



Τεχνολογικό Εκπαιδευτικό Ίδρυμα Δυτικής Ελλάδας

Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Μηχανικών Πληροφορικής Τ.Ε.

Πτυχιακή Εργασία

**Ανάπτυξη Εφαρμογής Android βασισμένη στην
πλατφόρμα StudentsWeb**

Αντώνιος Φρέρης

ΑΜ:2328

**Επιβλέπων Καθηγητής:
Αλεφραγκής Παναγιώτης**

Αντίρριο, Δεκέμβριος 2016



Τεχνολογικό Εκπαιδευτικό Ίδρυμα Δυτικής Ελλάδας

Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Μηχανικών Πληροφορικής Τ.Ε.

Πτυχιακή Εργασία

Ανάπτυξη Εφαρμογής Android βασισμένη στην πλατφόρμα StudentsWeb

Αντώνιος Φρέρης

AM:2328

Επιβλέπων Καθηγητής:
Αλεφραγκής Παναγιώτης

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την
2016.

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....

.....

.....



Τεχνολογικό Εκπαιδευτικό Ίδρυμα Δυτικής Ελλάδας

Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Μηχανικών Πληροφορικής Τ.Ε.

Copyright © –All rights reserved Φρέρης Αντώνιος, 2016.

Με την επιφύλαξη παντός δικαιώματος.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Το περιεχόμενο αυτής της εργασίας δεν απηχεί απαραίτητα τις απόψεις του Τμήματος, του Επιβλέποντα, ή της επιτροπής που την ενέκρινε.

Υπεύθυνη Δήλωση

Βεβαιώνω ότι είμαι συγγραφέας αυτής της πτυχιακής εργασίας, και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην πτυχιακή εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης, βεβαιώνω ότι αυτή η πτυχιακή εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Μηχανικών Πληροφορικής Τ.Ε. του ΤΕΙ Δυτικής Ελλάδας.

(Υπογραφή)

.....

Φρέρης Αντώνιος

Η πτυχιακή αυτή εργασία αυτή είναι αφιερωμένη
στην οικογένειά μου και σε όλους όσους με έχουν
βοηθήσει και με έχουν στηρίξει.

Περίληψη

Σαν φοιτητές η τεχνολογία είναι για τα καλά μέσα στην ζωή μας. Τα κινητά είναι ένα αναπόσπαστο κομμάτι από την καθημερινότητα μας. Η πλατφόρμα StudentsWeb δεν έχει mobile based έκδοση στα κινητά. Γεγονός που καθιστά την περιήγηση του Ιστότοπου δύσκολη.

Στόχος της πτυχιακής εργασίας είναι η ανάπτυξη ενός προγράμματος στην πλατφόρμα Android όπου θα είναι σε θέση (α) να ανταπεξέρχεται σε όλες τις ανάγκες του φοιτητή όσον αφορά την ηλεκτρονική γραμματεία (β) εμβαθύνοντας στα αντίστοιχα Α.Ε./Τ.Ε.Ι, (γ) δίνοντας του πρόσβαση στην πλατφόρμα StudentsWeb από το κινητό του καθώς και από μια πληθώρα λειτουργιών που δεν υπάρχουν στο StudentsWeb.

Λέξεις Κλειδιά

Εφαρμογή, Android, Πρόσβαση

Abstract

As students technology is for the good in our lives. Mobiles are an important part of our everyday life. The StudentsWeb platform does not support a mobile based version. Which makes browsing the Website extremely difficult.

This thesis's aim, is to develop a program in the Android platform that will be able (a) to cope with all the needs of the student regarding e-secretariat (b) deepening to the corresponding A.E.I / T.E.I, (c) allowing access to platform StudentsWeb from the mobile and a plurality of functions not found in StudentsWeb.

Keywords

Mobile, Platform, Access

Περιεχόμενα

Περίληψη	6
Abstract.....	7
Περιεχόμενα.....	8
Κεφάλαιο 1	11
Εισαγωγή	11
Εισαγωγή στο Android	12
Η αρχιτεκτονική του Android	13
Linux Kernel.....	14
Βιβλιοθήκες.....	14
Android Runtime	15
Application Framework	15
Applications	16
Βασικά στοιχεία εφαρμογών	16
Activities / Δραστηριότητες.....	16
Services / Υπηρεσίες.....	17
Broadcast Receivers	17
Content Providers	17
Additional Components.....	18
Κεφάλαιο 2	19
Σχεδιασμός	19
Λογική.....	20
Κεφάλαιο 3	26
Εισαγωγικά	26
Λεπτομέρειες Υλοποίησης.....	27
Άντληση δεδομένων από την πλατφόρμα	31
Ρυθμίσεις.....	35
Ελέγχοντας για αλλαγές της βαθμολογίας.....	38
Φτιάχνοντας το δικό μας API	39
Εισαγωγικά	39
Λεπτομέρειες	39
StudentsWeb και Doze Mode	44
Τι είναι το Doze Mode;	44
Υποστήριξη Doze Mode στο StudentsWeb	45
Κεφάλαιο 4	46

Βιβλιοθήκες.....	46
Jsoup	46
Google Analytics	47
Firebase	47
Fabric.io.....	50
Happy Apps	51
MPAndroidChart	52
Material Drawer.....	53
CircleImageView	54
Android About Page	54
Show/Hide Password EditText	54
Glide.....	55
Κεφάλαιο 5.....	56
Στοιχεία Χρήσης και Αξιολόγηση της Εφαρμογής	56
Η αρχή, η εξέλιξη, και το τώρα.....	57
Alpha Testing.....	59
Google Play Store.....	59
Google Play Developer Console	60
Υπογράφοντας την εφαρμογή.....	63
Κεφάλαιο 6.....	64
Περιορισμοί και προβλήματα	64
Κεφάλαιο 7.....	65
Συμπεράσματα και Μελλοντικές επεκτάσεις.....	65
Παράρτημα Α	68
Η εφαρμογή στο «Μικροσκόπιο»	68
build.gradle	68
/app/manifests	69
/app/res/	69
/app/assets.....	74
/app/java/.....	74
Παράρτημα Β	105
Περί βιβλιοθηκών	105
Jsoup	105
Google Analytics	106
Firebase	106

Fabric.io.....	107
Happy Apps	107
MPAndroidChart	108
Material Drawer.....	108
CircleImageView	108
Android About Page	108
Show/Hide Password EditText	108
Glide.....	108
Συντομογραφίες – Αρκτικόλεξα – Ακρωνύμια	110
Απόδοση ξενόγλωσσων όρων	111

Κεφάλαιο 1

Στο κεφάλαιο αυτό γίνεται μια εισαγωγή στην εφαρμογή μας, στον ανταγωνισμό που υπάρχει, αλλά και μια εισαγωγή στο οικοσύστημα του Android.

Εισαγωγή

Σε μία εποχή όπου πλέον όλα μπορούμε να τα κάνουμε από τα κινητά μας τηλέφωνα, η εφαρμογή StudentsWeb έρχεται για να βοηθήσει τον φοιτητή να έχει μία εφαρμογή για ότι έχει να κάνει με την σχολή του (ηλεκτρονική γραμματεία).

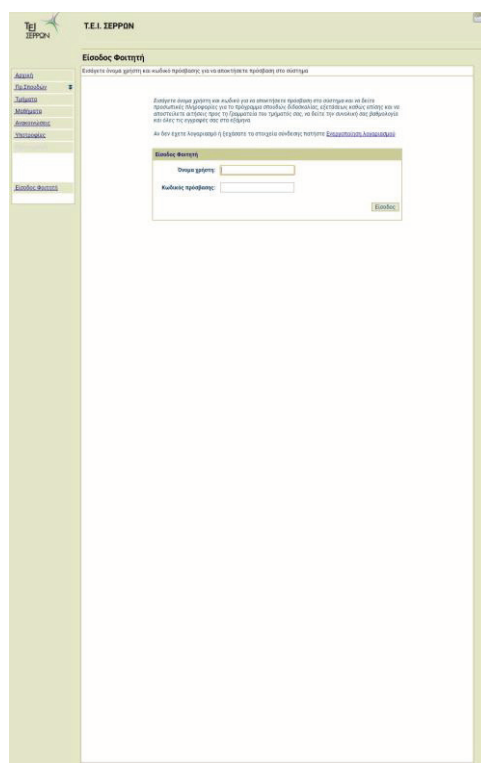
Ο ανταγωνισμός και αντίστοιχες εφαρμογές στο Play Store:

Υπάρχει άλλη μια αντίστοιχη ελληνική εφαρμογή μέσω της οποίας ο φοιτητής μπορεί να ελέγξει την βαθμολογία του, η οποία προσφέρει μόνο προβολή βαθμολογίας, στατιστικών και έλεγχο βαθμολογίας ανά προκαθορισμένο αριθμό λεπτών. Στο εξωτερικό υπάρχουν αντίστοιχες εφαρμογές για ιδρύματα. Η διαφορά με το εξωτερικό είναι, ότι εκείνες οι εφαρμογές είναι (τις περισσότερες φορές) σχεδιασμένες από το ίδιο το τμήμα επιτρέποντας πρόσβαση στο API/βάση δεδομένων τους, κάνοντας διαθέσιμη την εγγραφή σε μαθήματα, ή στο βαθμολόγιο κ.ά..

Η εφαρμογής μας σε σχέση με την πλατφόρμα και τον ανταγωνισμό:

Ο ιστότοπος της ηλεκτρονικής γραμματείας κάθε σχολής είναι τις περισσότερες φορές δυσανάγνωστος από τα κινητά τηλέφωνα (συγκεκριμένα εκείνος της cardisoft) διότι δεν υποστηρίζουν mobile based φυλλομετρητές (browsers). Η εφαρμογή μας, πέρα από το ότι έχει ένα πολύ ωραίο ματεριαλιστικό σχεδιασμό όπου είναι φιλικό προς τον χρήστη, παρέχει στον φοιτητή οτιδήποτε έχει να κάνει με την σχολή του. Από τα απλά πράγματα όπως φέρ' ειπείν να δει τα στοιχεία του, την βαθμολογία του, μέχρι και να είναι σε θέση να κατεβάζει/ανεβάζει σημειώσεις, και να συζητά με συμφοιτητές. Η εφαρμογή μας εμβαθύνει προσφέροντας πολλά παραπάνω από το ότι «δείχνει» ο ιστότοπος της ηλεκτρονικής γραμματείας καθώς δίνεται δυνατότητα να ειδοποιείται ο φοιτητής όταν βγαίνει κάποιος καινούριος βαθμός ενός μαθήματος επιτρέποντας του να επιλέγει εκείνος κάθε πότε θα γίνεται έλεγχος της βαθμολογίας. Επίσης του επιτρέπει να ανεβάζει σημειώσεις συγκεκριμένα για το τμήμα του, να κατεβάζει σημειώσεις φιλτραρισμένες για το τμήμα του, να συμμετέχει σε συζήτηση να βλέπει στατιστικά, το πρόγραμμα σπουδών του, να κατεβάζει τα αποδεικτικά των εγγραφών του κ.ά.. Έτσι έχοντας όλες αυτές τις δυνατότητες στο κινητό του τηλέφωνο και σε ένα περιβάλλον εφαρμογής το οποίο είναι μινιμαλιστικό,

ματεριαλιστικό και ευχάριστο, ο φοιτητής δεν έχει ανάγκη την άχαρη πλατφόρμα στο διαδίκτυο που στον φυλλομετρητή του κινητού είναι δυσανάγνωστη. Η εφαρμογή αυτή δημιουργήθηκε από φοιτητή για τον φοιτητή.



Εικόνα 1: Απεικόνιση της πλατφόρμας της ηλεκτρονικής γραμματείας σε φυλλομετρητή κινητού.

Παρόλο που η ανταγωνιστική εφαρμογή έχει αρκετά μεγάλη βάση χρηστών (για τα δεδομένα της Ελλάδας), μέσω διαφημίσεων (AdWords), promotion μέσω Facebook και άλλων κοινωνικών δικτύων, και μέσω των πολλών επιπλέον χαρακτηριστικών που διαθέτει η εφαρμογή μας υπολογίζεται ότι θα είμαστε σε θέση να αποκτήσουμε έναν σεβαστό αριθμό χρηστών.

Η εφαρμογή παρόλο που θα περιέχει και «συζήτηση» μεταξύ φοιτητών, δεν είναι από τις εφαρμογές όπου ο φοιτητής χρησιμοποιεί «καθημερινά». Πρέπει να είναι εκεί όταν και αν την χρειαστεί ο φοιτητής.

Εισαγωγή στο Android

Πλατφόρμα Android

Το Android είναι λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών

λογισμικού ανεπτυγμένων από την Google. Το Android είναι κατά κύριο λόγο σχεδιασμένο για συσκευές με οθόνη αφής, όπως τα έξυπνα τηλέφωνα και τα tablet, με διαφορετικό περιβάλλον χρήσης για τηλεοράσεις (Android TV), αυτοκίνητα (Android Auto) και ρολόγια χειρός (Android Wear). Παρόλο που έχει αναπτυχθεί για συσκευές με οθόνη αφής, έχει χρησιμοποιηθεί σε κονσόλες παιχνιδιών, ψηφιακές φωτογραφικές μηχανές, συνηθισμένους Η/Υ (π.χ. το HP Slate 21) και σε άλλες ηλεκτρονικές συσκευές.

Το Android είναι το πιο ευρέως διαδεδομένο λογισμικό στον κόσμο. Οι συσκευές με λειτουργικό σύστημα Android έχουν περισσότερες πωλήσεις από όλες τις συσκευές Windows, iOS και Mac OS X μαζί.

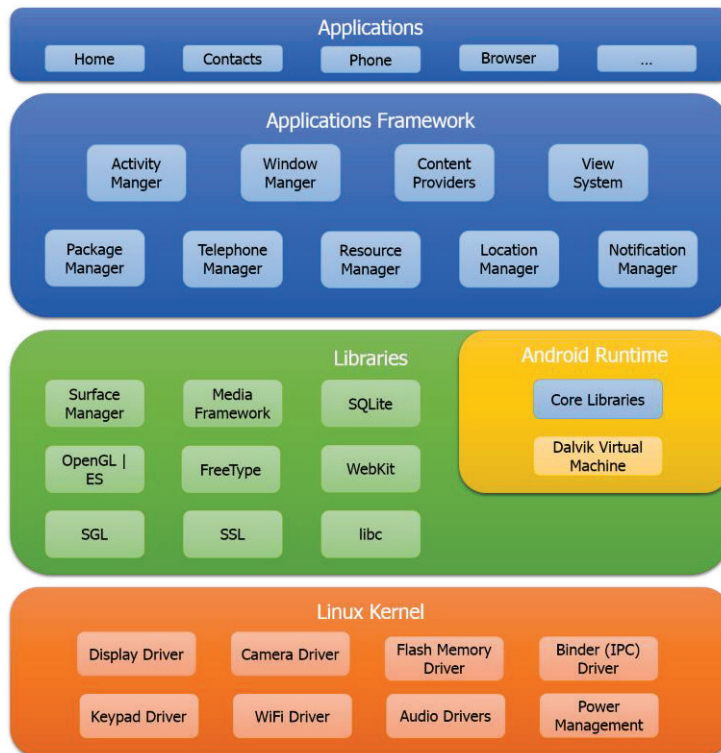
Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance, μιας κοινοπραξίας 48 τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής hardware, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη 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.

Η θέση του Android σε σχέση με τον ανταγωνισμό

Στο 2^ο τετράμηνο του 2016 το Android ήταν το λειτουργικό με το περισσότερο μερίδιο αγοράς το οποίο άγγιξε το 87,6%, έπειτα ακολούθησε το λογισμικό της Apple iOS με ποσοστό 11,7%, τα Windows Phone με ποσοστό 0,4% και άλλα λογισμικά όπως Taizen και BlackBerry με 0,3% [1].

Η αρχιτεκτονική του Android

Το λειτουργικό σύστημα Android βασίζεται σε μια «στοίβα» από στοιχεία (components) λογισμικού που χωρίζονται σε πέντε τμήματα και τέσσερις «στρώσεις». Παρακάτω μπορούμε να δούμε το διάγραμμα αρχιτεκτονικής [2].



Εικόνα 2: Διάγραμμα αρχιτεκτονικής του Android.

Linux Kernel

Στο κάτω-κάτω επίπεδο των «στρώσεων» έχουμε το Linux. Αυτό παρέχει ένα αφαιρετικό επίπεδο για το υλικό που χρησιμοποιείται στην συσκευή μας και παρέχει όλους τους βασικούς οδηγούς (drivers), όπως για παράδειγμα για την κάμερα, την οθόνη, την αφή, την δουλειά για την διαχείριση των δικτύων κ.ά. .

Βιβλιοθήκες

Στο επόμενο επίπεδο είναι μια «γκάμα» βιβλιοθηκών συμπεριλαμβανομένου βιβλιοθηκών ανοιχτού κώδικα, οι οποίες είναι υπεύθυνες για την ασφάλεια του ίντερνετ, την αποθήκευση δεδομένων εφαρμογών κ.ά. .

Android Βιβλιοθήκες

Αυτή η κατηγορία συμπεριλαμβάνει τις Java βιβλιοθήκες που είναι συγκεκριμένα για την ανάπτυξη του Android λειτουργικού. Μερικές από αυτές τις βασικές βιβλιοθήκες είναι:

- `android.app` – Παρέχει πρόσβαση στο μοντέλο εφαρμογής και είναι η βάση για όλες τις Android εφαρμογές
- `android.content` – Διευκολύνει την πρόσβαση του περιεχομένου
- `android.database` – Περιλαμβάνει SQLite διαχείριση βάσης και χρησιμοποιείται για να προβληθούν δεδομένα από εφαρμογές
- `android.opengl` – Ένα API σε γλώσσα Java για το OpenGL ES 3D
- `android.os` – Παρέχει εφαρμογές με πρόσβαση σε υπηρεσίες του λειτουργικού όπως μηνύματα, υπηρεσίες συστήματος κ.ά.
- `android.text` – Χρησιμοποιείται για να προβάσουμε και να χειριστούμε κείμενο στην συσκευή
- `android.view` – Χάριν σε αυτή γίνεται η δόμηση των διεπαφών του χρήστη
- `android.widget` – Μια πλούσια συλλογή από υπάρχοντα στοιχεία (όπως κουμπιά, λίστες, απεικονίσεις κ.ά.)

Android Runtime

Το τρίτο «τμήμα» της αρχιτεκτονικής, που βρίσκεται στο ίδιο επίπεδο με της βιβλιοθήκες, παρέχει ένα σημαντικό εργαλείο το οποίο ονομάζεται Dalvik Virtual Machine που ουσιαστικά είναι σαν ένα εικονικό μηχάνημα στην Java σχεδιασμένο και βελτιστοποιημένο για την πλατφόρμα Android.

Το Dalvik VM χρησιμοποιεί τα βασικά στοιχεία του Linux όπως διαχείριση μνήμης, πολλαπλά Threads εγγενώς στην γλώσσα προγραμματισμού Java. Το Dalvik VM επιτρέπει σε κάθε διεργασία να τρέχει το δικό της Dalvik VM

Παρέχει επίσης μια πληθώρα από βασικές βιβλιοθήκες που επιτρέπουν στους προγραμματιστές να γράψουν εφαρμογές χρησιμοποιώντας Java

Application Framework

Το Application Framework παρέχει «υπηρεσίες» σε μορφή Java, που οι προγραμματιστές είναι σε θέση να χρησιμοποιήσουν στις εφαρμογές τους.

Το Android Framework περιλαμβάνει:

- Activity Manager – Ελέγχει όλες τις πτυχές του κύκλου ζωής της εφαρμογής
- Content Providers – Επιτρέπει την κοινή χρήση δεδομένων μεταξύ άλλων εφαρμογών
- Resource Manager – Παρέχει πρόσβαση σε πόρους που δεν περιέχονται σε κώδικα, όπως χρώματα, κείμενα, διεπαφές
- Notifications Manager – Επιτρέπει την προβολή ειδοποιήσεων
- View System – Χρησιμοποιείται για την δημιουργία διεπαφών χρήστη

Applications

Όλες οι εφαρμογές Android είναι στην ανώτατη «στρώση». Η «εφαρμογή» μας βρίσκεται σε αυτό το σημείο.

Βασικά στοιχεία εφαρμογών

Τα βασικά στοιχεία εφαρμογών είναι τα δομικά στοιχεία μιας εφαρμογής Android. Αυτά τα στοιχεία δηλώνονται στο `AndroidManifest.xml` της εφαρμογής μας, και περιγράφει το κάθε στοιχείο και το πώς αλληλοεπιδρά με τα άλλα στοιχεία. Υπάρχουν τέσσερις κύριες συνιστώσες που μπορούν να χρησιμοποιηθούν και τις αναλύουμε παρακάτω.

Βασικό στοιχείο	Περιγραφή
Activities / Δραστηριότητες	Χειρίζονται το User Interface (UI) ώστε να αλληλοεπιδρά με τον χρήστη.
Services / Υπηρεσίες	Χειρίζονται στο παρασκήνιο ότι έχει να κάνει με την εφαρμογή.
Broadcast Receivers	Χειρίζονται την επικοινωνία μεταξύ του Android και των εφαρμογών
Content Providers	Χειρίζονται τα δεδομένα και θέματα που έχουν να κάνουν με την διαχείριση βάσεων δεδομένων

Activities / Δραστηριότητες

Μια δραστηριότητα αντιπροσωπεύει μια οθόνη με ένα User Interface (UI), όπου χάριν σε αυτή γίνονται όλες οι ενέργειες που κάνουμε στην οθόνη. Αν μια εφαρμογή έχει παραπάνω από μια δραστηριότητες τότε στο `AndroidManifest.xml` πρέπει να επιλέξουμε πια δραστηριότητα θα ανοίγει πρώτη μόλις ξεκινάμε την εφαρμογή μας.

Μια δραστηριότητα ενσωματώνεται σαν «υποκλάση» της κλάσης “Activity” και δηλώνεται έτσι:

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



```
}
```

Services / Υπηρεσίες

Τα Services «τρέχουν» στο παρασκήνιο κάνοντας διάφορες λειτουργίες (ανάλογα την εφαρμογή). Για παράδειγμα ένα Service μπορεί να παίζει μουσική στο παρασκήνιο ενώ ο χρήστης βρίσκεται σε άλλη εφαρμογή, ή να παίρνει δεδομένα από το ίντερνετ χωρίς να διακόπτει τον χρήστη χρησιμοποιώντας κάποια δραστηριότητα.

Ένα Service ενσωματώνεται σαν «υποκλάση» της κλάσης Service όπως παρακάτω:

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

Broadcast Receivers

Τα Broadcast Receivers απαντούν σε broadcast μηνύματα άλλων εφαρμογών ή από το σύστημα. Για παράδειγμα οι εφαρμογές είναι σε θέση να κάνουν κάποιο broadcast για να ενημερώσουν άλλες εφαρμογές ότι έχει γίνει κατέβαση δεδομένων για να χρησιμοποιηθεί και έτσι μέσω του broadcast receiver που «έλαβε» αυτό το broadcast να ξεκινήσει κάποια διαδικασία.

Ένα Broadcast Receiver ενσωματώνεται σαν «υποκλάση» της BroadcastReceiver κλάσης και κάθε μήνυμα σαν Intent Object

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

Content Providers

Ένα content provider παρέχει δεδομένα από μια εφαρμογή σε άλλες. Ο χειρισμός τέτοιων «αιτήσεων» γίνεται από την κλάση ContentResolver. Τα

δεδομένα μπορεί να είναι στα αρχεία του συστήματος, σε βάσεις δεδομένων είτε κάπου αλλού.

Ένας content provider ενσωματώνεται σαν «υποκλάση» της κλάσης ContentProvider και πρέπει να ενσωματώνει ένα στάνταρ API ώστε οι άλλες εφαρμογές να μπορούν να επικοινωνούν.

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

Additional Components

Υπάρχουν επίσης παραπάνω στοιχεία που χρησιμοποιούνται για την κατασκευή των προαναφερθέντων. Μερικά από αυτά είναι:

Στοιχεία	Περιγραφή
Fragments	Αντιπροσωπεύουν ένα τμήμα από το UI σε μια δραστηριότητα.
Views	Στοιχεία UI τα οποία απεικονίζονται στην οθόνη και περιλαμβάνουν κουμπιά, λίστες, φόρμες κ.ά.
Layouts	Ιεραρχίες που ελέγχουν την εμφάνιση.
Intents	«Μηνύματα» τα οποία συνδέουν μεταξύ τους «στοιχεία»/components.
Resources	Εξωτερικά στοιχεία όπως εικόνες, strings, διάφορες σταθερές κ.ά.
Manifest	Το αρχείο ρυθμίσεων για την εφαρμογή μας.

Κεφάλαιο 2

Στο κεφάλαιο αυτό παρουσιάζεται αναλυτικά ο σχεδιασμός και η λογική με βάση στην οποία δημιουργήθηκε η εφαρμογή StudentsWeb.

Σχεδιασμός

Η εφαρμογή StudentsWeb σχεδιάστηκε να είναι όσο το δυνατόν πιο κοντά στον ματεριαλιστικό σχεδιασμό του Android. Έγινε η προσπάθεια να χρησιμοποιηθούν τα νέα ματεριαλιστικά θέματα του Android (Μαύρο και Άσπρο θέμα), καθώς και τα νέα widgets και API's της Google.

Ο ματεριαλιστικός σχεδιασμός παρέχει νέα στυλ στην εφαρμογή μας. Με την χρήση νέων animations και widgets (στην εφαρμογή μας «κάρτες») σχεδιάζουμε για τον χρήστη ένα ευανάγνωστο και εύκολο στην χρήση περιβάλλον [3].

Σαν βασικά χρώματα και στα δυο θέματα, χρησιμοποιήθηκαν, στο άσπρο θέμα μπλε και άσπρο, και στο μαύρο, που δημιουργήθηκε κυρίως για τα άτομα που έχουν θέμα με τα μάτια, μπλε και μαύρο.

Ο λόγος που επιλέχθηκε το μπλε χρώμα δεν είναι τυχαίος. Πολλές εφαρμογές χρησιμοποιούν αυτό το χρώμα καθώς όπως λένε άρθρα και έρευνες, είναι το χρώμα της «Νιρβάνας». Το μπλε χρώμα που έχουν πολλά μέσα κοινωνικής δικτύωσης (Facebook, Twitter) και όχι μόνο, συμβολίζει άνεση, ουρανό, νερό, εμπιστοσύνη και ασφάλεια (π.χ. στολές αστυνομικών, περιπολικά κ.ά.). Ενώ άλλα χρώματα τείνουν να αποσπάνε την προσοχή από τον χρήστη, το μπλε είναι ένα χρώμα που «καλωσορίζει» τον χρήστη της εφαρμογής. Εμπνέει ένα κλίμα εμπιστοσύνης και σοβαρότητας ενώ συνάμα εξαφανίζεται στο παρασκήνιο επιτρέποντας στον χρήστη να αφοσιωθεί στις πληροφορίες που του απεικονίζονται στην οθόνη χωρίς να αποσπάται η προσοχή του [4].

Σκοπός της εφαρμογής μας όπως αναφέρθηκε και πιο πάνω είναι να είναι εκεί για τον φοιτητή όταν θα την χρειαστεί, εννοώντας πως ο φοιτητής με την «ηλεκτρονική γραμματεία» ασχολείται κυρίως σε περιόδους εξεταστικής ή δηλώσεων μαθημάτων. Η εφαρμογή σχεδιάστηκε να μην «ενοχλεί» τον χρήστη όταν αυτός δεν την χρειάζεται (χωρίς περιττές ειδοποιήσεις κ.ο.κ.). Επίσης δημιουργήθηκε για να εξυπηρετεί τους σκοπούς κυρίως της ηλεκτρονικής γραμματείας στο Τ.Ε.Ι. Δυτικής Ελλάδας αλλά γίνεται προσπάθεια να υποστηριχθούν όσα Α.Ε.Ι/Τ.Ε.Ι βασίζονται στην πλατφόρμα της cardisoft.

Λογική

Ο χρήστης καθώς εισέρχεται στην εφαρμογή, θα χρειαστεί να εισάγει το όνομα χρήστη και τον κωδικό του (αφού έχει επιλέξει το τμήμα του), καθώς επίσης έχει και την επιλογή να αποθηκεύσει τον κωδικό του και να κάνει αυτόματη επανασύνδεση ώστε να μην χρειαστεί να βάλει εκ νέου τα στοιχεία του την επόμενη φορά που θα ανοίξει την εφαρμογή. Στην οθόνη αυτή επίσης υπάρχει και το drawer(συρτάρι) όπου είναι το βασικό κομμάτι για να περιηγηθούμε στην εφαρμογή. Στην οθόνη «Είσοδος Χρήστη», το «συρτάρι» αυτό περιέχει κάποια πεδία που βοηθάνε τον φοιτητή (π.χ. το πρόγραμμα του εξαμήνου) χωρίς να χρειάζεται ο φοιτητής να κάνει είσοδο για να πάει στο συγκεκριμένο πεδίο (αρκεί να έχει επιλέξει την σχολή του).

