ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

ΚΟΙΝΩΝΙΚΟ ΔΙΚΤΥΟ ΖΩΓΡΑΦΙΩΝ ΣΕ ANDROID

SOCIAL NETWORK FOR PAINTINGS IN ANDROID

Διπλωματική Εργασία

Σούκας Ιωάννης - Σούκας Κλέοβις

Υπεύθυνος καθηγητής : Αλκιβιάδης Γ. Ακρίτας

Καθηγητής ΠΘ

Επιβλέπων Καθηγητής: Γεώργιος Σταμούλης

Καθηγητής ΠΘ

Βόλος, 2016

Ευχαριστίες

Με την ολοκλήρωση της διπλωματικής εργασίας, θα θέλαμε να ευχαριστήσουμε τον Κύριο Ακρίτα Αλκιβιάδη για την εμπιστοσύνη που μας έδειξε και την υποστήριξη κατά την εκπόνηση της παρούσας διπλωματικής καθώς και για τις πολύτιμες συμβουλές και την καθοδήγηση κατά την διάρκεια των σπουδών.

Επιπλέον ένα μεγάλο ευχαριστώ στους φίλους μας που ήταν δίπλα μας στα εύκολα και στα δύσκολα αυτής της διαδρομής, και φυσικά στην οικογένεια μας που χωρίς αυτούς δεν θα ήταν τίποτα εφικτό.

Περίληψη

Η παρούσα διπλωματική εργασία με τίτλο *“Social Network For Paintings In Android”* έχει ως στόχο την υλοποίηση μιας εφαρμογής για λειτουργικό σύστημα Google Android η οποία θα απευθύνεται σε λάτρεις της ζωγραφικής όλων των ηλικιών.

Πιο συγκεκριμένα ο κάθε χρήστης της εφαρμογής θα μπορεί να ζωγραφίσει πάνω στην οθόνη της συσκευής του με το δάχτυλό του μόνο ή με την βοήθεια κάποιας πένας αν προτιμάει και στη συνέχεια να ανεβάσει την ζωγραφιά του δημόσια στην γκαλερί της εφαρμογής όπου θα μπορούν άλλοι καλλιτέχνες (αλλά και ο ίδιος) να βλέπουν τα έργα που είναι ανεβασμένα και να σχολιάζουν ή να τα μαρκάρουν ως αγαπημένα.

Απώτερος σκοπός είναι η δημιουργία μιας κοινότητας καλλιτεχνών που θα ζωγραφίζουν με τα εργαλεία που παρέχονται στην εφαρμογή, θα εμπλουτίζουν την γκαλερί και θα παρέχουν ο ένας στον άλλον χρήσιμες συμβουλές βελτιώνοντας έτσι σταδιακά το επίπεδο των έργων τους.

Πίνακας Περιεχομένων

ΚΕΦΑΛΑΙΟ 1

1.1 Οργάνωση και δομή της εργασίας

Η παρούσα εργασία είναι δομημένη ως εξής:

Στο **κεφάλαιο 1,** γίνεται η εισαγωγή και η παρουσίαση της δομής της εργασίας.

Στα κεφάλαια 2,3 και 4 παρουσιάζεται το θεωρητικό υπόβαθρο της εφαρμογής, ειδικότερα:

Στο **κεφάλαιο 2**, παρουσιάζεται το λειτουργικό android, το android SDK, η Java και η εικονική μηχανή Dalvik (DVM) του android καθώς και τα βασικά στοιχεία μιας εφαρμογής android.

Στο **κεφάλαιο 3**, γίνεται λόγος για την αρχιτεκτονική πελάτη- εξυπηρετητή καθώς και το HTTP πρωτόκολλο.

Στο **κεφάλαιο 4**, αναφέρεται ο ρόλος της εφαρμογής android ως πελάτης και παρουσιάζεται η γλώσσα PHP ως η γλώσσα του εξυπηρετητή καθώς και η βάση δεδομένων MySQL.

Στο **κεφάλαιο 5**, παρουσιάζονται τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής.

Στο **κεφάλαιο 6**, περιγράφεται η υλοποίηση της εφαρμογής και γίνεται η παρουσίασή της.

Στο **κεφάλαιο 7**, γίνεται ο επίλογος και η αναφορά σε μελλοντικές επεκτάσεις.

Τέλος παρουσιάζεται η **βιβλιογραφία** και οι πηγές που χρησιμοποιήθηκαν για την συγγραφή της παρούσας διπλωματικής εργασίας.

ΚΕΦΑΛΑΙΟ 2

2.1 Τι είναι το android?

Το Android είναι ένα λειτουργικό σύστημα που αναπτύσσεται επί του παρόντος από τη Google, κυρίως για συσκευές με οθόνη αφής όπως smartphones και tablets. Παρόμοιες εκδόσεις του android, με διαφορετικό περιβάλλον χρήσης, έχουν αναπτυχθεί από τη Google και για τηλεοράσεις (Android TV), αυτοκίνητα (Android Auto) και ρολόγια χειρός (Android Wear). Παρόλο που έχει αναπτυχθεί για συσκευές με οθόνη αφής, έχει χρησιμοποιηθεί σε κονσόλες παιχνιδιών, ψηφιακές φωτογραφικές μηχανές, συνηθισμένους Η/Υ και σε άλλες ηλεκτρονικές συσκευές. Το Android είναι το πιο ευρέως διαδεδομένο λογισμικό στον κόσμο. Οι συσκευές με android ξεπερνάνε σε πωλήσεις τις αντίστοιχες συσκευές Windows, iOS και Mac OS X μαζί.

Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance, μιας κοινοπραξίας 48 τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής hardware, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές κινητής τηλεφωνίας. Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού. Το λογότυπο για το λειτουργικό σύστημα Android είναι ένα ρομπότ σε χρώμα πράσινου μήλου και σχεδιάστηκε από τη γραφίστρια Ιρίνα Μπλόκ.

2.2 Ανάπτυξη λογισμικού σε android

Οι εφαρμογές αναπτύσσονται συνήθως στη γλώσσα προγραμματισμού Java χρησιμοποιώντας το κιτ ανάπτυξης λογισμικού Android (SDK).

2.2.1 Android SDK

Το κιτ ανάπτυξης λογισμικού Android (SDK) περιλαμβάνει ένα ολοκληρωμένο σύνολο εργαλείων ανάπτυξης. Αυτά αποτελούνται από ένα πρόγραμμα εντοπισμού σφαλμάτων (debugger), βιβλιοθήκες, έναν εξομοιωτή συσκευής βασισμένο στο QEMU (Quick

Emulator), ένα δωρεάν και open-source λογισμικό που εκτελεί hardware virtualization, documentation, δείγματα κώδικα και tutorials. Προς το παρόν, οι πλατφόρμες ανάπτυξης που υποστηρίζονται, περιλαμβάνουν υπολογιστές με λειτουργικό σύστημα Linux (οποιαδήποτε σύγχρονη desktop έκδοση Linux), Mac OS X 10.5.8 ή νεότερη έκδοση και Windows XP ή νεότερη έκδοση. Από το Μάρτιο του 2015, το SDK δεν είναι διαθέσιμο στο ίδιο το λειτουργικό σύστημα Android, αλλά η ανάπτυξη λογισμικού είναι δυνατή με τη χρήση εξειδικευμένων εφαρμογών Android.

Μέχρι και τα τέλη του έτους 2014, το επίσημα υποστηριζόμενο ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) ήταν το Eclipse με την χρήση του Android Development Tools (ADT) Plugin, ωστόσο η ανάπτυξη λογισμικού android υποστηριζόταν πλήρως από το IntelliJ IDEA IDE (όλες τις εκδόσεις), καθώς και από το NetBeans IDE με τη χρήση plugin. Από το 2015, το Android Studio είναι το επίσημο ολοκληρωμένο περιβάλλον ανάπτυξης (IDE), ωστόσο, οι προγραμματιστές είναι ελεύθεροι να χρησιμοποιούν και άλλα IDEs. Επιπλέον, οι προγραμματιστές μπορούν να χρησιμοποιήσουν οποιοδήποτε πρόγραμμα επεξεργασίας κειμένου για να επεξεργαστούν Java και XML αρχεία, εν συνέχεια, να χρησιμοποιούν τα εργαλεία της γραμμής εντολών (τα Java Development Kit και Apache Ant είναι απαραίτητα για τη χρήση εργαλείων γραμμής εντολών) για τη δημιουργία, την κατασκευή (build) και τον εντοπισμό σφαλμάτων εφαρμογών Android, καθώς και τον έλεγχο συνδεδεμένων συσκευών Android.

Βελτιώσεις στο SDK του Android προχωράνε παράλληλα με τη συνολική ανάπτυξη και βελτίωση της πλατφόρμας Android. Το SDK υποστηρίζει επίσης παλαιότερες εκδόσεις της πλατφόρμας Android σε περίπτωση που οι προγραμματιστές θέλουν να αναπτύξουν τις εφαρμογές τους ώστε να είναι συμβατές και με παλαιότερες συσκευές. Οι προγραμματιστές μπορούν, αφού έχουν κατεβάσει την τελευταία έκδοση και την πλατφόρμα, να κατεβάσουν επίσης πλατφόρμες και εργαλεία παλαιότερων εκδόσεων για έλεγχο συμβατότητας.

Η εφαρμογή στο λειτουργικό σύστημα Android τελειώνει σε μορφή .apk και τοποθετείται στο φάκελο /data/app (ο οποίος είναι προσβάσιμος μόνο από root χρήστη για λόγους ασφαλείας). Ένα .apk αρχείο είναι ένα πακέτο που περιλαμβάνει τα .dex αρχεία(μεταγλωττισμένα byte code αρχεία που είναι εκτελέσιμα Dalvik αρχεία), αρχεία πόρων της εφαρμογής (resource files) και άλλα.

Το Android Debug Bridge (ADB) είναι ένα σύνολο εργαλείων που περιλαμβάνονται στο πακέτο του Android SDK. Αποτελείται από προγράμματα client και server-side που επικοινωνούν το ένα με το άλλο. Είναι ένα ευέλικτο εργαλείο που επιτρέπει την επικοινωνία με έναν εξομοιωτή συσκευής ή συνδεδεμένη συσκευή Android. Στο ADB μπορεί να υπάρξει πρόσβαση μέσω της διεπαφής γραμμής εντολών, ή μέσω κάποιας διεπαφής χρήστη γραφικού περιβάλλοντος (graphical user interface).

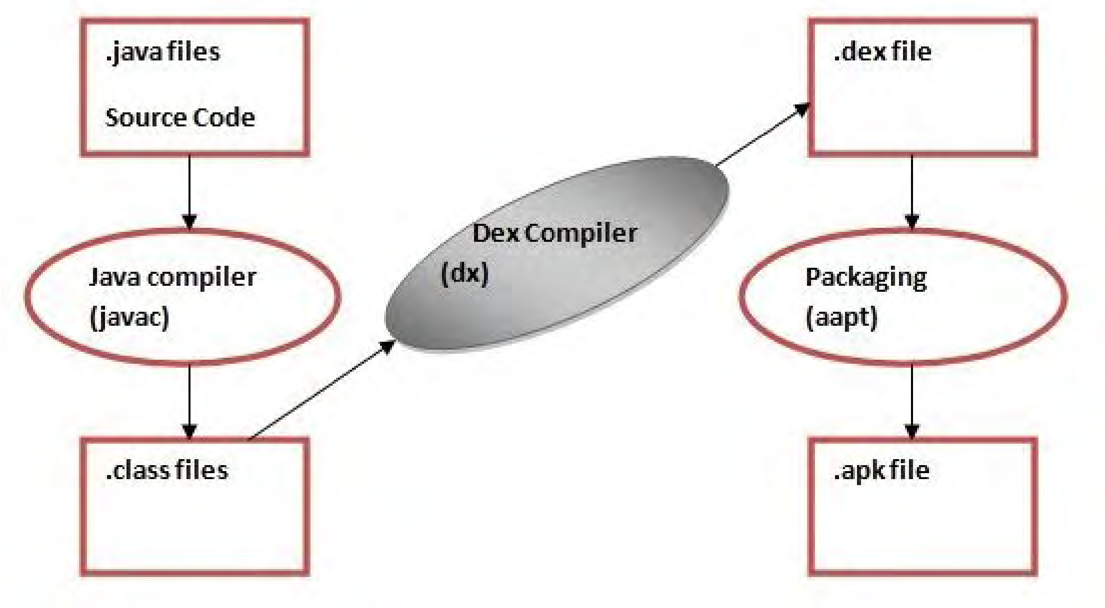
2.2.2 Η Java και η εικονική μηχανή Dalvik(DVM) του android

Το Android επαναχρησιμοποιεί μόνο τη σύνταξη και τη σημασιολογία της γλώσσας Java, και δεν παρέχει τις πλήρεις βιβλιοθήκες κλάσεων και API που περιλαμβάνονται στο Java SE ή ME. Η κλασική java χρησιμοποιεί JVM (Εικονική μηχανή java), ενώ το Android χρησιμοποιεί DVM για να τρέξει τις εφαρμογές του. Το Dalvik Virtual Machine (DVM) είναι μία εικονική μηχανή android που έχει βελτιστοποιηθεί για συσκευές κινητής τηλεφωνίας.

Η αρχιτεκτονική της JVM είναι βασισμένη στη στοίβα ενώ της DVM βασίζεται στους καταχωρητές. Η JVM χρησιμοποιεί java byte code και τρέχει τα .class αρχεία ενώ η DVM χρησιμοποιεί το δικό της byte code της και τρέχει τα αρχεία .Dex. Η DVM έχει σχεδιαστεί ώστε μια συσκευή να μπορεί να χρησιμοποιεί αποτελεσματικά πολλά στιγμιότυπα (instances) της εικονικής μηχανής και έχει βελτιστοποιηθεί ώστε να χρειάζεται ελάχιστη μνήμη, να έχει καλή απόδοση και να επιμηκύνει τη διάρκεια ζωής της μπαταρίας.

2.2.3 Τα βασικά στοιχεία μιας εφαρμογής

Οι εφαρμογές Android είναι γραμμένες σε γλώσσα προγραμματισμού Java. Τα εργαλεία του Android SDK μεταγλωττίζουν τον κώδικα και όποια άλλα δεδομένα και πόρους (resources) χρησιμοποιεί η εφαρμογή παράγοντας ένα τελικό αρχείο με κατάληξη .apk (το apk προέρχεται από το *Android package*).



(Source: http://www.javatpoint.com/dalvik-virtual-machine)

Το εργαλείο **javac (** μεταγλωττιστής της Java) μεταγλωττίζει τα αρχεία πηγαίου κώδικα της java (αρχεία .java) σε αρχεία .class.

Το εργαλείο **dx** (Dex compiler) είναι ένα εργαλείο κατασκευασμένο αποκλειστικά για την εκάστοτε πλατφόρμα (platform-specific tool) το οποίο παίρνει όλα τα .class αρχεία της εφαρμογής που έχουν μεταγλωττιστεί και παράγει ένα μοναδικό .dex αρχείο (εκτελέσιμο αρχείο Dalvik) το οποίο μπορεί να τρέξει στην DVM.

Το εργαλείο **Android Assets Packaging** χειρίζεται τη διαδικασία του packaging της εφαρμογής. **Android application package** (**APK**) είναι το package file format που χρησιμοποιείται από το λειτουργικό σύστημα Android για τη διανομή και εγκατάσταση τον εφαρμογών.

Ένα αρχείο APK περιλαμβάνει όλα τα επιμέρους τμήματα μιας εφαρμογής Android, όπως είναι ο κώδικας (.dex αρχεία), οι πόροι (app resources), τα assets, τα πιστοποιητικά και το αρχείο AndroidManifest.xml, και είναι το αρχείο που οι συσκευές Android χρησιμοποιούν για να εγκαταστήσουν την εφαρμογή. Όπως συμβαίνει και με πολλές μορφές αρχείων, τα αρχεία APK μπορούν να έχουν οποιοδήποτε όνομα, υπό την προϋπόθεση ότι το όνομα αρχείου τελειώνει σε ".apk".

Τα αρχεία APK εγκαθίστανται σε συσκευές Android ακριβώς όπως γίνεται και η εγκατάσταση ενός λογισμικού στον υπολογιστή. Όταν ένας χρήστης κατεβάζει και εγκαθιστά μια εφαρμογή Android είτε από επίσημη πηγή (όπως το Google Play), ή από κάποια άλλη ανεπίσημη πηγή, στην ουσία εγκαθιστά ένα αρχείο APK στη συσκευή του.

