

Τ.Ε.Ι ΠΕΙΡΑΙΑ

Υλοποίηση εφαρμογής κινητού τηλεφώνου με χρήση της πλατφόρμας Google Android

**ΚΟΥΤΣΚΩΣΤΑΣ ΧΡΗΣΤΟΣ
ΑΜ: 24775**

Εισηγητής Καθηγητής : ΑΓΓΕΛΟΠΟΥΛΟΣ ΙΩΑΝΝΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΑΥΤΟΜΑΤΙΣΜΟΥ Τ.Ε.**

ΙΟΥΛΙΟΣ 2017

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος ΚΟΥΤΣΟΚΩΣΤΑΣ ΧΡΗΣΤΟΣ,
του ΣΠΥΡΟΥ με αριθμό μητρώου 24775 φοιτητής του Τμήματος **Μηχανικών
Αυτοματισμού Τ.Ε.** του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την εκπόνηση της Πτυχιακής
Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του
συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο
περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή
μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν
λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου
συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την
ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα
του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η
Συνέλευση του Τμήματος με νέα απόφασης της, μετά από αίτηση του ενδιαφερόμενου, του
αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα
καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός
ημερολογιακού βμήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα
προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

Επίσης δηλώνω υπεύθυνα ότι έχω παρακολουθήσει το σεμινάριο συγγραφής και
εκπόνησης πτυχιακής εργασίας που διοργανώνεται από το Τμήμα Μηχανικών Αυτοματισμού
Τ.Ε. κατά το Χειμερινό Εξάμηνο του Ακ. Έτους 2016

Ο Δηλών

Ημερομηνία

Περίληψη

Στην παρούσα πτυχιακή εργασία γίνεται παρουσίαση της ανάπτυξης εφαρμογών έξυπνων κινητών λειτουργικού Android με τη χρήση του λογισμικού Android Studio καθώς και την χρήση των Android API (Application Programming Interface) και Android SDK (System Development Kit).

Στην εισαγωγή γίνεται μια παρουσίαση των εκδόσεων και των χαρακτηριστικών του Android. Στη συνέχεια γίνεται περιγραφή της ανάπτυξης μιας εφαρμογής με τη χρήση αυτών των εργαλείων.

Με την εφαρμογή αυτή ο χρήστης θα μπορεί να έχει γρήγορη πρόσβαση στις ανακοινώσεις και στην ηλεκτρονική γραμματεία του τμήματος. Επίσης μέσω αυτής θα παρέχονται στον σπουδαστή πληροφορίες για τα μαθήματα, τους καθηγητές και το τμήμα και στοιχεία επικοινωνίας με τη σχολή.

Πίνακας Περιεχομένων

Περίληψη.....	3
Λίστα Εικόνων.....	5
1.ΕΙΣΑΓΩΓΗ ΣΤΟ ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ ANDROID.....	6
1.1 Εισαγωγή στο Android.....	6
1.2 Εξέλιξη και ιστορική αναδρομή του Android.....	6
1.2.1 Android 1.5 Cupcake	7
1.2.2 Android 1.6 Donut	7
1.2.3 Android 2.0/2.1 Eclair.....	8
1.2.4 Android 2.2 Froyo.....	8
1.2.5 Android 2.3 Gingerbread	9
1.2.6 Android 3.0 Honeycomb.....	10
1.2.7 Android 4.0 Ice Cream Sandwich.....	10
1.2.8 Android 4.1/4.2/4.3 Jellybean.....	11
1.2.9 Android 4.4 KitKat	11
1.2.10 Android 5.0/5.1 Lollipop	12
1.2.11 Android 6.0 Marshmallow	12
1.3 Αρχιτεκτονική του Android	13
1.4 Εσωτερικό Εφαρμογής Android	14
1.4.1 AndroidManifest.xml.....	15
1.4.2 Φάκελος src.....	15
1.4.3 Φάκελος res.....	15
1.5 Ασφάλεια στο Android.	15
2. ANDROID STUDIO	17
2.1 Γενικά το Android Studio	17
2.2 Εγκατάσταση του Android Studio.	17
2.3 Android Standar Development Kit (SDK).....	19
2.4 Αποσφαλμάτωση (Debugging)	19
2.5 Android Virtual Device Manager(AVD)	20
2.6 Δημοσίευση Εφαρμογής.	20
3. Απαιτήσεις Εφαρμογής	21
4. Υλοποίηση της εφαρμογής Automation.....	21
4.1 Δημιουργία νέας εφαρμογής στο Android Studio.....	21

4.2 Υλοποίηση Κεντρικής Οθόνης.....	22
4.3 Υλοποίηση Ανακοινώσεων	25
4.5 Υλοποίηση Προγράμματος	30
4.6 Υλοποίηση Επικοινωνίας	32
4.7 Δήλωση των Activities στο AndroidManifest	33
5. Μελλοντική εξέλιξη	34
ΠΗΓΕΣ	35