Αφού ο φοιτητής πραγματοποιήσει επιτυχή είσοδο στην εφαρμογή, φορτώνονται τα στοιχεία του, και έπειτα μέσω του drawer είναι σε θέση να πλοηγηθεί στα αντίστοιχα στοιχεία που τον ενδιαφέρουν.

Τα πεδία που υπάρχουν (μέχρι την στιγμή που γράφεται η πτυχιακή) είναι τα εξής:

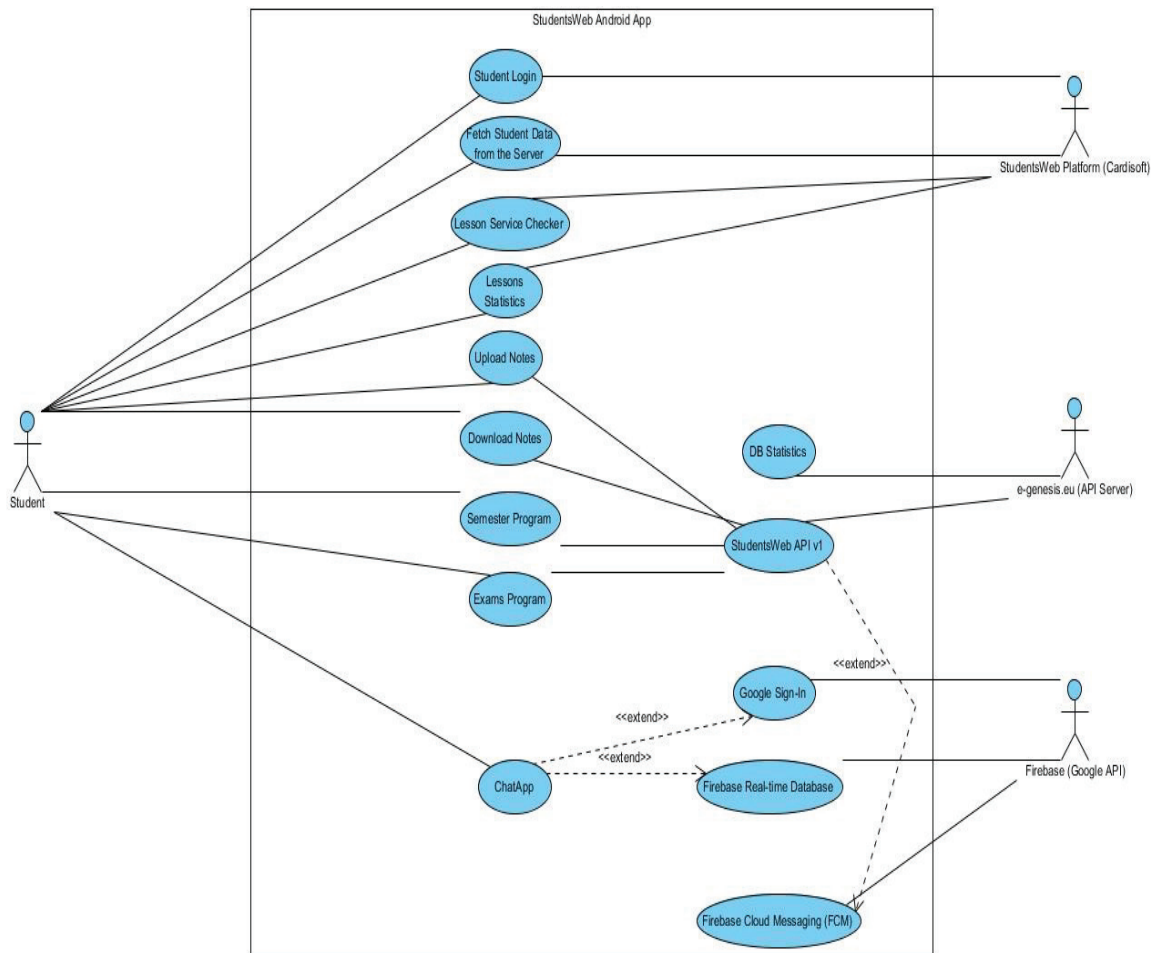
Για την οθόνη πριν την είσοδο του φοιτητή:

- Άνοιγμα του StudentsWeb στον Browser (φυλλομετρητή)
- Οδηγίες για να φτάσετε στο ΑΕΙ/ΤΕΙ
- Πρόγραμμα εαρινού/χειμερινού εξαμήνου 201χ-201x
- Επικοινωνία
- Κοινοποίηση

Για την κύρια οθόνη μετά την είσοδο του φοιτητή:

- Στοιχεία Φοιτητή
- Η δήλωση μου
- Τα εργαστήρια μου
- Αποτελέσματα
- Περασμένα Μαθήματα
- Βαθμολογίες (Αναλυτικά)
- Πρόγραμμα Σπουδών
- Στατιστικά
- Αποδεικτικά Ηλεκτρονικών Εγγραφών
- Ανεβάστε Σημειώσεις
- Κατεβάστε Σημειώσεις
- E-mail Καθηγητών
- Συχνές Ερωτήσεις για το ΑΕΙ/ΤΕΙ (FAQ)
- Άνοιγμα του StudentsWeb στον Browser (φυλλομετρητή)
- Πρόγραμμα εαρινού/χειμερινού εξαμήνου 201χ-201x

- Πρόγραμμα τελευταίας εξεταστικής
- Οδηγίες για να φτάσετε στο ΑΕΙ/ΤΕΙ
- Συζήτηση
- Κοινοποίηση
- Επικοινωνία
- Περί
- Αποσύνδεση



Εικόνα 3: Use case διάγραμμα της εφαρμογής.

Πιο συγκεκριμένα για κάθε δραστηριότητα:

- **Στοιχεία φοιτητή:** Σε αυτήν την δραστηριότητα ο φοιτητής βλέπει τις βασικές πληροφορίες του φοιτητικού του λογαριασμού. Εάν είναι περίοδος δηλώσεων θα του εμφανιστεί ανάλογη ειδοποίηση.
- **Η δήλωσή μου:** Σε αυτήν την δραστηριότητα ο φοιτητής μπορεί να δει την τρέχουσα δήλωσή του. Σε περίπτωση που δεν έχει κάνει δήλωση ή έχει στείλει κενή δήλωση εμφανίζεται το μήνυμα «Δεν υπάρχει τρέχουσα δήλωση».

- **Τα εργαστήρια μου:** Σε αυτήν την δραστηριότητα εμφανίζονται τα εργαστήρια στα οποία έχει εγγραφεί ο φοιτητής. Σε περίπτωση που δεν υπάρχουν εγγραφές σε εργαστήρια εμφανίζεται το μήνυμα «Δεν υπάρχουν δηλωμένα εργαστήρια. Ενώ αν η δήλωση είναι κενή εμφανίζεται το μήνυμα «Δεν υπάρχει τρέχουσα δήλωση».
- **Αποτελέσματα:** Εδώ εμφανίζονται τα «τελευταία» αποτελέσματα με την σειρά που είναι στις δηλώσεις του φοιτητή.
- **Περασμένα:** Εμφανίζονται μόνο τα μαθήματα των οποίων ο βαθμός είναι προβιβάσιμος (δηλαδή >5).
- **Βαθμολογίες (Αναλυτικά):** Στην δραστηριότητα αυτή εμφανίζονται αναλυτικά τα μαθήματα κάθε εξαμήνου και τα χαρακτηριστικά του κάθε μαθήματος, όπως βαθμός, τύπος κ.ά..
- **Οδηγός Σπουδών:** Στην δραστηριότητα αυτή εμφανίζονται ανά εξάμηνο τα μαθήματα στον οδηγό σπουδών του φοιτητή καθώς και τα χαρακτηριστικά τους.
- **Στατιστικά:** Στα στατιστικά έχουμε βάλει κάποια γραφήματα όπως BarCharts και PieCharts από την βιβλιοθήκη MPAndroidChart (βλ. βιβλιοθήκες), και απεικονίζουμε διάφορα στατιστικά, όπως τον Μ.Ο των εξαμήνων, το σύνολο περασμένων μαθημάτων ανά εξάμηνο κ.ά..
- **Αποδεικτικά Ηλ. Εγγραφών:** Αναλόγως αν το υποστηρίζει το αντίστοιχο Α.Ε./Τ.Ε.Ι, η εφαρμογή μας δίνει την δυνατότητα να κατεβάσουμε τα αποδεικτικά των δηλώσεων που έχουμε κάνει σε μορφή PDF στον φάκελο Downloads του κινητού μας.
- **Ανεβάστε Σημειώσεις:** Μας επιτρέπει να ανεβάσουμε σημειώσεις για την σχολή μας, ώστε να βοηθηθούν όλοι οι φοιτητές. Για λόγους ασφαλείας από την στιγμή που ανέβει το αρχείο στον server θα πρέπει να γίνει έγκριση από τον διαχειριστή του συστήματος ώστε π.χ. ο συμφοιτητής να είναι σε θέση να κατεβάσει τις σημειώσεις.
- **Κατεβάστε Σημειώσεις:** Σε αυτήν την δραστηριότητα μπορούμε να κατεβάσουμε στο κινητό μας τηλέφωνο τις σημειώσεις που έχουν ανεβάσει συμφοιτητές. Η κάθε σημείωση έχει το δικό της όνομα, ημερομηνία ανεβάσματος, το όνομα του χρήστη που την ανέβασε, και το εξάμηνο το οποίο αφορά.
- **Email Καθηγητών:** Στην δραστηριότητα αυτή απεικονίζονται όλα τα Email των καθηγητών του τμήματος που υπάρχουν. Υπάρχει επίσης κουμπί αποστολής email για κάθε καθηγητή το οποίο μας οδηγεί σε εφαρμογές όπως Gmail, Outlook κ.ά. (εφόσον είναι εγκατεστημένα) με

το πεδίο παραλήπτη να είναι ήδη συμπληρωμένο με το email του καθηγητή που επιλέξαμε.

- **Συχνές ερωτήσεις:** Εμφανίζονται τυχόν ερωτήσεις που μπορεί να έχει ο κάθε φοιτητής για το αντίστοιχο τμήμα.
- **Άνοιγμα του StudentsWeb στον Browser:** Ανοίγει στον φυλλομετρήτη του κινητού τον ιστότοπο της ηλεκτρονικής γραμματείας της σχολής.
- **Πρόγραμμα εαρινού εξαμήνου 2016-2017:** Δυναμικά από το API κατεβάζει το πρόγραμμα του εκάστοτε εξαμήνου ανάλογα με την σχολή που έχουμε προεπιλέξει.
- **Πρόγραμμα τελευταίας εξεταστικής:** Δυναμικά από το API κατεβάζει το πρόγραμμα της τελευταίας εξεταστικής του εκάστοτε τμήματος.
- **Οδηγίες για να φτάσετε στο A.E./T.E.I:** Ανοίγει είτε φυλλομετρητή είτε κάποια εφαρμογή με χάρτες και μας καθοδηγεί στο τμήμα μας.
- **Συζήτηση:** Όταν επισκεπτόμαστε για πρώτη φορά την δραστηριότητα αυτή μας εμφανίζεται ένα Google Sign-in κουμπί και αν το πατήσουμε μας παροτρύνει να συνδεθούμε με το Google Account μας. Ο λόγος που χρησιμοποιήθηκε το Google Account είναι για λόγους ασφαλείας και τυχόν κατάχρηση που μπορεί να γίνει. Σε περίπτωση, για παράδειγμα, που ο χρήστης δεν συμμορφώνεται μπορεί να γίνει μπλοκάρισμα του χρήστη και αφαίρεση των μηνυμάτων του.

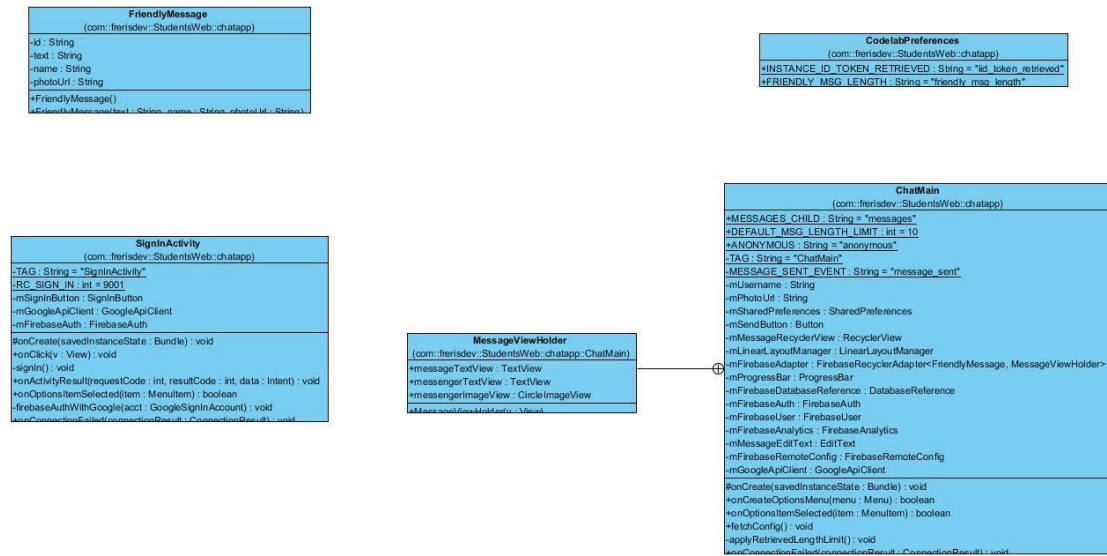
Από την στιγμή που συνδεόμαστε, μας εμφανίζεται μια πολύ απλή δραστηριότητα όπου ουσιαστικά μπορούμε να συζητήσουμε με άλλους συμφοιτητές της σχολής μας, για θέματα που αφορούν καθαρά και μόνο την σχολή.

Πάνω δεξιά στο 3-dot μενού έχουμε την έξοδο από την συζήτηση σε περίπτωση που δεν θέλουμε να λαμβάνουμε πλέον μέρος.

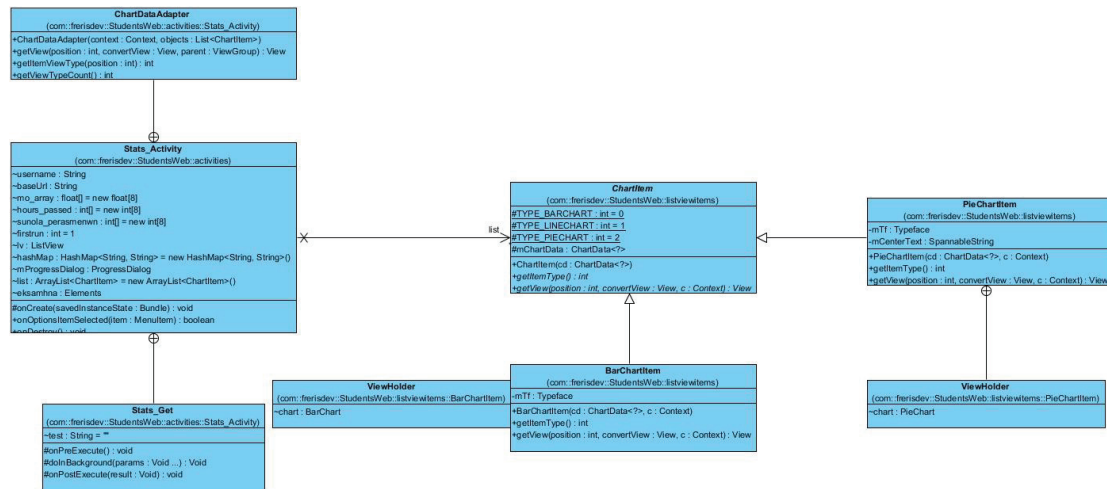
- **Κοινοποίηση:** Μπορούμε να μοιραστούμε την εφαρμογή StudentsWeb με τους φίλους μας σε περίπτωση που μας αρέσει.
- **Επικοινωνία:** Είναι μια απλή φόρμα την οποία μπορεί να χρησιμοποιήσει ο χρήστης για να στείλει τις απόψεις του, την βαθμολογία του, κάποια σφάλματα που τυχόν προέκυψαν, αλλά και προσθέσει τις οποίες δεν έχουν γίνει. Είναι αδύνατον να υπάρχουν π.χ. Email καθηγητών από όλα τα τμήματα. Έτσι ο φοιτητής σε περίπτωση που θέλει να προσθέσουμε κάτι μπορεί να επικοινωνήσει μαζί μας μέσω της φόρμας αυτής.

- **Περί:** Λίγα λόγια για την εφαρμογή, τον κύριο προγραμματιστή, τους παράγοντες που συνέβαλλαν στην ανάπτυξή της, τις βιβλιοθήκες κ.ά..
- **Αποσύνδεση:** Διαγράφονται όλα τα δεδομένα που έχει αποθηκεύσει ο χρήστης στην εφαρμογή, δηλαδή το «όνομα χρήστη», ο κωδικός, το τμήμα, κ.ά..

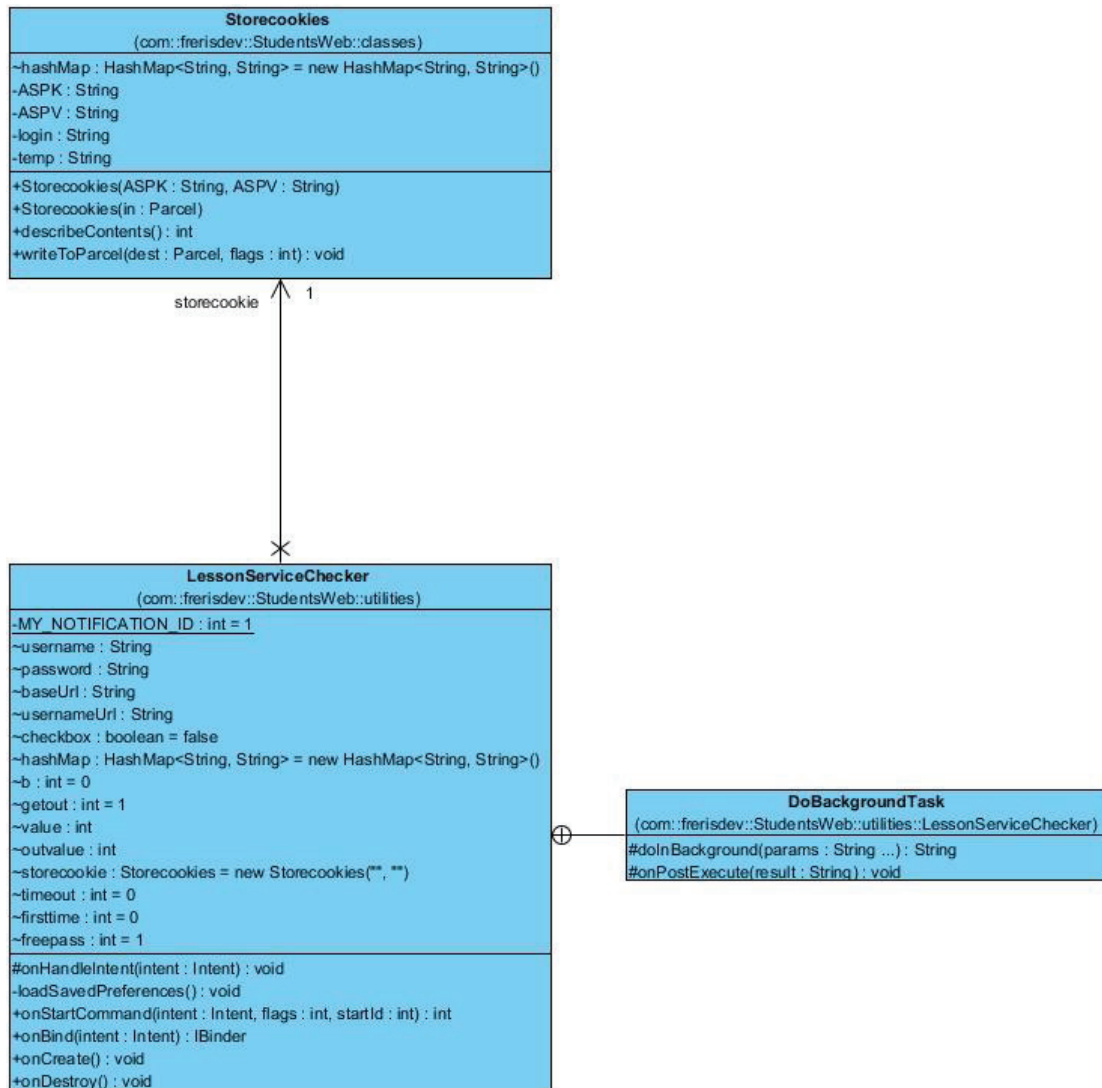
Παραθέτω ενδεικτικά αποσπάσματα από διαγράμματα κλάσεων



Εικόνα 4: Το διάγραμμα κλάσης της δραστηριότητας «Συζήτηση».



Εικόνα 5: Το διάγραμμα κλάσης της δραστηριότητας «Στατιστικά».



Εικόνα 6: Το διάγραμμα κλάσης της υπηρεσίας “LessonServiceChecker” που είναι υπεύθυνο για τον έλεγχο της βαθμολογίας στο παρασκήνιο.

Σημείωση: Λόγω του πολύ μεγάλου μεγέθους ολόκληρου του διαγράμματος κλάσης, ο ενδιαφερόμενος αναγνώστης παραπέμπεται να το αναγνώσει σε .png μορφή στο παραδοτέο ψηφιακό υλικό της εργασίας.

Κεφάλαιο 3

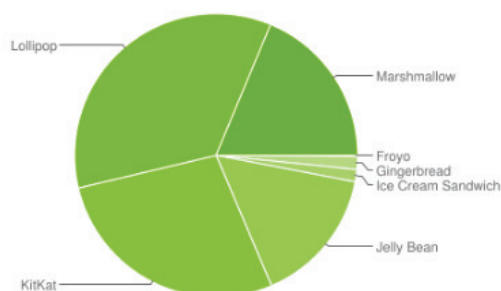
Στο κεφάλαιο αυτό αναφέρεται πως έγινε η υλοποίηση της εφαρμογής, καθώς και λεπτομέρειες υλοποίησης με βάση τον σχεδιασμό.

Εισαγωγικά

Η πτυχιακή αυτή αναπτύχθηκε με το Android Studio (έκδοση 2.0-2.1). Αφορά την πλατφόρμα Android και αναπτύχθηκε με σκοπό την υποστήριξη των εκδόσεων Android >4.0 (Ice Cream Sandwich) μέχρι την τελευταία 7.0-7.1 (Nougat). Πέρα από τα κινητά τηλέφωνα Android, υποστηρίζονται επίσης και Tablets.

Με αυτό το μεγάλο «εύρος» υποστήριξης καλύπτονται «σχεδόν» όλες οι συσκευές Android (98,4%), με τελευταία καταμέτρηση στις 5 Σεπτεμβρίου του 2016 [5].

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	1.5%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.4%
4.1.x	Jelly Bean	16	5.6%
4.2.x		17	7.7%
4.3		18	2.3%
4.4	KitKat	19	27.7%
5.0	Lollipop	21	13.1%
5.1		22	21.9%
6.0	Marshmallow	23	18.7%



*Data collected during a 7-day period ending on September 5, 2016.
Any versions with less than 0.1% distribution are not shown.*

Εικόνα 7: Κατακερματισμός εκδόσεων Android.

Χρησιμοποιήθηκε επίσης μια πληθώρα βιβλιοθηκών που μας βοήθησαν στην ευκολότερη υλοποίηση αυτής της εφαρμογής. Πιο κάτω αναφέρονται αναλυτικά οι βιβλιοθήκες καθώς και ο σκοπός που εξυπηρέτησαν στην εφαρμογή.

Λεπτομέρειες Υλοποίησης

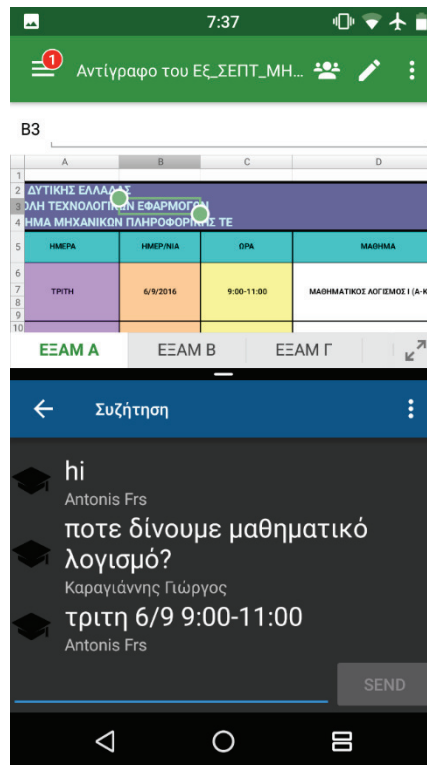
Το πακέτο της εφαρμογής έχει όνομα: “com.frerisdev.StudentsWeb”. Η εγκατάσταση της εφαρμογής μπορεί να γίνει μόνο στην εσωτερική μνήμη του κινητού για αποφυγή τυχόν προβλημάτων, το μέγεθος της εφαρμογής είναι περίπου 6Mb οπότε είναι σχεδόν «απίθανο» να υπάρξει πρόβλημα σε περίπτωση που ο χρήστης έχει συσκευή με λίγο αποθηκευτικό χώρο.

Η ελάχιστη έκδοση Android που απαιτείται είναι η «15» (4.0 Ice Cream Sandwich). Το TargetSdk είναι το «23» (6.0 Marshmallow). Παρόλα αυτά η εφαρμογή δουλεύει απροβλημάτιστα στην έκδοση «24» (7.0-7.1 Nougat).

Από την εφαρμογή απαιτούνται δικαιώματα Internet, αποθηκευτικού χώρου, και Wakelock διότι:

- Η εφαρμογή χρειάζεται να συνδεθεί στο Ίντερνετ για να λάβει δεδομένα από τον ιστότοπο.
- Ο αποθηκευτικός χώρος χρειάζεται ώστε να γίνει αποθήκευση τυχόν σημειώσεων, PDF αλλά και για να ανεβάσουμε ή κατεβάσουμε σημειώσεις.
- Το Wakelock μας χρειάζεται ώστε η εφαρμογή να είναι σε θέση να «ξυπνήσει» το κινητό μας τηλέφωνο όταν λάβουμε μια ειδοποίηση.

Έχει προστεθεί επίσης λειτουργία πολλαπλών παραθύρων. Μια λειτουργία που επιτρέπει να «τρέχουμε» στην οθόνη του κινητού μας παραπάνω από μία εφαρμογές (εάν και εφόσον υποστηρίζεται από το κινητό). Για όσα κινητά υποστηρίζουν αυτήν την λειτουργία, η εφαρμογή StudentsWeb είναι σε θέση να τρέχει παράλληλα με άλλες εφαρμογές.



Εικόνα 8: Παράδειγμα χρήσης πολλαπλών παραθύρων.

Επειδή κάθε Α.Ε./Τ.Ε.Ι είναι ίσως λίγο διαφορετικό από το άλλο και επειδή δεν παίρνουμε τα δεδομένα από κάποιο API αλλά από την κανονική σελίδα, πρέπει να είμαστε προσεκτικοί στο «που» και «πώς» θα «χτυπήσει» το κινητό μας τηλέφωνο την αντίστοιχη ιστοσελίδα. Για αυτό καθώς αρχίζει η ταυτοποίηση, το URL του φοιτητή χωρίζεται σε κομμάτια για την ευκολότερη παραμετροποίηση. Για παράδειγμα μια τυπική «κατασκευή» URL μέσα στην εφαρμογή γίνεται ως εξής :

- Το BaseURL: <https://studentsweb.teimes.gr/unistudent/>
- Το LoginURL: "login.asp?"
- Το UsernameURL: "userName" (σε πολλά τμήματα αυτό είναι "userName1" κ.ο.κ.)
- Το Username: Το όνομα χρήστη του φοιτητή
- Το Pwd: Ο κωδικός του φοιτητή

Εφόσον έχει «κατασκευαστεί» ο σύνδεσμος, είμαστε σε θέση να εισέλθουμε στον Ιστότοπο. Το Connection Timeout της σύνδεσης έχει οριστεί στα 45 δευτερόλεπτα (κυρίως για τις αργές συνδέσεις) και γίνεται μέσω της βιβλιοθήκης jsoup (βλ. βιβλιοθήκες).