Ένας χρήστης ή προγραμματιστής μπορεί επίσης να εγκαταστήσει ένα αρχείο APK απευθείας σε μια συσκευή (όχι μέσω λήψης από το δίκτυο) από έναν επιτραπέζιο υπολογιστή, χρησιμοποιώντας ένα πρόγραμμα επικοινωνίας, όπως το ADB, ή μέσα από μια εφαρμογή διαχείρισης αρχείων. Από προεπιλογή, η δυνατότητα εγκατάστασης από ανεπίσημες πηγές, μέσω επιτραπέζιου υπολογιστή και μέσω εφαρμογής διαχείρισης αρχείων, είναι απενεργοποιημένη για λόγους ασφαλείας. Ο χρήστης μπορεί να την ενεργοποιήσει από το menu ρυθμίσεων αλλάζοντας το "Unknown sources".

Συστατικά μέρη της εφαρμογής (App components)

Οι εφαρμογές Android αποτελούνται από ένα συνδυασμό διακριτών συστατικών (components) τα οποία μπορούν να κληθούν ξεχωριστά ή και ανεξάρτητα το ένα από το άλλο. Υπάρχουν τέσσερα διαφορετικά είδη συστατικών μιας εφαρμογής (types of app components). Αυτά είναι οι Δραστηριότητες (Activities), οι Υπηρεσίες (Services), οι Πάροχοι Περιεχομένων (Content Providers) και οι Δέκτες Μετάδοσης (Broadcast Receivers). Κάθε τύπος εξυπηρετεί ένα συγκεκριμένο σκοπό και έχει ένα ξεχωριστό κύκλο ζωής που καθορίζει το πώς το component αυτό δημιουργείται και καταστρέφεται. Κάθε ένα υπάρχει ως δική του οντότητα και παίζει ένα συγκεκριμένο ρόλο ως ένα μοναδικό δομικό στοιχείο που παίρνει μέρος στο να καθορισθεί η συνολική συμπεριφορά της εφαρμογής.

Intent

Μια μοναδική ιδιότητα του σχεδιασμού του συστήματος Android, είναι ότι οποιαδήποτε εφαρμογή, μπορεί να εκκινήσει- καλέσει συστατικό άλλης εφαρμογής. Από ένα component υπάρχει η δυνατότητα κλήσης άλλου component χρησιμοποιώντας μια *πρόθεση* (*intent*). Ένα Intent είναι ένα αντικείμενο μηνύματος που μπορεί να χρησιμοποιηθεί για να επικαλεσθεί μια ενέργεια από ένα άλλο component. Οι Προθέσεις (Intents) χρησιμοποιούνται για την επικοινωνία μεταξύ των components με διάφορους τρόπους, ωστόσο έχουμε τρεις θεμελιώδεις περιπτώσεις χρήσης των Intents, για να ξεκινήσουμε μια Δραστηριότητα (Activity), για να ξεκινήσουμε μια Υπηρεσία (Service) και για να ξεκινήσουμε έναν Δέκτη Εκπομπής (Broadcast Receiver). Ο άλλος τύπος συστατικού, Πάροχος Περιεχομένων (Content Provider), δεν ενεργοποιείται από τις προθέσεις.

Activity

Μία δραστηριότητα (activity) παρέχει μία οθόνη με την οποία ο χρήστης μπορεί να αλληλεπιδρά ώστε να κάνει κάποιες ενέργειες, π.χ., να καλέσει κάποιον αριθμό, να βγάλει μια φωτογραφία, να στείλει ένα mail, να δει κάποιον χάρτη και άλλα. Σε κάθε activity δίνεται ένα παράθυρο στο οποίο το activity μπορεί να σχεδιάσει το user interface του (διεπαφή χρήστη). Το παράθυρο αυτό συνήθως καλύπτει όλη την οθόνη αλλά μπορεί επίσης να είναι και μικρότερο από την οθόνη ή και να βρίσκεται πάνω από άλλα παράθυρα.

Μια εφαρμογή συνήθως αποτελείται από πολλαπλές δραστηριότητες που συνδέονται «χαλαρά» μεταξύ τους. Τυπικά, μία δραστηριότητα σε μια εφαρμογή ορίζεται ως η «κύρια» δραστηριότητα, η οποία παρουσιάζεται στο χρήστη κατά την έναρξη της εφαρμογής, για πρώτη φορά. Κάθε δραστηριότητα μπορεί στη συνέχεια να αρχίσει μια άλλη δραστηριότητα, προκειμένου να εκτελέσει διάφορες ενέργειες. Κάθε φορά που ξεκινά μια νέα δραστηριότητα, η προηγούμενη δραστηριότητα διακόπτεται, αλλά το σύστημα διατηρεί τη δραστηριότητα σε μια στοίβα (την "πίσω στοίβα"). Όταν ξεκινά μια νέα δραστηριότητα, ωθείται στην κορυφή της "πίσω στοίβας" και παίρνει την εστίαση του χρήστη πάνω της (takes the user focus). Η "πίσω στοίβα" υπακούει στον βασικό μηχανισμό στοίβας "last in, first out", έτσι, όταν ο χρήστης θέλει να επιστρέψει στην προηγούμενη δραστηριότητα και πατάει το "Πίσω" κούμπι, το τελευταίο activity που είχε το user focus, εξωθείται από τη στοίβα(και επί της ουσίας καταστρέφεται) και επανέρχεται η αμέσως προηγούμενη δραστηριότητα(παίρνει αυτήν το user focus τώρα).

Όταν ένα activity σταματάει διότι ξεκινά ένα νέο activity, το activity που διακόπηκε ειδοποιείται για την διακοπή αυτή (αλλαγή κατάστασης της δραστηριότητας) μέσω των callback methods του κύκλου ζωής της δραστηριότητας. Υπάρχουν αρκετά callback methods που επιδέχεται μία δραστηριότητα με βάση τις αλλάγες στην κατάστασής της(δηλαδή όταν το σύστημα τη δημιουργεί, όταν την σταματάει, όταν την επαναφέρει σε λειτουργία ή όταν την καταστρέφει) και κάθε callback δίνει την ευκαιρία στο χρήστη να εκτελέσει τις κατάλληλες ενέργειες. Για παράδειγμα όταν σταματάει μια δραστηριότητα, θα πρέπει να ελευθερώνονται τυχόν μεγάλα αντικείμενα που έχουν δεσμευτεί από τη δραστηριότητα, όπως αντικείμενα σύνδεσης στο δίκτυο ή αντικείμενα σύνδεσης με βάση δεδομένων. Όταν μια δραστηριότητα επανέρχεται στη λειτουργία, πρέπει να αποκτήσει ξανά τους αναγκαίους πόρους για τη λειτουργία της και να επαναφέρει τυχόν ενέργειες που είχαν διακοπεί. Αυτές οι αλλαγές κατάστασης είναι όλες μέρος του κύκλου ζωής της δραστηριότητας.

Service

Μια υπηρεσία χρησιμοποιείται για να εκτελεί μακροχρόνιες συνήθως εργασίες στο παρασκήνιο χωρίς να παρέχει κάποιο user interface στο χρήστη. Εάν κάποιο component της εφαρμογής εκκινήσει μια *υπηρεσία* (service), αυτή θα συνεχίσει να εκτελείται στο παρασκήνιο ακόμα και αν ο χρήστης μεταβεί σε άλλη εφαρμογή. Επιπλέον, ένα component μπορεί να συνδεθεί σε μια υπηρεσία (bind to a service) ώστε να αλληλεπιδρά μαζί της. Για παράδειγμα μια υπηρεσία μπορεί να χειρίζεται τις συναλλαγές του δικτύου, να παίζει μουσική, πραγματοποιεί λειτουργεία I/O σε αρχεία, να αλληλεπιδρά με έναν πάροχο περιεχομένου(content provider), και άλλα πολλά, όλα από το παρασκήνιο.

Μία υπηρεσία (service) μπορεί να πάρει δύο μορφές, “started” ή “bound”. Μία υπηρεσία είναι "started" (“started” service) όταν ένα component της εφαρμογής (π.χ. ένα activity) εκκινεί την υπηρεσία καλώντας την μέθοδο startService(). Μόλις ξεκινήσει η υπηρεσία, μπορεί να τρέχει στο παρασκήνιο επ’ αόριστον, ακόμη και αν καταστραφεί το component που την άρχισε. Συνήθως, ένα “started” service (υπηρεσία) εκτελεί μία μόνο λειτουργία και δεν επιστρέψει κάποιο αποτέλεσμα στον καλούντα. Για παράδειγμα, η υπηρεσία (service) θα μπορούσε να κατεβάσει ή να ανεβάσει ένα αρχείο μέσω του δικτύου. Όταν η λειτουργία αυτή ολοκληρώνεται, η υπηρεσία θα πρέπει να σταματάει από μόνη της.

Μία υπηρεσία είναι “bound” (“bound” service) όταν ένα component της εφαρμογής γίνεται “bind” (δεσμεύεται) σε αυτήν καλώντας bindService(). Ένα “bound” service προσφέρει ένα client-server interface το οποίο επιτρέπει τα components να αλληλεπιδρούν με την υπηρεσία αυτή, να στέλνουν requests και να λαμβάνουν αποτελέσματα Μία δεσμευμένη υπηρεσία (“bound” service) λειτουργεί για όσο υπάρχουν components που έχουν γίνει “bind” (έχουν δεσμεύθει) σε αυτήν. Σε μια υπηρεσία μπορούν να γίνουν "bind" πολλαπλά components. Η υπηρεσία παύει να λειτουργεί (καταστρέφεται) μόνο όταν αποδεσμευθούν από αυτήν όλα τα components που έχουν δεσμευθεί. Τέλος, μία υπηρεσία μπορεί να είναι "started", ώστε να τρέχει επ' αόριστον και να επιτρέπει ταυτόχρονα και binding.

Content provider

Οι πάροχοι περιεχομένου(content providers) διαχειρίζονται την πρόσβαση σε ένα δομημένο σύνολο δεδομένων. Ενθυλακώνουν τα δεδομένα, και παρέχουν μηχανισμούς για τον προσδιορισμό της ασφάλειας των δεδομένων. O content provider είναι η τυπική διεπαφή που συνδέει τα δεδομένα σε μια διαδικασία με τον κώδικα που τρέχει σε μια άλλη διαδικασία.

Όταν ο χρήστης θέλει να έχει πρόσβαση σε δεδομένα ενός content provider, χρησιμοποιεί το αντικείμενο ContentResolver του Context της εφαρμογής ώστε να επικοινωνεί με τον content provider ως client . Το Context είναι μια αφηρημένη κλάση η οποία υλοποιείται αυτόματα από το android system και επιτρέπει πρόσβαση σε κλάσεις και resources της συγκεκριμένης εφαρμογής, όπως και up-calls για λειτουργίες στα πλαίσια της εφαρμογής(π.χ. εκκίνηση κάποιου activity, λήψη ή αποστολή κάποιου intent κ.α.). Το αντικείμενο ContentResolver επικοινωνεί με το αντικείμενο πάροχο(provider object), το οποίο είναι ένα instance (στιγμιότυπο) μιας κλάσης που υλοποιεί το ContentProvider. Το αντικείμενο πάροχος λαμβάνει αιτήματα δεδομένων από τους πελάτες(clients), εκτελεί την απαιτούμενη ενέργεια, και επιστρέφει τα αποτελέσματα.

Ο χρήστης δεν χρειάζεται να υλοποιήσει το δικό του πάροχο εάν δε σκοπεύει να μοιραστεί δεδομένα με άλλες εφαρμογές, ωστόσο χρειάζεται να το κάνει εάν θέλει να παρέχει ιδιαιτερότητα στην αναζήτηση στην εφαρμογή του ή να αντιγράψει σύνθετα δεδομένα ή αρχεία από την εφαρμογή του σε άλλες.

Το ίδιο το Android περιλαμβάνει παρόχους περιεχομένου που διαχειρίζονται δεδομένα, όπως ήχο, βίντεο, εικόνες, και τα προσωπικά στοιχεία επικοινωνίας Με ορισμένους περιορισμούς, οι εν λόγω πάροχοι είναι προσβάσιμοι σε οποιαδήποτε εφαρμογή Android.

Broadcast receiver

Ένας broadcast receiver (δέκτης μετάδοσης) είναι ένα component το οποίο ανταποκρίνεται σε ευρείες μεταδόσεις ανακοινώσεων σε όλο το σύστημα (system-wide broadcast announcements). Πολλές από τις ευρείες μεταδόσεις ανακοινώσεων προέρχονται από το ίδιο το συστημα, π.χ. μια ανακοίνωση ότι η οθόνη του τηλεφώνου έκλεισε, η μπαταρία έπεσε σε χαμηλά επίπεδα, ή ότι μόλις απαθανατίστηκε μια φωτογραφία. Οι εφαρμογές μπορούν και αυτές να εκπέμψουν ευρείες ανακοινώσεις για να μεταδώσουν κάποιο γεγονός, για παράδειγμα μια εφαρμογή να ειδοποιεί τις υπόλοιπες ότι κάποια δεδομένα μόλις "κατέβηκαν" από το δίκτυο στη συσκευή και είναι έτοιμα προς χρήση. Οι broadcast receivers δεν παρέχουν κάποιο user interface, μπορούν ωστόσο να δημιουργήσουν μια ειδοποίηση στο status bar της συσκευής ώστε να ενημερωθεί ο χρήστης για το γεγονός που συνέβη. Γενικότερα ένας broadcast receiver έχει καθοδηγητικό ρόλο όσον αφορά τη ροή εκτέλεσης και αποτελεί απλά μια "πύλη" προς άλλα components. Για παράδειγμα ο broadcast receiver θα μπορούσε να προκαλέσει την εκκίνηση μιας υπηρεσίας (service) με βάση το γεγονός που μόλις μεταδόθηκε. Ένας broadcast receiver υλοποιείται ως subclass της BroadcastReceiver και κάθε εκπομπή παραδίδεται ως αντικείμενο Intent.

Πόροι της εφαρμογής(App resources)

Οι πόροι είναι τα πρόσθετα αρχεία και στατικό περιεχόμενο που μπορούμε να χρησιμοποιήσουμε στον κώδικά μας, όπως εικόνες bitmap, τα layouts που χρησιμοποιούμε, strings στο user interface της εφαρμογής, οδηγίες για animations, σταθερές τιμές και πολλά άλλα.

Το android παρέχει τη δυνατότητα για πολλά αντικείμενα- πόρους (resources) που χρησιμοποιεί η εφαρμογή (όπως layouts, buttons κ.α.) να δηλώνονται και προγραμματιστικά αλλά και σε μορφή xml φάκελο res. Η εξωτερίκευση των πόρων (όπως εικόνες, κείμενα, layouts κ.α.) από τον κώδικα της εφαρμογής και η τοποθέτηση τους στο φάκελο res είναι μια πρακτική που συνιστάται και από το ίδιο το android. Με αυτόν τον τρόπο ο χρήστης αποκτά το πλεονέκτημα του διαχωρισμού της λειτουργίας της εφαρμογής από την εμφάνισή της. Ένα άλλο μεγάλο πλεονέκτημα του διαχωρισμού του κώδικα από τα resources είναι ότι με αυτή τη πρακτική καθίσταται δυνατή η παροχή πολλών διαφορετικών εναλλακτικών πόρων, για τον ίδιο σκοπό, όπου σε κάθε περίπτωση χρησιμοποιείται ο πόρος που είναι πιο συμβατός με την εκάστοτε συσκευή στην οποία τρέχει η εφαρμογή (διαφορετικός πόρος για κάθε device configuration όπως γλώσσα, μέγεθος οθόνης κ.α.).

