:textStyle="bold"/>
44 </LinearLayout>
```

Εικόνα 89 : Layout_forgot_password.xml

7.2.6 Δημιουργία Μενού

- activity_main_drawer.xml



Εικόνα 90 : activity_main_drawer.xml

- MainActivity.java

Περιήγηση στο μενού και κώδικας για τις λειτουργίες κοινοποίηση, βαθμολογία, επικοινωνία.

```
public boolean onNavigationItemSelected(MenuItem item) {
    // Handle navigation view item clicks here.
    int id = item.getItemId();

    if (id == R.id.nav_login) {
        Intent login_activity = new Intent(this,
LoginActivity.class);
        startActivity(login_activity);
    } else if (id == R.id.nav_search_travel) {

    } else if (id == R.id.nav_cancel_ticket) {

    } else if (id == R.id.nav_history_travel) {

    } else if (id == R.id.nav_conditions) {

    } else if (id == R.id.nav_share) {
        Intent sharingIntent = new Intent(Intent.ACTION_SEND);
        sharingIntent.setAction(Intent.ACTION_SEND);
        sharingIntent.putExtra(Intent.EXTRA_TEXT, "Αγοράσε το
εισήτηριο σου και ταξίδεψε από την Κέρκυρα στις πόλεις της Ελλάδος
μέσω της εφαρμογής Bus Travel Corfu \n\n
https://play.google.com/store");/*replace url from app in play
store*/
        sharingIntent.setType("text/plain");
        startActivity(Intent.createChooser(sharingIntent, "Share
via"));
    } else if (id == R.id.nav_rate) {
        Intent intent = new Intent(Intent.ACTION_VIEW)
            .setData(Uri
                .parse("https://play.google.com/store"));/*replace url from app in
play store */
        startActivity(intent);
    } else if (id == R.id.nav_contact) {
        Intent Email = new Intent(Intent.ACTION_SEND);
        Email.setType("text/email");
        Email.putExtra(Intent.EXTRA_EMAIL, new
String[]{"your_email@gmail.com"});/*replace with your email*/
        Email.putExtra(Intent.EXTRA_SUBJECT, "Feedback from " +
Build.MODEL + " user");
        Email.putExtra(Intent.EXTRA_TEXT, "Dear Bus Travel team," +
        "");
        startActivity(Intent.createChooser(Email, "Send Feedback:"));
    } else if (id == R.id.nav_about) {

    }

    DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer_layout);
    drawer.closeDrawer(GravityCompat.START);
    return true;
}
```

Εικόνα 91: MainActivity.java

7.2.7 Δημιουργία Βάσεις Δεδομένων

Δημιουργία Βάσεις δεδομένων με το όνομα **bustravel.db**. Στην ουσία έχουμε δημιουργήσει τον πίνακα Users(Χρήστες) για να αποθηκευτεί ο κωδικός ασφαλείας του χρήστη και τα πεδία που χρειαζόμαστε. Στην εφαρμογή μας χρησιμοποιούμε την βιβλιοθήκη GreenDao για την βοήθεια της δημιουργίας της βάσης. Επίσης η βιβλιοθήκη αυτή προσφέρει ταχύτητα , ασφάλεια και κάποιες λειτουργίες της βάσης.

Προκειμένου να προσθέσουμε την βιβλιοθήκη στην εφαρμογή μας αρκεί να προσθέσουμε τον εξής κώδικα στο buildgradle αρχείο:

- build.gradle(Corfu_m-ticket)

```
1 // Top-level build file where you can add configuration options common to all sub-projects/modules.
2
3 buildscript {
4
5     repositories {
6         google()
7         jcenter()
8     }
9     dependencies {
10         classpath 'com.android.tools.build:gradle:3.1.4'
11         classpath 'org.greenrobot:greendao-gradle-plugin:3.2.2'
12
13         // NOTE: Do not place your application dependencies here; they belong
14         // in the individual module build.gradle files
15     }
16 }
17
18 allprojects {
19     repositories {
20         google()
21         jcenter()
22     }
23 }
24
25 task clean(type: Delete) {
26     delete rootProject.buildDir
27 }
28
```

Εικόνα 92 : build.gradle

- Build.gradle(Module app)

```
1 apply plugin: 'com.android.application'
2
3 android {
4     compileSdkVersion 27
5     defaultConfig {
6         applicationId "com.bustravelcorfu"
7         minSdkVersion 19
8         targetSdkVersion 27
9         versionCode 1
10        versionName "1.0"
11        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
12        vectorDrawables {
13            useSupportLibrary = true
14        }
15    }
16    buildTypes {
17        release {
18            minifyEnabled true
19            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
20        }
21    }
22 }
23
24 // In your app projects build.gradle file:
25 apply plugin: 'com.android.application'
26 apply plugin: 'org.greenrobot.greendao' // apply plugin
27
28 greendao {
29     schemaVersion 1
30     targetGenDir "src/main/java/"
31 }
32
33 dependencies {
34     implementation fileTree(dir: 'libs', include: ['*.jar'])
35     implementation 'com.android.support:appcompat-v7:27.1.1'
36     implementation 'com.android.support:design:27.1.1'
37     implementation 'com.android.support.constraint:constraint-layout:1.1.2'
38     testImplementation 'junit:junit4.12'
39     androidTestImplementation 'com.android.support.test:runner:1.0.2'
40     androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
41
42     implementation 'org.greenrobot:greendao:3.2.2' // add library
43     implementation 'net.zetetic:android-database-sqlcipher:3.5.6'
44     implementation 'com.paypal.sdk:paypal-android-sdk:2.16.0'
45
46     implementation 'com.android.support:preference-v7:27.1.1'
47     implementation 'com.android.support:cardview-v7:27.1.1'
48 }
49
50
```

Εικόνα 93: build.gradle

- SqliteDatabase.java

```

1 package com.bustravelcorfu.database;
2
3 import ...
4
5
6
7
8
9
10
11 public class SqliteDatabase extends SQLiteOpenHelper {
12     //Database Name
13     private static final String DATABASE_NAME = "bustravel.db";
14
15     //Database Version
16     private static final int DATABASE_VERSION = 1;
17     //Tables
18     private static final String TABLE_USERS = UsersDao.TABLENAME;
19     private static final String TABLE_HISTORY_BOOKING = History_BookingDao.TABLENAME;
20     private static final String TABLE_CANCEL_TICKET = CancelTicketsDao.TABLENAME;
21
22     //Users Columns
23     private static final String COLUMN_USER_ID = UsersDao.Properties.Id.columnName;
24     private static final String COLUMN_USERNAME = UsersDao.Properties.Username.columnName;
25     private static final String COLUMN_USER_PASSWORD = UsersDao.Properties.Password.columnName;
26     private static final String COLUMN_PIN_USER = UsersDao.Properties.Pinuser.columnName;
27
28     //History Booking Columns
29     private static final String COLUMN_HISTORY_BOOK_ID = History_BookingDao.Properties.Id.columnName;
30     private static final String COLUMN_DEPARTURE_POINT = History_BookingDao.Properties.Departure_point.columnName;
31     private static final String COLUMN_DESTINATION = History_BookingDao.Properties.Destination.columnName;
32     private static final String COLUMN_NUMBER_OF_TICKETS = History_BookingDao.Properties.Number_of_tickets.columnName;
33     private static final String COLUMN_DATE = History_BookingDao.Properties.Date.columnName;
34     private static final String COLUMN_TIME_TRAVEL = History_BookingDao.Properties.Time_travel.columnName;
35     private static final String COLUMN_NUMBER_OF_SEAT = History_BookingDao.Properties.Number_of_seat.columnName;
36     private static final String COLUMN_PRICE = History_BookingDao.Properties.Price.columnName;
37
38     //Cancel Tickets Columns
39     private static final String COLUMN_CANCEL_TICKET_ID = CancelTicketsDao.Properties.Id.columnName;
40     private static final String COLUMN_CA_DEPARTURE_POINT = CancelTicketsDao.Properties.Capoint.columnName;
41     private static final String COLUMN_CA_DESTINATION = CancelTicketsDao.Properties.Cadestination.columnName;
42     private static final String COLUMN_CA_TICKETS = CancelTicketsDao.Properties.Catickets.columnName;
43     private static final String COLUMN_CA_DATE = CancelTicketsDao.Properties.Cadate.columnName;
44     private static final String COLUMN_CA_TIME_TRAVEL = CancelTicketsDao.Properties.Catimetravel.columnName;
45     private static final String COLUMN_CA_NUMBER_SEAT = CancelTicketsDao.Properties.Canumberofseat.columnName;
46     private static final String COLUMN_CA_PRICE = CancelTicketsDao.Properties.Caprice.columnName;
47
48     private static SqliteDatabase instance;
49
50     public SqliteDatabase(Context context) {
51         super(context, DATABASE_NAME, null, DATABASE_VERSION);
52     }
53
54     @Override
55     public void onCreate(SQLiteDatabase db) {
56         String CREATE_USERS_TABLE = "CREATE TABLE " + TABLE_USERS + "(" + COLUMN_USER_ID + " INTEGER PRIMARY KEY," + COLUMN_USERNAME + " TEXT," + COLUMN_USER_PASSWORD + " TEXT," + COLUMN_PIN_USER + " TEXT)";
57         String CREATE_HISTORY_BOOKING_TABLE = "CREATE TABLE " + TABLE_HISTORY_BOOKING + "(" + COLUMN_HISTORY_BOOK_ID + " INTEGER PRIMARY KEY," + COLUMN_DEPARTURE_POINT + " TEXT," + COLUMN_DESTINATION + " TEXT," + COLUMN_NUMBER_OF_TICKETS + " TEXT," + COLUMN_DATE + " TEXT," + COLUMN_TIME_TRAVEL + " TEXT," + COLUMN_NUMBER_OF_SEAT + " TEXT," + COLUMN_PRICE + " TEXT)";
58         String CREATE_CANCEL_TICKETS_TABLE = "CREATE TABLE " + TABLE_CANCEL_TICKET + "(" + COLUMN_CANCEL_TICKET_ID + " INTEGER PRIMARY KEY," + COLUMN_CA_DEPARTURE_POINT + " TEXT," + COLUMN_CA_DESTINATION + " TEXT," + COLUMN_CA_TICKETS + " TEXT," + COLUMN_CA_DATE + " TEXT," + COLUMN_CA_TIME_TRAVEL + " TEXT," + COLUMN_CA_NUMBER_SEAT + " TEXT," + COLUMN_CA_PRICE + " TEXT)";
59         db.execSQL(CREATE_USERS_TABLE);
60         db.execSQL(CREATE_HISTORY_BOOKING_TABLE);
61         db.execSQL(CREATE_CANCEL_TICKETS_TABLE);
62     }
63
64     @Override
65     public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
66         db.execSQL("DROP TABLE IF EXISTS " + TABLE_USERS);
67         db.execSQL("DROP TABLE IF EXISTS " + TABLE_HISTORY_BOOKING);
68         db.execSQL("DROP TABLE IF EXISTS " + TABLE_CANCEL_TICKET);
69     }
70
71     public static SqliteDatabase get(Context ctx) {
72         if (instance == null) {
73             instance = new SqliteDatabase(ctx);
74         }
75         return instance;
76     }
77 }

```

Εικόνα 94 : SqliteDatabase.java

Δημιουργούμε έναν φάκελο **entities** και μέσα εκεί δημιουργούμε τους πίνακες που θέλουμε. Για να δημιουργήσουμε τον πίνακα Users δημιουργούμε μια κλάση με όνομα Users και πάνω από την κλάση δηλώνουμε ότι είναι @Entity. Στη συνέχεια δηλώνουμε μέσα στην κλάση τα πεδία του πίνακα κάτω από το @Id και πατάμε build. Το Greendao θα δημιουργήσει όλη την λειτουργία της βάσης μας.

Όλος ο κώδικας της βάσης και τα αρχεία greendao δημιουργούνται αυτόματα μετά το build.

```

package com.bustravelcorfu.entities;

import org.greenrobot.greendao.annotation.Entity;
import org.greenrobot.greendao.annotation.Id;
import org.greenrobot.greendao.annotation.Generated;

@Entity
public class Users {
    @Id
    private Long id;
    private String username;
    private String password;
    private String pinuser;
    @Generated(hash = 2021454661)
    public Users(Long id, String username, String password, String
pinuser) {
        this.id = id;
        this.username = username;
        this.password = password;
        this.pinuser = pinuser;
    }
    @Generated(hash = 2146996206)
    public Users() {
    }
    public Long getId() {
        return this.id;
    }
    public void setId(Long id) {
        this.id = id;
    }
    public String getUsername() {
        return this.username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public String getPassword() {
        return this.password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getPinuser() {
        return this.pinuser;
    }
    public void setPinuser(String pinuser) {
        this.pinuser = pinuser;
    }
}

```

Εικόνα 95 : Entities

Στην συνέχεια θα δημιουργήσουμε έναν Provider ο οποίος θα επικοινωνεί με την βάση μας. Όλα τα ερωτήματα θα γίνονται μέσω του Provider.

7.2.8 Δημιουργία AppContext

Δημιουργία αρχείου AppContext μέσα στο οποίο κάνουμε encryption την βάση μας.

```
1 package com.bustravelcorfu;
2
3 import ...
4
5 public class AppContext extends App {
6
7     @Override
8     public void onCreate() {
9         super.onCreate();
10         PrivateOpenHelper helper = new PrivateOpenHelper(context: this, name: "bustravelcorfu");
11         Database db = helper.getEncryptedWritableDb( password: "The true sign of intelligence is not knowledge but imagination");
12         mDaoSession = new DaoMaster(db).newSession();
13     }
14
15     @Override
16     public Class<?> getMainActivityClass() { return LoginActivity.class; }
17 }
18
```

Εικόνα 96 : AppContext.java

Δημιουργία αρχείου App.java μέσα στο οποίο έχουμε την σειρά λειτουργίας της εφαρμογής.