ProgressDialogs εμφανίζονται σε όλο το μήκος της εφαρμογής, ενημερώνοντας τον χρήστη ότι γίνεται προσπάθεια ανάκτησης των στοιχείων του από το StudentsWeb. Ο προσανατολισμός της οθόνης προσωρινά κλειδώνει μέχρι να απομακρυνθούν τα ProgressDialog ώστε να μην υπάρχουν απρόσμενα κλεισίματα, διότι κάθε φορά που περιστρέφουμε την

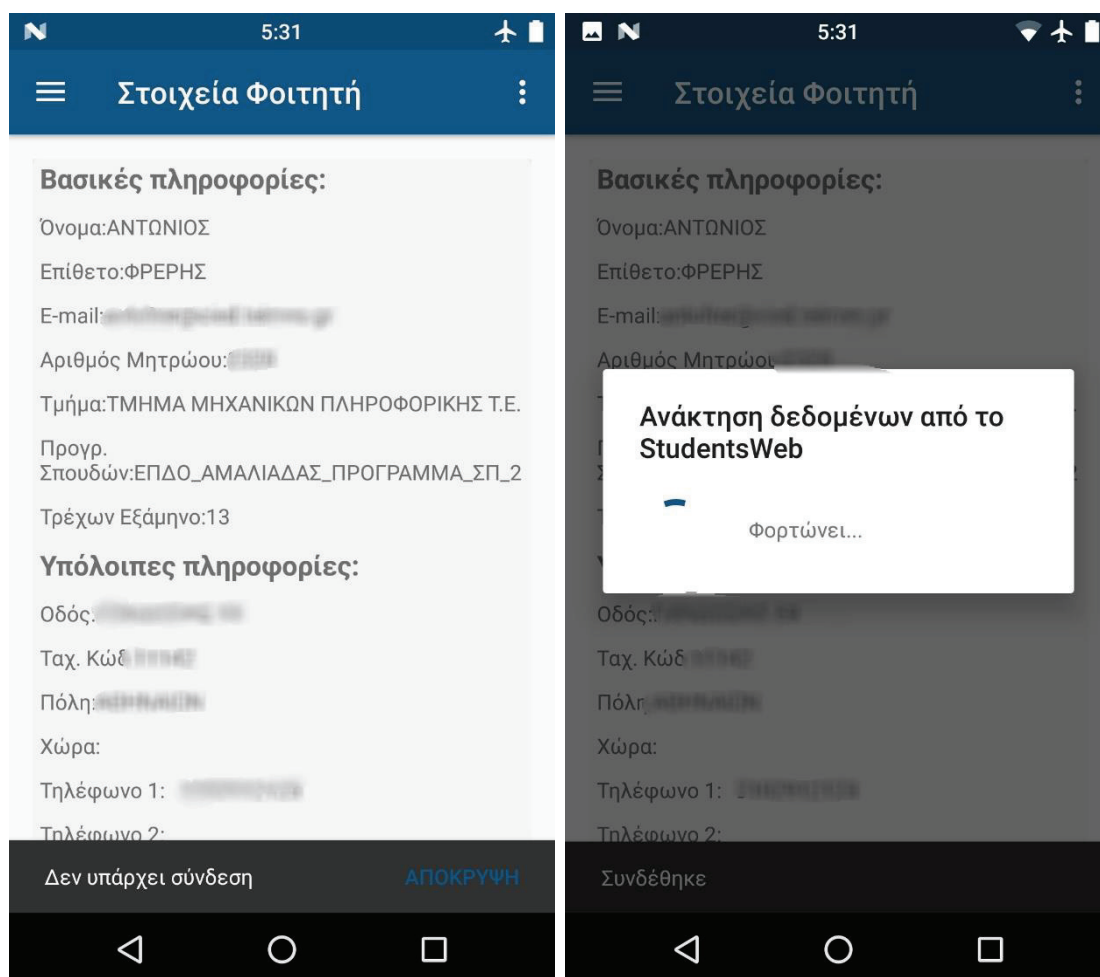
οθόνη και αλλάζει ο προσανατολισμός της, η εφαρμογή ουσιαστικά «καταστρέφει» την υπάρχουσα «δραστηριότητα»/Activity και την δημιουργεί εκ νέου.

Σημείωση: Αυτό συμβαίνει λόγω του Android που σε περίπτωση περιστροφής της οθόνης γίνεται «επανεκκίνηση» της δραστηριότητας στην οποία βρισκόμαστε.

Εφόσον τα στοιχεία μας επαληθευτούν, αποθηκεύουμε το “Cookie” το οποίο «λήγει» μετά από 15 λεπτά αδράνειας, και παίρνουμε τα «Στοιχεία φοιτητή» (studentMain.asp). Σε περίπτωση που υπάρξει πρόβλημα από πλευράς του ιστότοπου StudentsWeb server είτε πρόβλημα σύνδεσης του κινητού μας τηλεφώνου, ή έχουμε εισάγει λάθος στοιχεία, ο χρήστης ενημερώνεται με αντίστοιχο μήνυμα μέσω κάποιου Toast ή με Snackbar.

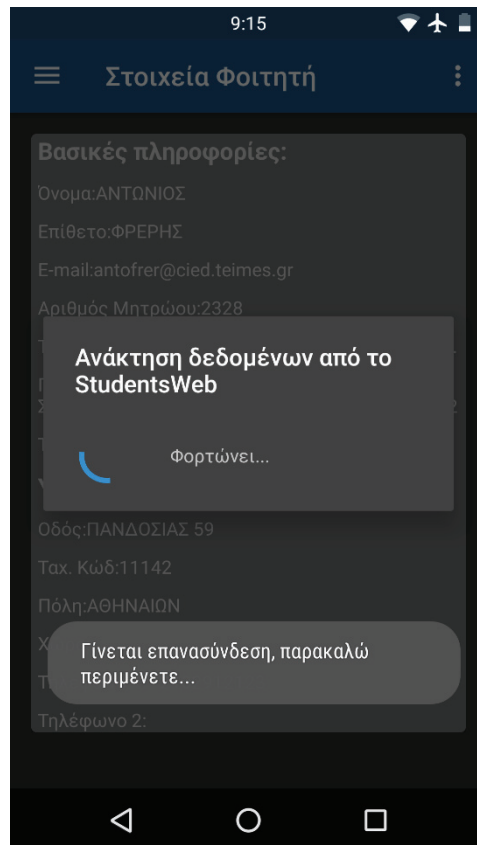
Εφόσον πλέον είμαστε στην «κεντρική δραστηριότητα»/MainActivity της εφαρμογής, φορτώνονται αυτομάτως τα «Στοιχεία Φοιτητή».

Ένα Thread ελέγχει κάθε 1-2 δευτερόλεπτα εάν υπάρχει σύνδεση στο διαδίκτυο. Αυτό γίνεται μόνο όταν η εφαρμογή StudentsWeb είναι στο προσκήνιο. Σε περίπτωση που δεν υπάρχει διαδίκτυο ένα ειδικό Snackbar μας ενημερώνει ότι έχουμε αποσυνδεθεί από το Ίντερνετ. Το Thread πέρα από τον έλεγχο που κάνει, είναι σε θέση να κάνει και «ανανέωση» σε περίπτωση που ξαναγίνει σύνδεση στο Ίντερνετ.



Εικόνα 9: Snackbar που μας ειδοποιεί για την κατάσταση του Ίντερνετ, και πραγματοποιεί αυτόματη ανανέωση μόλις εντοπιστεί Ίντερνετ.

Υπάρχουν επίσης, στην onPause() και στην onResume() της “MainActivity” δραστηριότητας μεταβλητές όπου «μετράνε» το πόσα λεπτά αδράνειας έχουν περάσει από την τελευταία ενέργεια του χρήστη έτσι ώστε να γίνει αυτόματη ανανέωση του cookie. Σε αντίθετη περίπτωση ο χρήστης δεν θα μπορούσε να έχει τις τελευταίες ενημερώσεις σε ότι αφορά τον λογαριασμό του διότι το cookie θα είχε λήξει από πλευράς του server της πλατφόρμας.



Εικόνα 10: Επανασύνδεση μετά την «λήξη» του cookie.

Εφόσον γίνει επιτυχημένη είσοδος (βγαίνει το ανάλογο Toast), φορτώνονται τα στοιχεία του φοιτητή και από εκεί και πέρα είμαστε έτοιμοι να πλοηγηθούμε στην εφαρμογή. Πάνω δεξιά υπάρχει το 3-dot μενού, που αν το πατήσουμε εμφανίζονται σαν επιλογή οι ρυθμίσεις (βλ. ρυθμίσεις). Αριστερά οι 3 παράλληλες γραμμές ανοίγουν το drawer μας, είτε πατώντας το κουμπί αυτό, είτε κάνοντας slide από την αριστερή πλευρά της οθόνης προς τα δεξιά.

Ανοίγοντας το drawer βλέπουμε όλες τις δραστηριότητες της εφαρμογής που αναφέραμε και στο στάδιο του σχεδιασμού. Θα αναφερθούμε σε αυτές εκτενέστερα.

Σε όλη την εφαρμογή έχουν χρησιμοποιηθεί Android RecyclerView και CardView. Το RecyclerView είναι μια πιο «εξειδικευμένη» ListView με

βελτιωμένη απόδοση και άλλα προτερήματα. Χρησιμοποιώντας RecyclerView και CardView μπορούμε να φτιάξουμε πολύ εύκολα λίστες και πλέγματα σε σχέση με ένα απλό ListView. Έτσι έχουμε ταυτόχρονα καλύτερη απόδοση και ένα πιο «όμορφο» design βασισμένο στα «πρότυπα» του ματεριαλιστικού design του Android.

Άντληση δεδομένων από την πλατφόρμα

Για να πάρουμε τα δεδομένα από την πλατφόρμα StudentsWeb, επειδή δεν έχουμε πρόσβαση σε κάποιο API, αυτό που κάνουμε είναι ότι μέσω της jsoup (βλ. βιβλιοθήκες) πραγματοποιούμε είσοδο στην πλατφόρμα, ταυτοποιώντας τα στοιχεία μας μέσω της jsoup και αποθηκεύοντας το cookie που μας δίνεται.

Πιο συγκεκριμένα, εφόσον έχουμε κατασκευάσει το URL μας το οποίο είναι τύπου :

[https://studentsweb.teimes.gr/unistudent/login.asp?userName="όνωμαχρήστη"&pwd="κωδικός"](https://studentsweb.teimes.gr/unistudent/login.asp?userName=)

(όπου «όνωμαχρήστη» και «κωδικός» τα στοιχεία του χρήστη), είμαστε σε θέση να στείλουμε τα στοιχεία μας προς την πλατφόρμα για ταυτοποίηση μέσω της εντολής:

```
Connection.Response response =  
Jsoup.connect(url).timeout(45000).method(Connection.Method.POST).execute();
```

Κάνοντας GET/POST στην παραπάνω ιστοσελίδα, εκείνη μας «απαντάει» με κάποια συγκεκριμένα cookies. Αποθηκεύουμε αυτά τα cookies ώστε να χρησιμοποιήσουμε τα ίδια καθώς κάνουμε περιήγηση στην εφαρμογή, και έπειτα ελέγχουμε εάν ο χρήστης έκανε επιτυχή είσοδο, ή όχι. Αυτό το ελέγχουμε κατά αυτόν τον τρόπο:

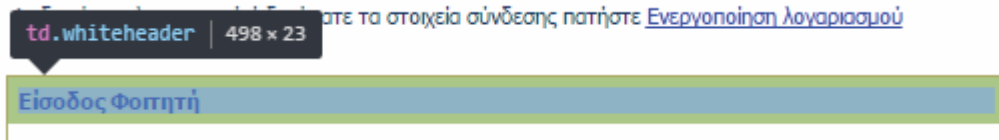
```
Elements Result = doc.select("table tbody tr td#main div form#login table tbody tr td  
table.tableborderColorOnly tbody tr td table.tableCell tbody tr td.whiteheader b");  
Elements Res_503 = doc.select("title");  
if (Result.text().toString().equals("Είσοδος Φοιτητή") ||  
Result.text().toString().equals("Σύντομη περιγραφή") ||  
Res_503.text().toString().equals("Service Unavailable")) {  
    res = 1;  
}
```

Μέσω του select που κάνουμε, γίνεται προσπάθεια να ανιχνευθεί αν έχει γίνει επιτυχής είσοδος. Σε περίπτωση που δεν έχει γίνει επιτυχής είσοδος, το

κείμενο του:

```
"table tbody tr td#main div form#login table tbody tr td table.tableborderColorOnly tbody tr td table.tableCell tbody tr td.whiteheader b"
```

Θα πρέπει να εμφανίζει το κείμενο «Είσοδος φοιτητή».



Εικόνα 11: Το `Result.text().toString()` μας επιστρέφει το κείμενο του πεδίου `td.whiteheader`.

Σε περίπτωση που αυτό το κείμενο αντιστοιχεί στους ελέγχους που κάνουμε, τότε δεν έχει γίνει επιτυχής είσοδος χρήστη

Σημείωση: Κάνουμε και έλεγχο για το αν έχει «πέσει» ο ιστότοπος ή όχι γιατί σε περιόδους δηλώσεων εργαστηρίων κλπ., πολλές φορές πέφτει η πλατφόρμα.

Αν δεν υπάρχει το «Είσοδος Φοιτητή» σαν κείμενο, αλλά και άλλοι έλεγχοι που κάνουμε, τότε έχει γίνει επιτυχής είσοδος. Η πλοήγηση στα βασικά στοιχεία του φοιτητή γίνεται μέσω της εντολής:

```
Document results = Jsoup.connect(baseUrl +  
"studentMain.asp").timeout(45000).cookies(hashMap).get();
```

Στο `baseUrl` έχουμε το : <https://studentsweb.teimes.gr/unistudent/> (ή το `baseUrl` άλλου τμήματος) και του υποδεικνύουμε να κατευθυνθεί στο <https://studentsweb.teimes.gr/unistudent/studentMain.asp> (όπου είναι η σελίδα με τα στοιχεία του φοιτητή) και να χρησιμοποιήσει τα cookies που είχαμε αποθηκεύσει πιο πριν.

Τα στοιχεία του φοιτητή όπως και άλλες πληροφορίες (βαθμολογίες, ονόματα μαθημάτων κ.ά.), τα παίρνουμε με τον ίδιο τρόπο.

Για παράδειγμα, για να πάρουμε το όνομα χρήστη του φοιτητή από την σελίδα των στοιχείων του, το κάνουμε επιλέγοντας το

```
"table tbody tr td div[style=MARGIN-TOP: 20px] table tbody tr[style=height:20]  
td[width=250]"
```


1. Για να πάρουμε το όνομα χρήστη όπως και άλλα στοιχεία μέσω της jsoup επιλέγουμε ακριβώς τον selector στον οποίο θέλουμε να βρεθούμε.

2. Σε περίπτωση που στο ίδιο selector υπάρχουν παραπάνω από ένα πεδία τότε μπορούμε να επιλέξουμε το πεδίο που θέλουμε ανάλογα με την σειρά εμφάνισης τους στο html αρχείο.

Εικόνα 12: Επεξήγηση του πως γίνεται η άντληση των δεδομένων.

Και επειδή είναι το «πρώτο» στοιχείο από τα πολλά που μπορεί να υπάρχουν, γινόμαστε πιο συγκεκριμένοι και ζητάμε το στοιχείο «0», κατά αυτόν τον τρόπο:

```
student.setUsername(USR.get(0).text());
```

Με αυτόν τον τρόπο (και πολλούς ελέγχους σε περίπτωση που προκύψει σφάλμα) είμαστε σε θέση να πάρουμε τα δεδομένα από την πλατφόρμα.

Αυτή η τακτική άντλησης δεδομένων από την πλατφόρμα, δεν είναι η καλύτερη δυνατή, διότι δεν αντλούμε τα δεδομένα από κάποιο API αλλά από τον ιστότοπο της πλατφόρμας με αποτέλεσμα «αν αλλάξει κάτι» στον κώδικα της ιστοσελίδας, να χρειαστεί να κάνουμε και εμείς αντίστοιχες αλλαγές (κάτι το οποίο δεν θα γινόταν αν χρησιμοποιούσαμε API). Το γεγονός όμως ότι δεν

έχουμε πρόσβαση στο API, το ότι ο ιστότοπος είναι στατικός, και αντίστοιχες σχολές με την δικιά μας μπορούν να υποστηριχθούν (με μικρό-αλλαγές) καθιστά αυτόν τον τρόπο αρκετά αποτελεσματικό και αποδοτικό.

Όπως αναφέραμε πιο πάνω λόγω της ποικιλομορφίας των εκδόσεων της πλατφόρμας που μπορεί να έχουν οι αντίστοιχες σχολές πρέπει να είμαστε σε θέση να αντιμετωπίσουμε οποιοδήποτε «σφάλμα». Για παράδειγμα, οι βαθμοί στο ένα τμήμα μπορεί να καταχωρούνται ως δεκαδικοί και σε άλλο ως συμβολοσειρές. Για αυτό, αλλά και για πολλά παρόμοια περιστατικά υπάρχουν εκατοντάδες έλεγχοι ώστε όσο είναι δυνατό να μην επηρεαστεί την εμπειρία χρήσης του χρήστη.

Άξιο αναφοράς είναι η δραστηριότητα «Οδηγός σπουδών», όπως και η «Αναλυτική βαθμολογία». Τόσο η πλατφόρμα, όσο και το κάθε τμήμα καταχωρεί διαφορετικά τον οδηγό σπουδών του. Έτσι σε πολλά τμήματα έλλειπαν πεδία (όπως π.χ. πόσες διδακτικές μονάδες έχει το μάθημα κλπ.). Επίσης πολύ οδηγοί σπουδών σε τμήματα «σταματάνε» στο 7^ο εξάμηνο. Χρησιμοποιήθηκαν τρία jsoup select με το πρώτο να «ανιχνεύει» πόσα εξάμηνα απεικονίζονται στον λογαριασμό του χρήστη, με το δεύτερο να αφορά τα μαθήματα, και το τρίτο να ασχολείται με τα χαρακτηριστικά του κάθε μαθήματος. Με αυτόν τον τρόπο έγινε προσπάθεια να βρεθεί κάποιο μοτίβο, και έτσι έγινε, ώστε να μην έχουμε σφάλματα και να έχουμε κανονικά αποτελέσματα. Στα πεδία που δεν είχαν οριστεί από το εκάστοτε τμήμα εμφανίζεται ένα μήνυμα του τύπου «Δεν Ορίστηκε».

Τα δεδομένα ανακτώνται/καταχωρούνται σε λίστες και κλάσεις, με Getters και Setters και απεικονίζονται στην εφαρμογή σε στυλ καρτών μέσω ειδικών adapter σε συνδυασμό με RecyclerView.

Για παράδειγμα, εάν θέλουμε να απεικονίσουμε τα αποδεικτικά των ηλεκτρονικών υπογραφών, δηλώνουμε ένα RecyclerView στην κύρια δραστηριότητα, και στην συνέχεια καλούμε μια AsyncTask, δίνοντας της όλα τα απαραίτητα στοιχεία (cookies, το RecyclerView που δηλώσαμε, όνομα σχολής, baseUrl, κ.ά.):

```
final RecyclerView recListProbAithsewn = (RecyclerView) findViewById(R.id.cardList);
ProbAithsewn newprobolhathsewn = new ProbAithsewn(MainActivity.this,
recListProbAithsewn, hashMap, swipeContainer, SXOLH, baseUrl);
newprobolhathsewn.execute();
```

Στην AsyncTask δηλώνουμε μία λίστα (Apodeiktika στο παράδειγμά μας) ανάλογα με την κλάση, (MyApodeiktika) που είναι σχετική με το Adapter (ApodeiktikaAdapter), και το AsyncTask (ProbAithsewn) που βρισκόμαστε και γίνεται άντληση δεδομένων μέσω της jsoup κατά αυτόν τον τρόπο:

```
List<MyApodeiktika> Apodeiktika = new ArrayList<MyApodeiktika>();
```

```
Elements Number = results.select("body table tbody tr #main #form1 div #mainTable tbody tr td table#tablemain tbody tr.TableCellBold td table tbody tr td.bottomBorderLight b");  
Elements Hmnia = results.select("body table tbody tr #main #form1 div #mainTable tbody tr td table#tablemain tbody tr.TableCellBold td table tbody tr td.bottomBorderLight");  
Elements Links = results.select("body table tbody tr #main #form1 div #mainTable tbody tr td table#tablemain tbody tr.TableCellBold td table tbody tr td.bottomBorderLight a.underline");
```

Ουσιαστικά παίρνουμε για το κάθε αποδεικτικό τρία στοιχεία. Το νούμερο, την ημερομηνία, και τον σύνδεσμο του αποδεικτικού. Τα αποδεικτικά τα καταχωρούμε στην λίστα “Apodeiktika” μέσω της κλάσης “MyApodeiktika” κατά αυτόν τον τρόπο:

```
Apodeiktika.add(new MyApodeiktika(Number.get(counter1).text().toString(),  
Hmnia.get(counter2).text().toString(), Links.get(counter3).attr("abs:href").toString()));
```

Εφόσον έχουμε αποθηκεύσει τα αποδεικτικά στην λίστα μας, καλούμε το “ApodeiktikoAdapter” προσθέτοντας σαν ιδιότητα την λίστα που δημιουργήσαμε:

```
ApodeiktikoAdapter ca = new ApodeiktikoAdapter(Apodeiktika,BASEURL);
```

Στο αποδεικτικό adapter κατασκευάζουμε τις «κάρτες», ώστε να προβληθούν στον χρήστη.

Έπειτα ορίζουμε το RecyclerView που είχαμε δηλώσει στην κύρια δραστηριότητα, έχοντας σαν ιδιότητα το “ApodeiktikoAdapter” ώστε να προβάλει τις καρτέλες μας.

Σημείωση: Αυτό είναι ένα παράδειγμα. Η ίδια διαδικασία γίνεται για κάθε στοιχείο που προβάλλεται από την πλατφόρμα StudentsWeb, αλλά και για στοιχεία που αντλούμε τον δικό μας εξυπηρετητή. Για περισσότερες πληροφορίες σχετικά με τις κλάσεις ο ενδιαφερόμενος μπορεί να κοιτάξει το Παράρτημα Α όπου εξηγούνται αναλυτικά όλα τα στοιχεία της εφαρμογής.

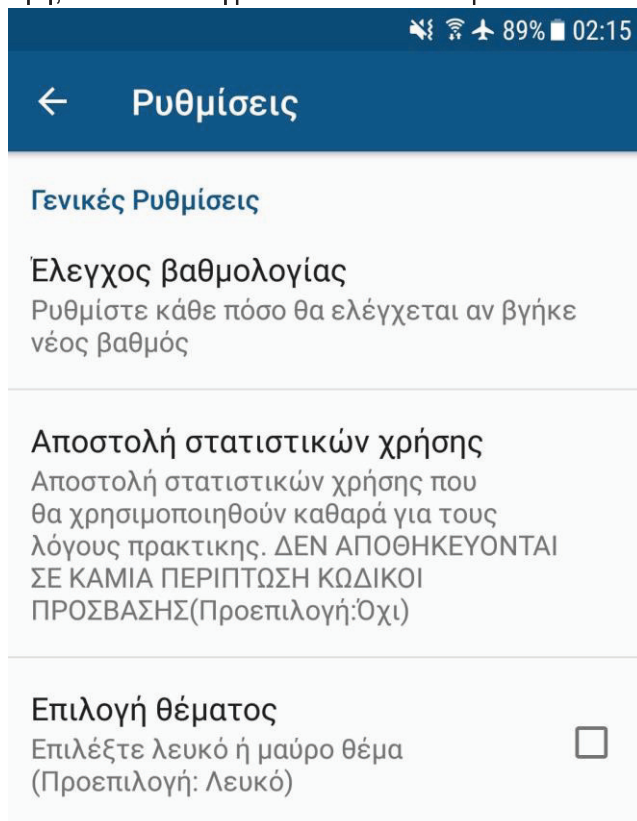
Ρυθμίσεις

Με το που γίνει είσοδος του χρήστη στην εφαρμογή, είμαστε σε θέση να καθορίσουμε κάποιες ρυθμίσεις (αν θέλουμε). Οι ρυθμίσεις που είμαστε σε θέση να αλλάξουμε είναι οι εξής:

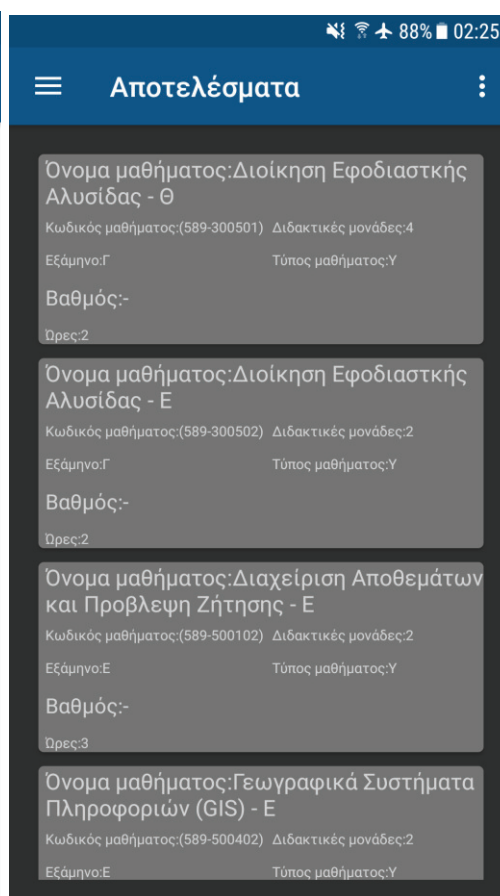
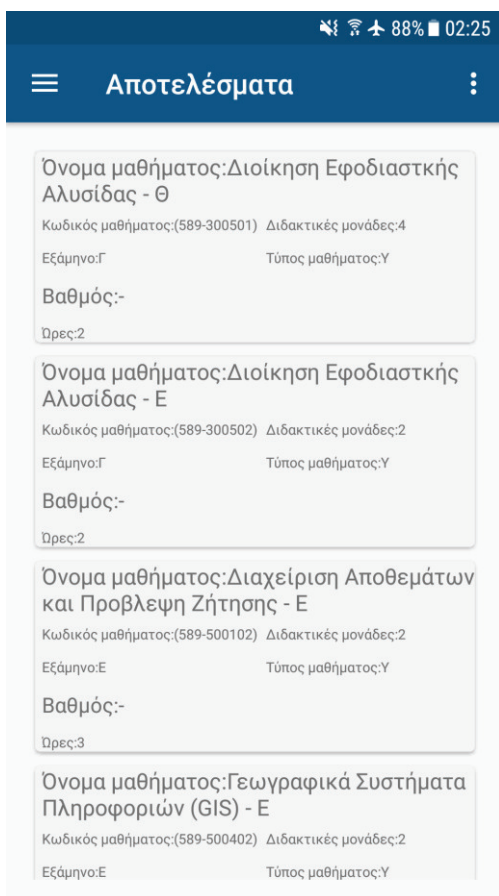
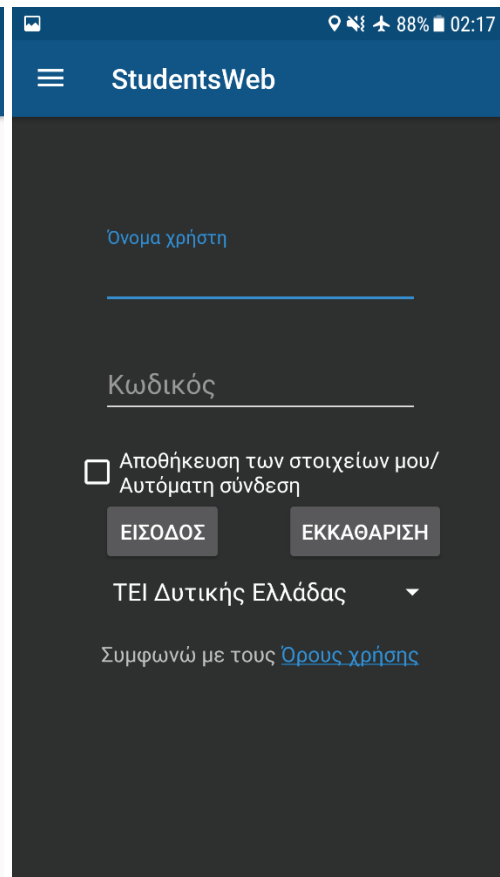
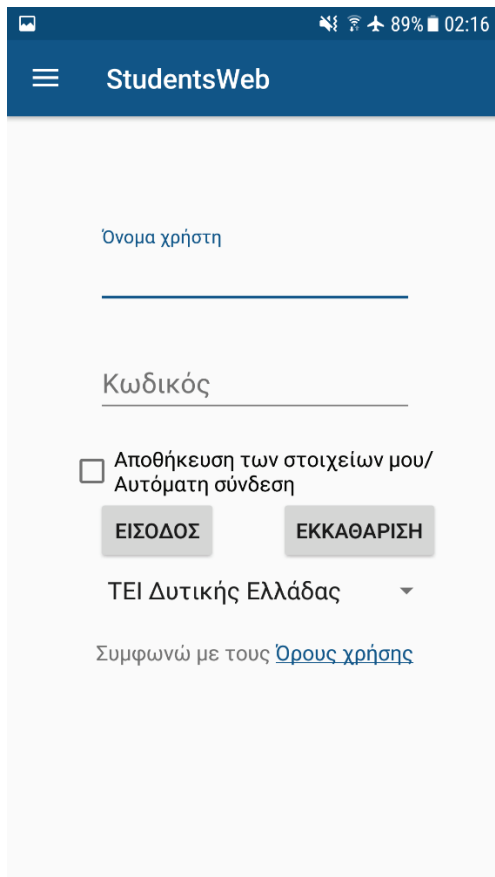
- Έλεγχος βαθμολογίας: Ρυθμίζουμε σε κάθε πόσο, και εάν, θα ελέγχει η υπηρεσία LessonServiceChecker το αν βγήκε καινούρια βαθμολογία ή όχι. Σαν προεπιλογή έχει οριστεί να μην γίνεται έλεγχος για νέα βαθμολογία, μέχρι να το ενεργοποιήσει ο χρήστης.
- Αποστολή στατιστικών χρήσης: Αποστολή στατιστικών από την συσκευή του χρήστη που θα χρησιμοποιηθούν κυρίως για την πτυχιακή εργασία (την συσκευή που χρησιμοποιεί ο χρήστης, το τμήμα

στο οποίο ανήκει κ.ά.). Σε καμιά περίπτωση δεν αποθηκεύονται κωδικοί πρόσβασης των χρηστών. Σαν προεπιλογή, σεβόμενοι τους χρήστες, έχουμε επιλέξει να μην γίνεται αποστολή στατιστικών.