Οι εναλλακτικοί πόροι μπορούν επίσης να χρησιμοποιηθούν και για διαχείριση διαφόρων runtime γεγονότων όπως είναι ο προσανατολισμός της οθόνης της συσκευής (π.χ. από portrait σε landscape) εάν ο χρήστης παρέχει βέβαια διαφορετικό resource για κάθε προσανατολισμό οθόνης. Τα εναλλακτικά αυτά resources τοποθετούνται σε κατάλληλους υποφακέλους του φακέλου res με βάση το είδος του resource και τα χαρακτηριστικά της συσκευής. Για κάθε είδος resource της εφαρμογής μπορεί να ορισθεί ένα προεπιλεγμένο (default) και πολλά εναλλακτικά resources. Το default resource χρησιμοποιείται όταν δεν υπάρχουν εναλλακτικά resources που ταιριάζουν με το εκάστοτε device configuration (π.χ. συσκευή με μέγεθος οθόνης για το οποίο δεν προσφέρεται κάποιο resource από τον προγραμματιστή, ή γλώσσα συσκευής για την οποία ο προγραμματιστής δεν παρέχει κείμενα strings του user interface στη γλώσσα αυτή).

Το android επιλέγει αυτόματα τα κατάλληλα resources ταιριάζοντας τις προδιαγραφές (device configuration) της εκάστοτε συσκευής που χρησιμοποιεί την εφαρμογή με τον κατάλληλο υποφάκελο του φακέλου res της εφαρμογής. Εάν δεν υπάρχει κάποιο resource, το android θα χρησιμοποιήσει το αντίστοιχο default. Όταν η εφαρμογή γίνεται compile, το εργαλείο aapt δημιουργεί την κλάση R, η οποία περιλαμβάνει αναγνωριστικά (IDs) για όλα τα resources στο φάκελο res της εφαρμογής. Για κάθε είδος resource, υπάρχει ένα R subclass (για παράδειγμα, το R.drawable για όλα τα drawable resources), και για κάθε resource αυτού του είδους υπάρχει ένα static integer(π.χ. για την εικόνα “icon” το R.drawable.icon). Αυτός ο ακέραιος αριθμός είναι το αναγνωριστικό το οποίο χρησιμοποιεί ο χρήστης για να αναφερθεί στο αντίστοιχο resource.

Σε ένα resource μπορεί να υπάρξει πρόσβαση είτε από τον κώδικα της εφαρμογής, είτε από τα αρχεία xml της εφαρμογής που αποθηκεύεται στον υποφάκελο. Αναφορικά κάποια resources είναι: Animation Resources, Color State List Resource, Drawable Resources, Layout Resource, Menu Resource, String Resources, Style Resource, ορισμός τιμών όπως boolean, integer, strings, dimensions και άλλα.

Αρχείο AndroidManifest.xml της εφαρμογής

Για να μπορεί να ξεκινήσει μια εφαρμογή είναι απολύτως απαραίτητο να έχει το αρχείο AndroidManifest.xml (xml αρχείο) αποθηκευμένο στο root directory της (.../app/src/main/AndroidManifest.xml) και ακριβώς με αυτό το όνομα. Το manifest (AndroidManifest.xml) παρέχει ουσιώδης πληροφορίες στο σύστημα android, πληροφορίες που αυτό πρέπει οπωσδήποτε να έχει πριν να εκκινήσει την εφαρμογή. Αναφορικά παραθέτονται κάποιες υπηρεσίες για της οποίες είναι υπεύθυνο το manifest:

- Ονομάζει και αναγνωρίζει το java πακέτο (java package) της εφαρμογής μέσα στο οποίο βρίσκονται οι java κλάσεις. Το όνομα του πακέτου χρησιμεύει ως μοναδικό αναγνωριστικό για την εφαρμογή.
- Περιγράφει όλα τα συστατικά μέρη(components) από τα οποία αποτελείται μια εφαρμογή δηλαδή τα activities, services, broadcast receivers και content providers της εφαρμογής.
- Ονοματίζει τις κλάσεις εκείνες που υλοποιούν κάθε ένα από τα παραπάνω components και δημοσιεύει τις δυνατότητές τους(για παράδειγμα, ποιές από τις προθέσεις(Intent) μπορούν να χειριστούν).Οι δηλώσεις αυτές στο AndroidManifest κάνουν γνωστό στο σύστημα android ποιά είναι τα components της εφαρμογής και υπό ποιές συνθήκες μπορούν αυτά να εκκινήσουν.
- Προσδιορίζει τα processes που θα χρειαστούν να υποστηρίξουν τα components της εφαρμογής.
- Προσδιορίζει τα δικαιώματα τα οποία πρέπει να έχει η τρέχον εφαρμογή για να μπορεί να αλληλεπιδρά με άλλες εφαρμογές.
- Ορίζει τα δικαιώματα που άλλες εφαρμογές πρέπει να έχουν για να χρησιμοποιήσουν την εφαρμογή μας.
- Δηλώνει το ελάχιστο επίπεδο του Android API που απαιτεί η εφαρμογή.
- Παραθέτει τις βιβλιοθήκες που χρειάζεται η εφαρμογή για να τρέξει.
- Και τέλος καλεί και χρησιμοποιεί τις Instrumentation κλάσεις που παρέχουν πληροφορίες και καταγράφουν την εκτέλεση και λειτουργία της εφαρμογής καθώς αυτή τρέχει και την αλληλεπίδρασή της με το android system. Οι δηλώσεις αυτές είναι παρούσες στο manifest μόνο κατά την ανάπτυξη και το testing της εφαρμογής και αφαιρούνται από το manifest όταν η εφαρμογή πρόκειται να δημοσιευθεί.

2.2.4 Αλληλεπίδραση του android OS με την εφαρμογή

Το λειτουργικό σύστημα Android είναι ένα σύστημα Linux πολλαπλών χρηστών στο οποίο κάθε εφαρμογή (app) αποτελεί ένα διαφορετικό χρήστη. Το σύστημα αποδίδει σε κάθε εφαρμογή ένα μοναδικό αναγνωριστικό χρήστη του Linux (Linux User ID). Το αναγνωριστικό χρησιμοποιείται μόνο από το σύστημα και είναι άγνωστο προς την εφαρμογή. Σε κάθε εφαρμογή ανατίθεται από το σύστημα ξεχωριστή εικονική μηχανή (VM), έτσι, ο κώδικας κάθε εφαρμογής τρέχει σε απομόνωση από τις άλλες. Κάθε εφαρμογή τρέχει στο δικό της Linux proccess (διεργασιά). Το Android ξεκινά τη διεργασία όταν πρέπει να εκτελεστεί κάποιο ή κάποια από τα components (συστατικά) της εφαρμογής, στη συνέχεια, τερματίζεται η διεργασία, όταν δεν είναι πλέον αναγκαία ή όταν το σύστημα πρέπει να ανακτήσει τη μνήμη για άλλες εφαρμογές. Με τον τρόπο αυτό, το σύστημα Android εφαρμόζει την αρχή των ελάχιστων προνομίων. Δηλαδή, κάθε εφαρμογή, έχει πρόσβαση μόνο στα στοιχεία που απαιτούνται για να κάνει τη δουλειά της και τίποτα περισσότερο. Αυτό δημιουργεί ένα πολύ ασφαλές περιβάλλον στο οποίο μια εφαρμογή δεν μπορεί να έχει πρόσβαση σε μέρη του συστήματος για τα οποία δεν της έχει δοθεί άδεια(permission). Ωστόσο, υπάρχουν τρόποι για μια εφαρμογή να μοιράζεται δεδομένα με άλλες εφαρμογές και να έχει πρόσβαση στις υπηρεσίες του συστήματος. Είναι δυνατόν να ορισθεί για δύο εφαρμογές να μοιράζονται το ίδιο αναγνωριστικό χρήστη του Linux, οπότε είναι σε θέση να έχουν πρόσβαση η μία στα αρχεία της άλλης. Για τη διατήρηση των πόρων του συστήματος, εφαρμογές με το ίδιο αναγνωριστικό χρήστη μπορεί επίσης να κανονισθεί να τρέξουν στην ίδια διαδικασία Linux και να μοιράζονται το ίδιο VM (οι εφαρμογές πρέπει να έχουν και την ίδια υπογραφή (signed with the same certificate)). Μια εφαρμογή μπορεί να ζητήσει άδεια για πρόσβαση στα δεδομένα της συσκευής, όπως επαφές του χρήστη, μηνύματα SMS, κάρτα μνήμης SD, κάμερα, Bluetooth, και πολλά άλλα. Ο χρήστης πρέπει να χορηγήσει ρητά αυτά τα δικαιώματα.

ΚΕΦΑΛΑΙΟ 3

3.1 Αρχιτεκτονική Client-Server

Στην επιστήμη των υπολογιστών το μοντέλο αρχιτεκτονικής λογισμικού πελάτη-διακομιστή αποτελεί μία συνήθη μέθοδο ανάπτυξης λογισμικού στην οποία ο πελάτης (ένα τμήμα λογισμικού) ζητά κάτι (π.χ. έναν πόρο, τα αποτελέσματα ενός υπολογισμού κ.ο.κ.) και ένα άλλο τμήμα λογισμικού, ο διακομιστής (ή εξυπηρετητής), του το επιστρέφει. Κάθε διακομιστής μπορεί να εξυπηρετεί πολλαπλούς πελάτες. Ο διακομιστής και ο πελάτης μπορούν να εκτελούνται σε διαφορετικές διεργασίες, οι οποίες με τη σειρά τους μπορούν να εκτελούνται σε διαφορετικούς υπολογιστές, οπότε απαιτείται ένα δίκτυο υπολογιστών για τη διαδιεργασιακή επικοινωνία μεταξύ τους. Σε αυτή την περίπτωση το μοντέλο πελάτη-διακομιστή αποτελεί μία από τις μεθόδους ανάπτυξης και λειτουργίας κατανεμημένων συστημάτων, όπου θεωρούμε τόσο τον πελάτη όσο και τον διακομιστή διαφορετικά τμήματα της ίδιας κατανεμημένης εφαρμογής (π.χ., με την έννοια αυτή, ο Παγκόσμιος Ιστός είναι μία μεγάλη κατανεμημένη εφαρμογή αποτελούμενη από τους πλοηγούς Web και το σύνολο των διακομιστών Web).

Γενικά, το client-server computing αναφέρεται σε μια βασική αλλαγή στο στυλ των υπολογιστών, την αλλαγή από τα συστήματα που βασίζονται στα μηχανήματα στα συστήματα που βασίζονται στον χρήστη. Ειδικότερα, ένα σύστημα client-server είναι ένα σύστημα στο οποίο το δίκτυο ενώνει διάφορους υπολογιστικούς πόρους, ώστε οι clients (ή αλλιώς front end) να μπορούν να ζητούν υπηρεσίες από έναν server (ή αλλιώς back end), ο οποίος προσφέρει πληροφορίες ή επιπρόσθετη υπολογιστική ισχύ. Με άλλα λόγια, στο client-server μοντέλο, ο client θέτει μια αίτηση και ο server επιστρέφει μια ανταπόκριση ή κάνει μια σειρά από ενέργειες. Ο server μπορεί να ενεργοποιείται άμεσα για την αίτηση αυτή ή να προσθέτει την αίτηση σε μια ουρά. Η άμεση ενεργοποίηση για την αίτηση μπορεί, για παράδειγμα, να σημαίνει ότι ο server υπολογίζει έναν αριθμό και τον επιστρέφει αμέσως στον client. Η τοποθέτηση της αίτησης σε μια ουρά μπορεί να σημαίνει ότι η αίτηση πρέπει να τεθεί σε αναμονή για να εξυπηρετηθεί. Ένα καλό παράδειγμα για αυτό είναι όταν εκτυπώνουμε ένα κείμενο σε ένα εκτυπωτή δικτύου. Ο server τοποθετεί την αίτηση σε μια ουρά μαζί με αιτήσεις εκτυπώσεων και από άλλους clients. Μετά επεξεργάζεται την αίτηση με βάση την σειρά προτεραιότητας, η οποία , σε αυτή την περίπτωση, καθορίζεται από τη σειρά με την οποία ο server παρέλαβε την απαίτηση.

3.2 Το πρωτόκολλο HTTP

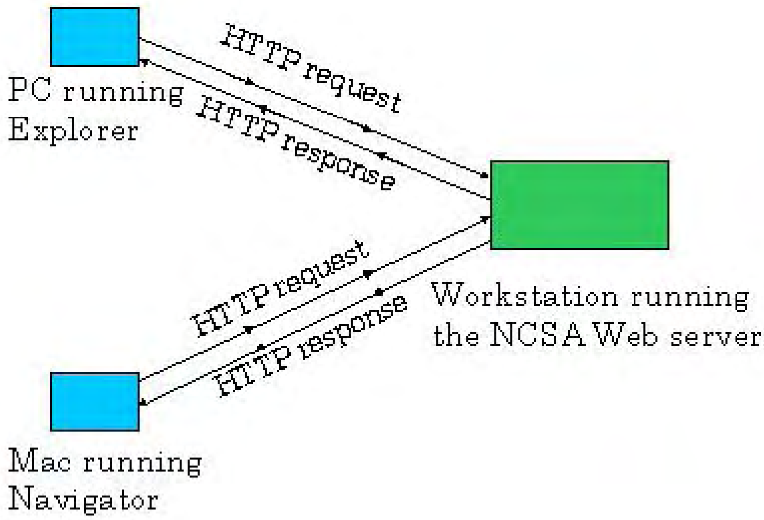
Το Πρωτόκολλο Μεταφοράς Υπερκειμένου HTTP (HyperText Transfer Protocol) είναι η καρδιά του Ιστού. Το HTTP ανήκει στο στρώμα εφαρμογών του Διαδικτύου και υλοποιείτε ως δύο προγράμματα: ένα πρόγραμμα πελάτη (client program) και ένα πρόγραμμα εξυπηρετητή (server program). Τα δύο αυτά προγράμματα εκτελούνται σε διαφορετικά μηχανήματα επικοινωνώντας μεταξύ τους ανταλλάσσοντας HTTP μηνύματα. Συγκεκριμένα το HTTP ορίζει τη δομή των μηνυμάτων αυτών καθώς και τον τρόπο ανταλλαγής τους ανάμεσα στον πελάτη και στον εξυπηρετητή.

3.2.1 Το HTTP στον παγκόσμιο ιστό

Πριν περιγράψουμε αναλυτικά το πρωτόκολλο HTTP πρέπει να αναφερθούμε σε κάποια βασική ορολογία του Ιστού. Μία Ιστοσελίδα (Web page) αποτελείται από αντικείμενα. Με τον όρο αντικείμενο (object) εννοούμε ένα απλό αρχείο, όπως ένα αρχείο HTML, ένα αρχείο εικόνας ή ένα αρχείο βίντεο, το οποίο μπορεί να προσπελαστεί μέσω ενός URL. Οι περισσότερες Ιστοσελίδες αποτελούνται από ένα βασικό αρχείο HTML και διάφορα σχετικά αντικείμενα. Αν υποθέσουμε ότι έχουμε μία Ιστοσελίδα που περιέχει ένα αρχείο HTML και 3 αρχεία εικόνων τότε λέμε ότι η Ιστοσελίδα έχει 4 αντικείμενα. Το βασικό αρχείο HTML αναφέρεται στα άλλα αντικείμενα της σελίδας μέσω των URL των αντικειμένων. Κάθε URL αποτελείται από δύο τμήματα: το όνομα του υπολογιστή – host στον οποίον είναι αποθηκευμένο το αρχείο και το όνομα του μονοπατιού (path) του αντικειμένου. Π.χ. το URL www.ntua.gr/index.htm έχει ως όνομα host το www.ntua.gr και ως όνομα μονοπατιού το index.htm.

Ο browser είναι ο αντιπρόσωπος του Ιστού: απεικονίζει στον χρήστη τη ζητούμενη Ιστοσελίδα και παρέχει πληθώρα χαρακτηριστικών πλοήγησης και παραμετροποίησης. Επίσης, στους browser υλοποιείτε και η πλευρά του πελάτη του πρωτοκόλλου HTTP. Ένας εξυπηρετητής Ιστού (Web server) αποθηκεύει τα αντικείμενα της Ιστοσελίδας, το καθένα από τα οποία έχει ως διεύθυνση ένα URL. Στους εξυπηρετητές Ιστού υλοποιείτε και η πλευρά του εξυπηρετητή του πρωτοκόλλου HTTP.

Το HTTP ορίζει τον τρόπο με τον οποίο οι πελάτες του Ιστού (π.χ. οι browsers) ζητούν (request) Ιστοσελίδες από τους εξυπηρετητές του Ιστού (π.χ. τους Web servers) και πως οι εξυπηρετητές μεταφέρουν τις Ιστοσελίδες στους πελάτες. Η βασική ιδέα της του πρωτοκόλλου αυτού φαίνεται στο παρακάτω σχήμα.