```
App
1 package com.bustravelcorfu;
2
3 import ...
4
5 public abstract class App extends Application {
6
7     private final static String TAG = Application.class.getSimpleName();
8     private static Context mAppContext;
9     protected static DaoSession mDaoSession;
10
11     @Override
12     public void onCreate() {
13         super.onCreate();
14
15         mAppContext = this;
16     }
17
18     public abstract Class<?> getMainActivityClass();
19
20     @ public static DaoSession getDaoSession() { return mDaoSession; }
21
22     @ public static Context getContext() { return mAppContext; }
23
24     @Override
25     public void onTerminate() { super.onTerminate(); }
26
27     public static Class<?> getStartActivityClass() {
28         return ((App)mAppContext).getMainActivityClass();
29     }
30 }
31
32
```

Εικόνα 97 : App.java

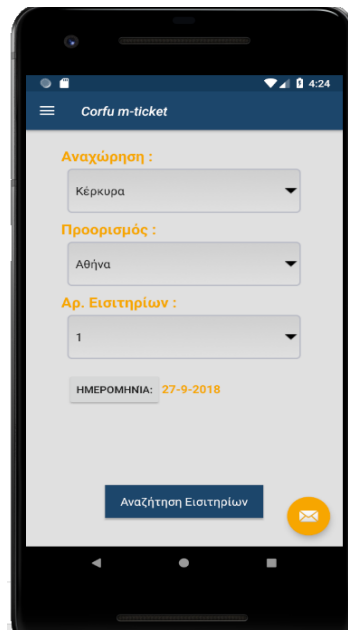
7.2.9 Δημιουργία BaseActivity.java

Δημιουργία BaseActivity.java στο οποίο ελέγχουμε το LoginActivity.java να ενεργοποιείται όταν η εφαρμογή είναι στο background και όταν η εφαρμογή γίνεται kill.

```
1 package com.bustravelcorfu.activities;
2
3 import ...
4
11
12 public abstract class BaseActivity extends AppCompatActivity {
13     protected static final String TAG = BaseActivity.class.getName();
14
15     public void onCreate(Bundle savedInstanceState) {
16
17         super.onCreate(savedInstanceState);
18         setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
19     }
20
21     public static boolean isAppWentToBg = false;
22     public static boolean isWindowFocused = false;
23     public static boolean isMenuOpened = false;
24     public static boolean isBackPressed = false;
25
26     @Override
27     protected void onStart() {
28         Log.d(TAG, "onStart isAppWentToBg " + isAppWentToBg);
29
30         applicationWillEnterForeground();
31
32         super.onStart();
33     }
34
35     private void applicationWillEnterForeground() {
36         if (isAppWentToBg) {
37             isAppWentToBg = false;
38             Intent intent = new Intent(getApplicationContext(), LoginActivity.class);
39             startActivity(intent);
40         }
41     }
42
43     @Override
44     protected void onStop() {
45         super.onStop();
46
47         Log.d(TAG, "onStop ");
48         applicationDidEnterBackground();
49     }
50
51     public void applicationDidEnterBackground() {
52         if (!isWindowFocused) {
53             isAppWentToBg = true;
54             /*Toast.makeText(getApplicationContext(),
55              "App is Going to Background", Toast.LENGTH_SHORT).show();*/
56         }
57     }
58
59     @Override
60     public void onBackPressed() {
61
62         if (this instanceof LoginActivity) {
63             Intent exit = new Intent(Intent.ACTION_MAIN);
64             exit.addCategory(Intent.CATEGORY_HOME);
65             exit.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
66             startActivity(exit);
67         } else {
68             isBackPressed = true;
69         }
70
71         Log.d(TAG,
72             "onBackPressed " + isBackPressed + " "
73             + this.getLocalClassName());
74         super.onBackPressed();
75     }
76
77     @Override
78     public void onWindowFocusChanged(boolean hasFocus) {
79
80         isWindowFocused = hasFocus;
81
82         if (isBackPressed && !hasFocus) {
83             isBackPressed = false;
84             isWindowFocused = true;
85         }
86
87         super.onWindowFocusChanged(hasFocus);
88     }
89
90     public boolean onOptionsItemSelected(MenuItem item) {
91         switch (item.getItemId()) {
92             case android.R.id.home:
93                 onBackPressed();
94                 break;
95         }
96         return true;
97     }
98 }
99
100
```

Εικόνα 98 : BaseActivity.java

Όταν ο χρήστης συνδεθεί με επιτυχία στην εφαρμογή εμφανίζεται η αρχική οθόνη που μπορεί να επιλέξει σημείο αναχώρησης και προορισμό, τον αριθμό των εισιτηρίων και την ημερομηνία του ταξιδιού.



Εικόνα 99 : Αναζήτηση Δρομολογίου

Στην συνέχεια πατώντας το κουμπί Αναζήτηση εισιτηρίων, η εφαρμογή τον ενημερώνει στην επόμενη οθόνη ότι υπάρχουν διαθέσιμες θέσεις και τότε ο χρήστης πρέπει να επιλέξει την ώρα που θέλει να ταξιδέψει, και στην πορεία να επιλέξει και τις θέσεις.

Μέσα στο activity μας κάνουμε ελέγχους έτσι ώστε ο χρήστης να μην μπορεί να επιλέξει περισσότερες ή λιγότερες θέσεις από τα εισιτήρια που έχει δηλώσει ότι θέλει να αγοράσει. Αφού έχει επιλέξει λοιπόν και τις θέσεις πατώντας στο κουμπί Επισκόπηση, κάνει μια επισκόπηση όλων των στοιχείων του ταξιδιού και αν συμφωνεί προχωράει στην αγορά μέσω της Pay pal.

Ο χρήστης καλείται να συνδεθεί στο λογαριασμό του στην paypal και αφού συνδεθεί μπορεί να κάνει την αγορά. Αν η αγορά γίνει με επιτυχία ο χρήστης ενημερώνεται με μήνυμα ότι η αγορά ήταν επιτυχής και μεταφέρεται στην αρχική οθόνη.

Για την ανάγκη της εφαρμογής να χρησιμοποιηθεί αγορά με Pay pal δημιουργήθηκε ένας λογαριασμός Gmail και ένας λογαριασμός Pay pal tester.

7.2.10 Διαδικασία Δημιουργίας Paypal

- Δημιουργία ενός λογαριασμού Business στην pay pal
- Επιβεβαίωση του email που δώσαμε
- Αφού συνδεθούμε στον paypal λογαριασμό μας μεταφερόμαστε στον πίνακα ελέγχου
- Και δημιουργούμε έναν Sandbox Account με στοιχεία όπως φαίνονται στην εικόνα παρακάτω
- Τέλος μεταφερόμαστε στο link My accounts and credentials και κάνουμε create app βάζοντας το όνομα της εφαρμογής και παίρνουμε το Client Id και το τοποθετούμε στον κώδικα μας.

The image shows two parts of the PayPal developer interface. The top part is the 'Create New App' page, which includes a sidebar with navigation links like Dashboard, My Apps & Credentials, My Account, Sandbox, Mock, and Live. The main content area is titled 'Create New App' and contains a form for 'Application Details'. The form has fields for 'App Name' (with the placeholder 'My Application Name') and 'Sandbox developer account' (with the email 'jperovich-facilitator@gmail.com'). Below these fields is a 'Create App' button. The bottom part of the image shows a separate form for creating a PayPal account. It includes fields for 'First name (optional)', 'Last name (optional)', 'Payment Methods', 'PayPal balance' (0.00 USD), 'Bank verified account' (radio buttons for Yes/No), 'Select payment card' (radio buttons for Discover/PayPal), 'Credit cards' (a dropdown menu showing 'Visa'), and 'Note (optional)'. A 'Create Account' button is at the bottom of this form.

Εικόνα 100 : Δημιουργία Λογαριασμού Paypal

Εφόσον ο χρήστης πραγματοποιήσει την αγορά τότε η εφαρμογή ρωτάει τον χρήστη αν θέλει να αποθηκεύσει την αγορά στο ιστορικό ενώ ταυτόχρονα τον ενημερώνει ότι η αγορά έγινε με επιτυχία.

7.2.11 MainActivity.java

```
1 package com.bustravelcorfu.activities;
2
3 import ...
4
35 public class MainActivity extends AppCompatActivity
36     implements NavigationView.OnNavigationItemSelectedListener {
37     private int mYear;
38     private int mMonth;
39     private int mDay;
40
41     private TextView mDateDisplay;
42     Button mPickDate;
43     Spinner destination_spinner;
44     Spinner point_spinner;
45     Spinner tickets_spinner;
46     Button but_overview;
47     static final int DATE_DIALOG_ID = 0;
48     private ProgressBar spinner;
49
50     @Override
51     protected void onCreate(Bundle savedInstanceState) {
52         super.onCreate(savedInstanceState);
53         setContentView(R.layout.activity_main);
54
55         Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
56         setSupportActionBar(toolbar);
57
58         DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
59         drawer.setBackgroundColor(getResources().getColor(R.color.colorBorder));
60         ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
61             this, drawer, toolbar, "Open navigation drawer", "Close navigation drawer");
62         drawer.addDrawerListener(toggle);
63         toggle.syncState();
64
65         NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
66         navigationView.setNavigationItemSelectedListener(this);
67
68         point_spinner = (Spinner) findViewById(R.id.point_spinner);
69         destination_spinner = (Spinner) findViewById(R.id.destination_spinner);
70         tickets_spinner = (Spinner) findViewById(R.id.ticket_spinner);
71         but_overview = (Button) findViewById(R.id.button_overview);
72         mDateDisplay = (TextView) findViewById(R.id.showMyDate);
73         mPickDate = (Button) findViewById(R.id.myDatePickerButton);
74         spinner = (ProgressBar) findViewById(R.id.progressBar1);
75
76         // Spinner departure point
77         List<String> departure_point = new ArrayList<>();
78         departure_point.add("Κέρκυρα");
79         departure_point.add("Θεσσαλονίκη");
80         departure_point.add("Πάτρα");
81         departure_point.add("Λάρισα");
82         departure_point.add("Αθήνα");
83         /* departure_point.add("Βόλος");
84         departure_point.add("Ιωάννινα");
85         departure_point.add("Τρίκαλα");
86         departure_point.add("Σέρρες");
87         departure_point.add("Αλεξανδρούπολη");
88         departure_point.add("Βάνθη");
89         departure_point.add("Κομοτηνή");
90         departure_point.add("Καβάλα");
91         departure_point.add("Λαμία"); */
92
93         //Spinner destination
94         final List<String> destination = new ArrayList<>();
95         destination.add("Αθήνα");
96         destination.add("Θεσσαλονίκη");
97         destination.add("Πάτρα");
98         destination.add("Λάρισα");
99         destination.add("Κέρκυρα");
100        /*destination.add("Βόλος");
101        destination.add("Ιωάννινα");
102        destination.add("Τρίκαλα");
103        destination.add("Σέρρες");
104        destination.add("Αλεξανδρούπολη");
105        destination.add("Βάνθη");
106        destination.add("Κομοτηνή");
107        destination.add("Καβάλα");
```

```

109 destination.add("Αθήνα");
110
111 //Spinner tickets number
112 List<String> ticket = new ArrayList<>();
113 ticket.add("1");
114 ticket.add("2");
115 ticket.add("3");
116 ticket.add("4");
117 ticket.add("5");
118 ticket.add("6");
119 ticket.add("7");
120 ticket.add("8");
121 ticket.add("9");
122 ticket.add("10");
123
124 // Creating adapter for spinner
125 ArrayAdapter<String> pointAdapter = new ArrayAdapter<>(context, this, android.R.layout.simple_dropdown_item_1line, departure_point);
126 ArrayAdapter<String> destinationAdapter = new ArrayAdapter<>(context, this, android.R.layout.simple_dropdown_item_1line, destination);
127 ArrayAdapter<String> ticketAdapter = new ArrayAdapter<>(context, this, android.R.layout.simple_dropdown_item_1line, ticket);
128
129 // attaching data adapter to spinner
130 point_spinner.setAdapter(pointAdapter);
131 destination_spinner.setAdapter(destinationAdapter);
132 tickets_spinner.setAdapter(ticketAdapter);
133
134 but_overview.setOnClickListener((v) -> { Validation(); });
135
136 point_spinner.setOnItemClickListener(new AdapterView.OnItemClickListener() {
137     @Override
138     public void onItemClick(AdapterView<?> parent, View view,
139         int position, long id) {
140
141     }
142
143     @Override
144     public void onNothingSelected(AdapterView<?> parent) {
145         // TODO Auto-generated method stub
146     }
147 });
148
149 mPickDate.setOnClickListener((v) -> { showDialog(DATE_DIALOG_ID); });
150
151 // get the current date
152 final Calendar c = Calendar.getInstance();
153 mYear = c.get(Calendar.YEAR);
154 mMonth = c.get(Calendar.MONTH);
155 mDay = c.get(Calendar.DAY_OF_MONTH);
156
157 // display the current date
158 updateDisplay();
159
160 FloatingActionButton button = findViewById(R.id.fab);
161 button.setOnClickListener((v) -> {
162     Intent Email = new Intent(Intent.ACTION_SEND);
163     Email.setType("text/email");
164     Email.putExtra(Intent.EXTRA_EMAIL, new String[]{"your_email@gmail.com"}); //replace with your email*/
165     Email.putExtra(Intent.EXTRA_SUBJECT, VALUE:"Feedback from " + Build.MODEL + " user");
166     Email.putExtra(Intent.EXTRA_TEXT, VALUE:"Dear Bus Travel team," + "");
167     startActivity(Intent.createChooser(Email, VALUE:"Send Feedback:"));
168 });
169
170 private void Validation() {
171     String _point = String.valueOf(point_spinner.getSelectedItem());
172     String _destination = String.valueOf(destination_spinner.getSelectedItem());
173
174     if (_point.equals("Ρέθυμνο") && _destination.equals("Ρέθυμνο")) {
175         dialogWarning();
176     } else if (_point.equals("Θεσσαλονίκη") && _destination.equals("Θεσσαλονίκη") || _point.equals("Αθήνα") && _destination.equals("Αθήνα")
177         || _point.equals("Θεσσαλονίκη") && _destination.equals("Πάτρα") || _point.equals("Θεσσαλονίκη") && _destination.equals("Λάρισα")) {
178         dialogWarning();
179     } else if (_point.equals("Πάτρα") && _destination.equals("Πάτρα") || _point.equals("Πάτρα") && _destination.equals("Αθήνα")
180         || _point.equals("Πάτρα") && _destination.equals("Θεσσαλονίκη") || _point.equals("Πάτρα") && _destination.equals("Λάρισα")) {
181         dialogWarning();
182     } else if (_point.equals("Λάρισα") && _destination.equals("Λάρισα") || _point.equals("Λάρισα") && _destination.equals("Αθήνα")
183         || _point.equals("Λάρισα") && _destination.equals("Πάτρα") || _point.equals("Λάρισα") && _destination.equals("Θεσσαλονίκη")) {
184         dialogWarning();
185     } else if (_point.equals("Αθήνα") && _destination.equals("Αθήνα") || _point.equals("Αθήνα") && _destination.equals("Θεσσαλονίκη")
186         || _point.equals("Αθήνα") && _destination.equals("Πάτρα") || _point.equals("Αθήνα") && _destination.equals("Λάρισα")) {
187

```