- Επιλογή θέματος: Ο χρήστης μπορεί να διαλέξει ανάμεσα σε δύο θέματα, είτε άσπρο είτε μαύρο. Το μαύρο συνίσταται σε εκείνους με ευαίσθητα μάτια ή εκείνους που τους αρέσουν τα πιο σκούρα θέματα. Σαν προεπιλογή, είναι επιλεγμένο το λευκό θέμα:



Εικόνα 13: Οι επιλογές στις ρυθμίσεις.



Εικόνα 14: Επίδειξη άσπρου και μαύρου θέματος.

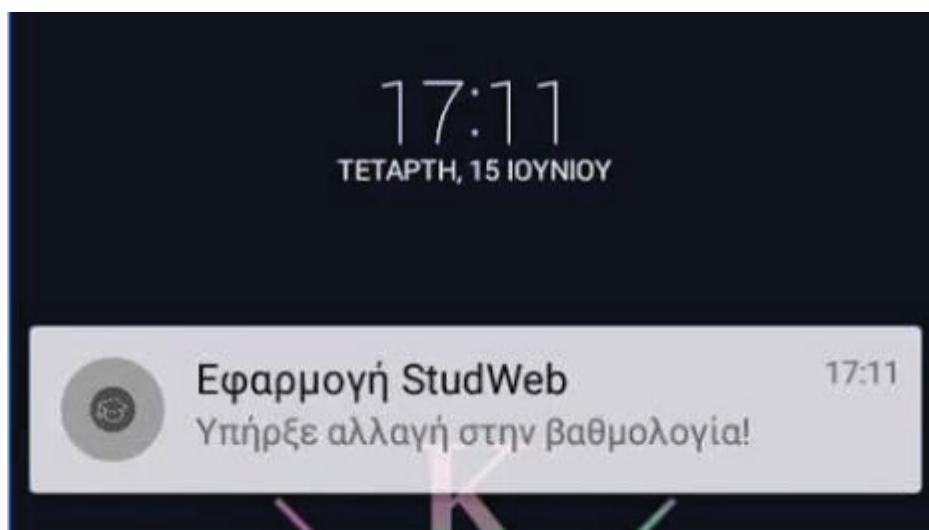
Ελέγχοντας για αλλαγές της βαθμολογίας

Ένα από τα χαρακτηριστικά που κάνουν αυτήν την εφαρμογή «μοναδική», είναι ότι υποστηρίζεται έλεγχος βαθμολογίας στο παρασκήνιο. Όταν δηλαδή η εφαρμογή δεν είναι στην κεντρική οθόνη του κινητού, μια υπηρεσία της εφαρμογής μας, επιτρέπει τον έλεγχο της βαθμολογίας. Ο έλεγχος της βαθμολογίας γίνεται ανά όσο έχουμε ορίσει εμείς στις ρυθμίσεις. Μπορεί να είναι από 30 λεπτά έως 1 εβδομάδα (ή αν θέλουμε να μην γίνεται και καθόλου έλεγχος).

Η λογική υλοποίησης αυτής της υπηρεσίας είναι πολύ απλή. Με το που ενεργοποιούμε το ανά πόσα λεπτά θα γίνεται έλεγχος βαθμολογίας στις ρυθμίσεις, παίρνουμε μια «αρχική εικόνα της βαθμολογίας». Την βαθμολογία αυτή την συγκρίνουμε με την καινούρια «εικόνα» που παίρνουμε την στιγμή που γίνεται ανανέωση/έλεγχος. Συγκρίνουμε αυτές τις δύο συμβολοσειρές, και αν υπάρχουν διαφορές σημαίνει ότι υπήρξε αλλαγή στην βαθμολογία.

Με το που γίνει ανίχνευση αλλαγής, ο χρήστης ειδοποιείται με μια ειδοποίηση (ηχητική και οπτική εάν υπάρχει λυχνία ειδοποίησης) που τον ενημερώνει για την αλλαγή του βαθμού του. Ο χρήστης με το που πατήσει στην ειδοποίηση, εισέρχεται αυτόματα στην εφαρμογή και γίνεται αυτόματη σύνδεση ώστε να δει τα αποτελέσματα του σε περίπτωση που το θέλει.

Σε περίπτωση που ο χρήστης κάνει εκκαθάριση στοιχείων/αποσύνδεση, ο έλεγχος της βαθμολογίας σταματάει καθώς άλλαξε ο χρήστης.



Εικόνα 15: Ειδοποίηση αλλαγής στην βαθμολογία από την υπηρεσία “LessonServiceChecker”.

Φτιάχνοντας το δικό μας API

Εισαγωγικά

Για τις ανάγκες της εφαρμογής StudentsWeb αποφασίστηκε να φτιαχτεί ένα API σαν Proof Of Concept. Χρησιμοποιήσαμε ένα server της GRNET που μας παρέχεται για ερευνητικούς ή ακαδημαϊκούς σκοπούς. Μέσω της υπηρεσίας φτιάχτηκε ένας εικονικός υπολογιστής ο οποίος μας βοηθάει στην λειτουργία της εφαρμογής StudentsWeb.

Επίσης αγοράστηκε ένα domain name το οποίο είναι το <http://e-genesis.eu> .

Μπορούσε να χρησιμοποιηθεί μια λύση χωρίς να «πληρώσουμε» καθώς η IP που μας δίνει το GRNET για τον server είναι στατική (θα μπορούσαμε να χρησιμοποιήσουμε π.χ. την υπηρεσία NO-IP ώστε να κάνουμε redirect μέσω εκείνης). Παρόλα αυτά σαν failsafe χρησιμοποιούνται και τα δύο. Σε περίπτωση που δεν θέλουμε να ανανεώσουμε την συνδρομή για το domain name που αγοράσαμε, έχουμε σαν backup επιλογή ήδη NO-IP domain name το οποίο είναι το <http://velocean.no-ip.info/> . Ο λόγος που αγοράστηκε domain name είναι για να έχουμε ένα πιο ευπρόσιτο και «εύηχο» όνομα για τον server/ιστότοπό μας.

Λεπτομέρειες

Στον Server μας ήμασταν σε θέση να συνδεθούμε μέσω VNC ή SSH client. Εγκαταστάθηκαν Apache, MySQL Server, FTP Server, WordPress και άλλα εργαλεία τα οποία μας βοήθησαν στην υλοποίηση του API, του Web Server κ.ά..

Έγινε σαν POC (Proof Of Concept) ένα υποτυπώδες API με versioning “v1” όπου είναι σε θέση να λαμβάνει και να δέχεται στοιχεία από την Android εφαρμογή μας.

Σημείωση: Ο λόγος που σχεδιάστηκε κατά αυτόν τον τρόπο το API (με versioning) είναι ώστε σε περίπτωση που αλλάξουμε το API (π.χ. “v2”, “v3”, κ.ο.κ.), να μην επηρεαστούν χρήστες που δεν έχουν κάνει ενημέρωση την εφαρμογή. Καθώς η εφαρμογή εξελίσσεται έτσι και το API της εφαρμογής βελτιώνεται και πρέπει να υπάρχει versioning σαν δικλείδα ασφαλείας, αλλά και για να συνεχιστεί η υποστήριξη παλαιότερων εκδόσεων εφαρμογής/API.

Δημιουργήθηκε μια βάση δεδομένων η οποία κρατάει (σε περίπτωση που συμφωνεί ο χρήστης μέσω της εφαρμογής) βασικά στοιχεία, όπως την ημερομηνία που κάνει είσοδο ο χρήστης, την μάρκα του κινητού τηλεφώνου που έχει και άλλα, τα οποία σκοπό έχουν να χρησιμοποιηθούν καθαρά για

στατιστικούς λόγους για στην πτυχιακή εργασία. Όπως αναφέρεται και στον χρήστη καθώς επιλέγει την αποστολή ή όχι των στοιχείων, σε καμία περίπτωση δεν αποθηκεύονται κωδικοί.

Ενδεικτικά ο κώδικας java όταν γίνεται αποστολή στατιστικών:

```
HttpPost httpPost = new HttpPost("http://e-genesis.eu/minisite/API/v1/login_api.php");
HttpClient httpClient = new DefaultHttpClient(getip.getparams());
List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>(4);
nameValuePairs.add(new BasicNameValuePair("ipaddress",
getip.getIpAddress()));nameValuePairs.add(new BasicNameValuePair("client",
deviceVersion)); nameValuePairs.add(new BasicNameValuePair("deviceName",
deviceName)); nameValuePairs.add(new BasicNameValuePair("UUID",
Installation.id(context.getApplicationContext())));httpPost.setEntity(new
UrlEncodedFormEntity(nameValuePairs));
HttpResponse response = httpClient.execute(httpPost);
```

Κώδικας php από την μεριά του server:

```
<?php
include 'dbconfig.php';
$ip = $client = $deviceName = $UUID= "";
$today = date("Y-m-d H:i:s");
$ip=$_POST["ipaddress"];
$client= $_POST["client"];
$deviceName= $_POST["deviceName"];
$UUID= $_POST["UUID"];
//send query to database and get results
$sql = "INSERT INTO Statistics
(logintime,ipaddress,client,deviceName,UUID) VALUES
('$today','.$ip.','.$client.','.$deviceName.','.$UUID.')";
$result = mysql_query("SET NAMES 'UTF8'", $conn);
$result = mysql_query($sql, $conn);
if (!$sql) {
    echo 'Could not run query: ' . mysqli_error($conn);
    exit;
}
exit;
$conn->close();
?>
```

Η καταχώρηση στην βάση SQL απεικονίζεται κατά αυτόν τον τρόπο:

id	logintime	ipaddress	client	deviceName	UUID
1459	2016-11-14 20:25:08	62.74.26.2	6.0.1	LG-D802	a38d85cb-49dc-4ed5-bdb8-a866acc14c4

Εικόνα 16: Απεικόνιση της καταχώρησης στην βάση δεδομένων.

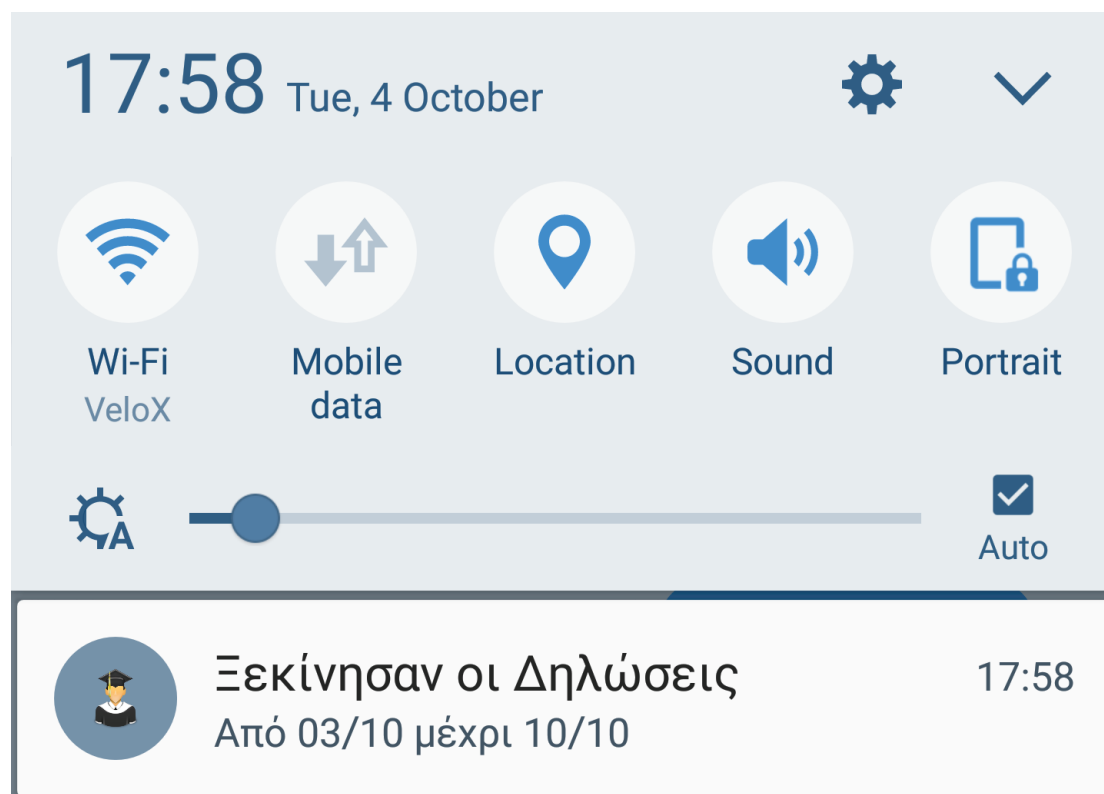
Η βάση δεδομένων επίσης πέρα των στατιστικών, μέσω του API που δημιουργήθηκε είναι σε θέση να στέλνει στην εφαρμογή ενημερώσεις (π.χ. για την κατάσταση του StudentsWeb σε περίπτωση που δεν λειτουργεί), να

κρατάει σχόλια που στέλνουν μέσω της εφαρμογής οι χρήστες όπως επίσης και μια πληθώρα ενεργειών οι οποίες περιγράφονται παρακάτω.

Όπως αναφέρθηκε παραπάνω η εφαρμογή μπορεί να χρησιμοποιηθεί για πολλά ιδρύματα, ξεχωριστά το ένα από το άλλο. Το κάθε «τμήμα» έχει κάποιο αναγνωριστικό. Για παράδειγμα το Τ.Ε.Ι. Μηχανικών Πληροφορικής στο Αντίρριο έχει το αναγνωριστικό «MP_TE» και το κάθε τμήμα κάτι αντίστοιχο. Αυτό επιτρέπει στον διαχειριστή ανάλογα με το τμήμα που έχει επιλέξει ο φοιτητής, να παρουσιάζει στον ενδιαφερόμενο φοιτητή στοχευμένα αποτελέσματα που αφορούν την σχολή του.

Η βάση δεδομένων ανάλογα με την σχολή που έχει επιλέξει ο φοιτητής είναι σε θέση μέσω του API μας να αποθηκεύει/παρέχει αποτελέσματα για συγκεκριμένες σημειώσεις της αντίστοιχης σχολής, πρόγραμμα εξαμήνου, πρόγραμμα εξεταστικής, οδηγίες για να φτάσει στην σχολή, συχνές ερωτήσεις για το αντίστοιχο ΑΕΙ/ΤΕΙ, τα email των καθηγητών και άλλα.

Στην βάση δεδομένων αποθηκεύονται και τα μοναδικά αναγνωριστικά κάθε συσκευής που εισέρχεται στην εφαρμογή (ανάλογα με το τμήμα που έχει επιλεγεί). Έτσι σε περίπτωση που για παράδειγμα υπάρξουν διορθωτικές δηλώσεις σε κάποια σχολή μπορούμε να μέσω του API που έχουμε στήσει αλλά και του FCM (Firebase Cloud Messaging) (βλ. Βιβλιοθήκες) να στείλουμε στοχευμένη ειδοποίηση στα κινητά των ατόμων αυτών που φοιτούν στην συγκεκριμένη σχολή (η ειδοποίηση μπορεί να έχει ή όχι ήχο) και να τους ενημερώσουμε για τις διορθωτικές δηλώσεις.



Εικόνα 17: Παράδειγμα χρήσης του FCM.

Το ανέβασμα σημειώσεων όπως αναφέρεται και στην εφαρμογή πρέπει να εγκριθεί από κάποιον διαχειριστή. Ο λόγος που έγινε αυτό είναι ότι πολλοί χρήστες μπορεί να καταχραστούν την υπηρεσία αυτή εις βάρος των διαχειριστών αλλά και των χρηστών. Έτσι δημιουργήθηκε μια υποτυπώδης πλατφόρμα για την έγκριση ή όχι αυτών των αρχείων. Με το που εγκριθούν τα αρχεία ο φοιτητής είναι σε θέση να κατεβάσει τις σημειώσεις του.

Πλατφόρμα ανεβάσματος αρχείων

Οι μεταφορτώσεις <u>Ανέβασμα νέων αρχείων</u>						
Όνομα Αρχείου	Εξάμηνο	Date Uploaded	Uploaded by	Approve	Reviewed	View
AI_LECTURE_NOTES.zip	A	2016-06-20 01:29:07	antofrer	Already approved	Yes	view file

Εικόνα 18: Μια «υποτυπώδης» πλατφόρμα διαχειριστή στον server του e-genesis.eu που μας δείχνει τα αρχεία, αν έχουν εγκριθεί ή όχι, καθώς και άλλα χαρακτηριστικά των ανεβασμένων σημειώσεων.

Το WordPress δημιουργήθηκε σαν ένα απλό ιστολόγιο/blog ώστε να ενημερώνονται οι χρήστες για τυχόν νέα, αναβαθμίσεις στην εφαρμογή, όρους χρήσης και διάφορα άλλα.



Εικόνα 19: Το “blog” του e-genesis.eu .

Οι server της GRNET έχουν πολύ καλό “uptime” το οποίο «αγγίζει» το 99% μέχρι την στιγμή που γράφεται αυτή η πτυχιακή, σύμφωνα με το HappyApps.io (βλ. βιβλιοθήκες). Παρόλα αυτά σε περίπτωση που υπάρχει πρόβλημα με τον server του «Ωκεανός» έχει γίνει πρόληψη η εφαρμογή να συνεχίζει να παραμένει λειτουργική (Βαθμολογίες, Αποτελέσματα κ.ά.), εκτός-προφανώς- από το κομμάτι που επικοινωνεί με το API που βρίσκεται στην υπηρεσία «Ωκεανός».

Ωστόσο έχει γίνει προσπάθεια ώστε σε περίπτωση που συμβεί κάτι στον server να υπάρχει «μηδενικό downtime». Για λόγους ασφαλείας, δημιουργούνται αντίγραφα ασφαλείας κάθε βδομάδα (και στα αρχεία API και στην βάση δεδομένων) ενώ γίνεται και “live” replication το οποίο όμως είναι σε

ρύθμιση **MASTER-MASTER** σε ένα Raspberry Pi σε ξεχωριστή τοποθεσία από αυτήν που βρίσκεται ο server της GRNET. Χρησιμοποιώντας ρύθμιση master-master τα δεδομένα αντιγράφονται και στους δύο server. Μπορούμε να διαβάσουμε/γράψουμε σε όποιον από τους δύο server «θέλουμε» και αυτόματα οι αλλαγές θα συγχρονιστούν και στους δύο server. Επειδή δεν υπάρχει κάποιος DNS / load balancer για να γίνει «ανίχνευση»/ανακατεύθυνση, εάν ο server της GRNET βρεθεί εκτός λειτουργίας, έχουμε προσθέσει προγραμματιστικά στον κώδικα της εφαρμογής μια μέθοδο όπου ελέγχει εάν ο server της GRNET είναι διαθέσιμος και επιλέγει αυτόν ως προκαθορισμένο server. Σε περίπτωση που ο πρώτος server είναι εκτός λειτουργίας η εφαρμογή επιλέγει το δεύτερο server που βρίσκεται σε ένα Raspberry Pi. Εφόσον έχουμε τα δεδομένα σε MASTER-MASTER ρύθμιση τα δεδομένα είναι πλήρως «ενημερωμένα» στον δεύτερο RPi server. Όταν ο πρώτος server επανέλθει online ο δεύτερος αντιγράφει τις «χαμένες» εγγραφές στον πρώτο, λόγω του “downtime”, και πλέον συνεχίζει να χρησιμοποιείται ο πρώτος, απροβλημάτιστα. Έχοντας master-master ρύθμιση στους server της βάσης έχουμε redundancy (πλεονασμό) σε περίπτωση προβλήματος αλλά και αποδοτικότητα (σε περίπτωση που ένας από τους δύο server δέχεται υψηλό φόρτο χρησιμοποιείται ο άλλος). Όλο αυτό είναι παραπάνω από αυτό που «χρειάζεται» αυτή η εφαρμογή καθώς ελάχιστο downtime είναι αποδεκτό για μια εφαρμογή τέτοιου είδους αλλά έγινε σαν POC.

Σημείωση: Το αν δέχεται ή όχι υψηλό φόρτο κάποιος από τους δύο server καθορίζεται μέσω ενός «Ping» που γίνεται καθώς δεν υπάρχει κάποιος load balancer για να γίνει ανακατεύθυνση στον server με το μικρότερο φόρτο συνολικά. Το γεγονός όμως ότι η εφαρμογή είναι εντός Ελλάδος και το GRNET είναι εντός δικτύου gr-ix (Greek Internet Exchange) μας επιτρέπει μέσω του χαμηλού latency του δικτύου να καταλάβουμε (όσο είναι δυνατόν) τουλάχιστον από άποψη φόρτου δικτύου, ποιος από τους δύο server μπορεί να «απαντήσει» πιο γρήγορα και έτσι επιλέγουμε αυτόν.

StudentsWeb και Doze Mode

Τι είναι το Doze Mode;

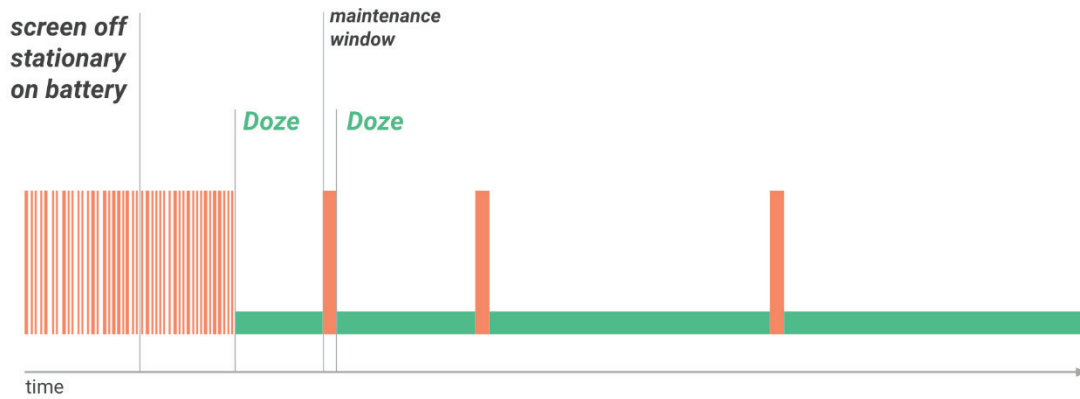
Με τον ερχομό του Android Marshmallow υπήρξαν πολλές αλλαγές στο λειτουργικό του Android για την βελτίωση της μπαταρίας. Η «μετάφραση» του Doze στα ελληνικά σημαίνει «λαγοκοιμάμαι», λέξη που «προδίδει» το τι κάνει το Doze. Το Doze δεν είναι τίποτε άλλο παρά από ένα «σετ» κανόνων τα οποία θα θέσουν την συσκευή σας σε κατάσταση αδράνειας (Deep Sleep) όταν δεν το χρησιμοποιείτε.

Το Doze ενεργοποιείται αυτόματα όταν δεν χρησιμοποιούμε πραγματικά την συσκευή. Λέγοντας δεν χρησιμοποιούμε δεν εννοούμε απλά να το έχουμε στην τσέπη μας. Για να ενεργοποιηθεί το Doze το κινητό πρέπει να έχει κλειστή οθόνη, οι σένσορες του κινητού να μην έχουν αλλαγές, και να μην φορτίζει το κινητό.

Αν ικανοποιούνται όλες αυτές οι συνθήκες, μετά από λίγο το κινητό πηγαίνει σε κατάσταση αδράνειας. Στο Doze το σύστημα προσπαθεί να συντηρήσει την μπαταρία «απαγορεύοντας» σε εφαρμογές να έχουν πρόσβαση στο δίκτυο και σε υπηρεσίες που χαλάνε πολύ επεξεργαστική ισχύ. Επίσης αναβάλλει ότι τυχόν syncs, jobs, alarms, μπορεί να είχαν οι εφαρμογές. Εφαρμογές όμως που ορίζουν ειδοποιήσεις «Υψηλής-προτεραιότητας» (όπως κλήσεις, SMS, κ.ά.) είναι ακόμη σε θέση να ξυπνήσουν το κινητό, βγάζοντάς το προφανώς από την κατάσταση Doze.

Αυτό είναι κάτι το οποίο θα μπορούσαν να το καταχραστούν οι προγραμματιστές και απλά να ορίσουν όλες τις εφαρμογές τους ως «Υψηλής-προτεραιότητας» για να γίνει όμως κάτι τέτοιο τα «Υψηλής-προτεραιότητας» μηνύματα θα πρέπει να περάσουν μέσω της Google. Σε περίπτωση που δει η Google ότι γίνεται κατάχρηση των GCM (Google Cloud Messaging) ή FCM μπορεί να διακόψει τις ειδοποιήσεις αυτές.

Αυτό όμως δεν σημαίνει ότι η εφαρμογή μας όσο το κινητό «κοιμάται» δεν μπορεί να δεχθεί ή να αποστείλει δεδομένα. Η Google χρησιμοποιεί ένα όπως το ονομάζει, “Idle Maintenance” το οποίο ουσιαστικά, για ένα μικρό χρονικό διάστημα βγάζει το κινητό από το Doze, ώστε όλες οι εφαρμογές που πρέπει να δεχθούν ή λάβουν δεδομένα τα οποία είχαν πρότινος αναβληθεί (jobs, syncs, alarms, κ.ά.), να το κάνουν. Με το που τελειώσει αυτό το «χρονικό διάστημα» του Idle Maintenance το κινητό επιστρέφει στην κατάσταση “Idle”.



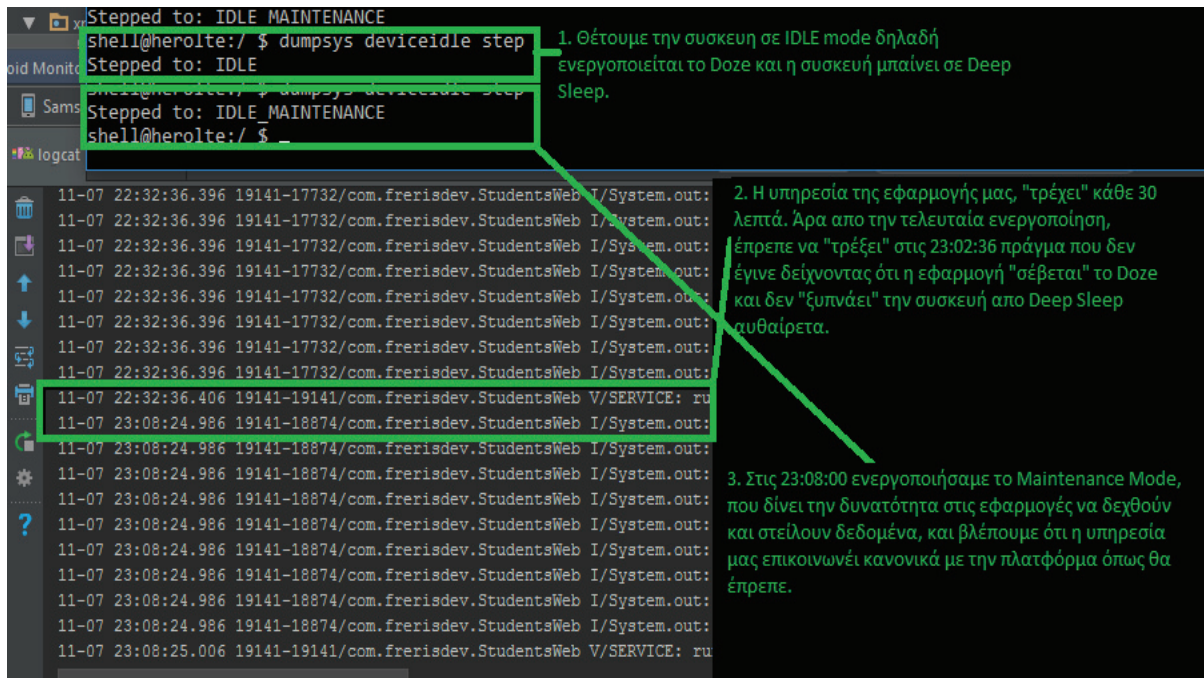
Εικόνα 20: Το επαναλαμβανόμενο maintenance window ώστε οι εφαρμογές να μπορούν να ολοκληρώσουν τις διεργασίες που είναι σε εξέλιξη.