Όταν ο χρήστης ζητά μία Ιστοσελίδα, ο browser στέλνει ένα μήνυμα HTTP αίτησης (HTTP request), για τα διάφορα αντικείμενα της σελίδας, στον εξυπηρετητή. Ο εξυπηρετητής όταν λάβει το μήνυμα αυτό ανταποκρίνεται με μηνύματα HTTP απόκρισης (HTTP response) στα οποία περιέχονται τα αιτούμενα αντικείμενα. Μέχρι το 1997 όλοι οι browsers και όλοι οι εξυπηρετητές Ιστού υλοποιούσαν την έκδοση 1.0 του HTTP, που για συντομία γράφεται HTTP/1.0. Από το 1998 όμως άρχισαν να υποστηρίζουν και το HTTP/1.1, το οποίο είναι συμβατό με το HTTP/1.0. Δηλαδή, ένας browser που υποστηρίζει το HTTP/1.0 μπορεί να επικοινωνήσει με ένας εξυπηρετητή Ιστού που υποστηρίζει το HTTP/1.1 και, αντιστρόφως, ένας browser που υποστηρίζει το HTTP/1.1 μπορεί να επικοινωνήσει με έναν εξυπηρετητή Ιστού που υποστηρίζει το HTTP/1.0.

Τόσο το HTTP/1.0 όσο και το HTTP/1.1 χρησιμοποιούν το TCP ως πρωτόκολλο μεταφοράς. Αφού ο πελάτης εγκαταστήσει μία σύνδεση TCP με τον εξυπηρετητή αρχίζει την αποστολή μηνυμάτων – αιτήσεων προς αυτόν και τη λήψη μηνυμάτων – αποκρίσεων από αυτόν. Λόγω της χρήσης του TCP το HTTP δεν χρειάζεται να ασχοληθεί καθόλου με τη μεταφορά των δεδομένων. Το μόνο που πρέπει να κάνει είναι να στείλει τις αιτήσεις μέσω της TCP σύνδεσης και να περιμένει τις αποκρίσεις. Το TCP εγγυάται την αξιόπιστη μεταφορά των δεδομένων καθώς και τον έλεγχο της συμφόρησης.

Οι εξυπηρετητές του HTTP δεν κρατάνε καθόλου στοιχεία για την κατάσταση του πελάτη. Επομένως, αν ένας πελάτης στείλει μία αίτηση για ένα αρχείο δύο φορές, ο εξυπηρετητής θα του στείλει το αρχείο αυτό δύο φορές. Τα πρωτόκολλα που δεν κρατάνε καθόλου πληροφορία για την κατάσταση του πελάτη ονομάζονται stateless.

3.2.2 Μορφή των HTTP μηνυμάτων

Το HTTP ορίζει μόνο δύο τύπους μηνυμάτων: τις HTTP αιτήσεις (requests) και τις HTTP αποκρίσεις (responses). Ακολουθεί ένα παράδειγμα ενός μηνύματος HTTP αίτησης.

GET /lessons/index.htm HTTP/1.1

Connection: close

User-agent: Mozilla/4.0

Accept: text/html, image/gif, image/jpeg

Accept-language:gr

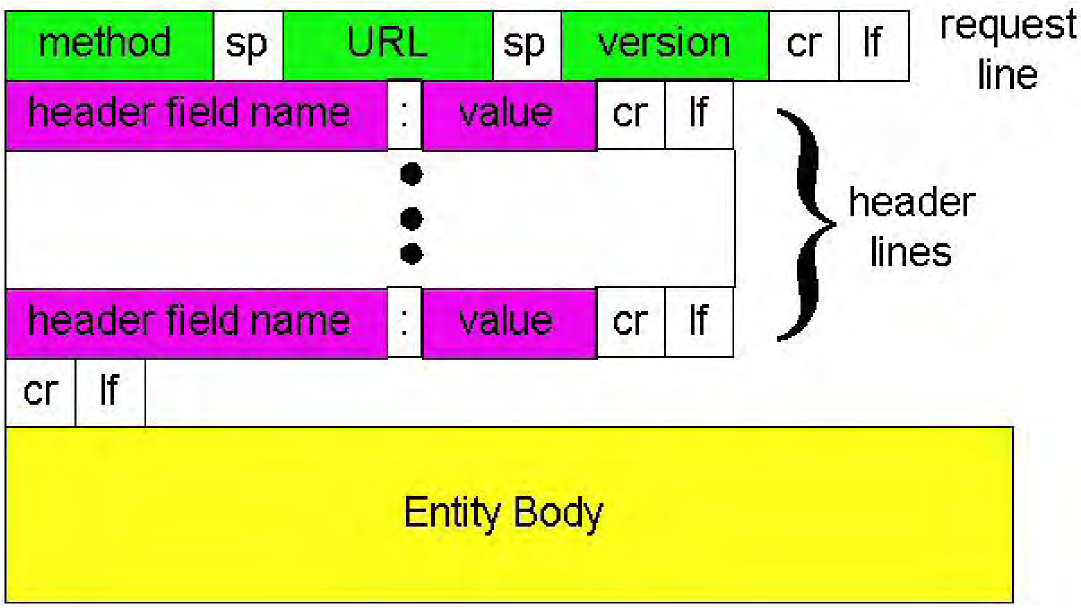
(extra carriage return, line feed)

Όπως φαίνεται και από το παραπάνω παράδειγμα οι HTTP αιτήσεις γράφονται με χαρακτήρες ASCII και μπορούν να διαβαστούν από τους ανθρώπους. Οι περισσότερες HTTP αιτήσεις αποτελούνται από 5 γραμμές κειμένου ακολουθούμενες από μία κενή γραμμή. Οι HTTP αιτήσεις περιέχουν τουλάχιστον μία γραμμή κειμένου, ενώ υπάρχουν και περιπτώσεις που αποτελούνται και από περισσότερες από 5 γραμμές κειμένου. Η πρώτη γραμμή κειμένου ονομάζεται γραμμή αίτησης (request line), ενώ οι επόμενες γραμμές επικεφαλίδας (header lines).

Η γραμμή αίτησης περιέχει τρία πεδία: το πεδίο μεθόδου, το πεδίο URL και το πεδίο HTTP έκδοσης. Το πεδίο μεθόδου μπορεί να έχει μία από τις ακόλουθες τιμές: GET, POST και HEAD. Η πιο συνηθισμένη μέθοδος στις HTTP αιτήσεις είναι η GET, με την οποία ζητείται από τον εξυπηρετητή η αποστολή του αρχείου που εμφανίζεται στο πεδίο URL. Στο URL δεν είναι απαραίτητο να υπάρχει και το όνομα του host, αφού ήδη έχει εγκατασταθεί μία σύνδεση με τον host αυτό (δηλαδή τον HTTP εξυπηρετητή). Τέλος, στο πεδίο έκδοσης HTTP περιγράφεται η έκδοση του HTTP που χρησιμοποιεί ο αιτών host. Στο παραπάνω παράδειγμα χρησιμοποιείται το HTTP/1.1.

Στη συνέχεια θα εξετάσουμε τις γραμμές επικεφαλίδας του παραπάνω παραδείγματος. Η γραμμή Connection:close λέει στον εξυπηρετητή να τερματιστεί η σύνδεση μετά την αποστολή του αρχείου, δηλαδή να μη γίνει χρήση μόνιμης σύνδεσης. Η επόμενη γραμμή που αρχίζει με User-agent: δηλώνει τον τύπο του browser του χρήστη. Στο παραπάνω παράδειγμα δηλώνεται ότι ο χρήστης χρησιμοποιεί τον browser Mozilla/4.0. Η γραμμή αυτή χρησιμοποιείται από τον εξυπηρετητή για την αποστολή διαφορετικών αντικειμένων ανάλογα με τον browser του χρήστη (τα οποία όμως έχουν το ίδιο URL). Η Accept: γραμμή δηλώνει τον τύπο των αντικειμένων που υποστηρίζει ο browser. Στο παραπάνω παράδειγμα δηλώνεται ότι ο browser υποστηρίζει αρχεία HTML και αρχεία εικόνων τύπου GIF και JPEG. Τέλος, η Accept-language: γραμμή δηλώνει την γλώσσα έκδοσης του αντικειμένου που επιθυμεί να λάβει ο browser. Αν ο εξυπηρετητής δεν έχει έκδοση του αντικειμένου για την γλώσσα αυτή τότε στέλνει την προκαθορισμένη έκδοση του αντικειμένου. Στο παραπάνω παράδειγμα αν ο εξυπηρετητής διαθέτει μία ελληνική έκδοση του αρχείου lessons/index.htm τότε πρέπει να τη στείλει, αλλιώς να στείλει την προκαθορισμένη έκδοση του αρχείου.

Η δομή του μηνύματος HTTP αίτησης φαίνεται στο παρακάτω σχήμα.



Από το παραπάνω σχήμα βλέπουμε ότι η γενική μορφή του μηνύματος ακολουθεί τη δομή του παραπάνω παραδείγματος. Το πεδίο Entity Body δεν χρησιμοποιείται για τη μέθοδο GET, χρησιμοποιείται όμως όταν γίνεται χρήση της μεθόδου POST. Ένα παράδειγμα χρήσης της μεθόδου POST είναι η αποστολή των δεδομένων που συμπλήρωσε ο χρήστης σε διάφορες φόρμες μίας Ιστοσελίδας. Στην περίπτωση αυτή τα δεδομένα αποστέλλονται στο πεδίο Entity Body και ανάλογα με τα δεδομένα αυτά ο χρήστης λαμβάνει μία κατάλληλη έκδοση της Ιστοσελίδας. Παρόμοια χρήση του πεδίου Entity Body γίνεται και στην περίπτωση χρήσης της μεθόδου HEAD. Η μέθοδος αυτή είναι χρήσιμη στον εντοπισμό σφαλμάτων (debugging) από τους developers των εξυπηρετητών Ιστού: όταν ο εξυπηρετητής λάβει αίτηση με χρήση της μεθόδου HEAD αποκρίνεται με ένα HTTP μήνυμα στο οποίο δεν περιλαμβάνεται το αρχείο. Παρακάτω φαίνεται ένα παράδειγμα μηνύματος HTTP απόκρισης.

HTTP/1.1 200 OK

Connection: close

Date: Thu, 06 Aug 1998 12:00:15 GMT

Server: Apache/1.3.0 (Unix)

Last-Modified: Mon, 22 Jun 1998 09:23:24 GMT

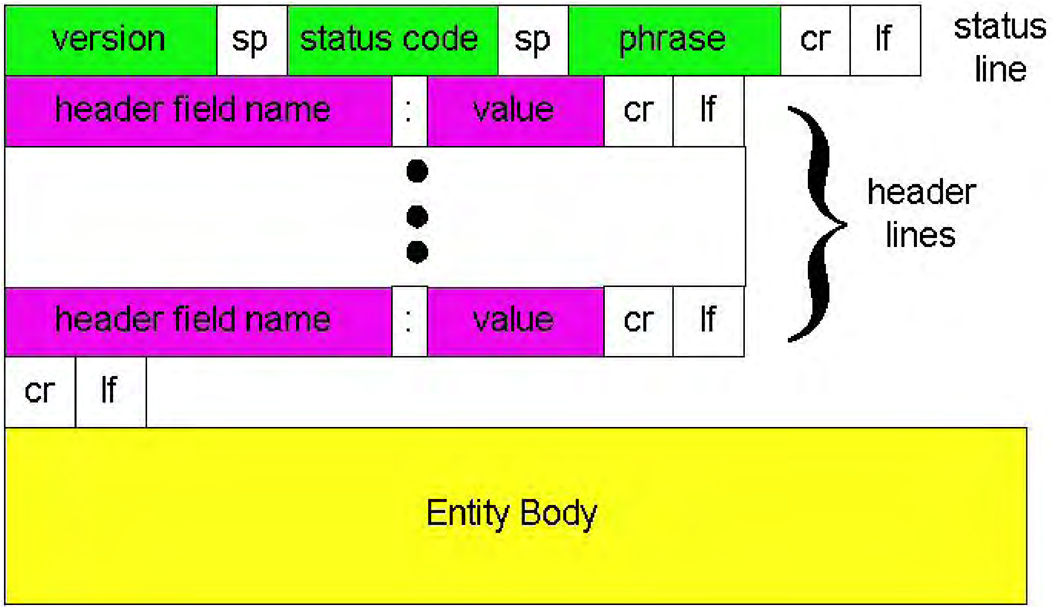
Content-Length: 6821

Content-Type: text/html

Το HTTP μήνυμα απόκρισης αποτελείται από τρία μέρη: την γραμμή κατάστασης (status line), τις γραμμές επικεφαλίδας (header lines) και το τμήμα περιεχομένου (entity body). Το τμήμα περιεχομένου περιέχει τα δεδομένα του αιτούμενου αρχείου, τα οποία στο παραπάνω παράδειγμα φαίνονται ως data data data data data data … Η γραμμή κατάστασης περιέχει τρία πεδία: το πεδίο HTTP έκδοσης, το πεδίο κωδικού κατάστασης και το πεδίο του αντίστοιχου μηνύματος κατάστασης. Στο παραπάνω παράδειγμα δηλώνεται ότι γίνεται χρήση του HTTP/1.1 και ότι δεν παρουσιάστηκε κάποιο σφάλμα (κωδικός κατάστασης 200 και αντίστοιχο μήνυμα κατάστασης OK).

Στο παραπάνω παράδειγμα περιλαμβάνονται έξι γραμμές επικεφαλίδας. Η γραμμή Connection:close δηλώνει ότι μετά την αποστολή του μηνύματος αυτού η TCP σύνδεση θα τερματιστεί. Η Date: γραμμή δηλώνει την ημερομηνία και την ώρα κατά την οποία δημιουργήθηκε και στάλθηκε η απόκριση, η Server: γραμμή δηλώνει τον τύπο του εξυπηρετητή Ιστού, η Last-Modified: δηλώνει την ημερομηνία και την ώρα κατά την οποία το αντικείμενο δημιουργήθηκε ή τροποποιήθηκε για τελευταία φορά, η γραμμή Content-Length: δηλώνει το μήκος των δεδομένων που αποστέλλονται (σε bytes), ενώ η γραμμή Content-Type: δηλώνει τον τύπο του αντικειμένου που περιέχεται στο τμήμα περιεχομένου του μηνύματος. Στο παραπάνω παράδειγμα δηλώνεται ότι χρησιμοποιείται ο Apache/1.3.0 (Unix) εξυπηρετητής Ιστού και ότι αποστέλλεται ένα αρχείο HTML μήκους 6821 bytes.

Σε περίπτωση που ο browser χρησιμοποιεί το HTTP/1.0 ακόμη και αν ο εξυπηρετητής Ιστού υποστηρίζει το HTTP/1.1, ο εξυπηρετητής Ιστού πρέπει να χρησιμοποιήσει μη μόνιμες συνδέσεις (δηλαδή στην απόκριση πρέπει να στείλει τη γραμμή Connection:close). Η δομή των HTTP μηνυμάτων απόκρισης φαίνεται στο παρακάτω σχήμα.



Από το παραπάνω σχήμα βλέπουμε ότι η γενική μορφή του μηνύματος ακολουθεί τη δομή του παραπάνω παραδείγματος. Στον παρακάτω πίνακα φαίνονται οι διάφοροι κωδικοί κατάστασης των HTTP αποκρίσεων με την αντίστοιχη σημασία τους.

Κωδικός κατάστασης	Μήνυμα κατάστασης	Επεξήγηση
200	OK	Επιτυχής αίτηση. Η πληροφορία επιστέφεται στην απόκριση
301	Moved Permanently	Το αιτούμενο αντικείμενο δεν υπάρχει. Η νέα τοποθεσία βρίσκεται στην επικεφαλίδα Location: Το λογισμικό του πελάτη πρέπει αυτόματα να αναζητήσει το αρχείο στο νέο URL.
400	Bad Request	Ο εξυπηρετητής δεν μπορεί να αναγνωρίσει το μήνυμα

		της αίτησης
404	Not Found	Το αιτούμενο αντικείμενο δεν υπάρχει στον εξυπηρετητή
505	HTTP Version Not Supported	Η αιτούμενη HTTP έκδοση δεν υποστηρίζεται από τον εξυπηρετητή

Το HTTP υποστηρίζει περισσότερους τύπους γραμμών επικεφαλίδας από αυτούς που περιγράφηκαν παραπάνω. Όμως, ο αριθμός των επικεφαλίδων που θα συμπεριληφθεί σε μία HTTP αίτηση εξαρτάται από τον τύπο και την έκδοση του browser (π.χ. αν ένας browser υποστηρίζει μόνο το HTTP/1.0 δεν μπορεί να συμπεριλάβει επικεφαλίδες του HTTP/1.1) αλλά και από τις ρυθμίσεις του χρήστη. Το ίδιο συμβαίνει και για τους τύπους των επικεφαλίδων που συμπεριλαμβάνουν οι εξυπηρετητές Ιστού στις HTTP αποκρίσεις.