```

219         if (point.equals(selectedPoint) && destination.equals(selectedDestination) || point.equals(selectedPoint) && destination.equals(selectedDestination)) {
220             dialogWarning();
221         } else {
222             spinner.setVisibility(View.GONE);
223             spinner.setVisibility(View.VISIBLE);
224             Intent intent = new Intent(getApplicationContext(), MainActivity.class);
225             intent.putExtra("point", String.valueOf(point_spinner.getSelectedItem()));
226             intent.putExtra("destination", String.valueOf(destination_spinner.getSelectedItem()));
227             intent.putExtra("tickets", String.valueOf(tickets_spinner.getSelectedItem()));
228             intent.putExtra("day", String.valueOf(mDay));
229             intent.putExtra("month", String.valueOf(mMonth + 1));
230             intent.putExtra("year", String.valueOf(mYear));
231             startActivity(intent);
232         }
233     }
234
235     private void updateDisplay() {
236         this.mDateDisplay.setText(
237             new StringBuilder()
238                 // Month is 0 based so add 1
239                 .append(mDay).append("-")
240                 .append(mMonth + 1).append("-")
241                 .append(mYear).append(" ")
242         );
243     }
244
245     @Override
246     protected Dialog onCreateDialog(int id) {
247         switch (id) {
248             case DATE_DIALOG_ID:
249                 return new DatePickerDialog(this,
250                     mDateSetListener,
251                     mYear, mMonth, mDay);
252             default:
253                 return null;
254         }
255     }
256
257     private DatePickerDialog.OnDateSetListener mDateSetListener =
258         (view, year, monthOfYear, dayOfMonth) -> {
259             mYear = year;
260             mMonth = monthOfYear;
261             mDay = dayOfMonth;
262             updateDisplay();
263         };
264
265     @Override
266     protected void onResume() {
267         super.onResume();
268     }
269
270     @Override
271     public void onBackPressed() {
272         DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
273         if (drawer.isDrawerOpen(GravityCompat.START)) {
274             drawer.closeDrawer(GravityCompat.START);
275         } else {
276             super.onBackPressed();
277             Intent exit = new Intent(Intent.ACTION_MAIN);
278             exit.addCategory(Intent.CATEGORY_HOME);
279             exit.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
280             startActivity(exit);
281         }
282     }
283
284     @Override
285     public boolean onCreateOptionsMenu(Menu menu) {
286         // Inflate the menu; this adds items to the action bar if it is present.
287         getMenuInflater().inflate(R.menu.main, menu);
288         return true;
289     }
290
291     @Override
292     public boolean onOptionsItemSelected(MenuItem item) {
293         int id = item.getItemId();
294         //noinspection SimplifiableIfStatement
295         /*if (id == R.id.action_settings) {
296             return true;
297         }*/
298         return super.onOptionsItemSelected(item);
299     }

```

```

282
283 @SuppressWarnings("StatementWithEmptyBody")
284 @Override
285 public boolean onNavigationItemSelected(MenuItem item) {
286     // Handle navigation view item clicks here.
287     int id = item.getItemId();
288
289     if (id == R.id.nav_search_travel) {
290         Intent main_activity = new Intent(getApplicationContext(), MainActivity.class);
291         startActivity(main_activity);
292
293     } else if (id == R.id.nav_cancel_ticket) {
294         Intent intent = new Intent(getApplicationContext(), CancelTicketActivity.class);
295         startActivity(intent);
296
297     } else if (id == R.id.nav_history_travel) {
298         Intent intent = new Intent(getApplicationContext(), HistoryBookingActivity.class);
299         startActivity(intent);
300
301     } else if (id == R.id.nav_conditions) {
302         Intent intent = new Intent(getApplicationContext(), ConditionsActivity.class);
303         startActivity(intent);
304
305     } else if (id == R.id.nav_share) {
306         Intent sharingIntent = new Intent(Intent.ACTION_SEND);
307         sharingIntent.setAction(Intent.ACTION_SEND);
308         sharingIntent.putExtra(Intent.EXTRA_TEXT, value: "Αγόρασε το εισιτήριό σου και ταξίδεψε από την Κέρκυρα");
309         sharingIntent.setType("text/plain");
310         startActivity(Intent.createChooser(sharingIntent, title: "Share via"));
311     } else if (id == R.id.nav_rate) {
312         Intent intent = new Intent(Intent.ACTION_VIEW)
313             .setData(Uri
314                 .parse("https://play.google.com/store")); /*replace url from app in play store */
315         startActivity(intent);
316     } else if (id == R.id.nav_contact) {
317         Intent Email = new Intent(Intent.ACTION_SEND);
318         Email.setType("text/email");
319         Email.putExtra(Intent.EXTRA_EMAIL, new String[]{"your_email@gmail.com"}); /*replace with your email*/
320         Email.putExtra(Intent.EXTRA_SUBJECT, value: "Feedback from " + Build.MODEL + " user");
321         Email.putExtra(Intent.EXTRA_TEXT, value: "Dear Bus Travel team," + "");
322         startActivity(Intent.createChooser(Email, title: "Send Feedback:"));
323     } else if (id == R.id.nav_about) {
324         Intent intent = new Intent(getApplicationContext(), AboutActivity.class);
325         startActivity(intent);
326
327     }
328
329     DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
330     drawer.closeDrawer(GravityCompat.START);
331     return true;
332 }
333
334 private void dialogWarning() {
335     AlertDialog alertDialog = new AlertDialog.Builder(context: MainActivity.this).create();
336     alertDialog.setTitle("Ειδοποίηση");
337     alertDialog.setMessage("Αναζητήστε εισιτήρια με σημείο αναχώρησης την Κέρκυρα προς τις πόλεις της Ελλάδας");
338     alertDialog.setButton(AlertDialog.BUTTON_NEUTRAL, text: "OK",
339         (OnClickListener) (dialog, which) -> { dialog.dismiss(); });
340     alertDialog.show();
341 }
342
343 }
344
345
346
347

```

Εικόνα 101: MainActivity.java

7.2.12 SelectBusActivity.java

```
1 package com.bustravelcorfu.activities;
2
3 import ...
4
20
21 public class SelectBusActivity extends AppCompatActivity {
22     Spinner mSelect_bus_spinner;
23     Button mSelect_Seats;
24     Toolbar mActionBarToolbar;
25     TextView mPoint_preview, mDestination_preview, mTickets_preview, mDate_preview;
26     String point, destination, tickets, day, month, year;
27
28     @Override
29     protected void onCreate(Bundle savedInstanceState) {
30         super.onCreate(savedInstanceState);
31         setContentView(R.layout.layout_select_bus);
32
33         LinearLayout linearLayout = (LinearLayout) findViewById(R.id.layout_select_bus);
34         linearLayout.setBackgroundColor(getResources().getColor(R.color.colorBorder));
35         mActionBarToolbar = (android.support.v7.widget.Toolbar) findViewById(R.id.toolbar);
36         setSupportActionBar(mActionBarToolbar);
37         mActionBarToolbar.setNavigationIcon(R.drawable.io_arrow_back_24dp);
38         setTitle("Επιλογή Λεωφορείου");
39
40         mSelect_bus_spinner = (Spinner) findViewById(R.id.select_bus_spinner);
41         mSelect_Seats = (Button) findViewById(R.id.button_select_seats);
42         mPoint_preview = (TextView) findViewById(R.id.point_preview);
43         mDestination_preview = (TextView) findViewById(R.id.destination_preview);
44         mTickets_preview = (TextView) findViewById(R.id.tickets_preview);
45         mDate_preview = (TextView) findViewById(R.id.date_preview);
46
47         point = getIntent().getStringExtra( name: "point");
48         destination = getIntent().getStringExtra( name: "destination");
49         tickets = getIntent().getStringExtra( name: "tickets");
50         day = getIntent().getStringExtra( name: "day");
51         month = getIntent().getStringExtra( name: "month");
52         year = getIntent().getStringExtra( name: "year");
53
54         mPoint_preview.setText(point);
55         mDestination_preview.setText(destination);
56
57         mDate_preview.setText(new StringBuilder()
58             .append(day).append("-")
59             .append(month).append("-")
60             .append(year).append(" "));
61
62         List<String> select_bus_list = new ArrayList<>();
63         select_bus_list.add("13:30π. σταθμός Υπερσπορίος Κιόκ");
64         select_bus_list.add("13:30π. σταθμός Υπερσπορίος Κιόκ");
65         select_bus_list.add("20:15π. σταθμός Υπερσπορίος Κιόκ");
66
67         ArrayAdapter<String> busAdapter = new ArrayAdapter<>( context: this, android.R.layout.simple_dropdown_item_1line, select_bus_list);
68         mSelect_bus_spinner.setAdapter(busAdapter);
69
70         mSelect_bus_spinner.setOnItemClickListener(new AdapterView.OnItemClickListener() {
71             @Override
72             public void onItemClick(AdapterView<?> parent, View view,
73                 int position, long id) {
74
75             }
76
77             @Override
78             public void onNothingSelected(AdapterView<?> parent) {
79                 // TODO Auto-generated method stub
80             }
81         });
82
83         mSelect_Seats.setOnClickListener((v) -> {
84             Intent select_seat_activity = new Intent(getApplicationContext(), SeatSelectionActivity.class);
85             select_seat_activity.putExtra( name: "point", point);
86             select_seat_activity.putExtra( name: "destination", destination);
87             select_seat_activity.putExtra( name: "tickets", tickets);
88             select_seat_activity.putExtra( name: "day", day);
89             select_seat_activity.putExtra( name: "month", month);
90             select_seat_activity.putExtra( name: "year", year);
91             select_seat_activity.putExtra( name: "time", String.valueOf(mSelect_bus_spinner.getSelectedItem()));
92             startActivity(select_seat_activity);
93         });
94
95     @Override
96     protected void onResume() {
97         super.onResume();
98     }
99
100     @Override
101     public void onBackPressed() {
102         super.onBackPressed();
103         Intent main_activity = new Intent(getApplicationContext(), MainActivity.class);
104         startActivity(main_activity);
105     }
106
107     @Override
108     public boolean onOptionsItemSelected(MenuItem menuItem) {
109
110         switch (menuItem.getItemId()) {
111             case android.R.id.home:
112                 onBackPressed();
113                 break;
114         }
115
116         return super.onOptionsItemSelected(menuItem);
117     }
118
119 }
120
121
```

Εικόνα 102 : SelectBusActivity.java

7.2.13 SeatSelectionActivity.java

```
1 package com.bustravelcorfu.activities;
2
3 import ...
4
19
20 public class SeatSelectionActivity extends AppCompatActivity {
21     ArrayList<Integer> number_seat;
22     Toolbar mActionBarToolbar;
23     String point, destination, tickets, day, month, year, time;
24     Button mButtonPreview;
25     int num = 0;
26     int mCols;
27     SeatAdapter mAdapter;
28
29     @Override
30     protected void onCreate(Bundle savedInstanceState) {
31         super.onCreate(savedInstanceState);
32         setContentView(R.layout.activity_seat_selection_screen);
33         mActionBarToolbar = (android.support.v7.widget.Toolbar) findViewById(R.id.toolbar);
34         setSupportActionBar(mActionBarToolbar);
35         mActionBarToolbar.setNavigationIcon(R.drawable.ic_arrow_back_24dp);
36         setTitle("Επιλογή Θέσεων");
37         number_seat = new ArrayList<>();
38         mCols = 4;
39         GridLayoutManager gridLayoutManager = new GridLayoutManager(getApplicationContext(), mCols);
40         RecyclerView recyclerView = findViewById(R.id.oidSeatRecyclerView);
41         recyclerView.setLayoutManager(gridLayoutManager);
42         recyclerView.setVisibility(View.VISIBLE);
43         mAdapter = new SeatAdapter(context, this, activity: SeatSelectionActivity.this);
44         recyclerView.setAdapter(mAdapter);
45         recyclerView.setHasFixedSize(true);
46
47
48         point = getIntent().getStringExtra(name: "point");
49         destination = getIntent().getStringExtra(name: "destination");
50         tickets = getIntent().getStringExtra(name: "tickets");
51         day = getIntent().getStringExtra(name: "day");
52         month = getIntent().getStringExtra(name: "month");
53         year = getIntent().getStringExtra(name: "year");
54         time = getIntent().getStringExtra(name: "time");
55
56         mButtonPreview = (Button) findViewById(R.id.buttonPreview);
57
58
59     @Override
60     protected void onResume() {
61         super.onResume();
62     }
63
64     @Override
65     public void onBackPressed() {
66         super.onBackPressed();
67         Intent intent = new Intent(getApplicationContext(), MainActivity.class);
68         startActivity(intent);
69     }
70
71     @Override
72     public boolean onOptionsItemSelected(MenuItem menuItem) {
73
74         switch (menuItem.getItemId()) {
75             case android.R.id.home:
76                 onBackPressed();
77                 break;
78         }
79
80         return super.onOptionsItemSelected(menuItem);
81     }
82
83     public void onListItemSelect(int position) {
84         Resources resources = getResources();
85         mAdapter.toggleSelection(position);
86         num = Integer.parseInt(String.valueOf(tickets));
87
88         boolean hasCheckedItems = mAdapter.getSelectedCount() > 0;
89         if (mAdapter.getSelectedCount() > num) {
90             Toast.makeText(context, this, "Εξέρε επιλέξει τις θέσεις σύμφωνα με τον αριθμό των εισιτηρίων", Toast.LENGTH_LONG).show();
91         } else {
92             // Toast.makeText(context, this, "Επιλογή θέσεων", Toast.LENGTH_SHORT).show();
93         }
94     }
95 }
```