Όσο περνάει η ώρα και το κινητό συνεχίζει να μην χρησιμοποιείται, τα maintenance window γίνονται λιγότερο συχνά ώστε να γίνει ακόμα μεγαλύτερη διατήρηση της μπαταρίας.

Με το που «ξυπνήσουμε» την συσκευή όλες οι εφαρμογές επιστρέφουν στην κανονική τους κατάσταση [6].

Υποστήριξη Doze Mode στο StudentsWeb

Στην εφαρμογή StudentsWeb υπάρχει υποστήριξη του doze χωρίς να γίνεται κατάχρηση του, ώστε να μην καταναλώνεται άσκοπα μπαταρία.



Εικόνα 21: Επίδειξη ότι το Doze Mode υποστηρίζεται στην εφαρμογή μας.

Σε περίπτωση που χρειαστεί κάποια «Υψηλής-προτεραιότητας» ειδοποίηση είμαστε σε θέση μέσω του Firebase Cloud Messaging να στείλουμε ειδοποίηση. Στις δοκιμές που έγιναν το κινητό «ξύπνησε» από το Doze Mode όπως και έπρεπε να συμβεί διότι η ειδοποίηση που εστάλη ήταν «Υψηλής-προτεραιότητας».

Κεφάλαιο 4

Στο κεφάλαιο αυτό αναφερόμαστε στις επιπλέον βιβλιοθήκες ή υπηρεσίες που χρησιμοποιήθηκαν στην ανάπτυξη αυτής της εφαρμογής.

Βιβλιοθήκες

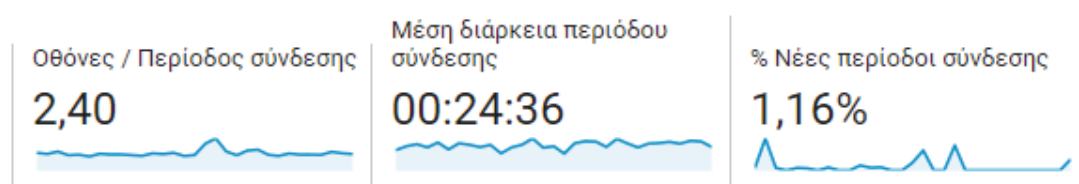
Jsoup

Πάνω στην βιβλιοθήκη jsoup στηρίχθηκε «όλη η εφαρμογή». Χάρη στη βιβλιοθήκη αυτή βασιστήκαμε ώστε να γίνεται άντληση δεδομένων από την πλατφόρμα StudentsWeb καθ' ότι δεν είχαμε πρόσβαση στο API της cardisoft. Χρησιμοποιούμε την jsoup για να συνδεθούμε, αναλύσουμε, αποθηκεύσουμε δεδομένα από την πλατφόρμα και να τα προβάλλουμε μέσω

της εφαρμογής μας στον χρήστη. Με την πλατφόρμα αυτή μπορούμε να επιλέξουμε συγκεκριμένα κομμάτι από την ιστοσελίδα μέσω CSS επιλογή και να βρούμε συγκεκριμένα στοιχεία που βρίσκονται στην ιστοσελίδα και αντίστοιχα να τα προβάλλουμε στην εφαρμογή μας.

Google Analytics

Το Google Analytics το χρησιμοποιούμε για να μπορούμε να βλέπουμε στατιστικά της εφαρμογής μας αλλά και αναφορές από χρήστες με σφάλματα και άλλα. Το Google Analytics προσφέρει μια «γκάμα» εργαλείων όπου μας επιτρέπει να επιβλέπουμε διάφορες αναφορές από την εφαρμογή μας. Μπορούμε να βλέπουμε στατιστικά σε πραγματικό χρόνο πόσοι χρήστες χρησιμοποιούν την εφαρμογή, σε ποιες οθόνες επικεντρώνονται κ.ά.. Είμαστε σε θέση να δούμε σε ποιες οθόνες επικεντρώνονται οι χρήστες, πόσο κάθονται, δημογραφικά στοιχεία και πολλά άλλα. Με βάση αυτά τα στατιστικά στοιχεία μπορούμε να βελτιστοποιήσουμε την εφαρμογή μας με βάση τα «μονοπάτια» που ακολουθούν οι χρήστες μας ώστε να προσφέρουμε στον τελικό χρήστη μια καλή εμπειρία.



Εικόνα 22: Απεικόνιση συμπεριφοράς χρηστών



Εικόνα 23: Γενικά στατιστικά εφαρμογής

Firebase

Στην εφαρμογή StudentsWeb εφαρμόζουμε το Firebase API μια υπηρεσία όπου είναι πολύ καινούρια και καινοτόμα.

Το Firebase API χωρίζεται στα:

- **Firebase Analytics**
- **Firebase Cloud Messaging (FCM)**

- **Firebase Auth**
- **Firebase Real-time Database**
- **Firebase Storage**
- **Firebase Hosting**
- **Firebase Remote Config**
- **Firebase Test Lab**
- **Firebase Crash Reporting**

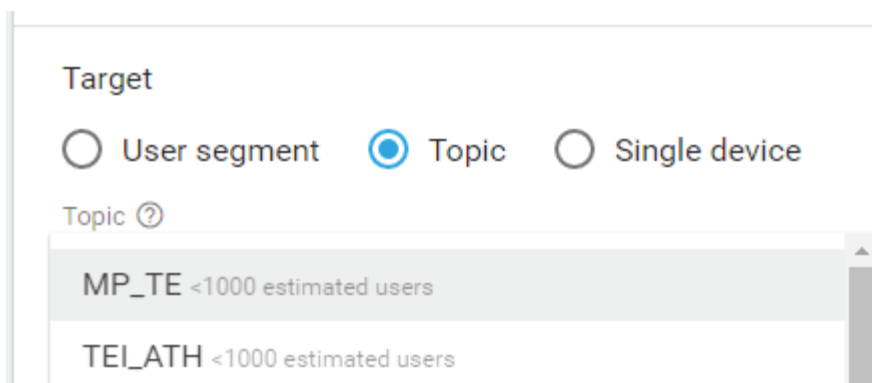
Οι υπηρεσίες που χρησιμοποιούμε εμείς στην εφαρμογή είναι οι:

Με το **Firebase Cloud Messaging(FCM)** που χρησιμοποιούμε στην εφαρμογή μας μπορούμε να στέλνουμε στενευμένες ειδοποιήσεις στους χρήστες (Π.Χ «Ξεκίνησε η περίοδος δηλώσεων»). Αναλόγως το τμήμα του χρήστη, γίνεται “Subscribe” σε συγκεκριμένο “topic” ώστε οι ειδοποιήσεις (σε περίπτωση που χρειαστούν να είναι όσο γίνεται πιο στοχευμένες).

Κώδικας στην java:

```
FirebaseMessaging.getInstance().subscribeToTopic(SXOLH);
```

Όπου “SXOLH” είναι η μεταβλητή της αντίστοιχης σχολής μας όπως π.χ. «MP_TE». Όταν θελήσουμε να στείλουμε ειδοποίηση μέσω της κονσόλας Firebase μπορούμε να στείλουμε σε όλο το εύρος της εφαρμογής, σε συγκεκριμένες εκδόσεις (είτε εκδόσεις εφαρμογής, είτε εκδόσεις Android και πολλά άλλα), συγκεκριμένα σε μεμονωμένα κινητά εφόσον έχουμε το FCM registration token το οποίο προηγουμένως έχουμε αποθηκεύσει στον server μας μέσω του δικού μας API στην βάση μας, αλλά και μέσω των “topic” που έχει κάνει “subscribe” ο χρήστης.



Εικόνα 24: Απεικόνιση των επιλογών που έχουμε μέσω του “subscribeToTopic();” που βάλαμε στην εφαρμογή

Με το **Firebase Auth** ταυτοποιούμε τον χρήστη μέσω μιας πληθώρας κοινωνικών μέσων δικτύωσης και όχι μόνο (Facebook, Google, Github κ.ο.κ.). Εμείς στην εφαρμογή μας χρησιμοποιούμε το Firebase Auth με το Google Login για την δραστηριότητα της «συζήτησης».

Με το **Firebase Real-time Database** είμαστε σε θέση να συγχρονίζουμε τα δεδομένα μας ανάμεσα σε όλες τις συσκευές. Η βάση δεδομένων είναι επίσης προσβιβάσιμη μέσω ενός REST API.

Με το **Firestore Storage** έχουμε ασφαλή ανεβάσματα και κατεβάσματα αρχείων. Μπορεί να χρησιμοποιηθεί για το ανέβασμα και κατέβασμα σημειώσεων. Προς το παρόν χρησιμοποιείται ο server που μας παρέχει το GRNET ο οποίος περιέχει το δικό μας API. Μελλοντικά μπορεί να γίνει εναλλαγή των υπηρεσιών.

Με το **Firestore Remote Config** μπορούμε ως προγραμματιστές να ορίσουμε κάποιες παραμέτρους που είτε θα αλλάξουν την συμπεριφορά στην εφαρμογή μας είτε εμφανισιακά χωρίς να χρειαστεί ο χρήστης να κατεβάσει κάποια αναβάθμιση. Σαν Proof of Concept στην εφαρμογή StudentsWeb έχουμε βάλει στην δραστηριότητα «Συζήτηση» remote config στον αριθμό των λέξεων. Που ουσιαστικά μπορούμε απομακρυσμένα να ορίσουμε τον μέγιστο αριθμό χαρακτήρων που μπορεί ο χρήστης να εισάγει σε κάθε πρόταση.

Με το **Firestore Test Lab** for Android είμαστε σε θέση να δοκιμάσουμε την εφαρμογή μας σε μια πληθώρα συσκευών Android για να ελέγξουμε αν υπάρχουν τυχόν απρόσμενα κλεισίματα, σφάλματα αλλά και στοιχεία τα οποία έπρεπε να εμφανίζονται διαφορετικά. Στην περίπτωση μας δοκιμάστηκε η τεχνολογία αυτή και ήταν πολύ χρήσιμη. Το γεγονός όμως ότι η εφαρμογή μας είχε κάποια «είσοδο» (δηλαδή έπρεπε να γίνει είσοδος στοιχείων για να προχωρήσουμε στις επόμενες οθόνες) κατέστησε αδύνατο στο Firestore Test Lab for Android να προχωρήσει πέρα από την αρχική –Login- δραστηριότητα. Αυτό βέβαια θα μπορούσε να λυθεί με τους εξής τρόπους:

- Να αγοράσουμε την «Premium» έκδοση του Firestore Test Lab for Android
- Να «Χτίσουμε» μια εφαρμογή που είχε ήδη μέσα στα πεδία «Όνομα χρήστη» και «Κωδικός» τα διαπιστευτήρια μας ώστε να γίνει «αυτόματα» η είσοδος.

Το πρώτο κομμάτι δεν έγινε λόγω έλλειψης πόρων (αρκετά δαπανηρό το κόστος). Το δεύτερο κομμάτι που ήταν και πολύ πιο εύκολο δεν έγινε κυρίως για λόγους ασφαλείας.

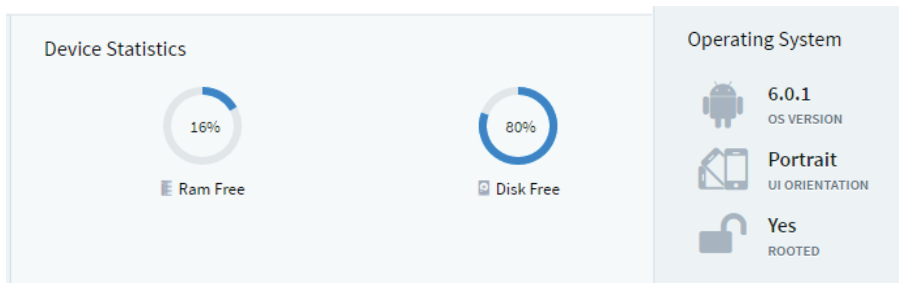
Παρόλα αυτά το κατάλληλο testing στην εφαρμογή έγινε από μια ομάδα Alpha Testers.

Με το **Firestore Crash Reporting** είμαστε σε θέση να έχουμε αναφορές από σφάλματα στην εφαρμογή μας. Μπορούμε να κατηγοριοποιήσουμε τα σφάλματα μας και να καταλάβουμε τι προξενεί το σφάλμα, σε πια έκδοση, αν είναι κάποιο συγκεκριμένο Smartphone κ.ο.κ. [9]. Χρησιμοποιήθηκε αρκετά στις Alpha εκδόσεις της εφαρμογής, παρόλα αυτά αντικαταστάθηκε από τα Crashlytics (θα αναφερθούμε πιο κάτω) της Farbric.io.

Fabric.io

Crashlytics & Answers

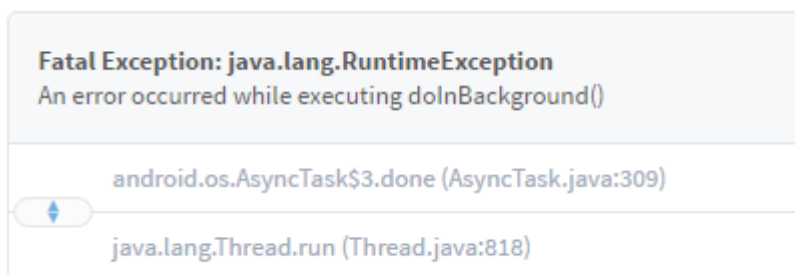
Την υπηρεσία **Crashlytics** την χρησιμοποιούμε σε όλο το εύρος της εφαρμογής. Σε περίπτωση απρόσμενου κλεισίματος/σφάλματος δημιουργείται ένα αρχείο με την καταγραφή του σφάλματος και αποστέλλεται στην υπηρεσία. Εμείς για το σφάλμα αυτό ενημερωνόμαστε είτε μέσω της εφαρμογής της Fabric (σε Android και iOS), μέσω Email, αλλά και μέσω του ιστότοπου τους. Έχει αποδειχθεί πολύ σημαντική σαν υπηρεσία διότι το σφάλμα εμφανίζεται «σχεδόν αμέσως» ώστε να μπορούμε να κάνουμε debugging. Επίσης εμφανίζονται και πολλά στοιχεία για το κινητό στο οποίο προέκυψε το σφάλμα.



Εικόνα 25: Στοιχεία της συσκευής που προέκυψε το σφάλμα.

Στην παραπάνω εικόνα βλέπουμε ότι το σφάλμα προκλήθηκε σε συσκευή η οποία είχε root δικαιώματα και είχε ελεύθερη μνήμη RAM 16%.

Τέτοιες ενδείξεις καθώς και η αναλυτική καταγραφή του σφάλματος, μας βοηθάει να ενημερωθούμε άμεσα για το πρόβλημα, να το εντοπίσουμε (αν υπάρχει) και να το διορθώσουμε.



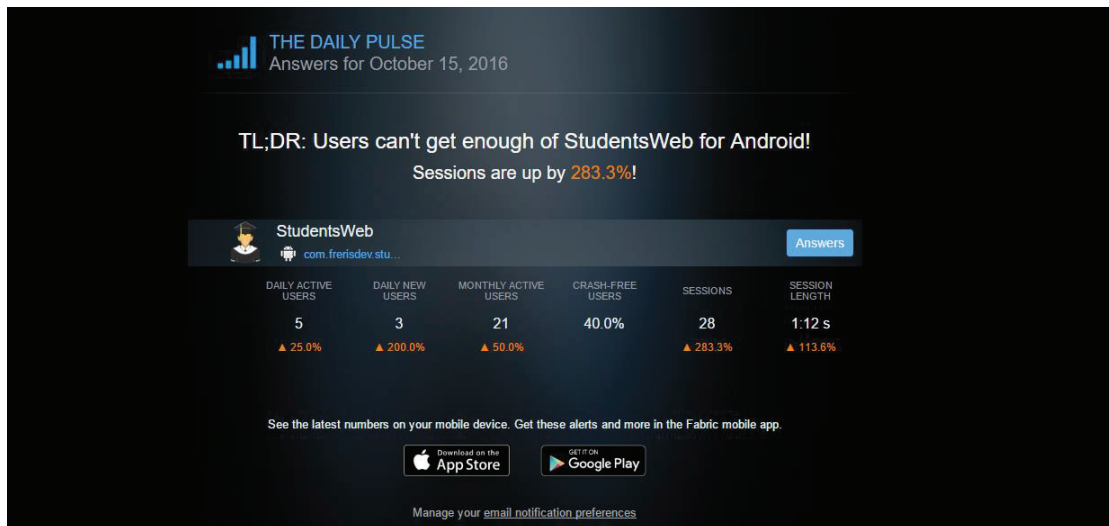
Εικόνα 26: Αναλυτικά στοιχεία για το που προέκυψε το σφάλμα και τί το προκάλεσε.

Επίσης με την υπηρεσία **Answers** που έχει η Fabric.io μπορούμε να βλέπουμε κάποια στατιστικά για την εφαρμογή μας με ταξινομήσεις κ.ά.. Στατιστικά όπως τους χρήστες που βρίσκονται στην εφαρμογή αυτήν την στιγμή, του καθημερινούς χρήστες, τις εκδόσεις Android και κινητές συσκευές που χρησιμοποιούν οι χρήστες που έχουν εγκαταστήσει την εφαρμογή κ.ά..



Εικόνα 27: Μηνιαίοι χρήστες και ποσοστό χρηστών που δεν έχουν σφάλματα (την σημερινή μέρα).

Καθημερινά επίσης αν το επιλέξουμε μπορεί να μας έρχεται ενημερωτικό Email για την κατάσταση της εφαρμογής (Αν υπάρχουν σφάλματα, πόσοι καθημερινούς χρήστες είχαμε, κ.ά.).

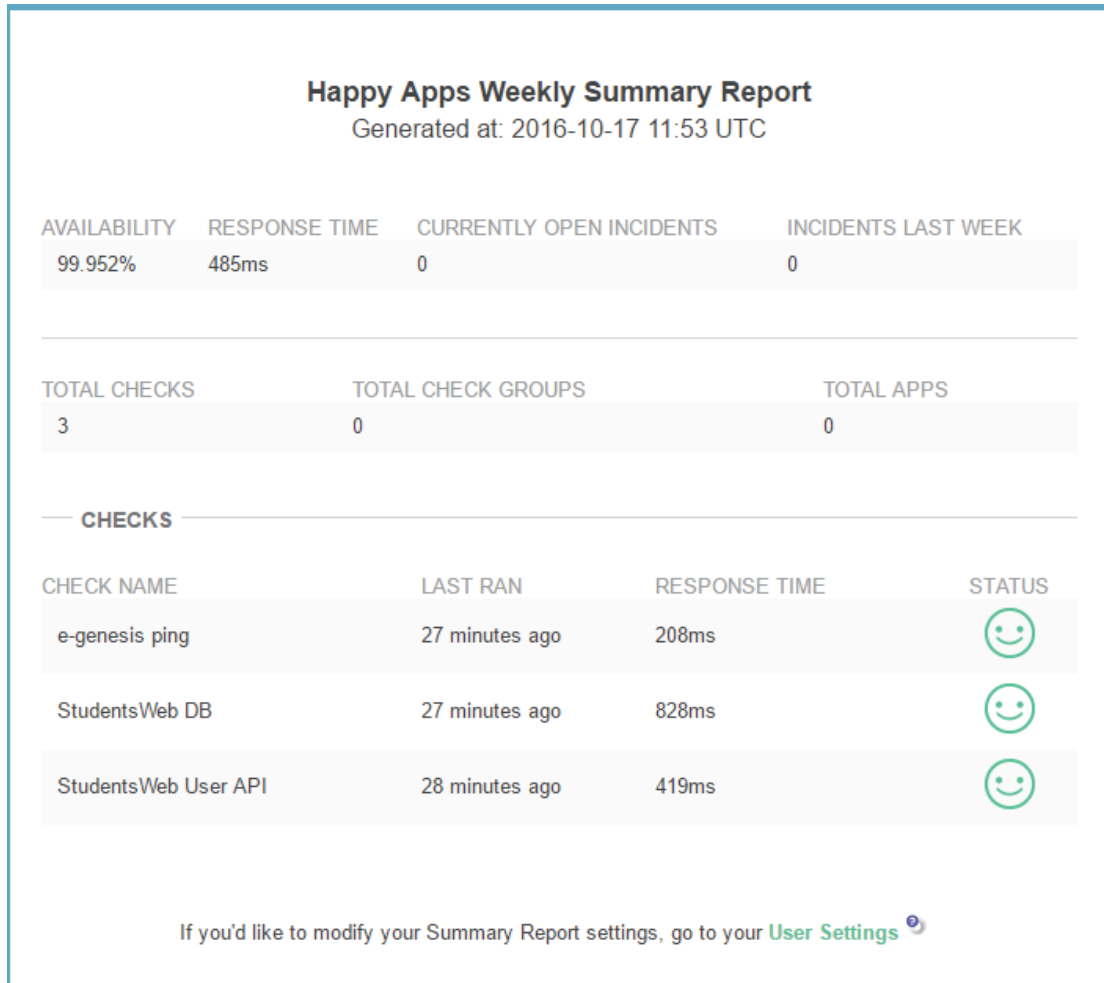


Εικόνα 28: Email το οποίο μας δείχνει κάποια βασικά «ζωτικά» στατιστικά, για να παρακολουθούμε την εφαρμογή.

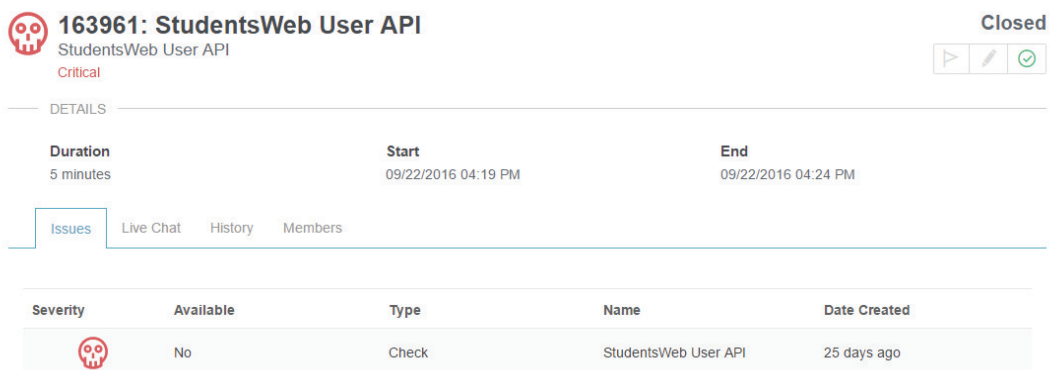
Happy Apps

Με την υπηρεσία Happy Apps όταν «κάτι δεν πάει καλά» στους servers μας ενημερωνόμαστε (στην δωρεάν έκδοση) μέσω email εντός 5 λεπτών δείχνοντας μας το πρόβλημα τότε συνέβη κλπ. Είναι πολύ αξιόπιστο όσο το χρησιμοποιώ και δεν με έχει «ενοχλήσει» άσκοπα.

Ανά εβδομάδα μας στέλνει (αν εμείς το θέλουμε) ένα ενημερωτικό Email για να μας δείξει την κατάσταση των servers/εφαρμογών [11].



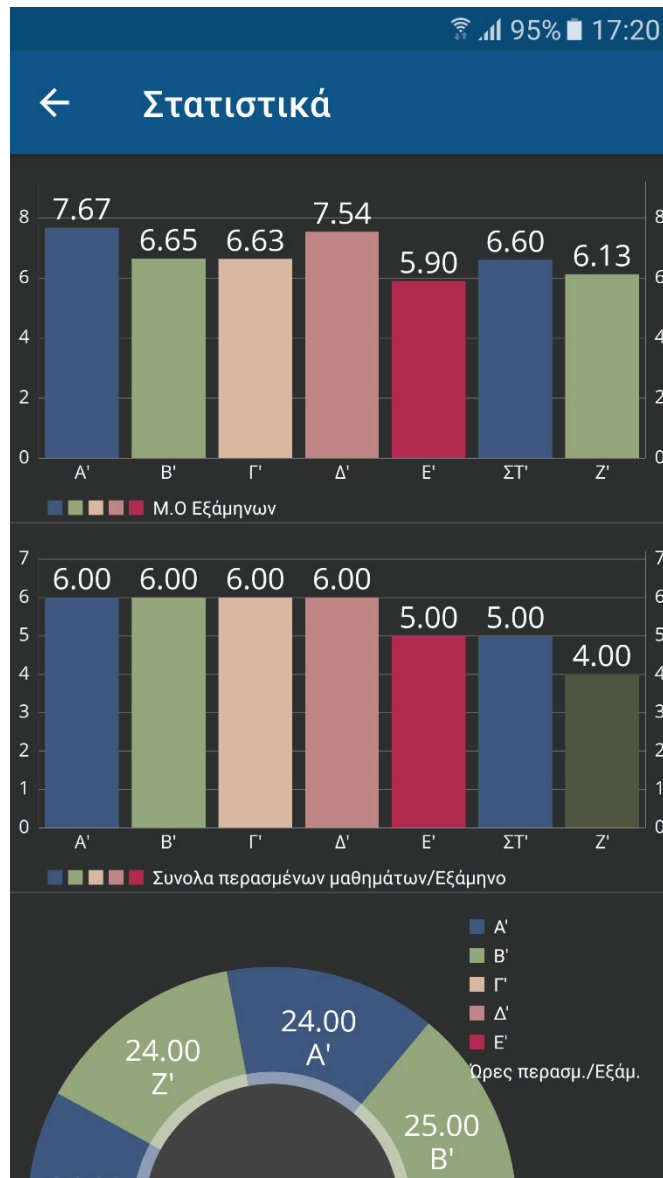
Εικόνα 29: Εβδομαδιαία αναφορά που μας ενημερώνει για την διαθεσιμότητα των υπηρεσιών μας.



Εικόνα 30: Απεικόνιση σφάλματος.

MPAndroidChart

Την βιβλιοθήκη MPAndroidChart την χρησιμοποιούμε σαν Proof Of Concept σε στατιστικά του εκάστοτε φοιτητή.



Εικόνα 31: Εικόνα στατιστικών ενός φοιτητή.

Τα χρώματα, οι μπάρες, τα γράμματα είναι όλα παραμετροποιήσιμα και στην διάθεση του προγραμματιστή για να τα φτιάξει όπως ακριβώς θέλει [13].

Material Drawer

Το Materialdrawer στην εφαρμογή μας χρησιμοποιείται για το κεντρικό μας μενού. Πάνω στο Materialdrawer έχουμε στηρίξει την πλοήγηση του χρήστη μας ώστε να προσφέρουμε μια εμπειρία «όσο πιο κοντινή γίνεται στο Android» (π.χ. Google Play Store). Ολοένα και περισσότερες εφαρμογές προσθέτουν drawer (συρτάρι) στην εφαρμογή τους γιατί είναι «βολικό», πάντα κρυμμένο (χωρίς να μπαίνει στην μέση όταν δεν το χρειαζόμαστε), άλλα

πάντα εκεί με μια κίνηση του χεριού μας (σέρνοντας από αριστερά προς τα δεξιά).

Η ενσωμάτωσή του ήταν πολύ εύκολη στην εφαρμογή και ένας από τους λόγους που χρησιμοποιήθηκε ήταν η εύκολη ενσωμάτωσή του, χρησιμοποιεί minimum target SDK 14, είναι βασισμένο στα πρότυπα του “material design” και για αυτό προτιμήθηκε έναντι του launcher με το default API της Google.

CircleImageView

Χρησιμοποιείται ώστε να είναι οι εικόνες των προφίλ στην δραστηριότητα «Συζήτηση» στρογγυλές, προς το παρόν σαν εικόνα προφίλ για κάθε χρήστη χρησιμοποιείται το λογότυπο της εφαρμογής[14].