ΚΕΦΑΛΑΙΟ 4

4.1 Η συσκευή android ως HTTP client

Όπως αναφέραμε προηγουμένως, για την επικοινωνία μέσω του HTTP πρωτοκόλλου απαιτείται ένας πελάτης ο οποίος θα στέλνει τις αιτήσεις σε έναν εξυπηρετητή και θα λαμβάνει απάντηση για κάθε αίτηση που στέλνει. Στο θεωρητικό κομμάτι που καλύψαμε παραπάνω αναφερθήκαμε στην κλασική περίπτωση που χρησιμοποιείται το HTTP πρωτόκολλο, αυτήν δηλαδή όπου ο πελάτης είναι ένας φυλλομετρητής (browser). Στην παρούσα εργασία τον ρόλο του client δεν έχει κάποιος φυλλομετρητής, αλλά η συσκευή android στην οποία θα τρέχει η εφαρμογή. Αυτό επιτυγχάνεται με την κατασκευή κατάλληλων HTTP αιτημάτων με την βοήθεια των HTTP βιβλιοθηκών που προσφέρονται για android λογισμικό. Τα αιτήματα αυτά (requests) είναι POST ή GET ανάλογα με την φύση του αιτήματος καθώς και το σώμα του (body), και μπορεί να περιλαμβάνουν strings, εικόνες, αριθμούς καθώς και διάφορα αρχεία. Στην περίπτωση που το αίτημα είναι τύπου GET τα στοιχεία που θέλουμε να μεταφέρουμε ενσωματώνονται στο URL που κατασκευάζουμε και αντιστοιχεί στη διεύθυνση του προγράμματος στον server το οποίο θα εξυπηρετήσει το συγκεκριμένο αίτημα όπως φαίνεται και στην παρακάτω εικόνα.

4.2 O php server

Για την εξυπηρέτηση των HTTP μηνυμάτων που αναφέραμε προηγουμένως απαιτείται ένας εξυπηρετητής (server), ένα πρόγραμμα δηλαδή το οποίο θα τρέχει στο πίσω άκρο (σε ένα μηχάνημα με στατική διεύθυνση IP) και θα είναι υπεύθυνο για την επεξεργασία του αιτήματος καθώς και για την παραγωγή απάντησης για το κάθε αίτημα. O PHP server είναι ένας εξυπηρετητής ο οποίος χρησιμοποιεί την γλώσσα PHP για την εξυπηρέτηση των αιτημάτων. Η PHP είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα-εφαρμογή στο άκρο του πελάτη (front-end). Η επικοινωνία μεταξύ πελάτη και PHP server γίνεται με την συγγραφή κατάλληλων scripts στον server τα οποία είναι διαμορφωμένα έτσι ώστε να παράγουν αντίστοιχη απάντηση στον κάθε χρήστη με βάση το αίτημα του. Ανάλογα το εκάστοτε αίτημα, μπορεί να εκτελεστούν διάφορες ενέργειες στον server όπως υπολογισμοί, δημιουργία, επεξεργασία ή διαγραφή αρχείων στον σκληρό δίσκο του μηχανήματος που φιλοξενεί τον server καθώς και καταχωρήσεις, διαγραφές ή αναζητήσεις σε μια βάση δεδομένων η οποία επίσης βρίσκεται στο μηχάνημα του εξυπηρετητή όπως φαίνεται στην εικόνα παρακάτω.

4.3 Η βάση δεδομένων MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων ανοικτού κώδικα (relational database management system - RDBMS), που χρησιμοποιεί την Structured Query Language (SQL), την πιο γνωστή γλώσσα για την προσθήκη, την πρόσβαση και την επεξεργασία δεδομένων σε μία Βάση Δεδομένων. Επειδή είναι ανοικτού κώδικα (open source), οποιοσδήποτε μπορεί να κατεβάσει τη MySQL και να την διαμορφώσει με βάση τις ανάγκες του, σύμφωνα πάντα με την γενική άδεια χρήσης. Η MySQL είναι γνωστή κυρίως για την ταχύτητα, την αξιοπιστία, και την ευελιξία που παρέχει.

Σε συνδυασμό με την γλώσσα PHP η MySQL παρέχει έναν ευέλικτο τρόπο αποθήκευσης δεδομένων σε μια σχεσιακή βάση δεδομένων στο μηχάνημα του εξυπηρετητή, παρέχοντας έτσι λύση στις ανάγκες αποθήκευσης στοιχείων των χρηστών της εφαρμογής και επαναχρησιμοποίησης αυτών στο μέλλον. Ειδικότερα, η σύνδεση στη βάση δεδομένων επιτυγχάνεται και αυτή μέσω ενός PHP script που τρέχει στον server, και έτσι γίνεται εφικτό στη συνέχεια το τρέξιμο ερωτημάτων (queries) προς την βάση. Τα ερωτήματα αυτά αφορούν καταχωρήσεις που θέλουμε να πραγματοποιήσουμε κάθε φορά ανάλογα με τις ανάγκες του πελάτη, όπως αποθήκευση στοιχείων, εικόνων, αρχείων κλπ, αλλά και ανάκτηση των στοιχείων αυτών με σκοπό την ταυτοποίηση των χρηστών (authentication).

ΚΕΦΑΛΑΙΟ 5

5.1 Εργαλεία που χρησιμοποιήθηκαν

Για την κατασκευή της εφαρμογής χρησιμοποιήθηκε το ολοκληρωμένο περιβάλλον ανάπτυξης Android Studio. Για την συγγραφή των scripts στον PHP server χρησιμοποιήθηκε ο Text Editor Atom, για την διαχείρηση της βάσης δεδομένων SQL χρησιμοποιήθηκε το εργαλείο PHP My Admin, για την κατασκευή των λογότυπων (logo) αλλά και των εικονιδίων της εφαρμογής χρησιμοποιήθηκε το Adobe Photoshop CS6, to MS Paint και το Android Asset Studio. Για debugging χρησιμοποιήθηκαν 2 συσκευές Android : Oukitel K4000 pro (Android 5.1) και Jiayu S3+ (Android 4.4) . Τέλος για cross - mobile testing χρησιμοποιήθηκε ο emulator Genymotion και για cross - tablet testing χρησιμοποιήθηκε το TestObject.

5.1.1 Σύντομη επισκόπηση του Android Studio

Το Android Studio είναι το επίσημο ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για την ανάπτυξη εφαρμογών Android, με βάση το IntelliJ IDEA. Το Android Studio είναι ελεύθερα διαθέσιμα υπό την άδεια χρήσης Apache 2.0.

Gradle

Το Android Studio χρησιμοποιεί Gradle (ένα λογισμικό ανοικτού κώδικα που χρησιμοποιείται για την αυτοματοποίηση της διαδικασίας "χτισίματος" λογισμικού) ως το θεμέλιο του συστήματος κατασκευής, με περισσότερα δυνατότητες για Android που παρέχονται από το Android Plugin για Gradle. Αυτό το σύστημα κατασκευής "τρέχει" ως ένα ολοκληρωμένο εργαλείο από το μενού του Android Studio, και ανεξάρτητα από τη γραμμή εντολών. Κάποια από τα πλεονεκτήματα που προσφέρει το συγκεκριμένο σύστημα κατασκευής (build system) είναι ότι ο χρήστης μπορεί να προσαρμόσει, να ρυθμίσει και να επεκτείνει τη διαδικασία κατασκευής(το build process της εφαρμογής), να δημιουργήσει πολλαπλά APKs για την εφαρμογή του με διαφορετικά χαρακτηριστικά χρησιμοποιώντας στην ουσία το ίδιο project, και τέλος να επαναχρησιμοποιήσει εύκολα τον κώδικα και τους πόρους της εφαρμογής του.

Τα αρχεία του build της εφαρμογής ονομάζονται build.gradle. Είναι απλά αρχεία κειμένου που χρησιμοποιούν το συντακτικό της γλώσσας Groovy για να διαμορφώσουν τη διαδικασία του build με στοιχεία που παρέχονται από το Android plugin για Gradle. Κάθε project στο Android Studio έχει ένα top-level αρχείο build, για όλο το project, έχει και ξεχωριστά module-level αρχεία build αρχεία για το κάθε module του project. Όταν ο χρήστης κάνει import κάποιο project το Android Studio παράγει αυτόματα τα απαραίτητα αρχεία build.

Modules

Ένα module είναι μια συλλογή αρχείων πηγαίου κώδικα της εφαρμογής και ρυθμίσεις για το build process της εφαρμογής. Τα modules επιτρέπουν στο προγραμματιστή να χωρίσει το project του σε διακριτά κομμάτια το καθένα από τα οποία παρέχει συγκεκριμένη λειτουργία και όλα μαζί συνδυασμένα καθορίζουν τη συνολική συμπεριφορά της εφαρμογής. Ένα project μπορεί να έχει ένα η πολλά modules, και κάποιο module μπορεί να χρησιμοποιήσει άλλα modules, και να δηλωθεί ως εξαρτημένο από αυτά. Κάθε module μπορεί να κατασκευαστεί, να δοκιμαστεί ή να αποσφαλματωθεί ως ανεξάρτητη οντότητα.

Τα modules είναι ιδιαίτερα χρήσιμα όταν ο προγραμματιστής θέλει να δημιουργήσει δικές του βιβλιοθήκες στο project που της χρησιμοποιεί η εφαρμογή, όπως επίσης και όταν θέλει να δημιουργήσει διαφορετικές υλοποιήσεις (διαφορετικά κομμάτια κωδικά και resources) για διαφορετικούς τύπους συσκευών(κινητά τηλέφωνα, ρολόγια android κ.α.) και να κρατήσει όλα τα αρχεία αυτά μαζί, στα πλαίσια ενός ενιαίου project και όχι πολλών, ώστε να μην χρειαστεί να επαναλάβει κομμάτια κώδικα ή resources που χρησιμοποιούνται από περισσότερες από μια υλοποιήσεις.

Ο χρήστης μπορεί να προσθέσει modules στο Android Studio επιλέγοντας

File > New > New Module. Το Android Studio παρέχει αρκετά διαφορετικά είδη modules.

Android app module

Αποτελεί ένα container για τον πηγαίο κώδικα της εφαρμογής,, τα αρχεία των πόρων, και τις ρυθμίσεις της εφαρμογής, όπως το αρχείο κατασκευής (build file) σε επίπεδο module και το αρχείο Android Manifest. Όταν δημιουργείτε ένα νέο project, το προεπιλεγμένο όνομα του module είναι "app".

Στο παράθυρο Create New Module του Android Studio διατίθονται αρκετά Modules τα οποία παρέχουν απαραίτητα αρχεία και κάποια πρότυπα κώδικα που είναι κατάλληλα για την αντίστοιχη εφαρμογή ή τύπο συσκευής. Τα modules αυτά είναι: Phone & Table Module, Android Wear Module, Android TV Module, Glass Module.

Library module

Αποτελεί ένα container για τον επαναχρησιμοποιήσιμο κώδικα, τον οποίο ο προγραμματιστής μπορεί να χρησιμοποιήσετε ως εξάρτηση(depndency) σε άλλa modules της εφαρμογής(δηλ. άλλα Modules να εξαρτώνται από αυτό για την λειτουργία τους) ή την εισαγωγή(import) του σε άλλα projects. Δομικά, το module της βιβλιοθήκης είναι το ίδιο με ένα app module, αλλά όταν "χτίζεται"(γίνεται build), δημιουργείται ένα συμπιεσμένο αρχείο κώδικα αντί ενός APK, με αποτέλεσμα να μην γίνεται η εγκατάστασή του στη συσκευή ως αυτόνομη εφαρμογή.

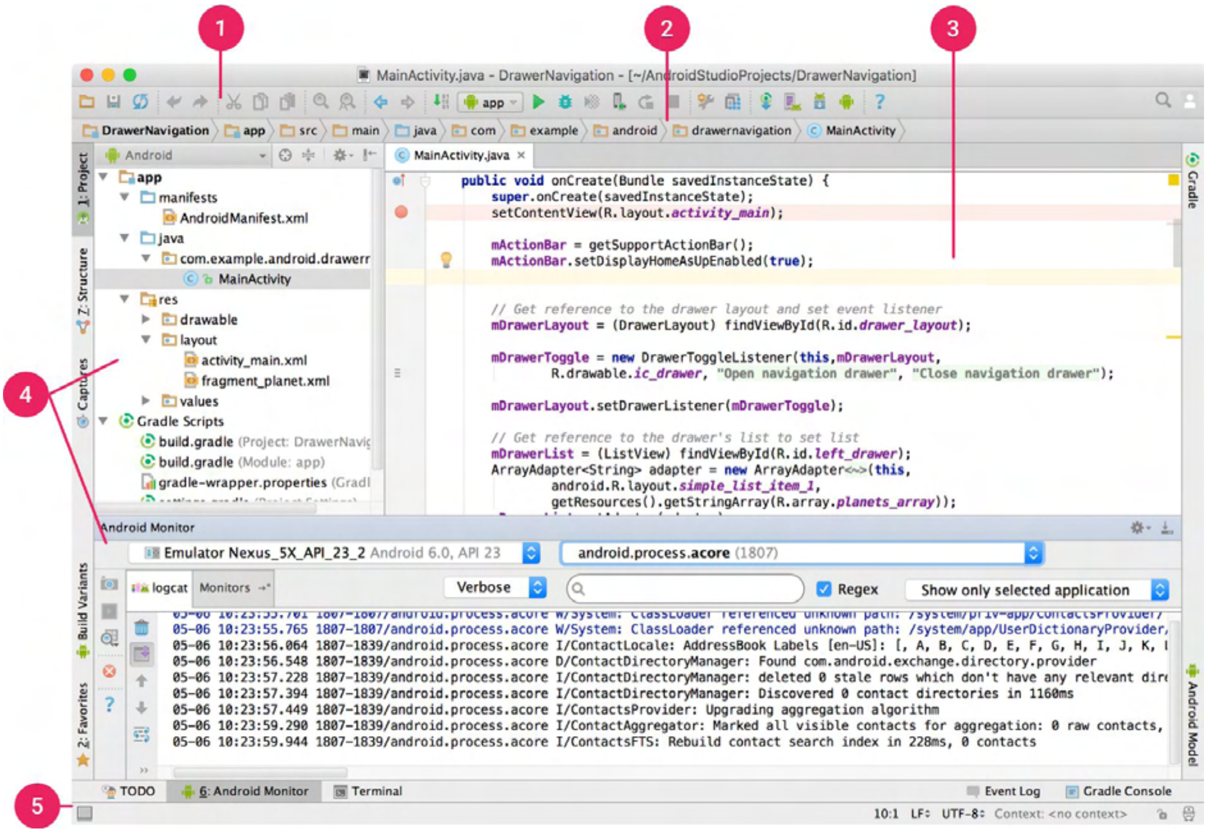
Στο παράθυρο Create New Module του Android Studio διατίθονται δύο library modules, Android Library και Java Library. Το Android Library module μπορεί να περιέχει όλα τα είδη αρχείων που υποστηρίζονται σε ένα Android Project συμπεριλαμβανομένου και του πηγαίου κώδικα, των πόρων, και του αρχείου android manifest. Το αποτέλεσμα του build process ενός τέτοιου module είναι ένα αρχείο Android Archive (AAR) το οποίο μπορεί να προστεθεί ως dependency(εξάρτηση) για άλλα app modules. To Java Library module μπορεί να περιέχει μόνο πηγαίο κώδικα Java. Το αποτέλεσμα του build process είναι ένα αρχείο Java ARchive (JAR) το οποίο μπορεί να προστεθεί ως dependency(εξάρτηση) για άλλα app modules ή άλλα Java projects.