```

124         if (hasCheckedItems) {
125             number_seat.add(position + 1);
126             mActionBarToolbar.setTitle(String.valueOf(mAdapter.getSelectedCount()) + " selected");
127             mActionBarToolbar.setBackgroundColor(resources.getColor(R.color.colorPrimaryDarkGrey));
128             mActionBarToolbar.setNavigationIcon(R.drawable.ic_close);
129         }
130     }
131     else {
132         if (position == 0) {
133             number_seat.remove(position);
134             number_seat.clear();
135             mAdapter.removeSelection();
136             mActionBarToolbar.setTitle("Corfu m-ticket");
137             mActionBarToolbar.setBackgroundColor(resources.getColor(R.color.colorPrimary));
138             mActionBarToolbar.setNavigationIcon(R.drawable.ic_arrow_back_24dp);
139         }
140         else {
141             number_seat.remove(index position - 1);
142             number_seat.clear();
143             mAdapter.removeSelection();
144             mActionBarToolbar.setTitle("Corfu m-ticket");
145             mActionBarToolbar.setBackgroundColor(resources.getColor(R.color.colorPrimary));
146             mActionBarToolbar.setNavigationIcon(R.drawable.ic_arrow_back_24dp);
147         }
148     }
149 }
150
151 mButtonPreview.setOnClickListener((v) = {
152     if (number_seat.size() < num || number_seat.size() > num) {
153         Toast.makeText(getApplicationContext(), "ΜΗΔΕ \"Επιλέξτε σωστά τον αριθμό των θέσεων σύμφωνα με τον αριθμό των εισηγτηρίων\", Toast.LENGTH_LONG).show();
154     }
155     else {
156         final int price = number_seat.size() * 48;
157         Intent intent = new Intent(getApplicationContext(), BookingActivity.class);
158         intent.putExtra(name: "point", String.valueOf(point));
159         intent.putExtra(name: "destination", String.valueOf(destination));
160         intent.putExtra(name: "tickets", String.valueOf(tickets));
161         intent.putExtra(name: "day", String.valueOf(day));
162         intent.putExtra(name: "month", String.valueOf(month));
163         intent.putExtra(name: "year", String.valueOf(year));
164         intent.putExtra(name: "time", String.valueOf(time));
165
166         intent.putExtra(name: "time", String.valueOf(time));
167         intent.putExtra(name: "seats", number_seat);
168         intent.putExtra(name: "price", price);
169         startActivity(intent);
170     }
171 });
172
173 }
174
175 public int ticket() { return num; }
176
177 }

```

Εικόνα 103 : SeatSelectionActivity.java

7.2.14 SeatAdapter.java

```
package com.bustravelcorfu.adapters;

import ...

public class SeatAdapter extends RecyclerView.Adapter<SeatHolder> {
    private Context mContext;
    private SparseBooleanArray mSelectedSeats;
    private SeatSelectionActivity mActivity;

    public SeatAdapter(Context context, SeatSelectionActivity activity) {
        this.mContext = context;
        this.mActivity = activity;
        mSelectedSeats = new SparseBooleanArray();
    }

    @NonNull
    @Override
    public SeatHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(mContext).inflate(R.layout.activity_grid_seat_item, parent, attachToRoot: false);
        return new SeatHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull final SeatHolder holder, final int position) {
        int seat_free = mContext.getResources().getIdentifier("com.bustravelcorfu:drawable/seat_free", "drawable", null);
        int seat_select = mContext.getResources().getIdentifier("com.bustravelcorfu:drawable/seat_select", "drawable", null);

        holder.itemView.setOnClickListener(v -> {
            mActivity.onListItemSelect(holder.getAdapterPosition());
        });

        if (mSelectedSeats.get(position)) {
            holder.mSeatText.setText(new StringBuilder().append("Select"));
            holder.mSeatImage.setImageResource(seat_select);
        } else {
            holder.mSeatText.setText(new StringBuilder().append("Seat ").append(String.valueOf(position + 1)));
            holder.mSeatImage.setImageResource(seat_free);
        }
    }

    @Override
    public int getItemCount() { return 50; }

    private void selectView(int position, boolean sel) {
        if (sel) {
            mSelectedSeats.put(position, sel);
        } else {
            mSelectedSeats.delete(position);
        }
        notifyDataSetChanged();
    }

    public void toggleSelection(int position) {
        selectView(position, !mSelectedSeats.get(position));
    }

    public int getSelectedCount() { return mSelectedSeats.size(); }

    private SparseBooleanArray getmSelectedIds() { return mSelectedSeats; }

    public void removeSelection() {
        mSelectedSeats = new SparseBooleanArray();
        notifyDataSetChanged();
    }
}
```

Εικόνα 104 : SeatAdapter.java

7.2.15 SeatHolder.java

```
1 package com.bustravelcorfu.holders;
2
3 import ...
4
5
6
7
8
9
10 public class SeatHolder extends RecyclerView.ViewHolder {
11     public TextView mSeatText;
12     public ImageView mSeatImage;
13     public SeatHolder(View itemView) {
14         super(itemView);
15         mSeatText = itemView.findViewById(R.id.seat_text);
16         mSeatImage = itemView.findViewById(R.id.seat_image);
17     }
18 }
19
```

Εικόνα 105 : SeatHolder.java

7.2.16 BookingActivity.java

```

1 package com.bustravelcorfu.activities;
2
3 import ...
4
51
52 public class BookingActivity extends AppCompatActivity {
53     TextView mPoint_preview, mDestination_preview, mTickets_preview, mDate_preview, mSeatPreview, mTimePreview, mPricePreview;
54     String point, destination, tickets, day, month, year, time;
55     ArrayList<Integer> number_seat;
56     int price;
57     Toolbar mActionBarToolbar;
58     private static final String CONFIG_CLIENT_ID = "Ad0yk3wGQoeMonKmfE1ep3RQzU5LVu_JQqK8CBpOpAj39jeB9WzFNLE0Kc3RcEayR4w34iVAcyyOPQv";
59     private static PayPalConfiguration config = new PayPalConfiguration()
60         .environment(PayPalConfiguration.ENVIRONMENT_SANDBOX)
61         .clientId(CONFIG_CLIENT_ID);
62
63     @Override
64     protected void onCreate(Bundle savedInstanceState) {
65         super.onCreate(savedInstanceState);
66         setContentView(R.layout.activity_booking);
67         mActionBarToolbar = (android.support.v7.widget.Toolbar) findViewById(R.id.toolbar);
68         setSupportActionBar(mActionBarToolbar);
69         mActionBarToolbar.setNavigationIcon(R.drawable.ic_arrow_back_24dp);
70         setTitle("Εργασιασ @corfu");
71
72         point = getIntent().getStringExtra("name: point");
73         destination = getIntent().getStringExtra("name: destination");
74         tickets = getIntent().getStringExtra("name: tickets");
75         day = getIntent().getStringExtra("name: day");
76         month = getIntent().getStringExtra("name: month");
77         year = getIntent().getStringExtra("name: year");
78
79         time = getIntent().getStringExtra("name: time");
80         price = getIntent().getIntExtra("name: price", 0);
81         number_seat = getIntent().getIntegerArrayListExtra("name: seats");
82         final Button button = (Button) findViewById(R.id.paypal_button);
83
84         mPoint_preview = (TextView) findViewById(R.id.point_preview_dialog);
85         mDestination_preview = (TextView) findViewById(R.id.destination_preview_dialog);
86         mTickets_preview = (TextView) findViewById(R.id.tickets_preview_dialog);
87         mDate_preview = (TextView) findViewById(R.id.date_preview_dialog);
88         mSeatPreview = (TextView) findViewById(R.id.seat_preview_dialog);
89         mTimePreview = (TextView) findViewById(R.id.time_preview_dialog);
90         mPricePreview = (TextView) findViewById(R.id.price_preview_dialog);
91
92         final StringBuilder builder = new StringBuilder();
93
94         mPoint_preview.setText(point);
95         mDestination_preview.setText(destination);
96         mTickets_preview.setText(tickets);
97         mTimePreview.setText(time);
98         mDate_preview.setText(new StringBuilder()
99             .append(day).append("-")
100             .append(month).append("-")
101             .append(year).append(" "));
102         for (int i = 0; i < number_seat.size(); i++) {
103             builder.append(String.valueOf(number_seat.get(i) + "k "));
104         }
105         mSeatPreview.setText(builder.toString());
106         mPricePreview.setText(new StringBuilder()
107             .append(String.valueOf(price)).append(" ")
108             .append("€"));
109     }
110
111     public void beginPayment(View view) {
112         Intent serviceConfig = new Intent(packageContext, this, PayPalService.class);
113         serviceConfig.putExtra(PayPalService.EXTRA_PAYPAL_CONFIGURATION, config);
114         startService(serviceConfig);
115
116         PayPalPayment payment = new PayPalPayment(new BigDecimal(price),
117             "EUR", "Τrip στην Corfu", PayPalPayment.PAYMENT_INTENT_SALE);
118     }
119 }

```

```

98
99 Intent paymentConfig = new Intent( packageContext: this, PaymentActivity.class);
100 paymentConfig.putExtra(PayPalService.EXTRA_PAYPAL_CONFIGURATION, config);
101 paymentConfig.putExtra(PaymentActivity.EXTRA_PAYMENT, payment);
102 startActivityForResult(paymentConfig, (requestCode: 0);
103
104
105 @Override
106 protected void onActivityResult(int requestCode, int resultCode, Intent data) {
107     if (resultCode == Activity.RESULT_OK) {
108         PaymentConfirmation confirm = data.getParcelableExtra(
109             PaymentActivity.EXTRA_RESULT_CONFIRMATION);
110         if (confirm != null) {
111             try {
112                 Log.i( tag: "bustravel", confirm.toJSONString().toString( indentSpaces: 4));
113                 String _point = mPoint_preview.getText().toString();
114                 String _destination = mDestination_preview.getText().toString();
115                 String _tickets = mTickets_preview.getText().toString();
116                 String _date = mDate_preview.getText().toString();
117                 String _time = mTimePreview.getText().toString();
118                 String _seats = mSeatPreview.getText().toString();
119                 String _price = mPricePreview.getText().toString();
120                 ContentValues values = new ContentValues();
121                 values.put(BusTravelProvider.COLUMN_CA_DEPARTURE_POINT, _point);
122                 values.put(BusTravelProvider.COLUMN_CA_DESTINATION, _destination);
123                 values.put(BusTravelProvider.COLUMN_CA_TICKETS, _tickets);
124                 values.put(BusTravelProvider.COLUMN_CA_DATE, _date);
125                 values.put(BusTravelProvider.COLUMN_CA_TIME_TRAVEL, _time);
126                 values.put(BusTravelProvider.COLUMN_CA_NUMBER_SEAT, _seats);
127                 values.put(BusTravelProvider.COLUMN_CA_PRICE, _price);
128
129                 getContentResolver().insert(BusTravelProvider.CONTENT_CANCEL_TICKET_INSERT, values);
130                 Toast.makeText( context: this, (text: "Η αγορά έγινε με επιτυχία", Toast.LENGTH_LONG).show();
131                 dialogSaveOnHistory();
132                 // TODO: send 'confirm' to your server for verification
133             } catch (JSONException e) {
134                 Log.e( tag: "bustravel", (msg: "no confirmation data: ", e);
135             }
136         }
137     }
138
139     } else if (resultCode == Activity.RESULT_CANCELED) {
140         Log.i( tag: "bustravel", (msg: "The user canceled.");
141     } else if (resultCode == PaymentActivity.RESULT_EXTRAS_INVALID) {
142         Log.i( tag: "bustravel", (msg: "Invalid payment / config set");
143     }
144
145     private void dialogExit() {
146         LayoutInflater inflater = LayoutInflater.from(this);
147         View dialogExitView = inflater.inflate(R.layout.dialog_exit_booking, root: null);
148         final AlertDialog dialogExit = new AlertDialog.Builder( context: this).create();
149         Button btnNo = dialogExitView.findViewById(R.id.btn_no);
150         Button btnYes = dialogExitView.findViewById(R.id.btn_yes);
151
152         btnNo.setOnClickListener((view) -> { dialogExit.dismiss(); });
153
154         btnYes.setOnClickListener((view) -> {
155             Intent intent = new Intent(getApplicationContext(), MainActivity.class);
156             startActivity(intent);
157             new Handler().post(() -> { dialogExit.dismiss(); });
158         });
159         dialogExit.setView(dialogExitView);
160         dialogExit.show();
161     }
162
163     @Override
164     public void onDestroy() {
165         // Stop service when done
166         stopService(new Intent( packageContext: this, PayPalService.class));
167         super.onDestroy();
168     }
169
170     @Override
171     public void onBackPressed() { dialogExit(); }
172
173     @Override
174     public boolean onOptionsItemSelected(MenuItem menuItem) {
175         switch (menuItem.getItemId()) {
176

```

```

203 View dialogView = inflater.inflate(R.layout.dialog_confirm_save_history, root: null);
204 final android.app.AlertDialog dialog = new android.app.AlertDialog.Builder( context: this).create();
205 Button btnNo = dialogView.findViewById(R.id.btn_no);
206 Button btnYes = dialogView.findViewById(R.id.btn_yes);
207
208 btnYes.setOnClickListener((v) -> {
209     String _point = mPoint_preview.getText().toString();
210     String _destination = mDestination_preview.getText().toString();
211     String _tickets = mTickets_preview.getText().toString();
212     String _date = mDate_preview.getText().toString();
213     String _time = mTimePreview.getText().toString();
214     String _seats = mSeatPreview.getText().toString();
215     String _price = mPricePreview.getText().toString();
216     ContentValues values = new ContentValues();
217     values.put(BusTravelProvider.COLUMN_DEPARTURE_POINT, _point);
218     values.put(BusTravelProvider.COLUMN_DESTINATION, _destination);
219     values.put(BusTravelProvider.COLUMN_NUMBER_OF_TICKETS, _tickets);
220     values.put(BusTravelProvider.COLUMN_DATE, _date);
221     values.put(BusTravelProvider.COLUMN_TIME_TRAVEL, _time);
222     values.put(BusTravelProvider.COLUMN_NUMBER_OF_SEAT, _seats);
223     values.put(BusTravelProvider.COLUMN_PRICE, _price);
224
225     getContentResolver().insert(BusTravelProvider.CONTENT_HISTORY_BOOK_INSERT, values);
226     Toast.makeText(getApplicationContext(), text: "Ανοθηκεύτηκε", Toast.LENGTH_LONG).show();
227     Intent intent = new Intent( packageContext: BookingActivity.this, MainActivity.class);
228     startActivity(intent);
229     new android.os.Handler().post(() -> { dialog.dismiss(); });
230 });
231
232 btnNo.setOnClickListener((v) -> {
233     dialog.dismiss();
234     Intent intent = new Intent( packageContext: BookingActivity.this, MainActivity.class);
235     startActivity(intent);
236 });
237 dialog.setView(dialogView);
238 dialog.show();
239 }
240
241 }
242

```

Εικόνα 106 : BookingActivity.java

7.2.17 HistoryBookingActivity.java