Android About Page

Μας επιτρέπει να δημιουργήσουμε μια «Περί» δραστηριότητα πολύ απλά και γρήγορα. Είναι παραμετροποιήσιμη και υποστηρίζει συγκεκριμένα Intents για άνοιγμα (π.χ. Ανοίγει την ιστοσελίδα στον Browser, ανοίγει την εφαρμογή του Email για αποστολή Email προς τον προγραμματιστή κ.ά.) [15].

Show/Hide Password EditText

Η βιβλιοθήκη αυτή βοηθάει στο να είμαστε σε θέση να προβάλουμε ή όχι τον κωδικό που έχουμε βάλει στην «Είσοδο». Είναι χρήσιμη σαν λειτουργία κυρίως για αυτούς που μπορεί να έχουν μεγάλα δάκτυλα, ή συσκευές με μικρή οθόνη και συνηθίζουν να κάνουν λάθη στην πληκτρολόγηση πράγμα που μπορεί να δυσαρεστήσει τους χρήστες [16].



Εικόνα 32: Επίδειξη του Show/Hide Password EditText

Glide

Έχει προστεθεί για να χρησιμοποιηθεί στην κατηγορία «Συζήτηση» ώστε να ανταλλάσσονται π.χ. σημειώσεις (εικόνες, σύνδεσμοι, κ.ά.). Δεν έχει ενεργοποιηθεί ακόμα για λόγους ασφαλείας (φοιτητές μπορεί να το καταχραστούν οπότε ίσως χρειαστούν διαχειριστές, πρέπει να γίνει επαναξιολόγηση στο Google Play λόγω “chatting” υπηρεσίας κ.ά.), εν δυνάμει όμως μπορεί να ενεργοποιηθεί [17].

Σημείωση: Για περαιτέρω πληροφορίες για τις βιβλιοθήκες, ο ενδιαφερόμενος μπορεί να ανατρέξει στο Παράρτημα Β.

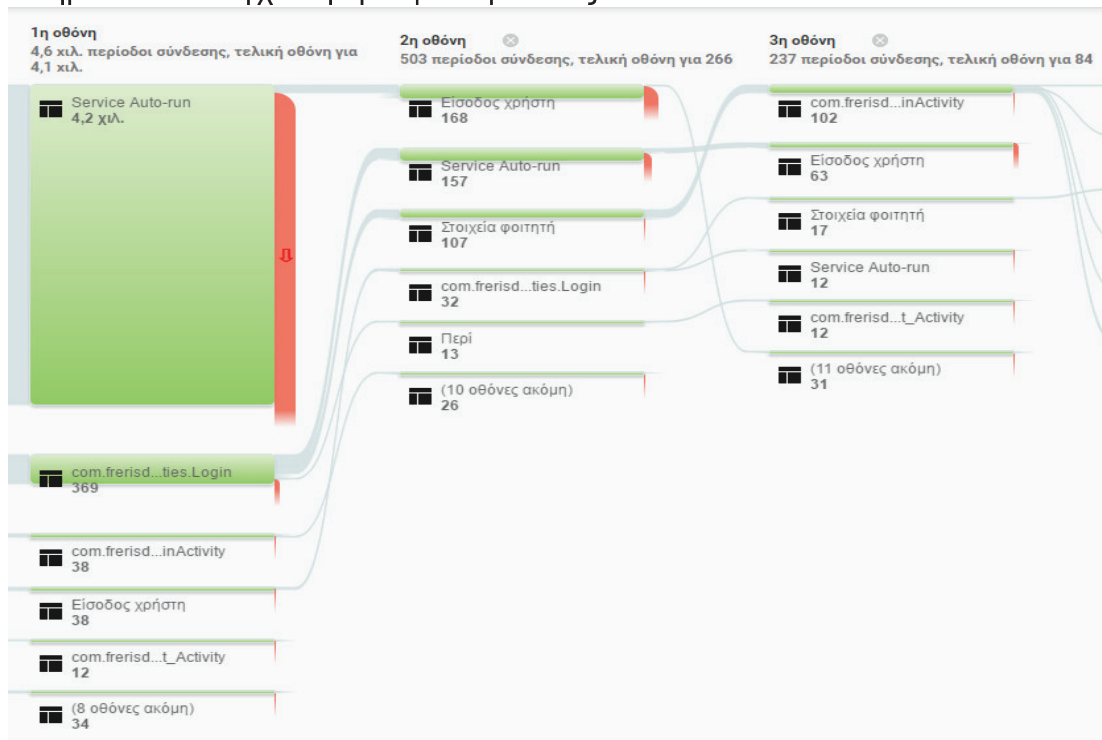
Κεφάλαιο 5

Στο κεφάλαιο αυτό θα αναφερθούμε στα στοιχεία χρήσης της εφαρμογής, την αξιολόγηση της, το στάδιο του Alpha/Beta Testing, αλλά και στην «εμπορική διάθεση» της εφαρμογής, στο Google Play Store.

Στοιχεία Χρήσης και Αξιολόγηση της Εφαρμογής

Κατά την διάρκεια της ανάπτυξης, συμμετείχαν πάνω από 22 συσκευές, ώστε να γίνει έλεγχος πολλών διαστάσεων οθονών, πολλών εκδόσεων, μοντέλων κ.ά.. Έγιναν πάνω από 1455 είσοδοι στην υπηρεσία και αυτό αφορά μόνο όσους επέλεξαν την αποστολή στατιστικών.

Από την στιγμή που ενεργοποιήθηκε το Google Analytics για την εφαρμογή, μέχρι και σήμερα όπως είπαμε και είχαμε προβλέψει πιο πάνω, βλέπουμε πως οι χρήστες θέλουν να είναι ενήμεροι για την βαθμολογία τους συνεχώς, αλλά την εφαρμογή την ανοίγουν πολύ πιο σπάνια από ότι έχουν βάλει την υπηρεσία να ελέγχει την βαθμολογία τους.



Εικόνα 33: Ροή συμπεριφοράς της εφαρμογής.

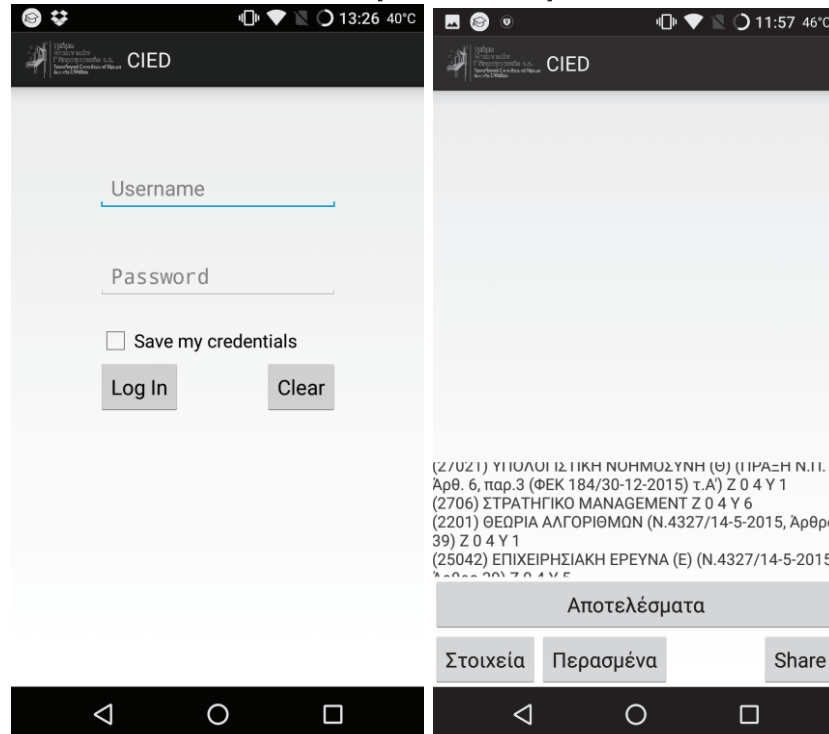


Εικόνα 34: (Ανοιχτό μπλε) Μέση περίοδος σύνδεσης του χρήστη και (σκούρο μπλε) οθόνες ανά περίοδο σύνδεσης.

Η αρχή, η εξέλιξη, και το τώρα

Η ανάπτυξη της εφαρμογής ξεκίνησε σαν χόμπι, και πέρασε αρκετά στάδια μέχρι να φτάσει στο σημείο που βρίσκεται σήμερα. Παρακάτω παραθέτω κάποιες εικόνες μέσα σε αυτόν τον ένα-ενάμιση χρόνο όπου βλέπουμε την εξέλιξη της εφαρμογής σταδιακά.

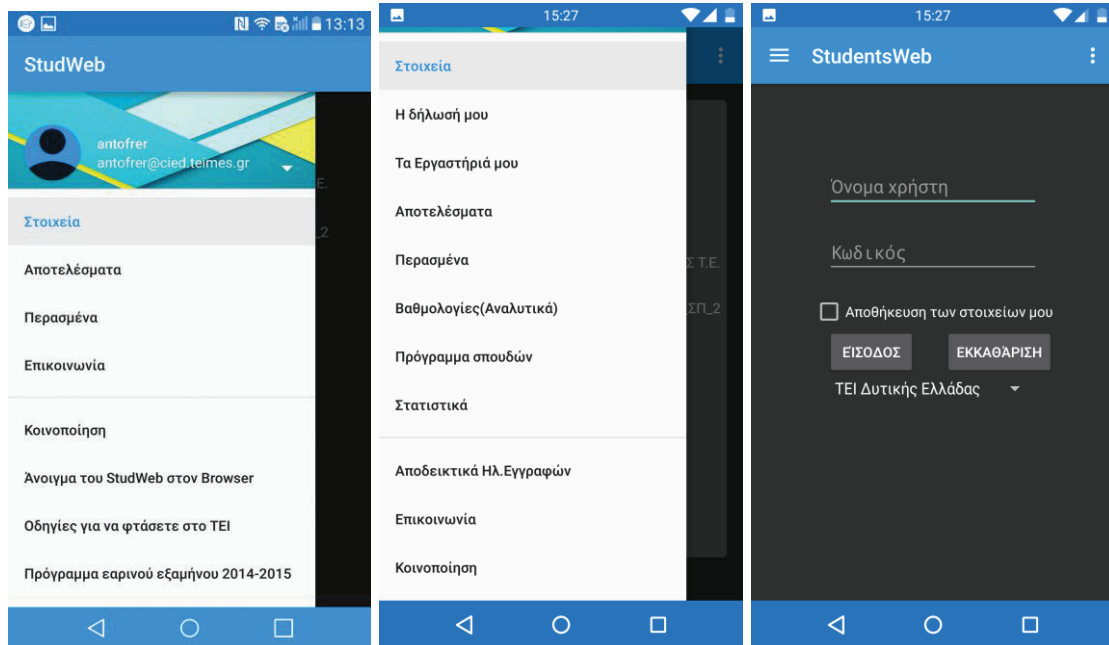
Pre-alpha έκδοση:



Εικόνα 35: Γραφικό περιβάλλον της εφαρμογής σε Pre-alpha έκδοση.

Στην Pre-alpha γινόταν απλά δοκιμή της βιβλιοθήκης jsour ώστε να δούμε αν γίνεται σωστά η ανάλυση, και δεν είχαμε ασχοληθεί με το γραφικό περιβάλλον της εφαρμογής καθόλου.

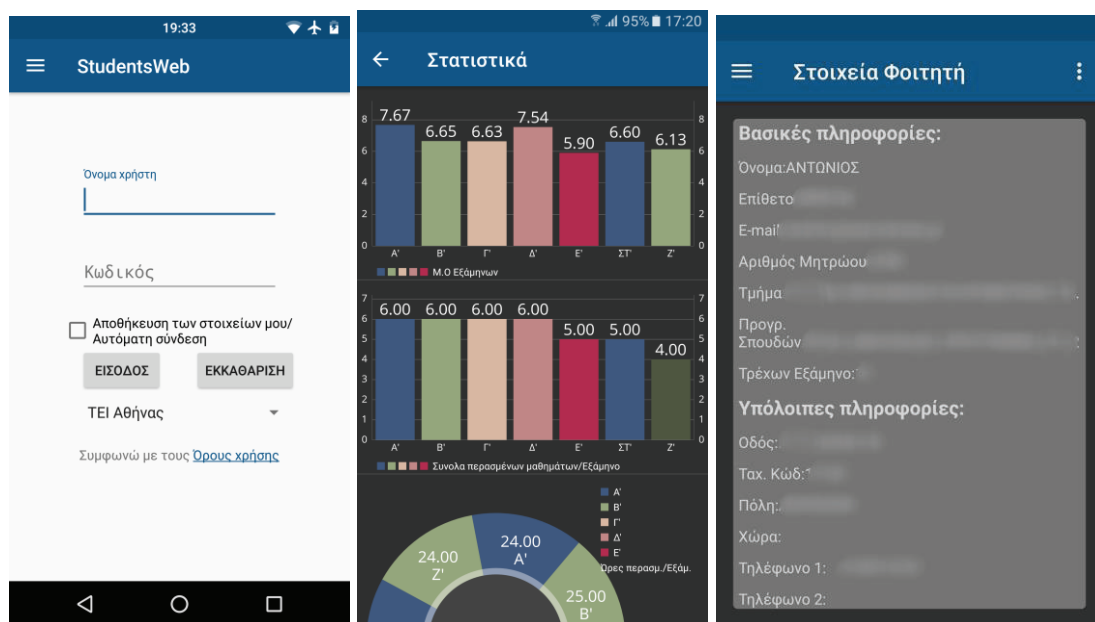
Alpha έκδοση:



Εικόνα 36: Γραφικό περιβάλλον της εφαρμογής σε Alpha έκδοση.

Στην Alpha έκδοση αρχικά είχαμε τα πολύ βασικά μενού όπως βλέπετε στην συνέχεια προστέθηκαν πολλά παραπάνω, όπως και επιλογή για το τμήμα που θέλουμε να επιλέξουμε. Το «θέμα» της εφαρμογής ήταν ένα ματεριαλιστικό «ανοιχτό» μπλε μαζί με σκούρο γκρι.

Beta-RC (Release Candidate)/Κανονική έκδοση:



Εικόνα 37: Γραφικό περιβάλλον της εφαρμογής σε Beta/Κανονική έκδοση.

Στις εκδόσεις αυτές τελειοποιήθηκαν όλα τα μενού διορθώθηκαν αμέτρητα bugs τα οποία μπορεί κάποια στιγμή στο μέλλον να «ενοχλούσαν» τον χρήστη, προστέθηκαν δύο θέματα (μαύρο-άσπρο), με το σκούρο μπλε να παίρνει την θέση του ανοιχτού μπλε κάνοντας την περιήγηση στην εφαρμογή πιο ξεκούραστη στα μάτια μας.

Alpha Testing

Το Alpha Testing διήρκησε αρκετά (κυρίως λόγω προσωπικού κωλύματος). Οι Alpha Testers ήταν φοιτητές (άλλοι από το ίδιο, άλλοι από διαφορετικό τμήμα) με διαφορετικές Android συσκευές και εκδόσεις ο καθένας.

Οι Alpha Testers ήταν οι εξής:

- Αθανάσιος Καραθανάσης
- Γιώργος Πετρόπουλος
- Ιωάννης Κολεύρης
- Κωνσταντίνος Ντουντούμης

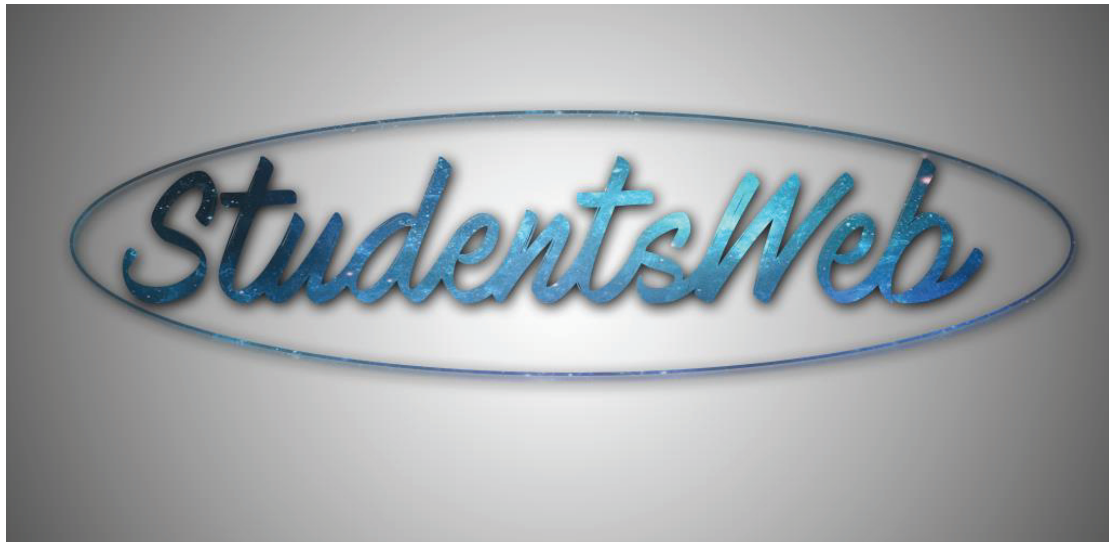
Επειδή χρησιμοποιούμε ουσιαστικά έναν έμμεσο τρόπο για να αντλήσουμε τα δεδομένα από το StudentsWeb (jsoup) και όχι κάποιο API της cardisoft, μέσα σε αυτήν την περίοδο κυκλοφόρησαν 23 αναβαθμίσεις στην εφαρμογή κάνοντας την ολοένα και πιο σταθερή με πολλές διορθώσεις και προσθήκες χαρακτηριστικών.

Οι Alpha Testers έκαναν αναφορά μέσω του Feedback που έχει το Google Play, μέσω προσωπικής συνομιλίας μαζί μου, αλλά και από τα δεδομένα που ερχόντουσαν από το CrashLytics.

Google Play Store

Η εφαρμογή μέχρι πριν λίγο καιρό δεν είχε βγει σε διάθεση και ήταν ακόμα σε Alpha φάση, και ο λόγος ήταν πως ήθελα να βγει κάτι ολοκληρωμένο προς τα έξω.

Έγινε μια υποτυπώδης δημιουργία γραφικού για τις ανάγκες του Google Play το οποίο είναι το εξής:



Εικόνα 38: Το banner της εφαρμογής στο Play Store.

Δημιουργήθηκε επίσης και ένα βίντεο που επιδεικνύει τις δυνατότητες της εφαρμογής το οποίο βρίσκεται εδώ:

<https://www.youtube.com/watch?v=NQVcZOS6IqM>

Σημείωση: Τα τραγούδια που προστέθηκαν, για να μην είναι τόσο «μονότονο» το βίντεο, είναι Royalty Free, και σε πολλά σημεία υπάρχει “blur” για λόγους ιδιωτικότητας.


Επίσης έχουν προστεθεί στο Play Store φωτογραφίες μέσα από την εφαρμογή για να μπορούν να βλέπουν οι χρήστες το γραφικό της περιβάλλον, πριν προβούν σε εγκατάσταση.

Τέλος ο χρήστης, πριν την εγκατάσταση μπορεί να διαβάσει την περιγραφή για να δει παραπάνω πληροφορίες για την εφαρμογή καθώς και επεξήγηση για τα δικαιώματα που ζητάει η εφαρμογή κ.ά..

Google Play Developer Console

Για την εγγραφή στο Google Play Developer Console χρειάστηκε να πληρώσουμε ένα «τέλος εγγραφής» της τάξης των 25 ευρώ, τα οποία είναι «εφ’ όρου ζωής» και δεν θα ζητηθούν άλλα χρήματα αργότερα.

Η εφαρμογή στην κονσόλα παρουσιάζεται κάπως έτσι:

ΟΝΟΜΑ ΕΦΑΡΜΟΓΗΣ	ΤΙΜΗ	ΕΝΕΡΓΕΣ / ΣΥΝΟΛΙΚΕΣ ΕΓΚΑΤΑΣΤΑΣΕΙΣ	ΜΕΣΗ ΒΑΘΜΟΛ. / ΣΥΝΟΛΟ #
 StudentsWeb 1.3.15	Δωρεάν	7 / 31	★ 5,00 / 1

Εικόνα 39: Απεικόνιση καταχωρημένης εφαρμογής στο Google Developer Console.

Εφόσον πατήσουμε μέσα στην κονσόλα για την συγκεκριμένη εφαρμογή, είμαστε σε θέση να παραμετροποιήσουμε, δούμε πολλά πεδία, όπως:

Πίνακας ελέγχου:

Εδώ βλέπουμε μια συνοπτική εικόνα για την εφαρμογή μας, με λεπτομερή στατιστικά κ.ά.

Στατιστικά στοιχεία:

Βλέπουμε πάλι στατιστικά στοιχεία πιο αναλυτικά, με γραφήματα, και περισσότερες επιλογές.

Απόκτηση χρήστη:

Είμαστε σε θέση να δούμε από πού προήλθαν οι εγκαταστάσεις, και επίσης σε περίπτωση που κάνουμε/χρησιμοποιούμε καμπάνιες AdWords να τις διαχειριστούμε από εκεί. Στο μέλλον στην εφαρμογή θα μπορούσαν να προστεθούν διαφημίσεις σε σημεία που δεν παρεμβαίνουν στην εμπειρία χρήσης του χρήστη, ώστε να υπάρχουν κάποια έσοδα για την συντήρηση των server και domain names.

Βαθμολογίες και κριτικές:

Βλέπουμε τις βαθμολογίες που έχουν αφήσει οι χρήστες για την εφαρμογή μας, μπορούμε να απαντήσουμε σε κριτικές που μπορεί να έχουν γίνει και να δούμε και στατιστικά για τις αξιολογήσεις από ποιες χώρες προήλθαν, εκδόσεις Android κ.ά..

Διακοπές λειτουργίας και μηνύματα ANR:

Βλέπουμε τυχόν σφάλματα τα οποία μπορεί να έχουν προκληθεί σε χρήστες και τυχόν μηνύματα που μπορεί να έχουν στείλει οι χρήστες (π.χ. τι έκαναν και προέκυψε το σφάλμα) ώστε να βοηθήσουν στην αντιμετώπιση του σφάλματος. Τα σφάλματα έχουν στατιστικά όπως για παράδειγμα σε πιά συσκευή έγιναν, την έκδοση της συσκευής, την έκδοση της εφαρμογής, το σύνολο των αναφορών σε περίπτωση που συμβαίνει σε πολλούς χρήστες κ.ά..

Συμβουλές βελτιστοποίησης:

Συμβουλές τις Google για να γίνει καλύτερη η εφαρμογή, φέρ' ειπείν να σχεδιαστεί για tablet, να την προωθήσουμε με AdWords, να την μεταφράσουμε σε πολλές γλώσσες, κ.ά..

Αναφορά πριν από την κυκλοφορία:

Με αυτήν την υπηρεσία μπορούμε πριν δημοσιεύσουμε την εφαρμογή μας σε κάποιο κανάλι διανομής, να είμαστε σε θέση να δοκιμάσουμε την εφαρμογή μας σε ένα μεγάλο εύρος εκδόσεων Android, συσκευών, αλλά και αναλύσεων ώστε πριν την δημοσιεύσουμε να κάνουμε έναν πρώιμο έλεγχο για τυχόν σφάλματα. Κατά την διάρκεια της ανάπτυξης της εφαρμογής χρησιμοποιήθηκε αυτή η υπηρεσία και διορθώθηκαν αρκετά σφάλματα, όμως επειδή η

εφαρμογή στην πρώτη οθόνη ζητάει όνομα χρήστη και κωδικό δεν ήμασταν σε θέση να περάσουμε πέρα από την είσοδο του χρήστη. Για να γινόταν κάτι τέτοιο ή θα έπρεπε να αγοράσουμε το πακέτο που μας πρόσφεραν (έναντι κάποιου αντίτιμου) ή να ανεβάσουμε μια έκδοση με τα στοιχεία μας ήδη καταχωρημένα, κάτι που δεν έγινε για λόγους ασφαλείας.

Διαγνωστικά στοιχεία FCM:

Χρησιμοποιείται για την αντιμετώπιση προβλημάτων μηνυμάτων που αποστέλλονται μέσω του Firebase Cloud Messaging.

Προσφορές:

Σε περίπτωση που η εφαρμογή μας ήταν επί πληρωμή ή είχε προϊόντα επί πληρωμής θα μπορούσαμε να χειριστούμε διάφορες προσφορές για προώθηση κ.ά..

APK:

Εδώ ανεβάζουμε τα αρχεία APK (Android application package). Υπάρχουν τρεις βασικές καρτέλες οι οποίες είναι:

- Δοκιμή Alpha – Συνήθως ασταθής, χρήσιμη στο να δείχνει τις δυνατότητες του προγράμματος.
- Δοκιμή Beta – Συνήθως ολοκληρωμένο από την άποψη των δυνατοτήτων, παρόλα αυτά είναι πιθανόν να περιέχει αρκετά σφάλματα (bugs).
- Παραγωγή – Το πρόγραμμα ολοκληρωμένο συνήθως γνωστό ως «έκδοση σταθερότητας» (stable release).

Η δοκιμή Alpha και Beta μπορούν να γίνουν είτε σαν κλειστές δοκιμές, δηλαδή να προσθέσουμε μεμονωμένους χρήστες μέσω διευθύνσεων ηλεκτρονικού ταχυδρομείου, ή μέσω Google Plus, είτε σαν ανοιχτές δοκιμές. Οι ανοιχτές δοκιμές επιτρέπουν σε χρήστες να συμμετάσχουν μέσω ενός συγκεκριμένου συνδέσμου.

Στην περίπτωση της εφαρμογής αυτής, ανέβηκαν 30 Beta εκδόσεις σε ανοιχτή δοκιμή και 2 στην «Παραγωγή».

Διαχείρισης κυκλοφοριών:

Σε περίπτωση που το εύρος της εφαρμογής είναι μεγάλο, ή για άλλους λόγους, για να αποφύγουμε τυχόν σφάλματα μεγάλου μεγέθους μπορούμε να διαθέσουμε την εφαρμογή σταδιακά σε περιοχές ώστε να αποφύγουμε τυχόν μαζικά σφάλματα και οι χρήστες δυσαρεστηθούν. Έτσι κυκλοφορούμε την εφαρμογή σε μια «μικρή» γεωγραφική περιοχή ώστε να είμαστε σε θέση να δούμε αν είναι σταθερή και μετά μπορούμε να την διαθέσουμε πλήρως στο κοινό.

Καταχώριση καταστήματος:

Σε αυτήν την επιλογή εισάγουμε όλα τα στοιχεία της εφαρμογής, το όνομα, την περιγραφή, τις μεταφράσεις το Play Store, τα γραφικά, το βίντεο προώθησης την κατηγοριοποίηση, τα στοιχεία επικοινωνίας μας, και πολλά άλλα.

Αξιολόγηση περιεχομένου:

Κάνοντας ένα ερωτηματολόγιο και υποβάλλοντας το στην Google για την εφαρμογή μας, η Google μας αξιολογεί την καταλληλότητα της εφαρμογής μας προς το κοινό.

Τιμολόγηση και διανομή:

Ορίζουμε αν η εφαρμογή είναι επί πληρωμή, σε πόσες χώρες θα γίνεται διανομή, αν περιέχει διαφημίσεις κ.ά..

Προϊόντα εντός εφαρμογής:

Σε περίπτωση που η εφαρμογή μας είχε προϊόντα εντός της θα τα ορίζαμε από εδώ. Για να γίνει κάτι τέτοιο χρειάζεται να ορίσουμε έναν λογαριασμό εμπόρου.

Υπηρεσίες και API:

Μπορούμε να χειριστούμε τα αναγνωριστικά των υπηρεσιών και API, όπως του Firebase Cloud Messaging, το κλειδί της άδειας της εφαρμογής κ.ά.

Υπογράφοντας την εφαρμογή

Το Android απαιτεί όλα τα APK να έχουν ψηφιακή υπογραφή με ένα πιστοποιητικό πριν εγκατασταθούν σε κάποια συσκευή.

Ένα ψηφιακό πιστοποιητικό περιέχει ένα δημόσιο κλειδί από ένα ζεύγος δημόσιου/ιδιωτικού κλειδιού όπως επίσης και άλλα metadata τα οποία ταυτοποιούν τον ιδιοκτήτη του κλειδιού. Ο ιδιοκτήτης του πιστοποιητικού έχει το αντίστοιχο ιδιωτικό κλειδί.

Όταν υπογράφουμε ένα APK, το εργαλείο υπογραφής προσάπτει το δημόσιο κλειδί στο APK. Το δημόσιο αυτό κλειδί υπηρετεί σαν «αποτύπωμα» που ενώνει μοναδικά το APK με το αντίστοιχο ιδιωτικό κλειδί. Αυτή η διαδικασία βοηθάει το Android να σιγουρέψει ότι οι μελλοντικές εκδόσεις APK είναι αυθεντικές, και προέρχονται από τον ιδιοκτήτη της εφαρμογής [7].