Google Cloud module

Αποτελεί ένα container για το Google Cloud backend κώδικα της εφαρμογής Το module αυτό προσθέτει τον απαιτούμενο κώδικα και εξαρτήσεις για ένα λειτουργικό, εκτελέσιμο backend που ο προγραμματιστής μπορεί να επεκτείνει με χαρακτηριστικά που επιθυμεί, όπως η αποθήκευση δεδομένων.

User Interface

Όπως φαίνεται και στην παρακάτω εικόνα, το κεντρικό παράθυρο του Android Studio αποτελείται από τα παρακάτω επιμέρους λογικά τμήματα.

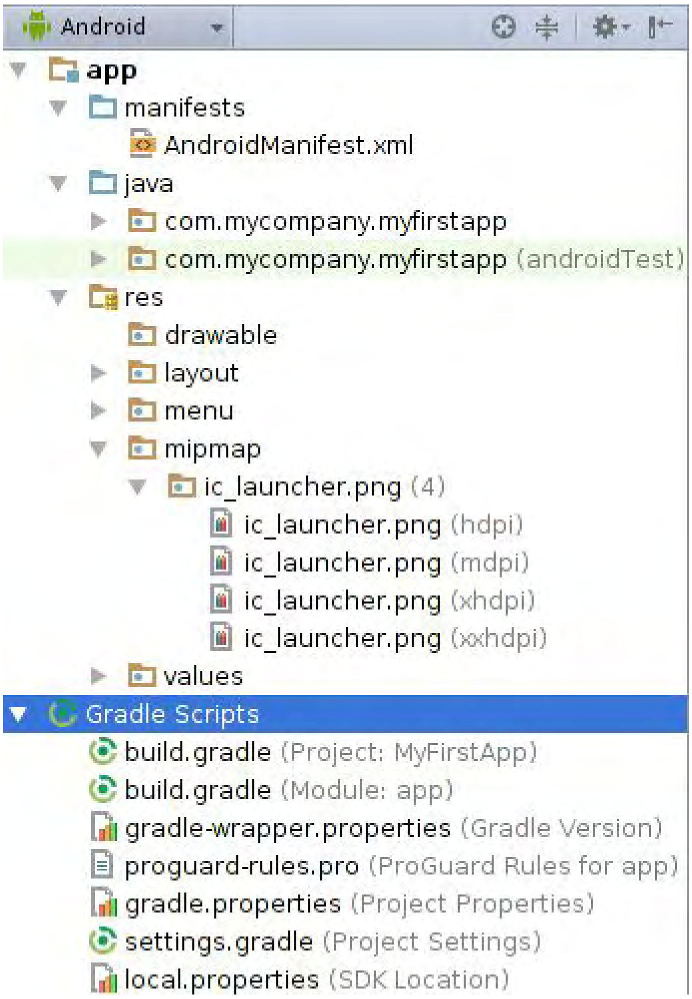


1. Η γραμμή εργαλείων επιτρέπει πληθώρα ενεργειών όπως το "τρέξιμο" της εφαρμογής και τη χρήση διάφορων εργαλείων android.
2. Η μπάρα πλοήγησης παρέχει εύκολη πρόσβαση σε όλα τα επιμέρους τμήματα του project, κάνοντας χρήση μιας ιεραρχικής δόμησης των φακέλων, όπως αυτά φαίνονται και στο παράθυρο του εργαλείου "Project".
3. Στο παράθυρο επεξεργασίας(editor window) ο χρήστης μπορεί να δημιουργεί και να τροποποιεί κώδικα. Το παράθυρο αυτό αλλάζει, με βάση το είδος του αρχείου που έχει ανοίξει για επεξεργασία.
4. Τα παράθυρα εργαλείων (Tool windows) παρέχουν πρόσβαση σε συγκεκριμένες ενέργειες όπως είναι η διαχείριση του project, η αναζήτηση, ο έλεγχος έκδοσης (version control) και άλλα. Τα παράθυρα αυτά μπορούν να συμπτυχθούν και να αναπτυχθούν.
5. Η γραμμή κατάστασης(status bar) εμφανίζει την κατάσταση του project καθώς και του ίδιου του IDE και τυχόν προειδοποιήσεων και μηνυμάτων.

Ο χρήστης έχει τη δυνατότητα να οργανώσει όπως θέλει την εμφάνιση του κεντρικού παραθύρου του Android Studio κρύβοντας ή μετακινώντας γραμμές εργαλείων και παράθυρα. Μπορεί επίσης να χρησιμοποιήσει πολλές συντομεύσεις πληκτρολογίου για άμεση πρόσβαση σε πολλά χαρακτηριστικά του IDE. Ο χρήστης μπορεί ανά πάσα στιγμή, να πραγματοποιήσει αναζήτηση σε όλο τον πηγαίο κώδικα, τις βάσεις δεδομένων, τα στοιχεία της διεπαφής χρήστη(user interface), και ούτω καθεξής, με διπλό πάτημα του πλήκτρου Shift, ή κάνοντας κλικ στο μεγεθυντικό φακό στην επάνω δεξιά γωνία του παραθύρου του Android Studio.

Η δομή των αρχείων ενός project στο Android Studio και η παρουσίασή τους στο User Interface:

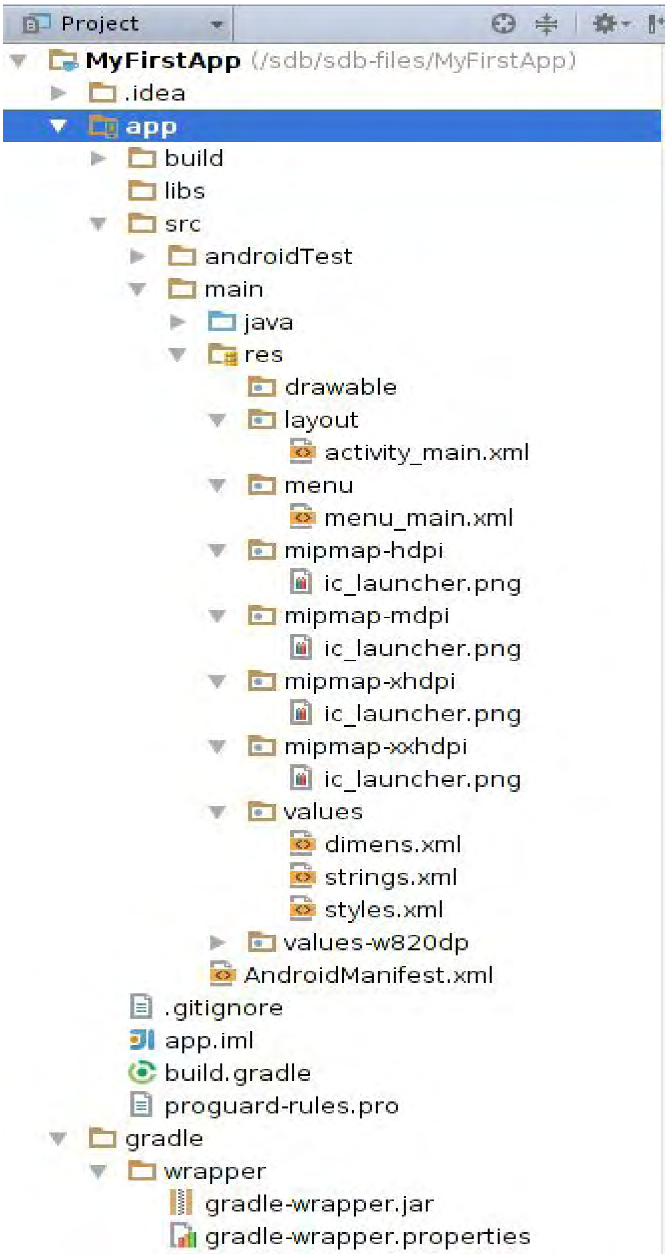
Android view



Από προεπιλογή το Android Studio εμφανίζει- παρουσιάζει τα αρχεία του project σε Android view. Αυτό δεν αντικατοπτρίζει την πραγματική ιεραρχία των φακέλων στο σκληρό δίσκο, αλλά είναι οργανωμένο με βάση τα επιμέρους modules και τους διαφορετικούς τύπους αρχείων με σκοπό την απλοποίηση της πλοήγησης του χρήστη μεταξύ σημαντικών αρχείων-κλειδιών του project, κρύβοντας ορισμένα αρχεία ή φακέλους που συνήθως δεν χρησιμοποιούνται Ο χρήστης, έχοντας επιλέξει το Android View, μπορεί να βρει τα αρχεία ρυθμίσεων για τη διαδικασία “χτισίματος”(build) της εφαρμογής, τα manifest αρχεία για κάθε module, όπως και όλα τα εναλλακτικά resources με βάση το είδος τους, ομαδοποιημένα στο ίδιο μέρος και όχι σε διαφορετικούς φακέλους όπως αυτά βρίσκονται πραγματικά στον σκληρό δίσκο.

Μέσα σε κάθε Android app module, τα αρχεία βρίσκονται ομαδοποιημένα στις ομάδες “manifests”, “java” kai “res”. Η ομάδα “manifests” περιέχει το αρχείο AndroidManifest.xml, η ομάδα “java” περιέχει τα αρχεία πηγαίου κώδικα Java, διαχωρισμένα με βάση το package name τους και η ομάδα “res” περιέχει όλους τους πόρους που δεν έχουν μορφή κώδικα, όπως είναι τα XML layouts, User interface strings, εικόνες bitmap κ.α., διαχωρισμένους σε κατάλληλες υποομάδες.

Project view



Για να δει ο χρήστης την πραγματική δομή των αρχείων του project, όπως αυτά βρίσκονται στο σκληρό, συμπεριλαμβανομένων και κάποιον αρχείων που αποκρύπτονται στο Android view, πρέπει να επιλέξει το “Project” στο παράθυρο του project, εκεί που ήταν προεπιλεγμένο το “Android”(είχαμε δηλαδή το Android view ως προεπιλογή). Με την επιλογή του Project view ο χρήστης έχει τη δυνατότητα να δει πολλά περισσότερα αρχεία και φακέλους, από τα οποία τα πιο σημαντικά είναι:

module-name/

build/

Περιέχει διαφορετικά build outputs.

libs/

Περιέχει ιδιωτικές βιβλιοθήκες της εφαρμογής.

src/

Περιέχει όλα τα αρχεία κώδικα και πόρων για το συγκεκριμένο module στους εξής υποφακέλους:

androidTest/

Περιέχει κώδικα για δοκιμές της εφαρμογής σε μια συσκευή Android.

main/

Περιέχει τα “κύρια” αρχεία: τον κώδικα και τους πόρους που είναι κοινά για όλα τα διαφορετικά builds.

AndroidManifest.xml

Περιγράφει τη φύση της εφαρμογής και κάθε ένα από τα συστατικά της μέρη.

java/

Περιέχει τα αρχεία πηγαίου κώδικα Java.

jni/

Περιέχει εγγενή κώδικα χρησιμοποιώντας το Java Native Interface (JNI).

gen/

Περιέχει τα αρχεία Java που δημιουργούνται από το Android Studio, όπως το αρχείο R.java.

res/

Περιέχει τους πόρους της εφαρμογής όπως τα drawable αρχεία, τα layout αρχεία, τα User Interface strings (κείμενα) κ. α.

assets/

Περιέχει αρχεία τα οποία θα πρέπει να μεταγλωττίσουν σε αρχείο .apk όπως είναι, χωρίς να υποστούν αλλαγές.

test/

Περιέχει κώδικα για τις τοπικές δοκιμές που εκτελούνται σε τοπική JVM.

build.gradle (module)

Αυτό καθορίζει το build configuration (διαμόρφωση κατασκευής) για το συγκεκριμένο module.

build.gradle (project)

Αυτό καθορίζει το build configuration (διαμόρφωση κατασκευής) για όλα τα modules. Αυτό το αρχείο είναι αναπόσπαστο μέρος του project και θα πρέπει να ελέγχεται συχνά για τυχόν αναθεωρήσεις όπως και όλος ο πηγαίος κώδικας της εφαρμογής.

Project Structure Settings

Ο χρήστης μπορεί να έχει πρόσβαση σε σημαντικές ρυθμίσεις του project στο File > Project Structure. Το Project Structure περιέχει τους εξής τομείς:

- SDK Location: Θέτει την τοποθεσία του σκληρού δίσκου, όπου βρίσκονται το JDK, το Android SDK, και το Android NDK που χρησιμοποιούνται από το project.
- Project: Ορίζει την έκδοση του Gradle, το plugin του Android για Gradle, και το όνομα της τοποθεσίας αποθήκευσης.
- Developer Services: Περιέχει ρυθμίσεις για τα Android Studio add-in components.
- Modules: Επιτρέπει στο χρήστη να επεξεργαστεί τα build configurations σε επίπεδο module, όπως είναι οι εξαρτήσεις(dependencies), το target και το minimum SDK, και το signature της εφαρμογής.

Το τμήμα Developer Services του Project Structure περιέχει σελίδες ρυθμίσεων(configuration pages) για αρκετές υπηρεσίες που ο χρήστης μπορεί να χρησιμοποιήσει στην εφαρμογή του. Αυτές είναι:

- AdMob: Σας επιτρέπει να ενεργοποιήσετε το AdMob component της Google, το οποία βοηθά τον προγραμματιστή να "κατανοήσει" τους χρήστες της εφαρμογής ώστε να τους προβάλει προσαρμοσμένες διαφημίσεις.
- Analytics: Επιτρέπει την εκκίνηση του Google Analytics το οποίο μετρά τις αλληλεπιδράσεις των χρηστών με την εφαρμογή σε διάφορες συσκεύες.
- Authentication: Επιτρέπει στους χρήστες της εφαρμογής να κάνουν sign-in στην εφαρμογή χρησιμοποιώντας τον Google λογαριασμό τους.
- Cloud: Επιτρέπει την ενεργοποίηση των cloud-based υπηρεσιών Firebase για την εφαρμογή.
- Notifications: Επιτρέπει τη χρήση Google Cloud Messaging για την επικοινωνία μεταξύ της εφαρμογής και του διακομιστή της.

Ενεργοποιώντας οποιαδήποτε από αυτές τις υπηρεσίες μπορεί να προκαλέσει το Android Studio να βάλει επιπρόσθετες εξαρτήσεις και να δώσει επιπλέον δικαιώματα στην εφαρμογή. Κάθε μια από τις παραπάνω σελίδες ρυθμίσεων(configuration pages) παραθέτει όλες τις εξαρτήσεις και τα δικαιώματα αυτά, καθώς και άλλες ενέργειες που το Android Studio θα πραγματοποιήσει αυτόματα, εφόσον ο προγραμματιστής θα επιλέξει να ενεργοποιήσει τη σχετική υπηρεσία.

Το τμήμα Modules του Project Structure επιτρέπει την αλλαγή των ρυθμίσεων για κάθε ένα από τα modules του project μας. Κάθε μια από τις σελίδες ρυθμίσεων του τμήματος Modules αποτελείται από τις παρακάτω καρτέλες ρυθμίσεων(tabs):

- Properties: Καθορίζει τις εκδόσεις του SDK και τα build tools που πρέπει να χρησιμοποιηθούν για την μεταγλώττιση του συγκεκριμένου module.
- Signing: Καθορίζει το πιστοποιητικό που πρέπει να χρησιμοποιηθεί ως υπογραφή του APK της εφαρμογής μας.
- Flavors: Επιτρέπει τη δημιουργία πολλαπλών build flavors("γεύσεις κατασκευής"), όπου κάθε flavor καθορίζει μια σειρά από ρυθμίσεις, όπως το ελάχιστο SDK και το SDK-στόχο του module, τον κωδικό έκδοσης και το όνομα έκδοσης. Για παράδειγμα, ο προγραμματιστής μπορεί να ορίσει ένα flavor που έχει ελάχιστο SDK το 15 και SDK-στόχο το 21, και ένα δεύτερο flavor που έχει ένα ελάχιστο SDK το 19 και SDK-στόχο το 23.
- Build Types: Επιτρέπει τη δημιουργία και επεξεργασία διάφορων build configurations. Κάθε module έχει από προεπιλογή δύο build types, debug και release. Ο προγραμματιστής μπορεί να ορίσει περισσότερα build types με βάση τις ανάγκες του.
- Dependencies: Εδώ παραθέτονται οι εξαρτήσεις του συγκεκριμένου module από άλλα modules, βιβλιοθήκες και αρχεία. Ο προγραμματιστής μπορεί να προσθέσει, να επεξεργαστεί και να διαγράψει εξαρτήσεις από αυτό το πλαίσιο.