```
1 package com.bustravelcorfu.activities;
2
3 import ...
4
5 public class HistoryBookingActivity extends AppCompatActivity {
6     Toolbar mActionBarToolbar;
7     RecyclerView recyclerView;
8     List<History_Booking> mListHistory;
9     HistoryAdapter mAdapter;
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_history_booking);
15
16         LinearLayoutManager layoutManager = new LinearLayoutManager( context, LinearLayoutManager.VERTICAL, reverseLayout: false);
17         recyclerView = (RecyclerView) findViewById(R.id.cidHistoryRecyclerView);
18         recyclerView.setLayoutManager(layoutManager);
19         recyclerView.setAdapter(mAdapter);
20         recyclerView.setHasFixedSize(true);
21         mActionBarToolbar = (android.support.v7.widget.Toolbar) findViewById(R.id.toolbar);
22         setSupportActionBar(mActionBarToolbar);
23         mActionBarToolbar.setNavigationIcon(R.drawable.ic_arrow_back_24dp);
24         setTitle("Ιστορία Κινητήσεων");
25
26         mListHistory = new ArrayList<>();
27
28         String URL = "content://com.bustravelcorfu.provider/" + BusTravelProvider.HISTORY_BOOKING_PATH + "";
29         final Uri _history = Uri.parse(URL);
30         final Cursor cursor = getContentResolver().query( _history, projection: null, selection: null, selectionArgs: null, sortOrder: null);
31
32         if (cursor != null) {
33             while (cursor.moveToNext()) {
34                 History_Booking historyBooking = new History_Booking();
35                 historyBooking.setId(cursor.getLong( columnIndex: 0));
36                 historyBooking.setDeparture_point(cursor.getString( columnIndex: 1));
37                 historyBooking.setDestination(cursor.getString( columnIndex: 2));
38                 historyBooking.setNumber_of_tickets(cursor.getString( columnIndex: 3));
39                 historyBooking.setDate(cursor.getString( columnIndex: 4));
40                 historyBooking.setTime_travel(cursor.getString( columnIndex: 5));
41                 historyBooking.setNumber_of_seat(cursor.getString( columnIndex: 6));
42                 historyBooking.setPrice(cursor.getString( columnIndex: 7));
43                 mListHistory.add(historyBooking);
44             }
45         }
46         if (mListHistory.size() > 0) {
47             recyclerView.setVisibility(View.VISIBLE);
48             mAdapter = new HistoryAdapter( context, this, mListHistory, activity: HistoryBookingActivity.this);
49             recyclerView.setAdapter(mAdapter);
50         } else {
51             RelativeLayout cidEmptyRecyclerView = (RelativeLayout) findViewById(R.id.cidEmptyRecyclerView);
52             cidEmptyRecyclerView.setVisibility(View.VISIBLE);
53         }
54     }
55
56     @Override
57     public boolean onOptionsItemSelected(MenuItem menuItem) {
58         switch (menuItem.getItemId()) {
59             case android.R.id.home:
60                 Intent mainActivity = new Intent( packageContext: this, MainActivity.class);
61                 startActivity(mainActivity);
62                 break;
63         }
64         return super.onOptionsItemSelected(menuItem);
65     }
66
67     @Override
68     public void onBackPressed() { super.onBackPressed(); }
69
70     @Override
71     protected void onResume() { super.onResume(); }
72 }
```

Εικόνα 107 : HistoryBookingActivity.java

7.2.18 HistoryHolder.java

```
HistoryHolder
1 package com.bustravelcorfu.holders;
2
3 import ...
4
5
6
7
8
9
10 public class HistoryHolder extends RecyclerView.ViewHolder {
11
12     public TextView mPoint_history;
13     public TextView mDestination_history;
14     public TextView mDate_history;
15     public TextView mPrice_history;
16     public ImageView mRemove_history;
17
18     public HistoryHolder(View itemView) {
19         super(itemView);
20
21         mPoint_history = (TextView) itemView.findViewById(R.id.point_history);
22         mDestination_history = (TextView) itemView.findViewById(R.id.destination_history);
23         mDate_history = (TextView) itemView.findViewById(R.id.date_history);
24         mPrice_history = (TextView) itemView.findViewById(R.id.price_history);
25         mRemove_history = (ImageView) itemView.findViewById(R.id.history_item_remove);
26     }
27 }
28
```

Εικόνα 108 : HistoryHolder.java

7.2.19 CancelTicketActivity.java

```
1 package com.bustravelcorfu.activities;
2
3 import ...
4
5 public class CancelTicketActivity extends AppCompatActivity {
6     Toolbar mActionBarToolbar;
7     RecyclerView recyclerView;
8     List<CancelTickets> mListCancelTickets;
9     CancelTicketAdapter mAdapter;
10    @Override
11    protected void onCreate(Bundle savedInstanceState) {
12        super.onCreate(savedInstanceState);
13        setContentView(R.layout.activity_cancel_ticket);
14        LinearLayoutManager layoutManager = new LinearLayoutManager(this);
15        recyclerView = (RecyclerView) findViewById(R.id.cidCancelRecyclerView);
16        recyclerView.setLayoutManager(layoutManager);
17        recyclerView.setAdapter(mAdapter);
18        recyclerView.setHasFixedSize(true);
19        mActionBarToolbar = (android.support.v7.widget.Toolbar) findViewById(R.id.toolbar);
20        setSupportActionBar(mActionBarToolbar);
21        mActionBarToolbar.setNavigationIcon(R.drawable.ic_arrow_back_24dp);
22        setTitle("Ακύρωση Εισιτηρίου");
23
24        mListCancelTickets = new ArrayList<>();
25
26        String URL = "content://com.bustravelcorfu.provider/" + BusTravelProvider.CANCEL_TICKET_PATH + "";
27        final Uri _cancel = Uri.parse(URL);
28        final Cursor cursor = getContentResolver().query(_cancel, projection: null, selection: null, selectionArgs: null, sortOrder: null);
29
30        if (cursor != null) {
31            while (cursor.moveToNext()) {
32                CancelTickets cancelTickets = new CancelTickets();
33                cancelTickets.setId(cursor.getLong(columnIndex 0));
34                cancelTickets.setCapoint(cursor.getString(columnIndex 1));
35                cancelTickets.setCdestination(cursor.getString(columnIndex 2));
36                cancelTickets.setCtickets(cursor.getString(columnIndex 3));
37                cancelTickets.setCdate(cursor.getString(columnIndex 4));
38                cancelTickets.setCtimetravel(cursor.getString(columnIndex 5));
39                cancelTickets.setCnumberofseat(cursor.getString(columnIndex 6));
40                cancelTickets.setCcaprice(cursor.getString(columnIndex 7));
41
42                mListCancelTickets.add(cancelTickets);
43            }
44        }
45
46        if (mListCancelTickets.size() > 0) {
47            recyclerView.setVisibility(View.VISIBLE);
48            mAdapter = new CancelTicketAdapter(this, mListCancelTickets, activity: CancelTicketActivity.this);
49            recyclerView.setAdapter(mAdapter);
50        } else {
51            RelativeLayout cidEmptyRecyclerView = (RelativeLayout) findViewById(R.id.cidEmptyRecyclerView);
52            cidEmptyRecyclerView.setVisibility(View.VISIBLE);
53        }
54    }
55
56    @Override
57    public boolean onOptionsItemSelected(MenuItem menuItem) {
58        switch (menuItem.getItemId()) {
59            case android.R.id.home:
60                Intent mainActivity = new Intent(packageContext: this, MainActivity.class);
61                startActivity(mainActivity);
62                break;
63        }
64        return super.onOptionsItemSelected(menuItem);
65    }
66
67    @Override
68    public void onBackPressed() { super.onBackPressed(); }
69
70    @Override
71    protected void onResume() { super.onResume(); }
72 }
```

Εικόνα 109 : CancelTicketActivity.java

7.2.20 CancelTicketAdapter.java

```
1 package com.bustravelcorfu.adapters;
2
3 import ...
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18 public class CancelTicketAdapter extends RecyclerView.Adapter<CancelTicketHolder> {
19     private List<CancelTickets> mListCancelTickets;
20     private Context mContext;
21     private CancelTicketActivity mActivity;
22
23     public CancelTicketAdapter(Context context, List<CancelTickets> cancelTicketsList, CancelTicketActivity activity) {
24         this.mContext = context;
25         this.mListCancelTickets = cancelTicketsList;
26         this.mActivity = activity;
27     }
28
29     @NonNull
30     @Override
31     public CancelTicketHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
32         View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.cancel_ticket_list_item, parent, attachToRoot: false);
33         return new CancelTicketHolder(view);
34     }
35
36     @Override
37     public void onBindViewHolder(@NonNull final CancelTicketHolder holder, int position) {
38         final CancelTickets cancelTickets = mListCancelTickets.get(position);
39         final String date = cancelTickets.getCadate();
40         Date _date = Calendar.getInstance().getTime();
41         DateFormat df = new SimpleDateFormat("d-M-yyyy ", Locale.getDefault());
42         final String today = df.format(_date);
43
44         holder.mPoint_history.setText(new StringBuilder()
45             .append(cancelTickets.getCapoint())
46             .append(" - "));
47
48         holder.mDestination_history.setText(cancelTickets.getCadestination());
49         holder.mDate_history.setText(cancelTickets.getCadate());
50         holder.mPrice_history.setText(cancelTickets.getCaprice());
51
52         holder.itemView.setOnClickListener((v) -> {
53             Intent intent = new Intent(mContext, PreviewCancelActivity.class);
54             intent.putExtra("name", "point", cancelTickets.getCapoint());
55
56             intent.putExtra("name", "tickets", cancelTickets.getCancelTickets());
57             intent.putExtra("name", "date", cancelTickets.getCadate());
58             intent.putExtra("name", "time", cancelTickets.getCatime travel());
59             intent.putExtra("name", "seats", cancelTickets.getCanumberofseat());
60             intent.putExtra("name", "price", cancelTickets.getCaprice());
61             mContext.startActivity(intent);
62         });
63
64         holder.mCancel_ticket.setOnClickListener((v) -> {
65             if(date.equals(today)){
66                 Toast.makeText(mContext, "Δεν μπορείτε να ακυρώσετε το εισιτήριο σας, λόγω διαθεσιμότητας", Toast.LENGTH_LONG).show();
67             }else {
68                 dialogCancelTicket(holder, holder.getAdapterPosition());
69             }
70         });
71
72         holder.mRemoveItem.setOnClickListener((v) -> {
73             dialogRemoveCancelTicket(holder, holder.getAdapterPosition());
74         });
75     }
76
77     private void dialogCancelTicket(final CancelTicketHolder holder, final int position) {
78         final CancelTickets cancelTickets = mListCancelTickets.get(position);
79         final long id = cancelTickets.getId();
80         LayoutInflater inflater = LayoutInflater.from(mActivity);
81         View dialogView = inflater.inflate(R.layout.dialog_cancel_ticket, root: null);
82         final AlertDialog dialog = new AlertDialog.Builder(mActivity).create();
83         Button btnNo = dialogView.findViewById(R.id.btn_no);
84         Button btnYes = dialogView.findViewById(R.id.btn_yes);
85         btnYes.setOnClickListener((v) -> {
86             mContext.getContentResolver().delete(BusTravelProvider.CONTENT_CANCEL_TICKET_DELETE, String.valueOf(id), null);
87             mListCancelTickets.remove(holder.getAdapterPosition());
88             notifyItemRemoved(holder.getAdapterPosition());
89             Toast.makeText(mContext, "Το εισιτήριό σας ακυρώθηκε. Η επιχείρησή σας χρεώθηκε με την προγραμματισμένη εντός 3 εργάσιμων ημερών.", Toast.LENGTH_LONG).show();
90             new android.os.Handler().post(() -> { dialog.dismiss(); });
91         });
92
93         dialog.show();
94     }
95
96     private void dialogRemoveCancelTicket(final CancelTicketHolder holder, final int position) {
97         final CancelTickets cancelTickets = mListCancelTickets.get(position);
98         final long id = cancelTickets.getId();
99         LayoutInflater inflater = LayoutInflater.from(mActivity);
100         View dialogView = inflater.inflate(R.layout.dialog_confirm_delete, root: null);
101         final AlertDialog dialog = new AlertDialog.Builder(mActivity).create();
102         Button btnNo = dialogView.findViewById(R.id.btn_no);
103         Button btnYes = dialogView.findViewById(R.id.btn_yes);
104         btnYes.setOnClickListener((v) -> {
105             mContext.getContentResolver().delete(BusTravelProvider.CONTENT_CANCEL_TICKET_DELETE, String.valueOf(id), null);
106             mListCancelTickets.remove(holder.getAdapterPosition());
107             notifyItemRemoved(holder.getAdapterPosition());
108             Toast.makeText(mContext, "Εισιτήριο διατεθειμένο", Toast.LENGTH_LONG).show();
109             new android.os.Handler().post(() -> { dialog.dismiss(); });
110         });
111
112         btnNo.setOnClickListener((v) -> { dialog.dismiss(); });
113         dialog.setView(dialogView);
114         dialog.show();
115     }
116
117     @Override
118     public int getItemCount() { return mListCancelTickets.size(); }
119 }
```

Εικόνα 110 : CancelTicketAdapter.java

7.2.20 CancelTicketHolder.java

```
package com.bustravelcorfu.holders;

import ...

public class CancelTicketHolder extends RecyclerView.ViewHolder {

    public TextView mPoint_history;
    public TextView mDestination_history;
    public TextView mDate_history;
    public TextView mPrice_history;
    public ImageView mCancel_Ticket;
    public ImageView mRemoveItem;

    public CancelTicketHolder(View itemView) {
        super(itemView);

        mPoint_history = (TextView) itemView.findViewById(R.id.point_history);
        mDestination_history = (TextView) itemView.findViewById(R.id.destination_history);
        mDate_history = (TextView) itemView.findViewById(R.id.date_history);
        mPrice_history = (TextView) itemView.findViewById(R.id.price_history);
        mCancel_Ticket = (ImageView) itemView.findViewById(R.id.cancel_item_remove);
        mRemoveItem = (ImageView) itemView.findViewById(R.id.item_remove);
    }
}
```

Εικόνα 111 : CancelTicketHolder.java

7.2.21 AboutActivity.java