Κεφάλαιο 6

Στο κεφάλαιο αυτό αναφέρουμε τους περιορισμούς και τα προβλήματα τα οποία συναντήσαμε.

Περιορισμοί και προβλήματα

Οι περιορισμοί που συνάντησα κατά την διάρκεια ανάπτυξης της εφαρμογής ήταν ότι, δεν μπορούσα να βρω τρόπο να κάνω ανανέωση/δήλωση νέου εξαμήνου από το κινητό. Και αυτό, λόγω του ότι η βιβλιοθήκη jsoup απλά «κατεβάζει τον ιστότοπο» και από εκεί και πέρα εγώ είμαι σε θέση να προβάλλω τα δεδομένα. Δεν μπόρεσα να καταφέρω να κάνω το πρόγραμμα να στέλνει δηλώσεις προς τον server της cardisoft. Κάτι το οποίο με πρόσβαση στο API (όπως ανέφερα πιο πάνω) πιστεύω μπορεί να γίνει πολύ εύκολα.

Άλλο ένα πρόβλημα που αντιμετώπισα αλλά υπήρξε λύση, ήταν οι διάλογοι φόρτωσης (ProgressDialogs) όταν ανακτούσαμε δεδομένα από την πλατφόρμα. Περιστρέφοντας το κινητό η εφαρμογή έκλεινε απρόσμενα όπως αναφέρθηκε και στο στάδιο της υλοποίησης. Λύση στο πρόβλημα αυτό ήταν να «απενεργοποιήσουμε» για σύντομο χρονικό διάστημα (μέχρι να τελειώσει ο διάλογος φόρτωσης) την περιστροφή οθόνης έτσι ώστε να μην υπάρχει απρόσμενο κλείσιμο και στην συνέχεια να την ενεργοποιήσουμε ξανά. Προσωπικά πιστεύω ότι στα επόμενα SDK κάτι τέτοιο πρέπει να γίνεται αυτόματα έτσι ώστε ο προγραμματιστής να μην χρειάζεται να κάνει τέτοια «μπαλώματα» και να «καταναλώνει» πολύτιμο χρόνο από τα Project του.

Κεφάλαιο 7

Στο κεφάλαιο αναφερόμαστε στα συμπεράσματα και τις μελλοντικές επεκτάσεις της εφαρμογής.

Συμπεράσματα και Μελλοντικές ΕΠΕΚΤΑΣΕΙΣ

Συμπεράσματα:

Σε μια εποχή που περισσότερο χρησιμοποιούμε τα κινητά μας παρά τους σταθερούς Η/Υ, όλα πλέον τα βλέπουμε/διαχειριζόμαστε on-the-go. Η εφαρμογή StudentsWeb έρχεται για να βοηθήσει τον φοιτητή με ένα Material-like περιβάλλον στο οικοσύστημα του Android, επιτρέποντας στον φοιτητή να έχει από μία και μόνο εφαρμογή, οτιδήποτε χρειάζεται για το αντίστοιχο Α.Ε.Ι/Τ.Ε.Ι του, από την προβολή βαθμολογίας μέχρι και κατέβασμα σημειώσεων.

Μελλοντικές επεκτάσεις:

Η εφαρμογή ήταν προγραμματισμένο να σχεδιαστεί σε Android αλλά και σε iOS. Κάτι το οποίο δεν πραγματοποιήθηκε διότι η Apple ζήτηγε για να «γραφτούμε» σαν developers στον ιστότοπο της περίπου 100 ευρώ. Το Google Play Developer Console ζήτησε περίπου 25 ευρώ χωρίς έξτρα χρεώσεις στο μέλλον (τα οποία και δόθηκαν) για την υλοποίηση της πτυχιακής.

Μελλοντικά θα μπορούσε να γίνει και **υποστήριξη της iOS πλατφόρμας** καθότι υπάρχουν πολλοί εν δυνάμει χρήστες στο οικοσύστημα αυτό.

Η εφαρμογή StudentsWeb θα μπορούσε να κερδίσει ακόμα **παραπάνω κοινό κάνοντας καμπάνιες Adwords** ώστε με αυτόν τον τρόπο να προσελκύσει περισσότερο κόσμο. Ο ανταγωνισμός θα μπορούσε να «χτυπηθεί» στο ότι η εφαρμογή εστιάζει, πέρα από τους βαθμούς, και στο κατέβασμα σημειώσεων, στην συζήτηση μεταξύ φοιτητών, ενημέρωση ειδοποίηση σε περίπτωση που βγήκε κάποιο αποτέλεσμα μαθήματος κ.ο.κ..

Θα μπορούσαμε επίσης να έρθουμε σε επικοινωνία με πολλούς διαχειριστές των τμημάτων αυτών, ώστε να μας δώσουν πρόσβαση στο API. Κάτι που θα κάνει την άντληση δεδομένων από την ηλεκτρονική γραμματεία πολύ πιο εύκολη και ασφαλή (από την άποψη σφαλμάτων).

Μια άλλη καινοτομία που ήταν στα σχέδια και μπορεί να αναπτυχθεί είναι αν η εκάστοτε γραμματεία δίνει ένα xml/json πρόγραμμα εξαμήνου το οποίο

μπορούμε να το επεξεργαστούμε ώστε να έχουμε μέσα στην εφαρμογή ένα «ημερολογιακό πρόγραμμα» το οποίο θα μας δείχνει κάθε πότε έχει μάθημα ο φοιτητής.

Ένα χαρακτηριστικό το οποίο θα ακολουθήσει σε λίγο καιρό είναι η **δυνατότητα της offline χρήσης**. Ο φοιτητής θα είναι σε θέση να βλέπει τις βαθμολογίες του, την δήλωσή του, το πρόγραμμα σπουδών του και πολλά άλλα χωρίς να χρειάζεται να είναι συνδεδεμένος στο διαδίκτυο. Η εφαρμογή θα ανανεώνεται αυτόματα όταν ο χρήστης (είναι στο διαδίκτυο και) το θελήσει.

Αν υπάρχει κάποια συνεννόηση για το πως θα είναι δομημένος ο τίτλος του μαθήματος (π.χ. «Υπολογιστική Νοημοσύνη (Ε) Δευτέρα 9:00-11:00) στα εργαστήρια του κάθε τμήματος, θα μπορούμε να είμαστε σε θέση να προτείνουμε στον χρήστη να δέχεται **ειδοποιήσεις για το πότε έχει εργαστήριο**. (π.χ. η εφαρμογή να τον ειδοποιεί 2-3-4 ώρες πριν), ώστε να μην χάσει το μάθημα.

Ένα χαρακτηριστικό το οποίο θα προστεθεί ανάλογα με την ζήτηση που θα υπάρχει είναι το φιλτράρισμα των σημειώσεων ανά εξάμηνο/καθηγητή/χρονολογία.

Δημιουργία κάποιου είδους **leaderboard** που θα μπορεί για παράδειγμα ο φοιτητής να βλέπει την επίδοσή του (ανά εξάμηνο, μάθημα, χρονιά) έτσι ώστε να ξέρει που περίπου «βρίσκεται» σε σχέση με τους άλλους, πράγμα που μπορεί να τον παροτρύνει να βελτιώσει την επίδοσή του στα μαθήματα λόγω συναγωνισμού. Παράλληλα θα μπορούσε να υπάρχει ένα σύστημα το οποίο θα του δίνει **επιβραβεύσεις / achievements** (επιτεύγματα) ανάλογα με τον στόχο που πραγματοποιεί. Για παράδειγμα όταν πάρει το πρώτο του 10άρι ή όταν περάσει 3 ή παραπάνω μαθήματα στην σειρά κ.ά..

Παράρτημα Α

Η εφαρμογή στο «Μικροσκόπιο»

build.gradle

Στο build.gradle (Project) έχουμε προσθέσει τα εξής dependencies:

- classpath 'com.android.tools.build:gradle:2.2.2'
- classpath 'com.google.gms:google-services:3.0.0'

Το πρώτο αφορά τα «εργαλεία» με τα οποία κάνει compile η εφαρμογή, και το δεύτερο είναι σημαντικό ώστε να δουλεύουν τα Google Analytics, η υπηρεσία Firebase κ.ά.. Το συγκεκριμένο “dependency” απαιτεί να βάλουμε στον αρχικό κατάλογο του Project της εφαρμογής μας, ένα αρχείο το οποίο ονομάζεται “google-services.json” και περιέχει API κλειδιά και άλλα αναγνωριστικά για την εφαρμογή μας.

Στο build.gradle (Module:app) τα dependencies που δηλώσαμε ήταν τα εξής:

- 'com.android.support:cardview-v7:24.0.0' Υπεύθυνο για τα CardView
- 'com.android.support:appcompat-v7:24.0.0' Υπεύθυνο για την υποστήριξη των «νεότερων» βιβλιοθηκών της Google σε παλαιότερες συσκευές
- 'com.mikepenz:materialdrawer:5.3.6' Ματεριαλιστικό Drawer/συρτάρι (βλ. Βιβλιοθήκες)
- 'com.github.PhilJay:MPAndroidChart:v2.2.4' Βιβλιοθήκη για Στατιστικά (βλ. Βιβλιοθήκες)
- 'com.github.medyo:android-about-page:1.1.0' Βιβλιοθήκη για το «Περί» της εφαρμογής μας (βλ. Βιβλιοθήκες)
- 'com.google.android.gms:play-services-analytics:9.2.0' Βιβλιοθήκη για τα Google Analytics (βλ. Βιβλιοθήκες)
- 'com.google.firebase:firebase-core:9.2.0' Βιβλιοθήκη για τα βασικά μέρη της υπηρεσίας Firebase (βλ. Βιβλιοθήκες)
- 'com.google.android.gms:play-services-auth:9.2.0' Βιβλιοθήκη που είναι υπεύθυνη για την ταυτοποίηση με λογαριασμό Google
- 'com.github.scottyab:showhidepasswordedittext:0.8' Βιβλιοθήκη για διαφορετικό πεδίο κωδικού (βλ. Βιβλιοθήκες)

/app/manifests

Κάθε εφαρμογή πρέπει να έχει το `AndroidManifest.xml`. Το αρχείο αυτό δίνει ουσιώδεις πληροφορίες για την εφαρμογή στο λειτουργικό σύστημα Android, όπου πρέπει να υπάρχει πριν εκτελεστεί η εφαρμογή [8].

Μέσα σε όλα τα άλλα το `AndroidManifest.xml` κάνει τα εξής:

- Ονομάζει το Java πακέτο για την εφαρμογή. Το πακέτο αυτό παρέχει ένα χαρακτηριστικό αναγνωριστικό για την εφαρμογή
- Περιγράφει τα βασικά στοιχεία της εφαρμογής συμπεριλαμβάνοντας δραστηριότητες, υπηρεσίες, broadcast receivers, και content providers τα οποία συνθέτουν την εφαρμογή. Επίσης παραθέτει τις κλάσεις που χρησιμοποιούν αυτά τα στοιχεία και τα «δημοσιοποιεί» στο λειτουργικό Android να ώστε να είναι ενήμερο σε περίπτωση που ληφθεί π.χ. κάποιο Intent μήνυμα να είναι σε θέση να ενημερώσει τα στοιχεία της εφαρμογής
- Δηλώνει τα δικαιώματα που η εφαρμογή έχει ώστε να έχει πρόσβαση σε προστατευμένα μέρη του API και να αλληλοεπιδρά με άλλες εφαρμογές, όπως και επίσης δικαιώματα που χρειάζονται για να δουλέψουν σωστά τα στοιχεία της εφαρμογής
- Δηλώνει το μικρότερο επίπεδο του Android API που υποστηρίζει η εφαρμογή
- Δηλώνει την έκδοση της εφαρμογής, τις αναλύσεις που υποστηρίζει κ.ά.

/app/res/

Εξωτερικεύουμε τους πόρους όπως εικόνες, συμβολοσειρές κ.ά., έτσι ώστε να είμαστε σε θέση να τα διαχειριζόμαστε ανεξάρτητα από την εφαρμογή. Επίσης μπορούμε ανάλογα με τον τύπο της συσκευής (κινητό ή tablet), την έκδοση android, το μέγεθος της οθόνης και άλλων χαρακτηριστικών να έχουμε διαφορετικές προβολές οθόνης κ.ά. [9].

Σημείωση: Αποθήκευση xml αρχείων κατευθείαν στον φάκελο `/res/` θα προκαλέσει σφάλμα.

/app/res/drawable

Εδώ προστέθηκαν όλα τα «γραφικά» τα οποία απεικονίζονται στην εφαρμογή. Γραφικά όπως το logo της εφαρμογής, το εικονίδιο του λογαριασμού του χρήστη, κ.ά.. Πιο συγκεκριμένα:

- `grades.png` – Εμφανίζεται όταν υπάρχει ειδοποίηση για αλλαγή της βαθμολογίας.
- `hat.png` – Εμφανίζεται στην δραστηριότητα «συζήτηση».
- `headerbg.png` – Η φωτογραφία που είναι στο background του drawer.
- `ic_launcher.png` – Το κύριο εικονίδιο της εφαρμογής. Εμφανίζεται στο launcher αλλά και όταν λαμβάνουμε FCM.
- `icon512.png` – Εικονίδιο που εμφανίζεται στο λογαριασμό του χρήστη.

/app/res/layout

Το layout ή αλλιώς διάταξη οθόνης καθορίζει την αρχιτεκτονική ενός γραφικού περιβάλλοντος, ή δραστηριότητας ή ενός στοιχείου.

Σχεδόν όλα τα αρχεία περιλαμβάνουν Toolbar. Πιο συγκεκριμένα το καθένα:

- **`activity_about.xml`** – Το γραφικό περιβάλλον για την δραστηριότητα «Περί».
- **`activity_chat.xml`** – Το γραφικό περιβάλλον για την δραστηριότητα «συζήτηση». Περιέχει RecyclerView και ένα LinearLayout το οποίο περιλαμβάνει ένα EditText, και ένα κουμπί για το γράψιμο και την αποστολή μηνύματος.
- **`activity_contact.xml`** – Το γραφικό περιβάλλον για την δραστηριότητα «Επικοινωνία». Περιέχει ένα Button για την αποστολή, τρία TextView για τις περιγραφές των πεδίων, δύο EditText για να γράψει ο χρήστης τα σχόλια του, ένα RatingBar για να αξιολογήσει την εφαρμογή και ένα CheckBox για να συμφωνήσει με τους όρους χρήσης.
- **`activity_login.xml`** – Το γραφικό περιβάλλον για την δραστηριότητα «Είσοδος χρήστη» / “Login”. Αποτελείται από 2 TextInputLayout που προσφέρουν ένα ωραίο animation όταν ο χρήστης πατάει πάνω στο κείμενο για να εισάγει τα στοιχεία του, ένα EditText για το όνομα χρήστη, ένα ShowHidePasswordEditText το οποίο είναι το πεδίο για τον κωδικό και μας επιτρέπει αν θέλουμε να εμφανίσουμε τον κωδικό του χρήστη ή όχι για περισσότερη ευκολία, δύο Buttons το ένα για την είσοδο και το άλλο για τον καθαρισμό των πεδίων. Επίσης ένα CheckBox που μας επιτρέπει την αποθήκευση στοιχείων/αυτόματη σύνδεση, δύο TextView στο ένα προβάλλονται τυχόν ανακοινώσεις και στο άλλο οι όροι χρήσης, και ένα Spinner όπου διαλέγει ο χρήστης το τμήμα του.
- **`activity_main.xml`** – Το γραφικό περιβάλλον για την κύρια δραστηριότητα της εφαρμογής. Περιέχει SwipeRefreshLayout

(υπεύθυνο για την «χειρονομία» / gesture που τραβάμε προς τα κάτω και κάνει ανανέωση την δραστηριότητα), έναν CoordinatorLayout (όπου χάριν σε αυτό προβάλλουμε τα Snackbar) και ένα FrameLayout το οποίο ενσωματώνει ένα RecyclerView.

- **activity_sign_in.xml** – Το γραφικό περιβάλλον για την δραστηριότητα «Είσοδος με Google Account», που εμφανίζεται πριν την δραστηριότητα «Συζήτηση», όταν δεν έχουμε κάνει είσοδο με τα στοιχεία μας. Περιέχει ένα TextView και ένα κουμπί `gms.common.SignInButton` που αφορά το Google Sign In.
- **activity_tab_animation.xml** – Χρησιμοποιείται στις δύο δραστηριότητες που περιέχουν στυλ καρτελών, στην «Αναλυτική βαθμολογία» και «Οδηγός σπουδών». Περιέχουν CoordinatorLayout, AppBarLayout, Toolbar, TabLayout και ViewPager.
- **activity_upload_to_server.xml** – Η δραστηριότητα «Ανέβασμα Σημειώσεων» αποτελείται από ένα RelativeLayout το οποίο ενσωματώνει δύο TextView, ένα που ενημερώνει τον φοιτητή για την πορεία του αρχείου του και ένα που τον παροτρύνει να επιλέξει εξάμηνο. Επίσης υπάρχει ένα Spinner για την επιλογή του εξαμήνου, και ένα κουμπί με το οποίο γίνεται το ανέβασμα του αρχείου.
- **apod_dhlwshs.xml** – Χρησιμοποιείται στην δραστηριότητα «Αποδεικτικά Ηλ. Εγγραφών» και περιλαμβάνει CardView διότι θέλουμε τα αποτελέσματα σε στυλ καρτών. Έπειτα Δημιουργούμε ένα RelativeLayout το οποίο ενσωματώνει, δύο TextView τα οποία είναι ο αριθμός του αποδεικτικού και η ημερομηνία που δημιουργήθηκε, καθώς και ένα κουμπί για να κατεβάσουμε το αποδεικτικό.
- **basikes_plhrofories.xml** – Χρησιμοποιείται στην δραστηριότητα «Στοιχεία Φοιτητή» και περιλαμβάνει CardView διότι θέλουμε τα αποτελέσματα σε στυλ καρτών. Έπειτα Δημιουργούμε ένα RelativeLayout το οποίο ενσωματώνει, δεκαπέντε TextView τα οποία περιγράφουν τα στοιχεία του φοιτητή.
- **chart.xml** – Το γραφικό περιβάλλον των «Στατιστικά». Περιλαμβάνει ένα <ListView>, πάνω στο οποίο προστίθενται τα γραφήματα.
- **dhlwsh.xml** – Χρησιμοποιείται στις δραστηριότητες «Η Δήλωση μου» και «Τα Εργαστηριά μου». Δηλώνουμε CardView διότι θέλουμε τα αποτελέσματα σε στυλ καρτών. Έπειτα Δημιουργούμε ένα RelativeLayout το οποίο ενσωματώνει, οκτώ TextView, το όνομα του μαθήματος, τον κωδικό, τις διδακτικές μονάδες, το εξάμηνο, τον τύπο, τις ώρες, τα ECTS, και αν έχει δηλωθεί σε προηγούμενη δήλωση ή όχι
- **email.xml** – Αφορά την δραστηριότητα «Email Καθηγητών». Δηλώνουμε CardView διότι θέλουμε τα αποτελέσματα σε στυλ καρτών. Έπειτα Δημιουργούμε ένα RelativeLayout το οποίο ενσωματώνει, δύο TextView, το όνομα του καθηγητή και το Email του, καθώς και ένα κουμπί αποστολής Email προς τον καθηγητή.
- **filedownload.xml** – Το χρησιμοποιούμε στην δραστηριότητα «Κατέβασμα Σημειώσεων». Δηλώνουμε CardView διότι θέλουμε τα αποτελέσματα σε στυλ καρτών. Έπειτα Δημιουργούμε ένα

RelativeLayout το οποίο ενσωματώνει, τέσσερα TextView, για το όνομα του αρχείου, την ημερομηνία ανεβάσματος, το όνομα χρήστη από το οποίο ανέβηκε η σημείωση, το εξάμηνο που αφορά και το κουμπί της λήψης των σημειώσεων.

- **item_message.xml** – Το κάθε μήνυμα που απεικονίζεται στην δραστηριότητα «Συζήτηση» έχει αυτό το «στυλ». Έχουμε ένα LinearLayout το οποίο περιέχει, το εικονίδιο που έχουμε ορίσει για τον χρήστη μέσω της βιβλιοθήκης του Hdodenhof – CircleImageView, και δηλώνεται ως εξής:

```
<de.hdodenhof.circleimageview.CircleImageView/>
```

Στην συνέχεια έχουμε δεύτερο LinearLayout όπου ενσωματώνουμε δύο TextView, το ένα για το μήνυμα αυτό καθ' αυτό, και το άλλο για να γράψουμε τον χρήστη οπού το απέστειλε.

- **list_item_barchart.xml** – Δηλώνει το στυλ γραφήματος (ραβδόγραμμα), δηλώνοντας από την βιβλιοθήκη MPAndroidChart το εξής:

```
<com.github.mikephil.charting.charts.BarChart/>
```

- **list_item_piechart.xml** – Δηλώνει το στυλ γραφήματος (πίτα) ,δηλώνοντας από την βιβλιοθήκη MPAndroidChart το εξής:

```
<com.github.mikephil.charting.charts.PieChart/>
```

- **odhgos_spoudwn.xml** – Το xml αυτό «ενσωματώνεται» στο activity_tab_animation.xml και χρησιμοποιείται για την δραστηριότητα «Οδηγός Σπουδών». Αποτελείται από τέσσερα TextView, το όνομα, τον κωδικό, τις διδακτικές μονάδες και τον τύπο μαθήματος.
- **probolh_dhlwsewn.xml** – Χρησιμοποιείται στις δραστηριότητες «Αποτελέσματα» και «Περασμένα» και αποτελείται από επτά TextView τα οποία είναι το όνομα του μαθήματος, ο κωδικός του, οι διδακτικές μονάδες, το εξάμηνο, ο τύπος, ο βαθμός και οι ώρες.
- **toolbar.xml** – Προσθέτει μια μπάρα στο πάνω σημείο της εφαρμογής μας όπου μας ενημερώνει σε πια δραστηριότητα βρισκόμαστε.
- **xrhsimes_plhrofories.xml** – Το γραφικό περιβάλλον των «Χρήσιμες πληροφορίες / F.A.Q.». Περιλαμβάνει ένα RelativeLayout το οποίο ενσωματώνει μέσα του τρία TextView, τον τίτλο του θέματος, τις πληροφορίες, και το όνομα του χρήστη που τις επεξεργάστηκε.

/app/res/menu

Εδώ μπαίνουν τα xml αρχεία για το «μενού» της εφαρμογής μας. Χρησιμοποιήθηκαν 2 xml, ένα στην κύρια δραστηριότητα και ένα στην δραστηριότητα συζήτηση. Πιο συγκεκριμένα:

- **main.xml** – Το οποίο έχει μια καταχώρηση, και αυτή είναι η «ρυθμίσεις».
- **chat_menu.xml** – Έχει δύο καταχωρήσεις «Έξοδος» και «κατέβασε το νέο configuration». Με την έξοδο πραγματοποιούμε έξοδο με τον Google λογαριασμό, μας από την «συζήτηση». Με το «κατέβασε το νέο configuration» παίρνουμε από το Firebase τις τελευταίες ρυθμίσεις για την συζήτηση (περιορισμός λέξεων στην προκειμένη περίπτωση).

/app/res/values

Εδώ καθορίζουμε διάφορες τιμές όπως ακέραιους, χρώματα, πίνακες, διαστάσεις κ.ά.. Πιο συγκεκριμένα στην εφαρμογή:

- **arrays.xml** – Ένας «πίνακας» όπου καθορίζουμε τις τιμές που θα έχει ο έλεγχος βαθμολογίας (και το κείμενο), αν ο χρήστης επιθυμεί αποστολή δεδομένων στις ρυθμίσεις κ.ά..
- **colors.xml** – Περιέχει όλα τα χρώματα που χρησιμοποιούνται στα θέματα.
- **dimens.xml** – Τέσσερα αρχεία dimens.xml το καθένα για διαφορετική ανάλυση. Μας επιτρέπει να καθορίζουμε διάφορες μεταβλητές ανάλογα το πόσο μεγάλη είναι η οθόνη του κινητού ή tablet.
- **strings.xml** – Περιέχει όλες τις συμβολοσειρές της εφαρμογής που μπορούμε να ανακαλέσουμε στην εφαρμογή, ή σε άλλα αρχεία xml.
- **styles.xml** - Τέσσερα αρχεία styles.xml το καθένα για διαφορετική έκδοση Android. Μας επιτρέπει να καθορίζουμε διάφορες μεταβλητές ανάλογα την έκδοση Android του κινητού. Χρησιμοποιείται για την παραμετροποίηση κάποιων μεταβλητών στο θέμα/στυλ, διότι σε παλαιότερες εκδόσεις πολλές φορές οι κάρτες και τα χρώματα απεικονίζονται διαφορετικά από ότι στις νεότερες εκδόσεις.

/app/res/xml

Xml αρχεία τα οποία είναι κυρίως αρχεία ρυθμίσεων. Πιο συγκεκριμένα στην εφαρμογή μας:

- **app_tracker.xml** – Περιλαμβάνει ρυθμίσεις για το Google Analytics. Το αναγνωριστικό της εφαρμογής, το πότε θα γίνει παύση της σύνδεσης και άλλες ρυθμίσεις που μπορούμε να παραμετροποιήσουμε ανάλογα με τις προτιμήσεις μας.
- **preferences.xml** – Περιλαμβάνει την απεικόνιση που θα έχουν οι ρυθμίσεις στην εφαρμογή. Έχουμε 2 <ListPreference> όπου το ένα

αφορά το πεδίο έλεγχός βαθμολογίας, και το άλλο το αν θα γίνεται αποστολή στατιστικών χρήσης ή όχι.

Σε κάθε <ListPreference> υπάρχουν χαρακτηριστικά όπως:

```
android:entries="@array/entries_list_preference"
```

```
android:entryValues="@array/entryvalues_list_preference"
```

```
android:key="list_preference"
```

Τα οποία είναι «οι πίνακες» που είχαμε ορίσει πιο πάνω στο `/res/values/arrays.xml` και περιγράφουν τις επιλογές της λίστας.

Σημείωση: Το παραπάνω παράδειγμα αφορά μόνο την «λίστα» για τον έλεγχο της βαθμολογίας

Υπάρχει επίσης και το <CheckBoxPreference> το οποίο είναι για την επιλογή άσπρου ή μαύρου θέματος.

/app/assets

Εδώ περιλαμβάνονται όλες οι γραμματοσειρές που χρησιμοποιήσαμε για την εφαρμογή. Χρησιμοποιήθηκε η OpenSans σαν γραμματοσειρά και μόνο για όσον αφορά τα γραφήματα. Για όλη την άλλη εφαρμογή, την γραμματοσειρά την καθορίζει το λειτουργικό του κινητού (δηλαδή την προκαθορισμένη γραμματοσειρά που έχει κάθε κινητό Android).

/app/java/

/app/java/

`com.frerisdev.StudentsWeb/`

Σχεδόν παντού γίνεται χρήση της κλάσης store που αποθηκεύονται όλα τα δεδομένα της εφαρμογής έτσι ώστε να ελέγξουμε το χρώμα θέματος που έχει επιλέξει ο χρήστης και να το προβάλλουμε.

/app/java/ com.frerisdev.StudentsWeb/activitie S

About_Activity.java:

Μέσω της βιβλιοθήκης “Medyo-About Page” δημιουργούμε μια δραστηριότητα και στην συνέχεια προσθέτουμε σύμφωνα με την βιβλιοθήκη αυτή “items” τα οποία εμφανίζονται και περιγράφουν στον χρήστη τους συντελεστές που βοήθησαν την εφαρμογή.



Σας ευχαριστώ θερμά που με στηρίζετε και κατεβάσατε την εφαρμογή μου. Κάνω οτι μπορώ για να την βελτιώσω. :)

Έκδοση 1.3.10

Κύριος προγραμματιστής:
Αντώνιος Φρέρης

Συνδεθείτε, και στείλτε μου τις απόψεις σας

✉ [Επικοινωνήστε με E-mail](#)

🔗 [Επισκεφθείτε την ιστοσελίδα μου](#)

▶ [Αξιολογήστε την εφαρμογή στο Play Store](#)

📷 [Ακολουθήστε με στο Instagram!](#)

Ένα μεγάλο ευχαριστώ σε όσους βοήθησαν στην ανάπτυξη αυτού του προγράμματος.

Alpha Testers:
Αθανάσιος Καραθανάσης
Γιώργος Πετρόπουλος
Ιωάννης Κολεούρης
Κωνσταντίνος Ντουντούμης

Βιβλιοθήκες που χρησιμοποιήθηκαν:

Loopj - Async Http Client
BumpTech - Glide
Medyo - Android About Page
PhilJav - MP Android Chart

Εικόνα 40: Απεικόνιση της δραστηριότητας «Περί».