ΚΕΦΑΛΑΙΟ 6

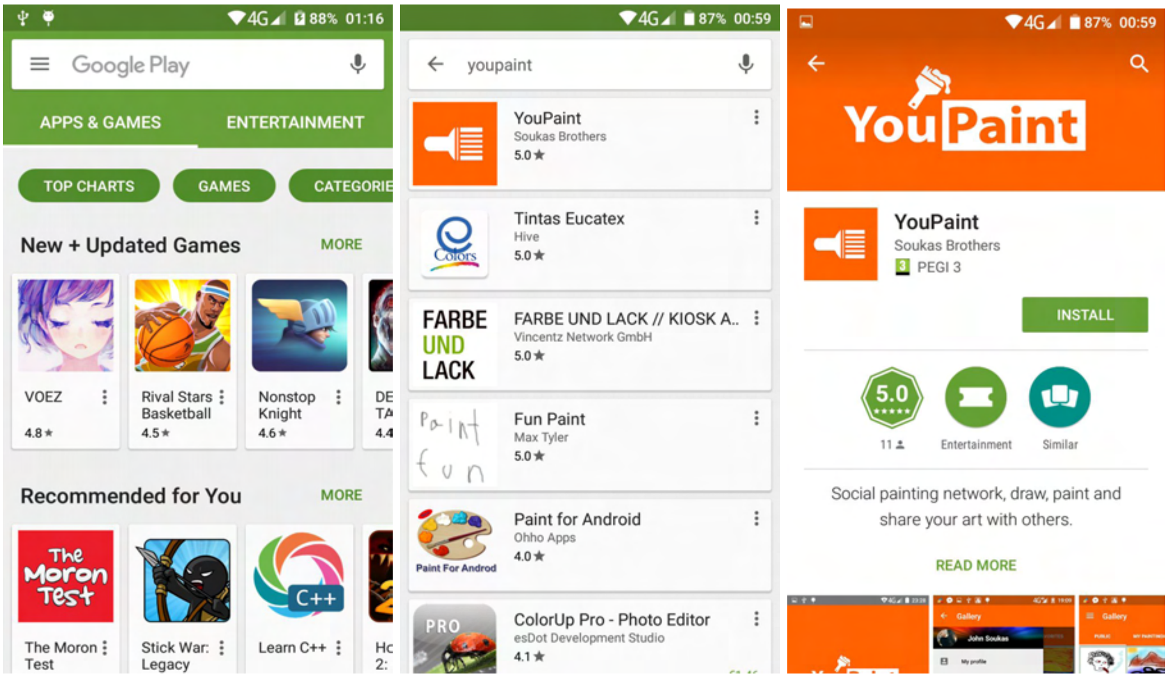
6.1 Περιγραφή υλοποίησης της εφαρμογής YouPaint

Η εφαρμογή υλοποιήθηκε σε Android Studio (version 1.5.1) και έγινε compile σε SDK version 23 (Android 6.0) με minimum SDK version την 15 που σημαίνει ότι υποστιρίζει όλες τις συσκευές με έκδοση Android 4.0.3 και μεταγενέστερη (περίπου το 94.5% των συσκευών). Για build automation χρησιμοποιήθηκε το Gradle 23.0.2.

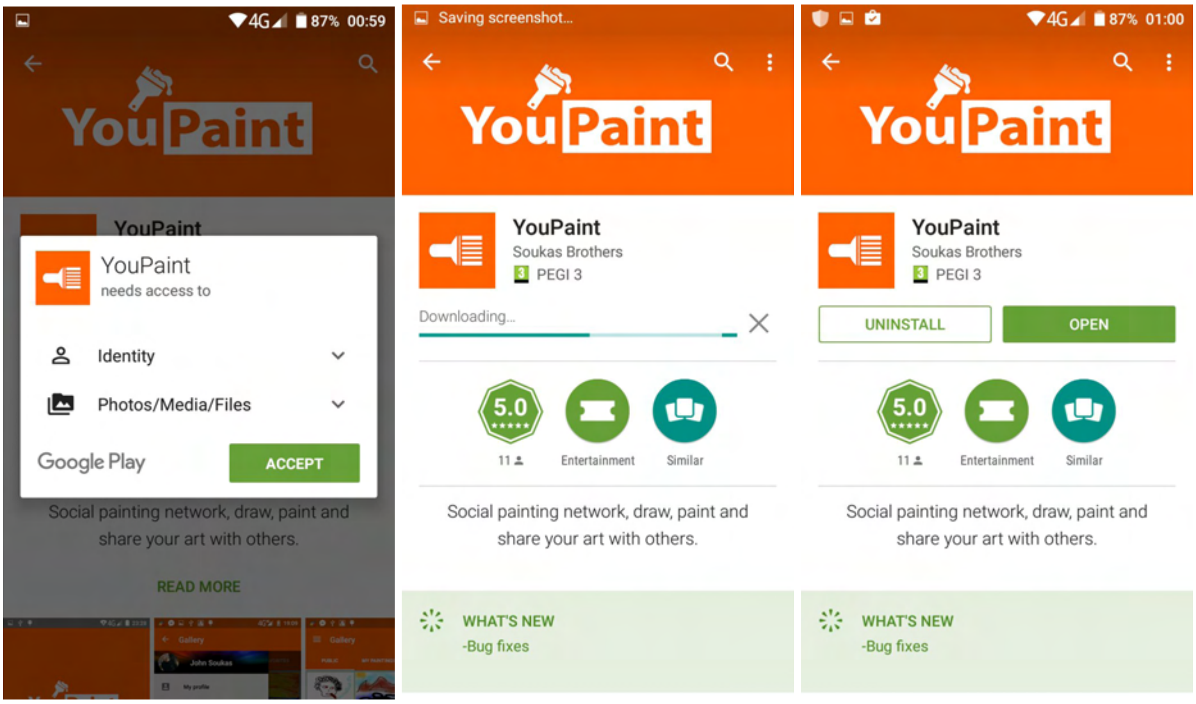
Για το στήσιμο του server εξυπηρέτησης της εφαρμογής χρησιμοποιήθηκε ο PHP server 5.5.35 και για το στήσιμο της βάσης δεδομένων της εφαρμογής χρησιμοποιήθηκε βάση MySQL 5.1.71 με phpMyAdmin 3.5.2.2.

6.2 Παρουσίαση της εφαρμογής YouPaint

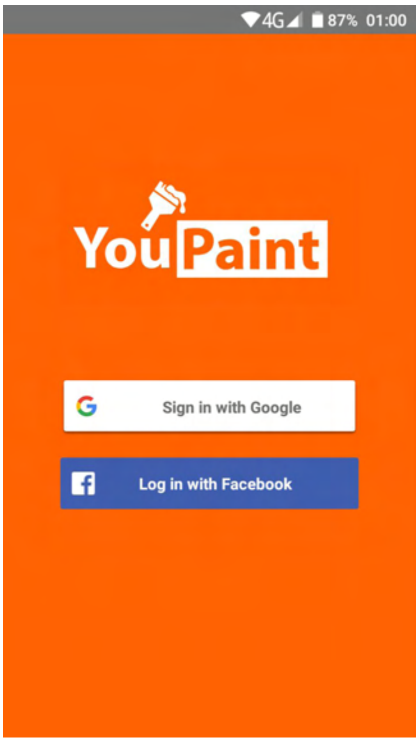
Η εφαρμογή βρίσκεται ανεβασμένη στο Play Store της Google στην τελευταία της έκδοση (v1.2.3 αυτή τη στιγμή). Μπορεί κανείς να την βρει κάνοντας αναζήτηση με το όνομα της.



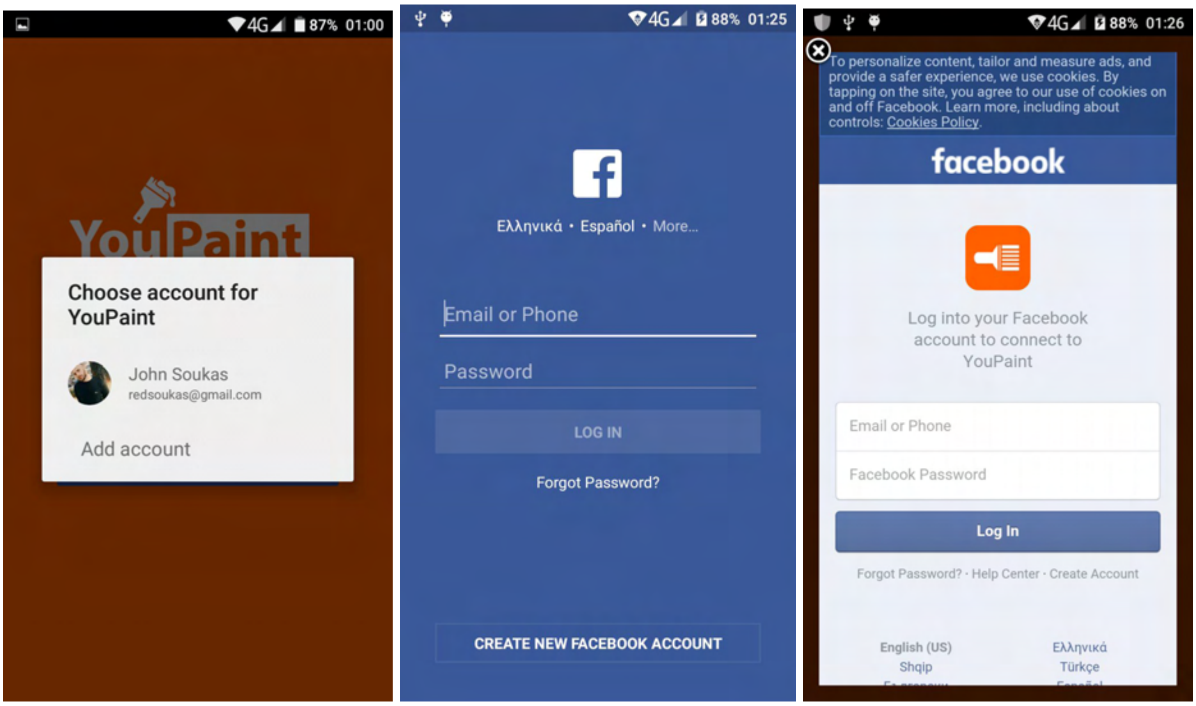
Παρακάτω φαίνονται screenshots από την εγκατάσταση της εφαρμογής.



Παρακάτω φαίνεται η οθόνη υποδοχής η οποία περιέχει το λογότυπο της εφαρμογής και 2 κουμπιά για σύνδεση μέσω Google και σύνδεση μέσω Facebook.

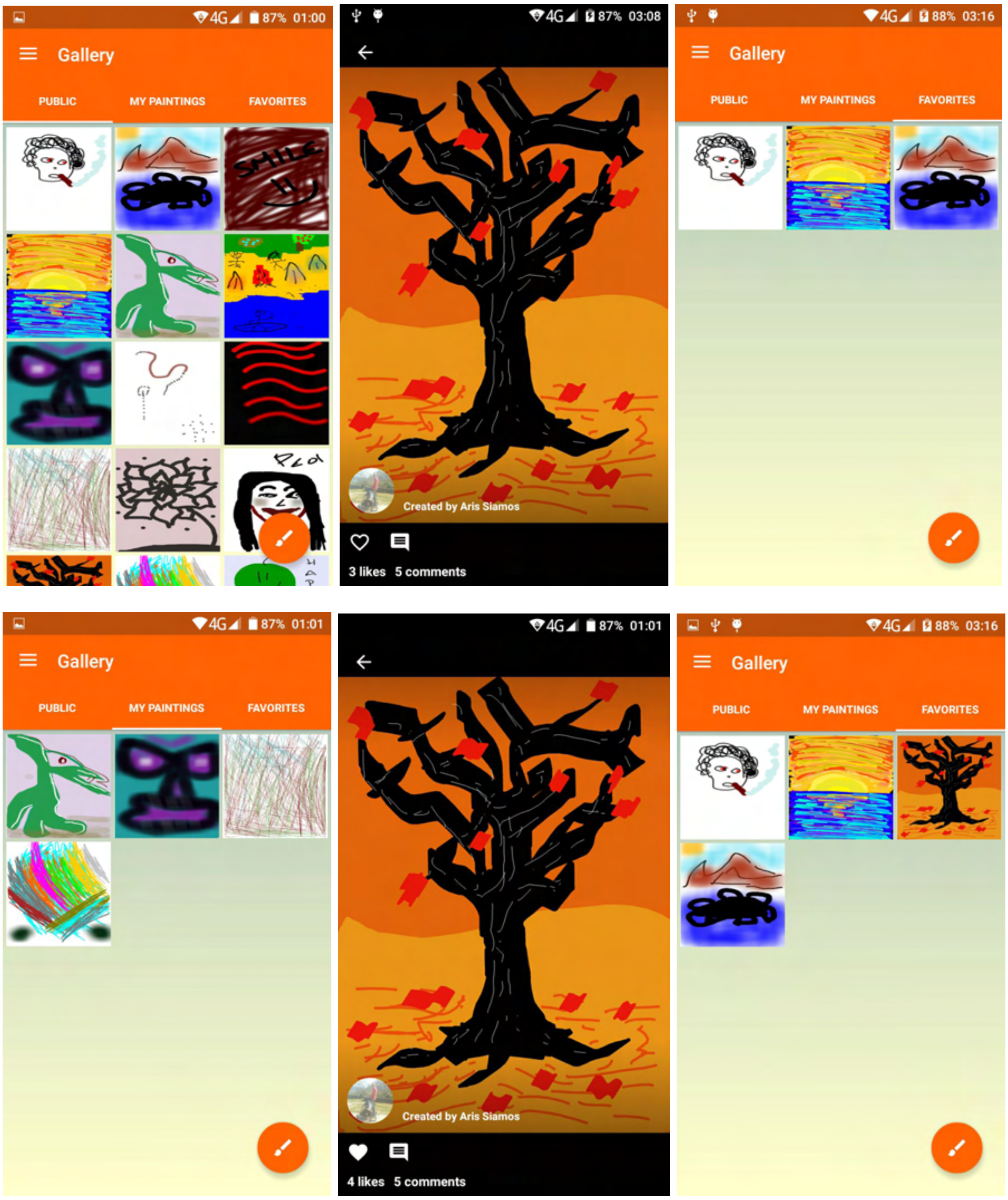


Διαλέγοντας το Google login εμφανίζεται η οθόνη επιλογής του Google λογαριασμού που επιθυμεί ο χρήστης να χρησιμοποιήσει για την σύνδεση. Αν διαλέξει να συνδεθεί μέσω Facebook, τότε θα μεταφερθεί στην εφαρμογή Facebook για android προκειμένου να βάλει τα στοιχεία του ή στο custom tab (εικόνα στα δεξιά παρακάτω) αν δεν έχει εγκατεστημένη την native εφαρμογή του Facebook.