```
package com.bustravelcorfu.activities;

import ...

public class AboutActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_about);
        Objects.requireNonNull(getSupportActionBar()).setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        setTitle("HAnpopeptec");

        ImageView mEmail = findViewById(R.id.send_email);
        ImageView mRate = findViewById(R.id.rate);
        TextView mVersion = findViewById(R.id.version);

        String version = "";
        PackageManager pm = this.getPackageManager();
        try {
            PackageInfo pi = pm.getPackageInfo(this.getPackageName(), PackageManager.GET_META_DATA);
            version = pi.versionName;
        } catch (PackageManager.NameNotFoundException e) {
            e.printStackTrace();
        }
        String appName = "Corfu m-ticket" + " " + version;
        mVersion.setText(appName);
    }

    public void rateApplication (View view) {
        Intent intent = new Intent(Intent.ACTION_VIEW)
            .setData(Uri
                .parse("https://play.google.com/store")); /*replace url from app in play store */
        startActivity(intent);
    }

    public void sendFeedbackmail (View view){
        Intent Email = new Intent(Intent.ACTION_SEND);
        Email.setType("text/email");
        Email.putExtra(Intent.EXTRA_EMAIL, new String[]{"infotravelbus08@gmail.com"});
        Email.putExtra(Intent.EXTRA_SUBJECT, value: "Feedback from " + Build.MODEL + " user");
        Email.putExtra(Intent.EXTRA_TEXT, value: "Hallo," + "");
        startActivity(Intent.createChooser(Email, title: "Send Feedback:"));
    }

    @SuppressWarnings("RestrictedApi")
    @Override
    public boolean onCreateOptionsMenu(Menu menu) { return super.onCreateOptionsMenu(menu); }

    @Override
    public void onBackPressed() { super.onBackPressed(); }

    @Override
    public boolean onOptionsItemSelected(MenuItem menuItem) {
        switch (menuItem.getItemId()) {
            case android.R.id.home:
                onBackPressed();
                break;
        }

        return super.onOptionsItemSelected(menuItem);
    }
}
```

Εικόνα 112 : AboutActivity.java

7.2.22 ConditionsActivity.java

```
package com.bustravelcorfu.activities;

import ...

public class ConditionsActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_conditions);
        Objects.requireNonNull(getSupportActionBar()).setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    }

    @SuppressWarnings("RestrictedApi")
    @Override
    public boolean onCreateOptionsMenu(Menu menu) { return super.onCreateOptionsMenu(menu); }

    @Override
    public void onBackPressed() { super.onBackPressed(); }

    @Override
    public boolean onOptionsItemSelected(MenuItem menuItem) {

        switch (menuItem.getItemId()) {
            case android.R.id.home:
                onBackPressed();
                break;
        }

        return super.onOptionsItemSelected(menuItem);
    }
}
```

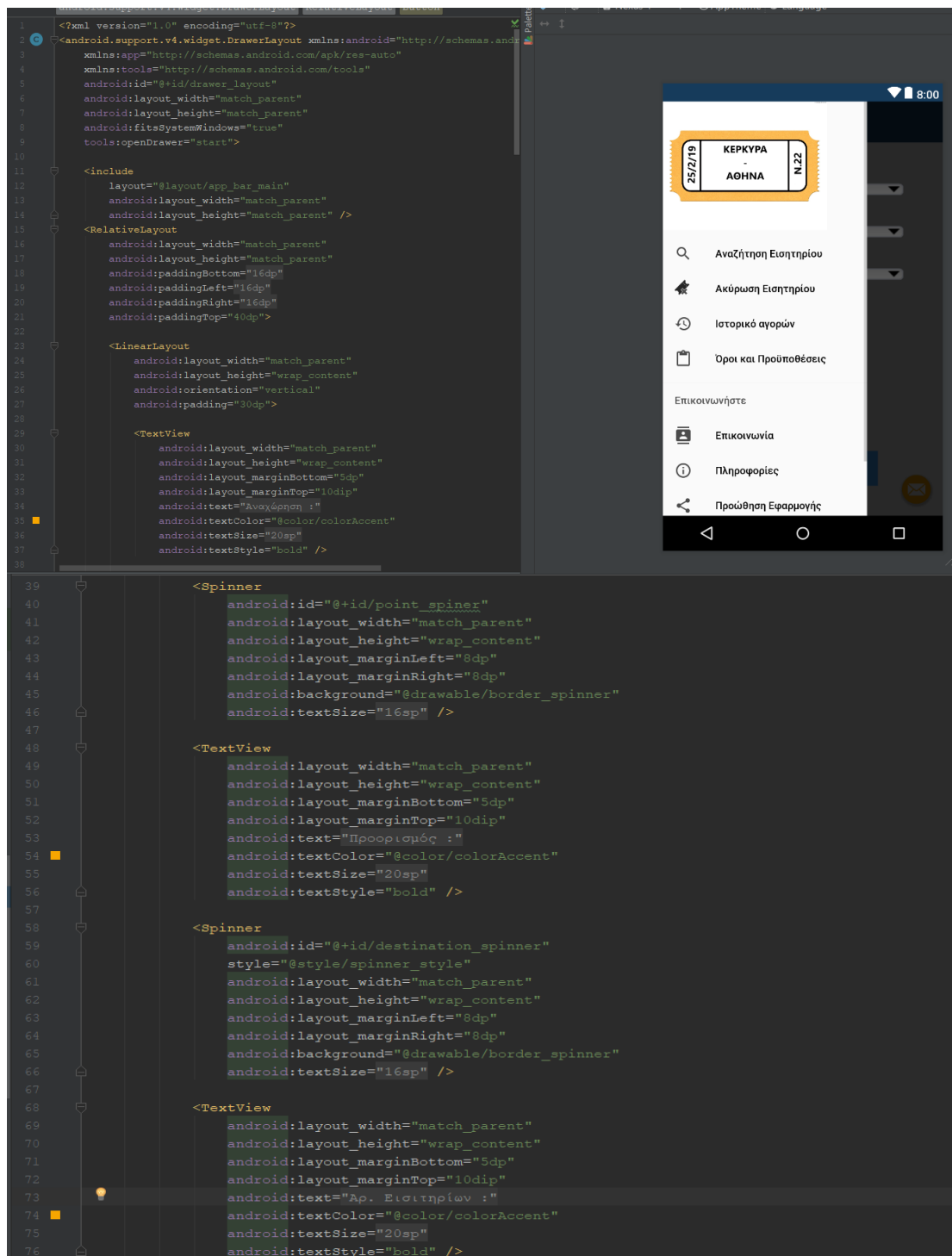
Εικόνα 113 : ConditionsActivity.java

7.2.23 SqliteDatabase.java

```
1 package com.bustravelcorfu.database;
2
3 import ...
4
5
6
7
8
9
10
11 public class SqliteDatabase extends SQLiteOpenHelper {
12     //Database Name
13     private static final String DATABASE_NAME = "bustravel.db";
14
15     //Database Version
16     private static final int DATABASE_VERSION = 1;
17
18     //Tables
19     private static final String TABLE_USERS = UsersDao.TABLENAME;
20     private static final String TABLE_HISTORY_BOOKING = History_BookingDao.TABLENAME;
21     private static final String TABLE_CANCEL_TICKET = CancelTicketsDao.TABLENAME;
22
23     //Users Columns
24     private static final String COLUMN_USER_ID = UsersDao.Properties.Id.columnName;
25     private static final String COLUMN_USERNAME = UsersDao.Properties.Username.columnName;
26     private static final String COLUMN_USER_PASSWORD = UsersDao.Properties.Password.columnName;
27     private static final String COLUMN_PIN_USER = UsersDao.Properties.Pinuser.columnName;
28
29     //History Booking Columns
30     private static final String COLUMN_HISTORY_BOOK_ID = History_BookingDao.Properties.Id.columnName;
31     private static final String COLUMN_DEPARTURE_POINT = History_BookingDao.Properties.Departure_point.columnName;
32     private static final String COLUMN_DESTINATION = History_BookingDao.Properties.Destination.columnName;
33     private static final String COLUMN_NUMBER_OF_TICKETS = History_BookingDao.Properties.Number_of_tickets.columnName;
34     private static final String COLUMN_DATE = History_BookingDao.Properties.Date.columnName;
35     private static final String COLUMN_TIME_TRAVEL = History_BookingDao.Properties.Time_travel.columnName;
36     private static final String COLUMN_NUMBER_OF_SEAT = History_BookingDao.Properties.Number_of_seat.columnName;
37     private static final String COLUMN_PRICE = History_BookingDao.Properties.Price.columnName;
38
39     //Cancel Tickets Columns
40     private static final String COLUMN_CANCEL_TICKET_ID = CancelTicketsDao.Properties.Id.columnName;
41     private static final String COLUMN_CA_DEPARTURE_POINT = CancelTicketsDao.Properties.Capoint.columnName;
42     private static final String COLUMN_CA_DESTINATION = CancelTicketsDao.Properties.Cadestination.columnName;
43     private static final String COLUMN_CA_TICKETS = CancelTicketsDao.Properties.Catickets.columnName;
44     private static final String COLUMN_CA_DATE = CancelTicketsDao.Properties.Cadate.columnName;
45     private static final String COLUMN_CA_TIME_TRAVEL = CancelTicketsDao.Properties.Catimetravel.columnName;
46     private static final String COLUMN_CA_NUMBER_SEAT = CancelTicketsDao.Properties.Canumberofseat.columnName;
47
48     private static final String COLUMN_CA_PRICE = CancelTicketsDao.Properties.Caprice.columnName;
49
50     private static SqliteDatabase instance;
51
52     public SqliteDatabase(Context context) {
53         super(context, DATABASE_NAME, factory: null, DATABASE_VERSION);
54     }
55
56     @Override
57     public void onCreate(SQLiteDatabase db) {
58         String CREATE_USERS_TABLE = "CREATE TABLE " + TABLE_USERS + "(" + COLUMN_USER_ID + " INTEGER PRIMARY KEY," + COLUMN_USERNAME + " TEXT," + COLUMN_USER_PASSWORD + " TEXT," + COLUMN_PIN_USER + " TEXT)";
59         String CREATE_HISTORY_BOOKING_TABLE = "CREATE TABLE " + TABLE_HISTORY_BOOKING + "(" + COLUMN_HISTORY_BOOK_ID + " INTEGER PRIMARY KEY," + COLUMN_DEPARTURE_POINT + " TEXT," + COLUMN_DESTINATION + " TEXT," + COLUMN_NUMBER_OF_TICKETS + " TEXT," + COLUMN_DATE + " TEXT," + COLUMN_TIME_TRAVEL + " TEXT," + COLUMN_NUMBER_OF_SEAT + " TEXT," + COLUMN_PRICE + " TEXT)";
60         String CREATE_CANCEL_TICKETS_TABLE = "CREATE TABLE " + TABLE_CANCEL_TICKET + "(" + COLUMN_CANCEL_TICKET_ID + " INTEGER PRIMARY KEY," + COLUMN_CA_DEPARTURE_POINT + " TEXT," + COLUMN_CA_DESTINATION + " TEXT," + COLUMN_CA_TICKETS + " TEXT," + COLUMN_CA_DATE + " TEXT," + COLUMN_CA_TIME_TRAVEL + " TEXT," + COLUMN_CA_NUMBER_SEAT + " TEXT," + COLUMN_CA_PRICE + " TEXT)";
61         db.execSQL(CREATE_USERS_TABLE);
62         db.execSQL(CREATE_HISTORY_BOOKING_TABLE);
63         db.execSQL(CREATE_CANCEL_TICKETS_TABLE);
64     }
65
66     @Override
67     public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
68         db.execSQL("DROP TABLE IF EXISTS " + TABLE_USERS);
69         db.execSQL("DROP TABLE IF EXISTS " + TABLE_HISTORY_BOOKING);
70         db.execSQL("DROP TABLE IF EXISTS " + TABLE_CANCEL_TICKET);
71     }
72
73     public static SqliteDatabase get(Context ctx) {
74         if (instance == null)
75             instance = new SqliteDatabase(ctx);
76         return instance;
77     }
78 }
79
```

Εικόνα 114 : SqliteDatabase.java

7.2.24 activity_main.xml



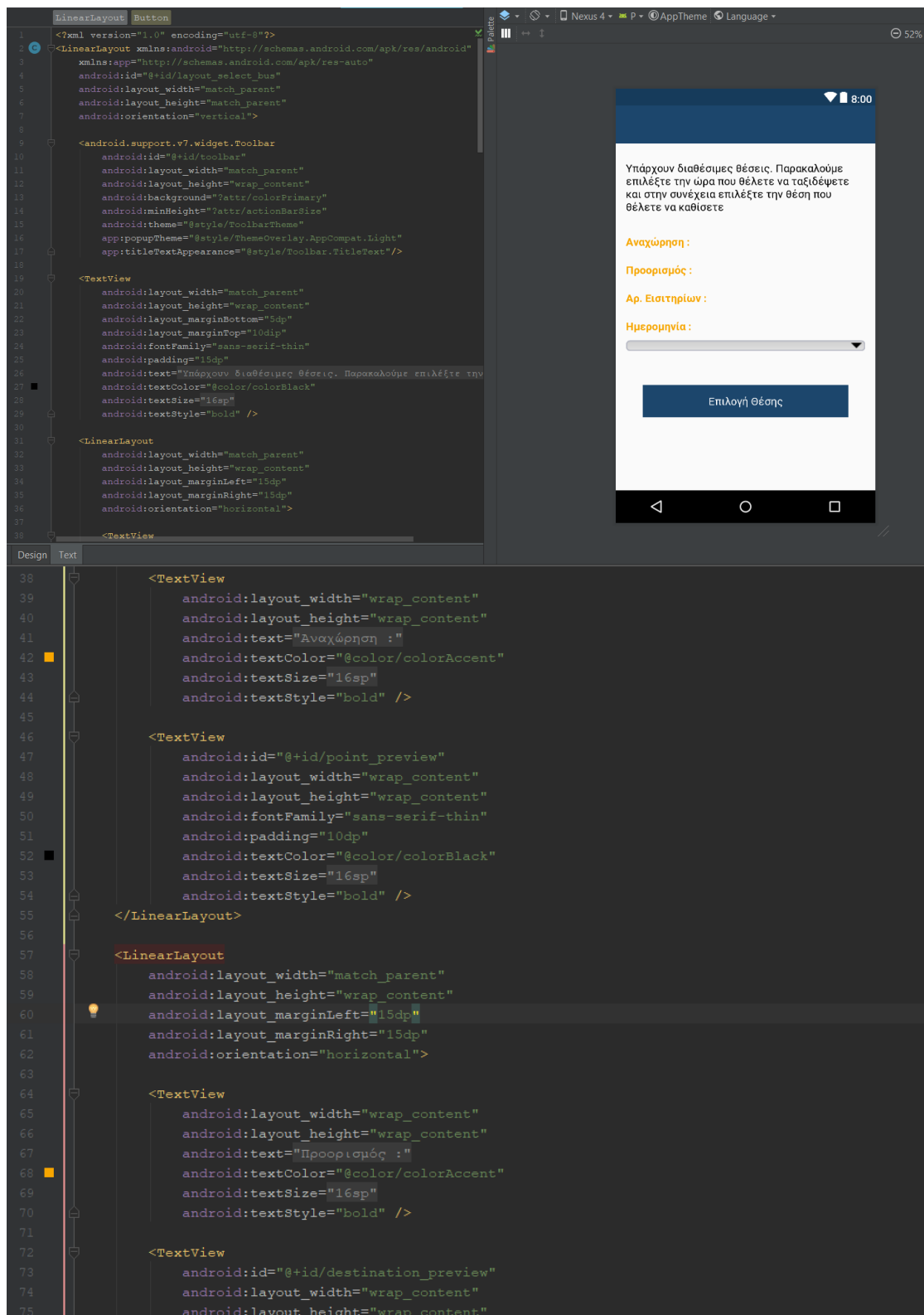
```