AnalutikhBathmologia_Activity.java:

Ξεκινάει ένα καινούριο Activity. Το MainActivity γίνεται onPause(), δεν καταστρέφεται. Στο καινούριο αυτό Activity το οποίο είναι με καρτέλες, δηλώνουμε 8 λίστες από List <AnalutikhBathmologia> μια για κάθε εξάμηνο. Αρχικοποιούμε τις καρτέλες ανάλογα τα εξάμηνα του χρήστη (ο φοιτητής μπορεί να είναι μικρό εξάμηνο και να μην έχει φτάσει στο 8^ο εξάμηνο, οπότε γίνεται έλεγχος και γίνονται οι ανάλογες καρτέλες αρχικοποίηση). Καθώς φορτώνει η δραστηριότητα, κλειδώνει η περιστροφή (για να μην έχουμε απρόσμενα κλεισίματα) και παίρνουμε τα δεδομένα από την πλατφόρμα StudentsWeb. Μόλις τελειώσει η φόρτωση και αντλήσουμε τα δεδομένα από την πλατφόρμα τα δεδομένα μας παρουσιάζονται κατά αυτόν τον τρόπο.



Εικόνα 41: Απεικόνιση της αναλυτικής βαθμολογίας.

AppCompatActivity.java:

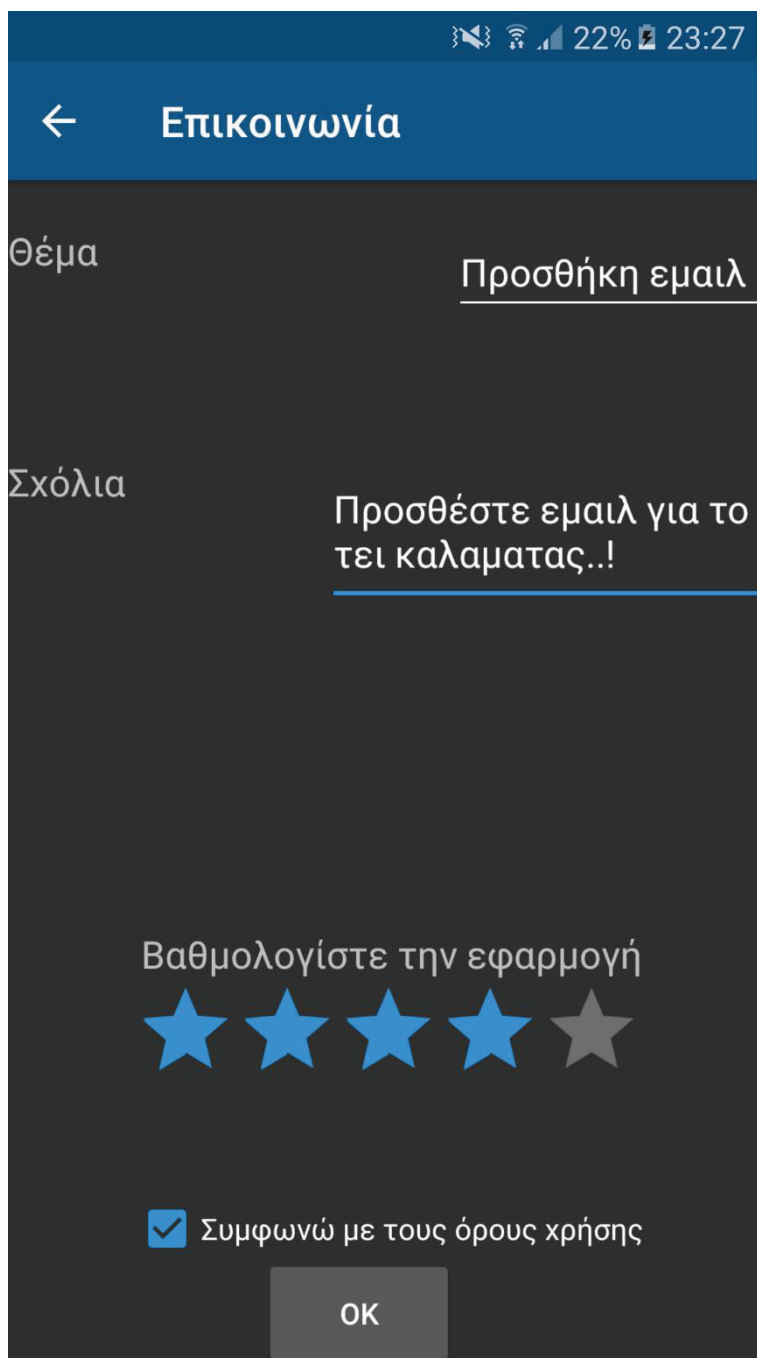
Χρησιμοποιείται ως βοηθητική βιβλιοθήκη για την κλάση Settings. Προσθέτει ActionBar και υποστήριξη σε παλαιότερες εκδόσεις (Lollipop, KitKat κ.ά.).

Contact_Activity.java:

Μια απλή δραστηριότητα η οποία έχει δύο πεδία για να γράψει ο χρήστης το «θέμα» και τα «σχόλιά» του, μια RatingBar για να κάνει αξιολόγηση και ένα CheckBox που θα συμφωνεί με τους όρους χρήσης.

Υπάρχουν έλεγχοι αν υπάρχει Ίντερνετ, αν ο χρήστης προσπαθήσει κατά λάθος ή επίτηδες να ξαναστείλει το ίδιο ερώτημα προς τον server τον σταματάει με αντίστοιχο μήνυμα, αν έχουν εισαχθεί όλα τα πεδία κ.ά. .

Αν περάσουν όλοι οι έλεγχοι και δεν υπάρξει θέμα ξεκινάει ένα AsyncTask το οποίο στέλνει στο συγκεκριμένο link/API: http://e-geneisis.eu/minisite/API/v1/contact_api.php την φόρμα συμπλήρωσε ο χρήστης στην εφαρμογή. Η φόρμα αυτή αποθηκεύεται στην βάση δεδομένων του server μας στο GRNET ή στο RPi.



Εικόνα 42: Απεικόνιση της δραστηριότητας «Επικοινωνία».

7 | login_gues | Προσθήκη email | Προσθέστε email για το τει καλαματας..! | 4 | 2016-11-06 23:27:14 |

Εικόνα 43: Απεικόνιση καταχώρησης στην βάση δεδομένων.

Αν ο χρήστης στείλει φόρμα επικοινωνίας πριν την είσοδο το όνομα χρήστη που στέλνεται είναι το "login_guest", ειδάλλως αν έχει εισέλθει και στείλει φόρμα επικοινωνίας θα σταλεί το username του.

Από μεριάς API γίνεται μέσω rph σύνδεση με την βάση δεδομένων, παίρνουμε τα δεδομένα από την φόρμα επικοινωνίας, και τα καταχωρούμε στην βάση με UTF-8 κωδικοποίηση, έτσι ώστε να απεικονίζονται τα ελληνικά και να μην έχουμε μη αναγνωρίσιμους χαρακτήρες.

Κομμάτι κώδικα στην php:

```
$sql="INSERT INTO Contact (username,topic,message,rating,date)
VALUES
('$username','$topic','$message','$rating','$dt)";
$result = mysql_query($sql, $conn);
$result = mysql_query("SET NAMES 'UTF8'", $conn);
```

Login.java:

Είναι η δραστηριότητα με την οποία ξεκινά η εφαρμογή και για να ξεκινήσει σαν πρώτη δραστηριότητα προσθέσαμε στο αρχείο AndroidManifest.xml το παρακάτω:

```
<intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
```

Με το που ανοίξουμε την εφαρμογή κάνει αρχικοποίηση το CrashLytics γίνεται έλεγχος αν υπάρχουν αποθηκευμένες ρυθμίσεις και στην συνέχεια παίρνουμε από το API του server μας τυχόν ανακοινώσεις που μπορεί να υπάρχουν. Σε περίπτωση που υπάρχει πρόβλημα του server ή δεν υπάρχει σύνδεση, η εφαρμογή συνεχίζει και λειτουργεί απροβλημάτιστα.

Σύνδεσμος για το API: http://e-genesis.eu/minisite/API/v1/announce_get.php

Κομμάτι κώδικα στην php που παίρνει τις ανακοινώσεις:

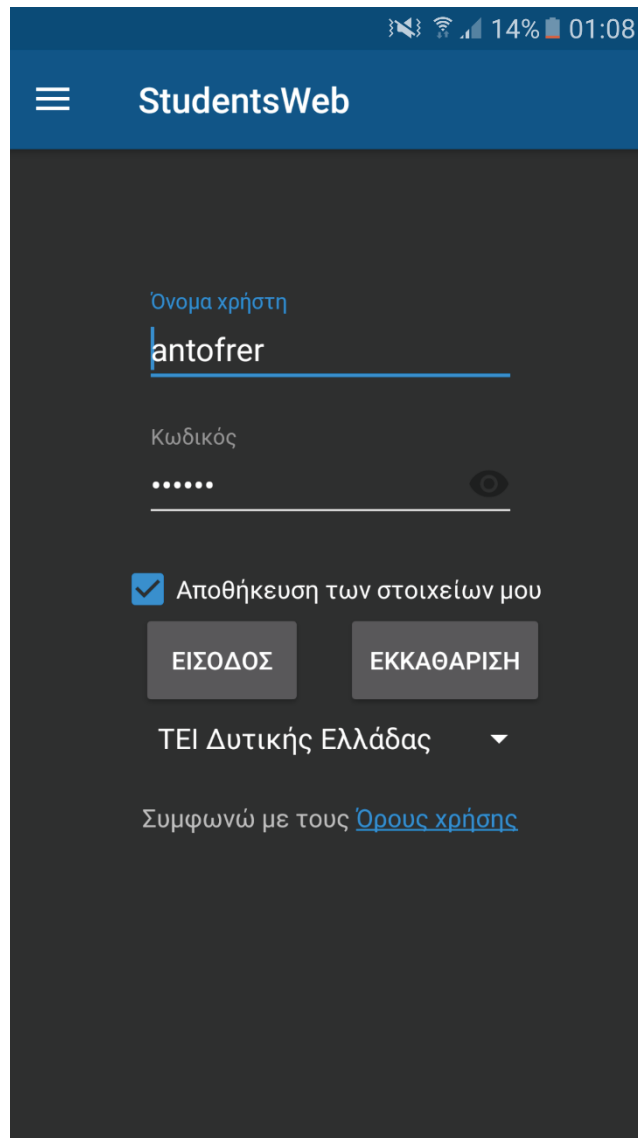
```
include 'dbconfig.php';
$sql="SELECT DISTINCT message FROM Announce";
$result = mysql_query($sql, $conn);
$row = mysql_fetch_array($result);
```

Όπως αναφέρθηκε και πιο πάνω υπάρχει αριστερά το drawer σε περίπτωση που χρειαστούμε κάτι από το συγκεκριμένο μενού. Στο κέντρο της οθόνης έχουμε τα πεδία «Όνομα χρήστη» και «κωδικός χρήστη» που πρέπει να συμπληρώσει ο χρήστης, καθώς και το τμήμα του. Έπειτα αν θέλει μπορεί με το CheckBox να αποθηκεύσει τα στοιχεία του για την επόμενη φορά.

Υπάρχουν δύο κουμπιά:

- Είσοδος – Ξεκινάει το LoginStart (AsyncTask), το οποίο το αναλύουμε παρακάτω.
- Εκκαθάριση – Διαγράφει τα αποθηκευμένα στοιχεία του χρήστη στην εφαρμογή.

Σε περίπτωση που δεν υπάρχει σύνδεση στο διαδίκτυο ή τα δεδομένα που έβαλε ο χρήστης είναι λάθος, ή δεν λειτουργεί προσωρινά η πλατφόρμα StudentsWeb ο χρήστης ενημερώνεται με αντίστοιχο Toast.



Εικόνα 44: Απεικόνιση της δραστηριότητας Login.

Σημείωση: Χάριν στην βιβλιοθήκη του χρήστη του GitHub “scottyab” – “showhidepasswordedittext” είμαστε σε θέση να αποκαλύπτουμε και να αποκρύπτουμε οποιαδήποτε στιγμή τον κωδικό μας έτσι ώστε σε περίπτωση που πατήσουμε κατά λάθος άλλο κουμπί να βρούμε το λάθος που κάναμε και να το διορθώσουμε.

MainActivity.java:

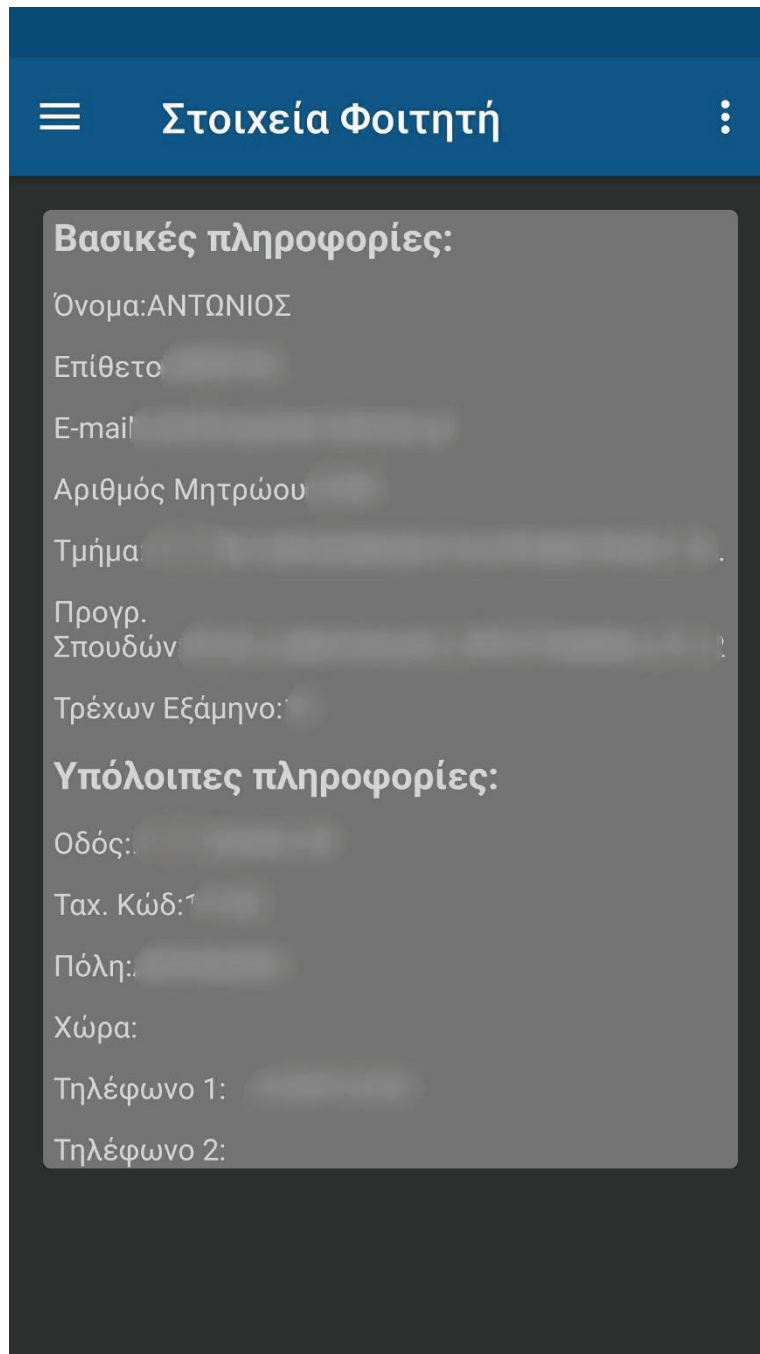
Μετά την επιτυχή είσοδο φορτώνεται η MainActivity, και στην onCreate() «τρέχει» κατευθείαν η AsyncTask “Στοιχεία” όπου λαμβάνει τα στοιχεία του φοιτητή και τα προβάλλει. Παράλληλα τρέχει ένα Thread το οποίο ελέγχει κάθε 1 δευτερόλεπτο, εάν υπάρχει σύνδεση στο ίντερνετ. Σε περίπτωση που διακοπεί η σύνδεση στο ίντερνετ ο χρήστης ειδοποιείται με ένα Snackbar. Όταν ο χρήστης επανασυνδεθεί και εντοπιστεί ίντερνετ η εφαρμογή ανανεώνει αυτόματα την συγκεκριμένη δραστηριότητα στην οποία βρίσκεται ώστε τα δεδομένα να είναι ανανεωμένα. Στην εφαρμογή επίσης έχει ενσωματωθεί SwipeRefresh. Το SwipeRefresh μας επιτρέπει κάνοντας ένα gesture

(χειρονομία) προς τα κάτω στην δραστηριότητα μας να ανανεώνει την δραστηριότητα στην οποία βρισκόμαστε (χρήσιμο όταν περιμένουμε για παράδειγμα να βγει κάποιος βαθμός ή περιμένουμε κάποια άλλη αλλαγή).

Τα cookies είναι ρυθμισμένα από την πλατφόρμα StudentsWeb να λήγουν περίπου στα 15 λεπτά, για να αντιμετωπίσουμε το πρόβλημα του «ληγμένου» cookie έχουμε βάλει μετρητές όπου μετράνε την αδράνεια (χρόνος της onPause()-onResume(=χρόνος αδράνειας) και σε περίπτωση που υπερβούμε το όριο των 15 λεπτών γίνεται αυτόματα επανασύνδεση.

Σε περίπτωση που έχουμε έκδοση μεγαλύτερη ή ίση με Marshmallow και επιλέξουμε την επιλογή «Αποδεικτικά Ηλ. Εγγραφών», «Κατεβάστε σημειώσεις» ή «Ανεβάστε σημειώσεις», πριν γίνει η φόρτωση αυτών, μας βγαίνει ένα μήνυμα παροτρύνοντας μας να δώσουμε πρόσβαση στον αποθηκευτικό χώρο ώστε η εφαρμογή να είναι σε θέση να αποθηκεύσει τα αρχεία τα οποία θα κατεβάσει. Σε περίπτωση που πατηθεί άρνηση, βγαίνει αντίστοιχο Snackbar ενημερώνοντας τον χρήστη ότι έγινε άρνηση του δικαιώματος και ότι δεν θα είναι σε θέση να αποθηκεύσει τα αρχεία που ίσως ήθελε να κατεβάσει. Βγαίνει επίσης Toast το οποίο κατευθύνει τον χρήστη να πάει στις ρυθμίσεις για να ενεργοποιήσει χειροκίνητα το δικαίωμα πρόσβασης στον αποθηκευτικό χώρο.

Τέλος στην Beta περίοδο παρατηρήθηκε ότι πολλοί χρήστες «έβγαιναν» από την εφαρμογή «κατά λάθος» για αυτό προστέθηκε μια δικλείδα ασφαλείας στην οποία ο χρήστης πρέπει να πατήσει δυο φορές γρήγορα το κουμπί «πίσω» για να εξέλθει από την εφαρμογή.

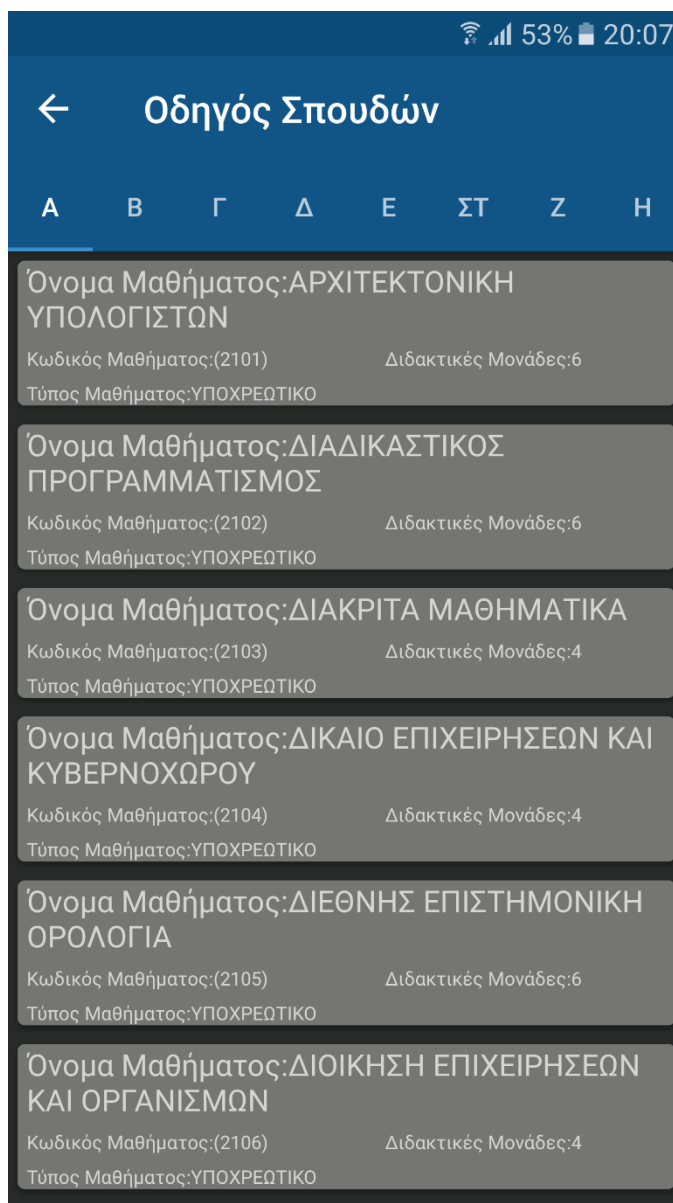


Εικόνα 45: Η κύρια δραστηριότητα ενώ έχει ήδη φορτωθεί το AsyncTask “Στοιχεία” (μαύρο θέμα).

OdhgosSpoudwn_Activity.java:

Ο σχεδιασμός είναι ίδιο με αυτόν της «αναλυτικής βαθμολογίας». Ξεκινάει ένα καινούριο Activity. Το MainActivity γίνεται onPause(), δεν καταστρέφεται. Στο καινούριο αυτό Activity το οποίο είναι με καρτέλες, δηλώνουμε 8 λίστες από List <OdhgosSpoudwn> μια για κάθε εξάμηνο. Αρχικοποιούμε τις καρτέλες ανάλογα τα εξάμηνα του χρήστη (μερικά τμήματα δεν «έχουν» 8^ο εξάμηνο για αυτό γίνεται έλεγχος καθώς παίρνουμε τα δεδομένα από την πλατφόρμα και

στην συνέχεια γίνεται η αρχικοποίηση των καρτελών). Καθώς φορτώνει η δραστηριότητα, κλειδώνει η περιστροφή (για να μην έχουμε απρόσμενα κλεισίματα) και παίρνουμε τα δεδομένα από την πλατφόρμα StudentsWeb. Μόλις τελειώσει η φόρτωση και αντλήσουμε τα δεδομένα από την πλατφόρμα τα δεδομένα μας παρουσιάζονται κατά αυτόν τον τρόπο.



Εικόνα 46: Απεικόνιση της δραστηριότητας, «Οδηγός Σπουδών».

Stats_Activity.java:

Σε αυτήν την δραστηριότητα αναλύουμε σε στατιστικά, τον Μ.Ο εξαμήνων, τα μαθήματα περασμένα ανά εξάμηνο και το πόσες ώρες έχουν περαστεί σε κάθε εξάμηνο. Τα δύο πρώτα παρουσιάζονται σε μορφή ραβδογράμματος, και το τελευταίο σε σχήμα πίτας. Για τα γραφήματα χρησιμοποιούμε την

βιβλιοθήκη MPAndroidChar-PhilJay όπου περιγράφουμε κάθε κλάση που χρησιμοποιούμε αναλυτικότερα παρακάτω (βλπ. listviewitems).

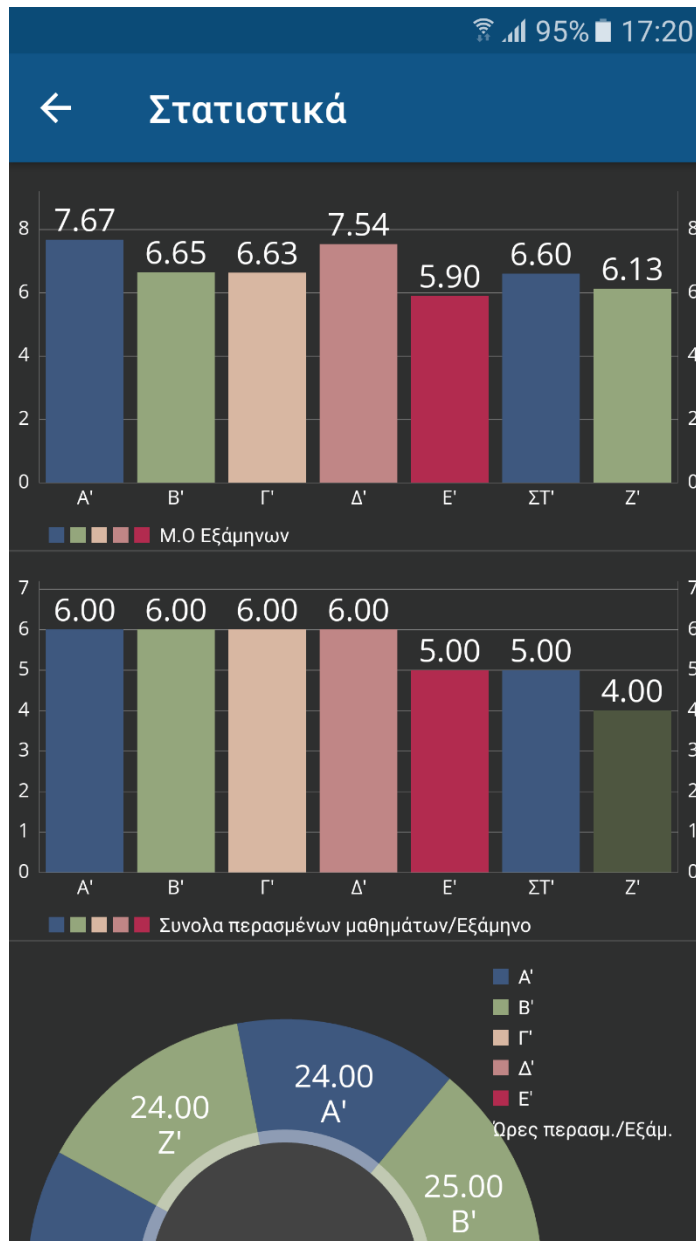
Για κάθε εξάμηνο γίνεται κάποιου είδους ανάλυση έτσι ώστε σε περίπτωση που π.χ. ο χρήστης είναι Β εξαμήνου να εμφανιστούν 2 εξάμηνα και να μην υπάρξει σφάλμα.

Τα γραφήματα τα παράγουμε ως εξής:

```
list.add(new BarChartItem(data, getApplicationContext()));  
list.add(new BarChartItem(data2, getApplicationContext()));  
list.add(new PieChartItem(data1, getApplicationContext()));
```

Έτσι γίνεται η είσοδος όλων των δεδομένων που έχουμε αναλύσει από την πλατφόρμα στα γραφήματα, οι αντίστοιχες περιγραφές και καταχώρηση των μεταβλητών γίνεται ως εξής:

```
//είσοδος μεταβλητής για το PieDataSet (παράδειγμα μιας εισόδου)  
entries1.add(new Entry(hours_passed[7], 7));  
//Αρχικοποιούμε το PieDataSet και στην συνέχεια εισαγάγουμε τις μεταβλητές και μια περιγραφή  
PieDataSet dataset1 = new PieDataSet(entries1, "Ώρες περασμ./Εξάμ.");  
//Λίστα με τις περιγραφές  
ArrayList<String> labels1 = new ArrayList<String>();  
//Παράδειγμα εισόδου λίστας  
labels1.add("Α");  
//Δημιουργούμε το PieData με τις προηγούμενες περιγραφές και PieDataSet  
PieData data1 = new PieData(labels1, dataset1);  
//Δημιουργούμε το γράφημα με τα δεδομένα μας  
list.add(new PieChartItem(data1, getApplicationContext()));
```



Εικόνα 47: Επίδειξη της δραστηριότητας Stats_Activity.

UploadToServer_Activity.java:

Με την δραστηριότητα “UploadToServer_Activity”, ο φοιτητής είναι σε θέση να ανεβάζει στον server του «Ωκεανός» τις σημειώσεις που μπορεί να έχει ώστε να βοηθήσει και άλλους συμφοιτητές. Με το `showFileChooser()` ανοίγουμε την διαχείριση αρχείων, ώστε ο χρήστης να επιλέξει το αρχείο που επρόκειτο να ανεβάσει. Μετά την επιλογή του αρχείου επιλέγει και το εξάμηνο που αντιστοιχεί για το συγκεκριμένο αρχείο. Τα στοιχεία (όνομα αρχείου, ημερομηνία, όνομα χρήστη που το ανέβασε κ.ά.) αποστέλλονται στην διεύθυνση: <http://e-genesis.eu/minisite/API/v1/UploadToServer.php>

Το αρχείο αποστέλλεται στον server κατά αυτόν τον τρόπο:

// Καθορισμ

ERROR: stackunderflow
OFFENDING COMMAND: exch

STACK:

/_ct_na