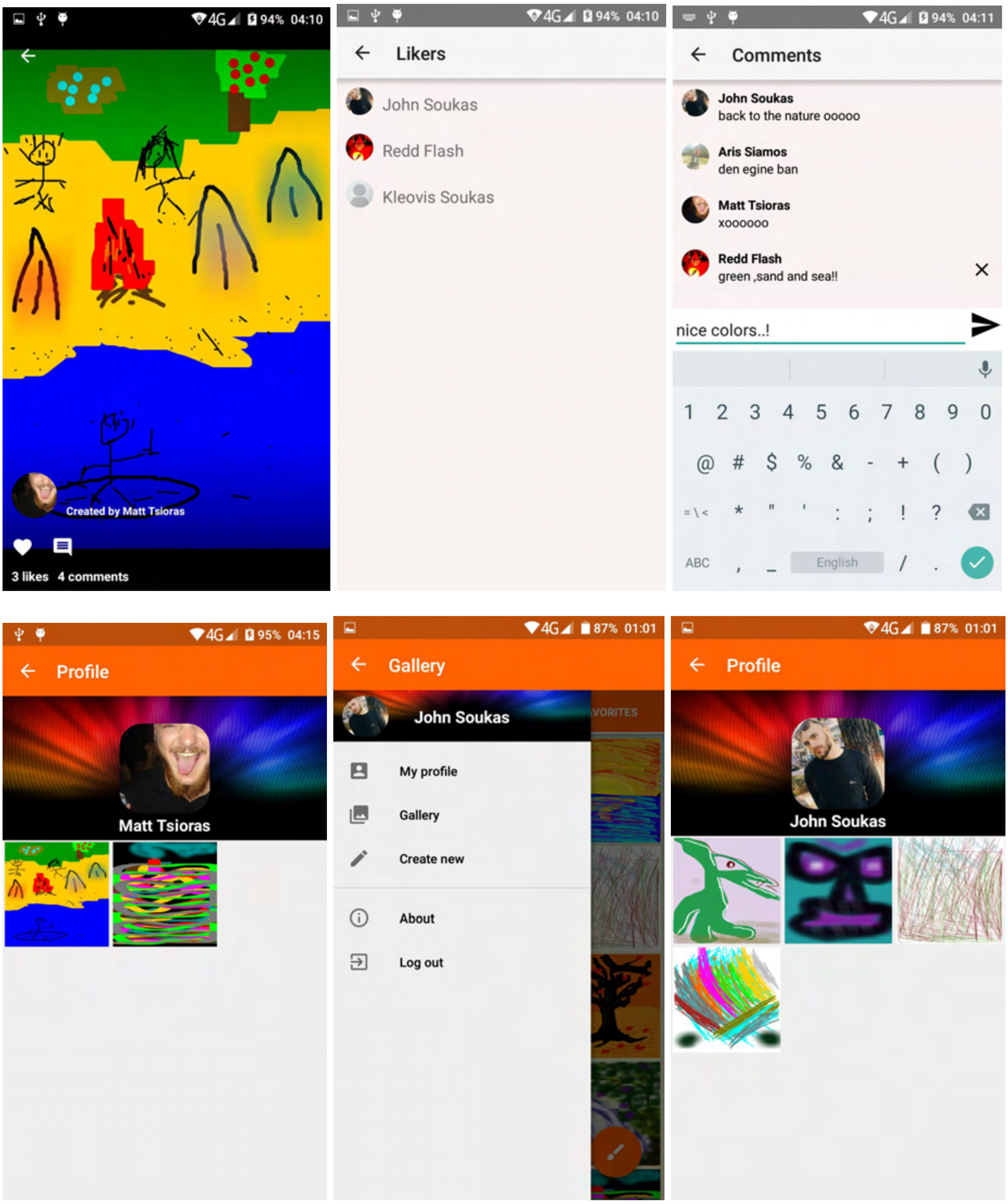


Να τονίσουμε σε αυτό το σημείο πως δεν χρησιμοποιούνται ούτε φαίνονται πουθενά οι κωδικοί του χρήστη σε κανένα επίπεδο της εφαρμογής. Τα στοιχεία του απλώς χρησιμοποιούνται για ταυτοποίηση (authentication) μέσω των server της Google και του Facebook αντίστοιχα χρησιμοποιώντας τα αντίστοιχα API's που προσφέρονται από τις εταιρίες, η εφαρμογή χρησιμοποιεί μόνο την εικόνα προφίλ του χρήστη και το όνομά του για να δημιουργήσει το αντίστοιχο δημόσιο profile του στο κοινωνικό δίκτυο των ζωγραφιών.

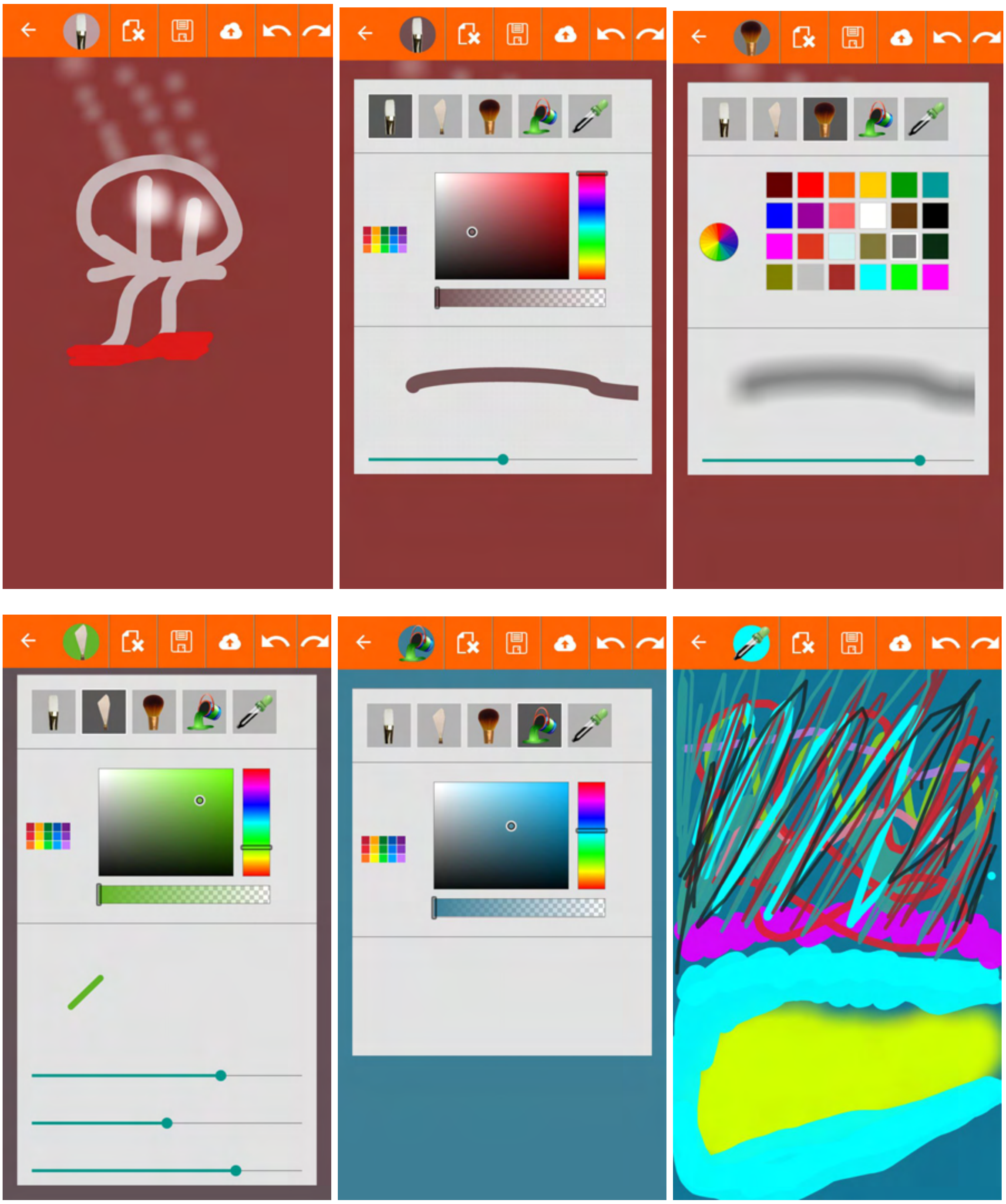
Αφού συνδεθεί ο χρήστης μεταφέρεται στην οθόνη της Gallery όπου υπάρχουν 3 υποκατηγορίες, στην πρώτη (public) φαίνονται όλες οι ζωγραφιές που έχουν κάνει οι χρήστες μέσω της εφαρμογής, στην δεύτερη κατηγορία (my paintings) φαίνονται οι ζωγραφιές που έχει κάνει ο ίδιος ο χρήστης και τέλος στην κατηγορία favorites φαίνονται οι ζωγραφιές που έχει μαρκάρει ως αγαπημένες πατώντας το αντίστοιχο κουμπί όπως φαίνεται παρακάτω.



Ο χρήστης μπορεί πατώντας πάνω στην εικόνα προφίλ ή το όνομα του καλλιτέχνη να πλοηγηθεί στο προφίλ του όπου φαίνονται οι ζωγραφιές που έχει κάνει. Επίσης έχει την δυνατότητα να εκφράσει τις σκέψεις του για τη ζωγραφιά σχολιάζοντας και να δείξει τον ενθουσιασμό του πατώντας like.

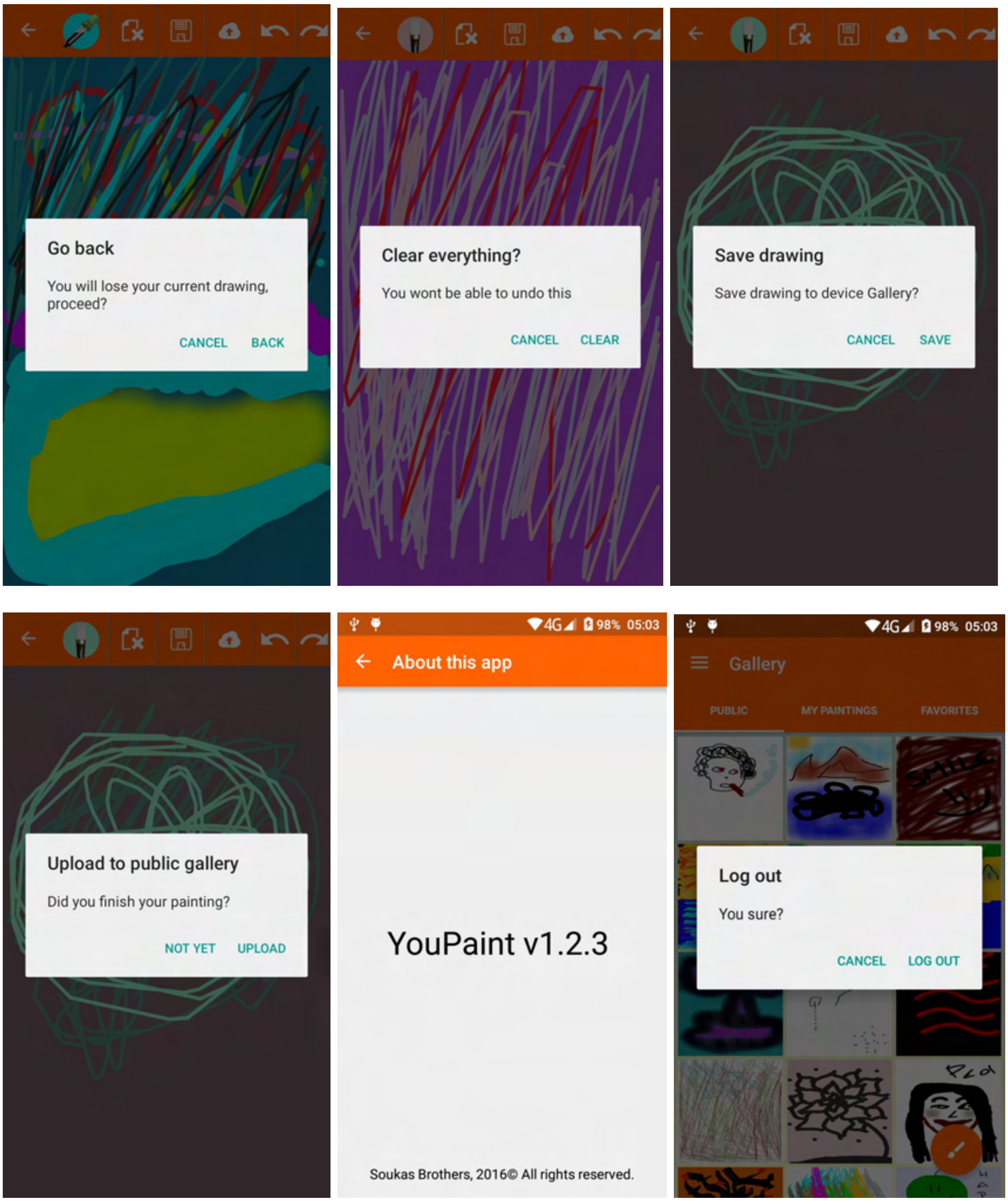


Πατώντας πάνω στο κουμπί Create new του menu ή στην συντόμευσή του (το πορτοκαλί πινέλο), ο χρήστης μεταφέρεται στην οθόνη - καμβά, όπου μπορεί να ξεκινήσει να ζωγραφίζει. Τα εργαλεία που προσφέρονται είναι 2 βούρτσες, μια σπάτουλα, ένας color picker και ένα γέμισμα καμβά με χρώμα. Επίσης προσφέρονται όλα τα χρώματα σε όλες τις αποχρώσεις με δυνατότητα διαμόρφωσης της διαφάνειας (opacity).



Τα κουμπιά της μπάρας δράσης είναι (από αριστερά προς τα δεξιά):

Back button, άνοιγμα- κλείσιμο παλέτας με τα εργαλεία και τα χρώματα, καθαρισμός καμβά, αποθήκευση ζωγραφιάς στην συσκευή, ανέβασμα της ζωγραφιάς στην δημόσια gallery και τέλος τα κουμπιά undo και redo. Τέλος όταν αποσυνδεθεί ο χρήστης μεταφέρεται στην αρχική οθόνη.


Go back
You will lose your current drawing, proceed?
CANCEL BACK
Clear everything?
You wont be able to undo this
CANCEL CLEAR
Save drawing
Save drawing to device Gallery?
CANCEL SAVE
Upload to public gallery
Did you finish your painting?
NOT YET UPLOAD
98% 05:03
About this app
YouPaint v1.2.3
Soukas Brothers, 2016© All rights reserved.
98% 05:03
Gallery
PUBLIC
MY PAINTINGS
FAVORITES
Log out
You sure?
CANCEL LOG OUT

ΚΕΦΑΛΑΙΟ 7

7.1 Επίλογος, μελλοντικές επεκτάσεις

Στόχος της εφαρμογής ήταν η δημιουργία μιας πλατφόρμας που θα επιτρέπει στους χρήστες να ζωγραφίσουν χρησιμοποιώντας τα απλά εργαλεία ζωγραφικής σε καμβά χωρίς την προσθήκη έτοιμων εικόνων και stickers όπως γίνεται σε αντίστοιχες εφαρμογές στο Google Play Store, προωθώντας έτσι την αγνή και καθαρή τέχνη και δίνοντας την ευκαιρία στους καλλιτέχνες να αναδειχτούν μέσω της εφαρμογής.

Στο μέλλον και ανάλογα με τον όγκο των χρηστών θα μπορούσαν να προστεθούν καινούρια features όπως εβδομαδιαίοι διαγωνισμοί καλύτερης ζωγραφιάς, collaborative painting, καινούρια εργαλεία ζωγραφικής, καθώς και paint battles με χρονικό όριο και με live βαθμολόγηση από χρήστες. Θα μπορούσε να προστεθεί και ένα σύστημα κατάταξης των καλλιτεχνών ανάλογα με την επίδοσή τους στους διαγωνισμούς.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] https://en.wikipedia.org/wiki/Android_(operating_system)

[2] https://en.wikipedia.org/wiki/PHP

[3] https://en.wikipedia.org/wiki/MySQL

[4] https://en.wikipedia.org/wiki/Android_Studio

[5] https://semfe.gr/files/users/376/texnologies_diadiktyou-kefalaio10.pdf

[6] https://en.wikipedia.org/wiki/Client%E2%80%93server_model

[7] https://en.wikipedia.org/wiki/Android_software_development

[8] https://en.wikipedia.org/wiki/Gradle

[9] https://developer.android.com/guide/index.html

[10] https://developer.android.com/studio/command-line/adb.html

[11] https://developer.android.com/guide/components/fundamentals.html

[12] https://developer.android.com/guide/components/intents-filters.html

[13] https://developer.android.com/reference/android/app/Activity.html

[14] https://developer.android.com/reference/android/app/Service.html

[15] https://developer.android.com/reference/android/content/ContentProvider.html

[16] https://developer.android.com/reference/android/content/BroadcastReceiver.html

[17] https://developer.android.com/guide/topics/manifest/manifest-intro.html

[18] https://developer.android.com/studio/intro/index.html

[19] https://developer.android.com/studio/projects/index.html

[20] https://developer.android.com/guide/topics/ui/overview.html

[21] https://developer.android.com/guide/topics/resources/overview.html

[22] https://developer.android.com/guide/topics/resources/providing-resources.html

[23] https://developer.android.com/studio/build/index.html

[24] https://developer.android.com/studio/publish/app-signing.html