78     <Spinner
79         android:id="@+id/ticket_spinner"
80         style="@style/spinner_style"
81         android:layout_width="match_parent"
82         android:layout_height="wrap_content"
83         android:layout_marginLeft="8dp"
84         android:layout_marginRight="8dp"
85         android:background="@drawable/border_spinner"
86         android:textSize="16sp" />
87
88     <LinearLayout
89         android:layout_width="match_parent"
90         android:layout_height="wrap_content"
91         android:layout_marginLeft="8dp"
92         android:layout_marginRight="8dp"
93         android:layout_marginTop="20dp"
94         android:orientation="horizontal">
95
96         <Button
97             android:id="@+id/myDatePickerButton"
98             android:layout_width="wrap_content"
99             android:layout_height="wrap_content"
100             android:text="Ημερομηνία:" />
101
102         <TextView
103             android:id="@+id/showMyDate"
104             android:layout_width="wrap_content"
105             android:layout_height="wrap_content"
106             android:textColor="@color/colorAccent"
107             android:textSize="16sp"
108             android:textStyle="bold" />
109     </LinearLayout>
110
111 </LinearLayout>
112 <LinearLayout
113     android:layout_width="0px"
114     android:layout_height="0px"
115     android:focusable="true"
116
117     <TextView
118         android:id="@+id/button_overview"
119         style="@style/Base.TextAppearance.AppCompat.Small"
120         android:layout_width="wrap_content"
121         android:layout_height="wrap_content"
122         android:layout_alignParentBottom="true"
123         android:layout_centerHorizontal="true"
124         android:layout_marginBottom="25dp"
125         android:background="@color/colorPrimary"
126         android:fontFamily="sans-serif-thin"
127         android:paddingLeft="20dp"
128         android:paddingRight="20dp"
129         android:text="Αναζήτηση Εισιτηρίων"
130         android:textAllCaps="false"
131         android:textColor="@color/colorWhite"
132         android:textSize="16sp"
133         android:textStyle="bold" />
134 </RelativeLayout>
135
136 <android.support.design.widget.NavigationView
137     android:id="@+id/nav_view"
138     android:layout_width="wrap_content"
139     android:layout_height="match_parent"
140     android:layout_gravity="start"
141     android:fitsSystemWindows="true"
142     app:headerLayout="@layout/nav_header_main"
143     app:menu="@menu/activity_main_drawer" />
144 </android.support.v4.widget.DrawerLayout>
145
146
147
148
149
150
151
152
153
154
155
156
157

```

Εικόνα 115 : Activity_main.xml

7.2.25 Layout_select_bus.xml



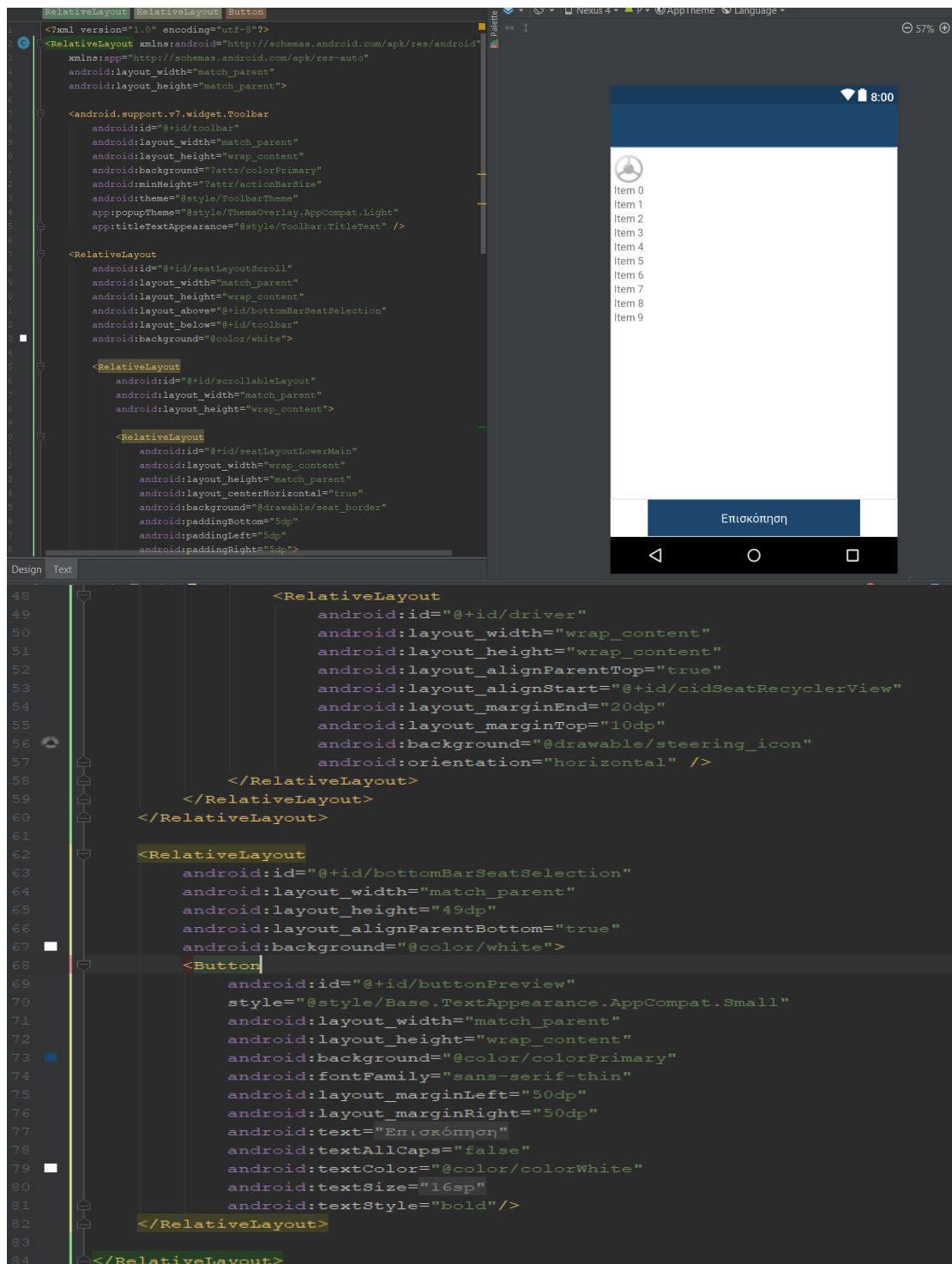
```

76         android:fontFamily="sans-serif-thin"
77         android:padding="10dp"
78         android:textColor="@color/colorBlack"
79         android:textSize="16sp"
80         android:textStyle="bold" />
81     </LinearLayout>
82
83     <LinearLayout
84         android:layout_width="match_parent"
85         android:layout_height="wrap_content"
86         android:layout_marginLeft="15dp"
87         android:layout_marginRight="15dp"
88         android:orientation="horizontal">
89
90         <TextView
91             android:layout_width="wrap_content"
92             android:layout_height="wrap_content"
93             android:text="Αρ. Εισιτηρίων : "
94             android:textColor="@color/colorAccent"
95             android:textSize="16sp"
96             android:textStyle="bold" />
97
98         <TextView
99             android:id="@+id/tickets_preview"
100             android:layout_width="wrap_content"
101             android:layout_height="wrap_content"
102             android:fontFamily="sans-serif-thin"
103             android:padding="10dp"
104             android:textColor="@color/colorBlack"
105             android:textSize="16sp"
106             android:textStyle="bold" />
107     </LinearLayout>
108
109     <LinearLayout
110         android:layout_width="match_parent"
111         android:layout_height="wrap_content"
112         android:layout_marginLeft="15dp"
113         android:layout_marginRight="15dp"
114
115
116
117
118
119
120
121
122
123
124
125
126     <Spinner
127         android:id="@+id/select_bus_spinner"
128         android:layout_width="match_parent"
129         android:layout_height="wrap_content"
130         android:layout_marginLeft="15dp"
131         android:layout_marginRight="15dp"
132         android:background="@drawable/border_spinner"
133         android:textSize="16sp" />
134
135
136     <Button
137         android:id="@+id/button_select_seats"
138         style="@style/Base.TextAppearance.AppCompat.Small"
139         android:layout_width="match_parent"
140         android:layout_height="wrap_content"
141         android:layout_marginLeft="40dp"
142         android:layout_marginRight="40dp"
143         android:layout_marginTop="50dp"
144         android:background="@color/colorPrimary"
145         android:fontFamily="sans-serif-thin"
146         android:text="Επιλογή Θέσης"
147         android:textAllCaps="false"
148         android:textColor="@color/colorWhite"
149         android:textSize="16sp"
150         android:textStyle="bold" />
151 </LinearLayout>

```

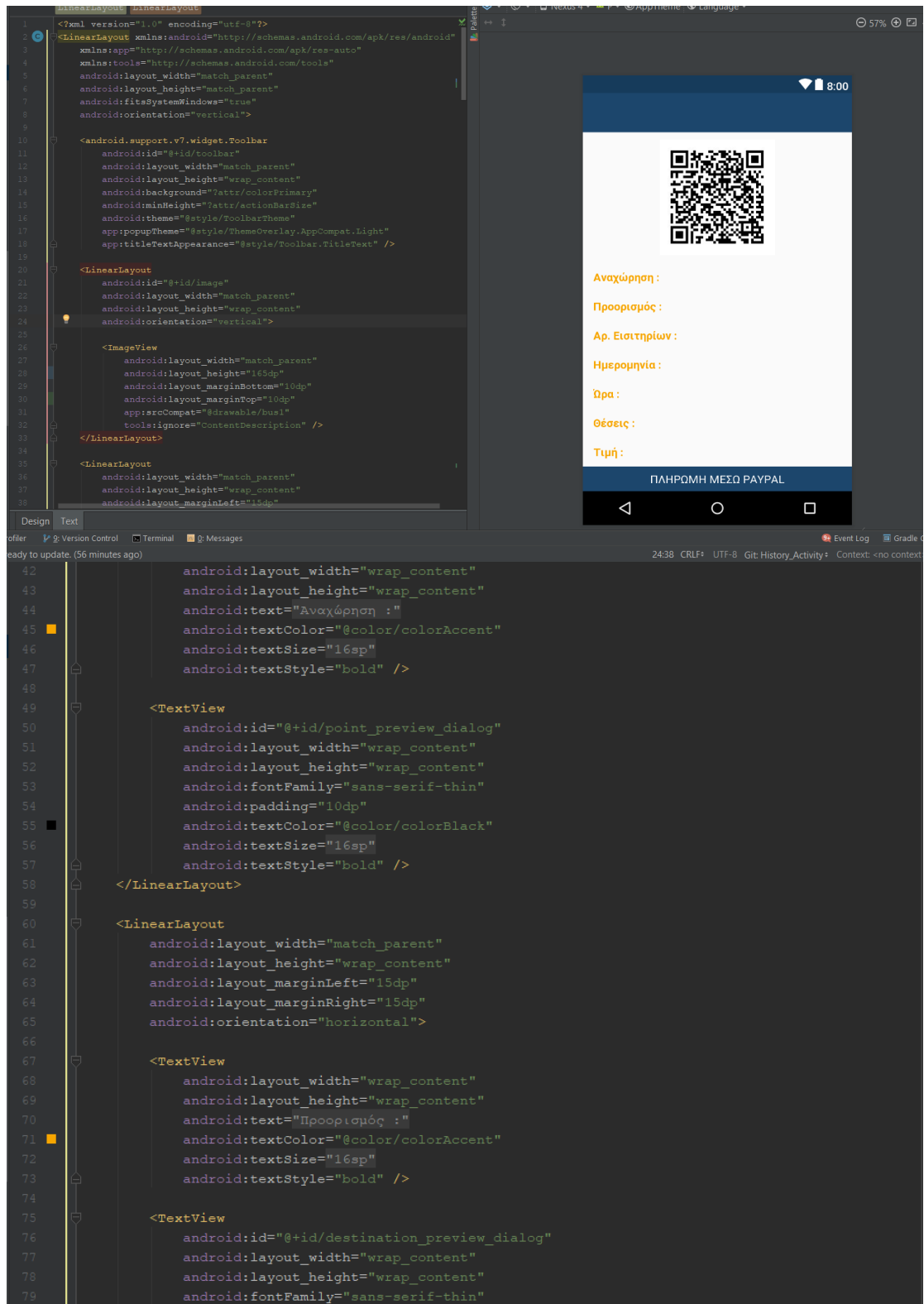
Εικόνα 116 : Layout_select_bus.xml

7.2.26 Activity_seat_selection_screen.xml



Εικόνα 117 :Activity_seat_selection_screen.xml

7.2.27 Activity_booking.xml



Εικόνα 118 : activity_booking.xml

7.2.27 AndroidManifest.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.bustravelcorfu">
4
5     <!-- for most things, including card.io & paypal -->
6     <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
7     <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
8     <uses-permission android:name="android.permission.INTERNET" />
9
10    <application
11        android:name=".AppContext"
12        android:allowBackup="false"
13        android:icon="@mipmap/ic_launcher"
14        android:label="Corfu m-ticket"
15        android:roundIcon="@mipmap/ic_launcher_round"
16        android:supportRtl="false"
17        android:theme="@style/AppTheme">
18        <activity
19            android:name=".activities.LoginActivity"
20            android:label="Corfu m-ticket"
21            android:screenOrientation="portrait"
22            android:theme="@style/AppTheme">
23            <intent-filter>
24                <action android:name="android.intent.action.MAIN" />
25
26                <category android:name="android.intent.category.LAUNCHER" />
27            </intent-filter>
28        </activity>
29        <activity
30            android:name=".activities.MainActivity"
31            android:label="Corfu m-ticket"
32            android:screenOrientation="portrait"
33            android:theme="@style/AppTheme.NoActionBar" />
34        <activity
35            android:name=".WelcomeScreen"
36            android:screenOrientation="portrait"
37            android:theme="@style/CustomWelcomeScreenTheme" />
38        <activity
39            android:name=".activities.SelectBusActivity"
40            android:screenOrientation="portrait"
41            android:theme="@style/AppTheme" />
42        <activity
43            android:name=".activities.ForgotPasswordActivity"
44            android:screenOrientation="portrait"
45            android:theme="@style/AppTheme" />
46        <activity
47            android:name=".activities.ResetPasswordActivity"
48            android:label="Επαναφορά Κωδικού"
49            android:screenOrientation="portrait" />
50        <activity
51            android:name=".activities.SeatSelectionActivity"
52            android:screenOrientation="portrait"
53            android:theme="@style/AppTheme" />
54        <activity
55            android:name=".activities.BookingActivity"
56            android:screenOrientation="portrait"
57            android:theme="@style/AppTheme" />
58        <activity
59            android:name=".activities.HistoryBookingActivity"
60            android:screenOrientation="portrait"
61            android:theme="@style/AppTheme" />
62        <activity
63            android:name=".activities.PreviewHistoryActivity"
64            android:screenOrientation="portrait"
65            android:theme="@style/AppTheme" />
66        <activity
67            android:name=".activities.CancelTicketActivity"
68            android:screenOrientation="portrait"
69            android:theme="@style/AppTheme" />
70        <activity
71            android:name=".activities.AboutActivity"
72            android:screenOrientation="portrait"
73            android:theme="@style/AppThemeAbout" />
74        <activity android:name=".activities.ContitionsActivity"
75            android:screenOrientation="portrait"
76            android:theme="@style/AppThemeAbout"
```

```

76         android:theme="@style/AppThemeAbout"
77         android:label="@string/app_name_about" />
78     <activity android:name=".activities.PreviewCancelActivity"
79             android:screenOrientation="portrait"
80             android:theme="@style/AppTheme" />
81
82     <service
83         android:name="com.paypal.android.sdk.payments.PayPalService"
84         android:exported="false" />
85
86     <activity android:name="com.paypal.android.sdk.payments.PaymentActivity" />
87     <activity android:name="com.paypal.android.sdk.payments.LoginActivity" />
88     <activity android:name="com.paypal.android.sdk.payments.PaymentMethodActivity" />
89     <activity android:name="com.paypal.android.sdk.payments.PaymentConfirmActivity" />
90     <activity android:name="com.paypal.android.sdk.payments.PayPalFuturePaymentActivity" />
91     <activity android:name="com.paypal.android.sdk.payments.FuturePaymentConsentActivity" />
92     <activity android:name="com.paypal.android.sdk.payments.FuturePaymentInfoActivity" />
93     <activity
94         android:name="io.card.payment.CardIOActivity"
95         android:configChanges="keyboardHidden|orientation" />
96     <activity android:name="io.card.payment.DataEntryActivity" />
97
98     <provider
99         android:name=".BusTravelProvider"
100         android:authorities="com.bustravelcorfu.provider"
101         android:exported="false" />
102 </application>
103
104 </manifest>

```

Εικόνα 119 : androidmanifest.xml

7.2.27 Build.gradle

```

1  apply plugin: 'com.android.application'
2
3  android {
4      compileSdkVersion 27
5      defaultConfig {
6          applicationId "com.bustravelcorfu"
7          minSdkVersion 19
8          targetSdkVersion 27
9          versionCode 1
10         versionName "1.0"
11         testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
12         vectorDrawables {
13             useSupportLibrary = true
14         }
15     }
16     buildTypes {
17         release {
18             minifyEnabled true
19             proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
20         }
21     }
22 }
23
24 // In your app projects build.gradle file:
25 apply plugin: 'com.android.application'
26 apply plugin: 'org.greenrobot.greendao' // apply plugin
27
28 greendao {
29     schemaVersion 1
30     targetGenDir "src/main/java/"
31 }
32
33 dependencies {
34     implementation fileTree(dir: 'libs', include: ['*.jar'])
35     implementation 'com.android.support:appcompat-v7:27.1.1'
36     implementation 'com.android.support:design:27.1.1'
37     implementation 'com.android.support.constraint:constraint-layout:1.1.2'
38     testImplementation 'junit:junit:4.12'
39     androidTestImplementation 'com.android.support.test:runner:1.0.2'
40     androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
41
42     implementation 'org.greenrobot:greendao:3.2.2' // add library
43     implementation 'net.zetetic:android-database-sqlcipher:3.5.6'
44     implementation 'com.paypal.sdk:paypal-android-sdk:2.16.0'
45
46     implementation 'com.android.support:preference-v7:27.1.1'
47     implementation 'com.android.support:cardview-v7:27.1.1'
48
49 }

```

Εικόνα 120 : Build.gradle

8. Σύνοψη

Στο κεφάλαιο αυτό θα αναφερθούμε στα συμπεράσματα που προέκυψαν από την υλοποίηση της εφαρμογής μας καθώς και στις μελλοντικές επεκτάσεις της. Θα προτείνουμε νέες ιδέες και υπηρεσίες προκειμένου να βελτιωθεί η εμπειρία του Κερκυραίου πολίτη και να μειωθεί το άγχος και η ταλαιπωρία των ταξιδιών.

8.1 Συμπεράσματα

Στην συγκεκριμένη πτυχιακή εργασία ασχοληθήκαμε με την ανάπτυξη μιας εφαρμογής για smartphones, την Corfu m-ticket. Η υλοποίηση της πραγματοποιήθηκε πάνω στην βάση της πλατφόρμας Android Studio, σε γλώσσα προγραμματισμού Java. Στην αρχή της πτυχιακής ασχοληθήκαμε με την ανάλυση του Android και του Android Studio. Είδαμε την αρχιτεκτονική του λογισμικού Android και τα στάδια τα οποία το απαρτίζουν. Στην συνέχεια αναλύσαμε την εφαρμογή σε θεωρητικό επίπεδο, και αφού ολοκληρώθηκε και αυτή η διαδικασία, τότε ξεκίνησε η υλοποίηση της, όσο αναφορά το προγραμματιστικό κομμάτι.

Γενικά η έμπνευση και ο σχεδιασμός της εφαρμογής σε θεωρητικό επίπεδο και η αποτύπωση της ιδέας στο χαρτί υπήρξε το εύκολο κομμάτι της προσπάθειας ολοκλήρωσης της συγκεκριμένης εφαρμογής. Το δύσκολο κομμάτι της, ήταν η υλοποίηση και ανάπτυξη της, καθώς υπήρξε μία πολύ απαιτητική και χρονοβόρα διαδικασία. Χρειάστηκαν πολλές ώρες μελέτης, αναζήτησης στο διαδίκτυο και παρακολούθησης online μαθημάτων σε διάφορες πλατφόρμες προκειμένου να φτάσουμε στο σημείο να ανταποκριθούμε στις απαιτήσεις της πτυχιακής εργασίας. Επίσης πολλές ώρες δαπανήθηκαν στο να υλοποιηθεί ένα περιβάλλον στην εφαρμογή, το οποίο όταν θα το αντίκριζε ο χρήστης να του ήταν φιλικό και πάνω από όλα εύχρηστο. Παρόλες τις δυσκολίες όμως, το αποτέλεσμα που καταλήξαμε είναι ικανοποιητικό και θεωρούμε ότι άξιζε ο κόπος, και η γνώση που αποκομίστηκε από αυτή την προσπάθεια θα μας βοηθήσει και στο μέλλον.

8.2 Μελλοντικές Επεκτάσεις

Η εφαρμογή Corfu m-ticket δημιουργήθηκε για την καλύτερη εξυπηρέτηση των επιβατών που έχουν σκοπό να ταξιδέψουν με τα υπεραστικά ΚΤΕΛ Κέρκυρας. Πρόκειται για μία εφαρμογή που βοηθάει τους πολίτες να κάνουν κρατήσεις σε δρομολόγια από και προς την Κέρκυρα. Μελλοντικός σκοπός της συγκεκριμένης εφαρμογής είναι από την στιγμή που υπάρχουν φοιτητές στην πόλη της Κέρκυρας, να δοθεί η δυνατότητα και σε αυτούς να κάνουν κρατήσεις εισιτηρίων με την έκπτωση που τους αντιστοιχεί. Το ίδιο ισχύει και για τις ευπαθείς ομάδες που δικαιούνται έκπτωση.

Η εφαρμογή στο στάδιο που βρίσκεται δεν περιέχει όλες τις στάσεις του δρομολογίου που κάνουν τα Υπεραστικά λεωφορεία, ένας ακόμα μελλοντικός στόχος είναι αυξηθούν οι πόλεις για τις οποίες θα μπορεί ο χρήστης να κάνει κράτηση εισιτηρίου. Μία ακόμα προέκταση είναι η προσθήκη ειδοποιήσεων push προκειμένου να γνωστοποιείται στο χρήστη κάποιο συμβάν σχετικά με το δρομολόγιο του, όπως συμβαίνει με μεγάλες αεροπορικές εταιρείες. Τέλος, η δημοσίευση της εφαρμογής στην αγορά της Google (Play Store) είναι ένας από τους επιθυμητούς στόχους, προκειμένου να αξιοποιηθεί από το ευρύ κοινό και να διευκολύνει τους ταξιδιώτες.

9.Βιβλιογραφία

1. Εισαγωγή στον Προγραμματισμό Android , I.N.ΕΛΛΗΝΑΣ - N.I.ΕΛΛΗΝΑΣ, Εκδόσεις Τζιόλα, 2014
2. Android ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ, Paul Deitel - Harvey Deitel - Abbey Deitel, Εκδόσεις: Μ. Γκιούρδας, 2014
3. Ανάπτυξη Εφαρμογών με το Android, Lauren Darcey – Shane Conder, Εκδόσεις: Μ. Γκιούρδας, 2011
4. Android για Προγραμματιστές Μάθετε μέσα από Εφαρμογές, Paul Deitel - Harvey Deitel - Abbey Deitel – Michael Morgano, Εκδόσεις: Μ. Γκιούρδας, 2012
5. Burnette, Ed. Hello, Android: introducing Google’s mobile development platform. Pragmatic Bookshelf, 2009
6. Πλήρες Εγχειρίδιο της Java 7, Rogers Cadenhead, Εκδόσεις: Μ. Γκιούρδας, 2013
7. Βασικές Αρχές Τεχνολογίας Λογισμικού 8, Sommerville, Εκδόσεις ΚΛΕΙΔΑΡΙΘΜΟΣ, 2009
8. Android Programming: The Big Nerd Ranch Guide (3rd Edition) (Big Nerd Ranch Guides)
9. Οδηγός της XML με παραδείγματα, Benoit Marchal, Β.Γκιούρδας Εκδοτική, 2001
10. <https://developer.android.com/guide/>
11. http://dione.lib.unipi.gr/xmlui/bitstream/handle/unipi/9857/Darla_Ekaterini.pdf?sequence=1&isAllowed=y (Ντάρλα Αικατερίνη , Μεταπτυχιακή Διατριβή, Εφαρμογή Android: Έκδοση ηλεκτρονικού εισιτηρίου για τα ΜΜΜ της Αθήνας,2016)
12. <https://www.tutorialspoint.com/android/>
13. <http://www.greeceandroid.gr/development>
14. Udemy (Course- Android O & Java - The Complete Android Development Bootcamp) <https://www.udemy.com/android-app-development-with-java/>
15. Udemy(The Complete Android N developer Course)<https://www.udemy.com/complete-android-n-developer-course/learn/v4/t/lecture/5690148?start=0>
16. Πτυχιακή εργασία Ανάπτυξη εφαρμογής Android για την παροχή υπηρεσιών δρομολόγησης και καθοδήγησης στους χρήστες για τις καθημερινές τους δραστηριότητες Αρβανίτης Κώστας, 2014

http://nefeli.lib.teicrete.gr/browse/stef/epp/2014/ArvanitisAntonios/attached-document-1455101244-556850-16943/ArvanitisKonstantinos_KalykakisEleftherios2014.pdf

17. <https://github.com/samuelrafini/MovieApp>
18. <https://github.com/parthkamdar/mymatatu>
19. <https://www.youtube.com/user/eddydn71>
20. Android Studio Download Page <https://developer.android.com/studio/index.html>
21. <https://developer.android.com/training/volley/simple.html>
22. <https://androidresearch.wordpress.com>
23. Stack Overflow, <https://stackoverflow.com/>
24. <https://el.wikipedia.org>
25. <https://www.theatlans.com/charts/N1e4Qloq>
26. <https://www.statista.com/>
27. <https://developer.android.com/design/>
28. <https://developer.paypal.com/developer/applications/>
29. https://github.com/codepath/android_guides/wiki/Using-an-ArrayAdapter-with-ListView
30. <https://developer.android.com/guide/topics/manifest/manifest-intro>
31. <https://developer.android.com/reference/android/widget/Spinner>
32. <https://stackoverflow.com/questions/16796182/whats-the-difference-between-baseadapter-and-arrayadapter>