Λίστα Εικόνων

Εικόνα 1.1 Android Cupcake.....	07
Εικόνα 1.2 Android Donut	07
Εικόνα 1.3 Android 2.0/2.1 Éclair.....	08
Εικόνα 1.4 Android 2.2 Froyo	08
Εικόνα 1.5 Android 2.3 Gingerbread	09
Εικόνα 1.6 Android 3.0 Honeycomb	10
Εικόνα 1.7 Android 4.0 Ice Cream Sandwich.....	10
Εικόνα 1.8 Android 4.1/4.2/4.3 Jellybean read.....	11
Εικόνα 1.10 Android KitKat	11
Εικόνα 1.11 Android Lollipop.....	12
Εικόνα 1.12 Android Marshmallow.....	13
Εικόνα 1.12 Αρχιτεκτονική Android	13
Εικόνα 2.1: Καλωσόρισμα στον Οδηγό Εγκατάστασης	17
Εικόνα 2.2: Επιλογή τοποθεσίας εγκατάστασης.....	18
Εικόνα 2.3: Ολοκλήρωση εγκατάστασης.....	19
Εικόνα 4.1: Τα τρία στάδια δημιουργίας νέου project.....	22
Εικόνα 4.2 Κάθετη και Οριζόντια απεικόνιση της κεντρικής οθόνης.....	25
Εικόνα 4.3 Οθόνη Ανακοινώσεων.....	29
Εικόνα 4.4 Εμφάνιση προγράμματος.....	32
Εικόνα 4.5 Οθόνη δημιουργίας νέου στοιχείου στο πρόγραμμα.....	33

1.ΕΙΣΑΓΩΓΗ ΣΤΟ ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ ANDROID

1.1 Εισαγωγή στο Android

Το Android είναι ένα ανοιχτού κώδικα λειτουργικό σύστημα, για φορητές συσκευές όπως smartphones, tablets, smartwatches καθώς πλέον και για Smart TVs, το οποίο βασίζεται στον πυρήνα του Linux. Αρχικά αναπτύχθηκε από την Google και η πρώτη παρουσίασή του έγινε στις 5 Νοεμβρίου 2007 καθώς επίσης η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα Android υπό τους όρους της Apache License μιας ελεύθερης άδειας λογισμικού.

Οι προγραμματιστές μπορούν να δημιουργήσουν εφαρμογές με σχεδόν απεριόριστη λειτουργικότητα χρησιμοποιώντας μια ποικιλία απο βιβλιοθήκες και εργαλεία αυξάνοντας έτσι την παραγωγικότητάς τους καθώς και την ποιότητα των εφαρμογών

1.2 Εξέλιξη και ιστορική αναδρομή του Android

Το android είναι ανοιχτού κώδικα λογισμικό γεγονός που συνέβαλε στην ραγδαία εξέλιξή του καθώς έχουμε ήδη 12 κύριες εκδόσεις μέσα σε 7 χρόνια. Η κωδική ονομασία των διάφορων εκδόσεων του Android προέρχεται από ονομασίες γλυκών. Έτσι έχουμε τις εξείς εκδόσεις:

1.2.1 Android 1.5 Cupcake



Εικόνα 1.1 : Android Cupcake

Η έκδοση αυτή κυκλοφόρησε στις 30 Απριλίου 2009 και υποστήριζε ένα νέο εικονικό πληκτρολόγιο και την μεταφόρτωση αρχείων video και εικόνων στο Youtube και το Picasa.

1.2.2 Android 1.6 Donut



Εικόνα 1.2 : Android Donut

Η έκδοση αυτή παρουσιάστηκε στις 15 Σεπτεμβρίου 2009. Υποστηρίζει την επιλογή πολλαπλών αρχείων ενώ ταυτόχρονα έχει βελτιωμένο γκάλερι, φωτογραφική μηχανή και Android Market. Ο χρήστης έχει επίσης τη δυνατότητα για βελτιωμένη αναζήτηση που

συμπεριλαμβάνει την αναζήτηση σελιδοδεικτών, ιστορικού, επαφών αλλά και στο διαδίκτυο μέσω αρχικής οθόνης.

1.2.3 Android 2.0/2.1 Eclair



Εικόνα 1.3 : Android Eclair

Παρουσιάστηκε στις 26 Οκτωβρίου 2009 και επανεκδόθηκε σε Android 2.1 Eclair τον Ιανουάριο του 2010. Σε αυτή την έκδοση έχουμε την υποστήριξη πολλαπλών χρηστών, αποστολής email και του Bluetooth 2.1. Η camera είναι βελτιωμένη καθώς διαθέτει φλας και ψηφιακό ζοομ.

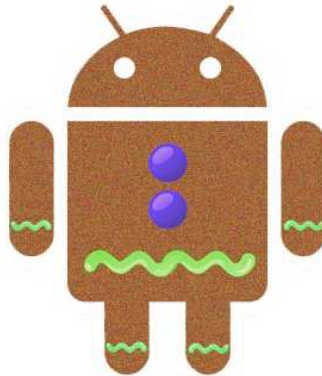
1.2.4 Android 2.2 Froyo



Εικόνα 1.4: Android Froyo

Η παρουσίαση της έκδοσης αυτής έγινε το Μάιο του 2010. Σε αυτή την έκδοση έχουμε την υποστήριξη Tethering και τη λειτουργία Hotspot καθώς και την ύπαρξη του Adobe Flash 10.1. Τέλος υπάρχει η δυνατότητα της εγκατάστασης εφαρμογών στην έξωτερική κάρτα μνήμης και η μεταφορά τους εκεί από τη μνήμη τηλεφώνου.

1.2.5 Android 2.3 Gingerbread



Εικόνα 1.5: Android Gingerbread

Αυτή η έκδοση εμφανίστηκε τον Δεκέμβριο του 2010 και επανεκδόθηκε σε Android 2.3.3 τον Φεβρουάριο του 2011. Σε αυτή την έκδοση έχουν γίνει βελτιώσεις στο User Interface κάνοντάς το πιο απλό και ταχύτερο. Υποστηρίζεται πλέον η δυνατότητα για copy-paste σε όλο το σύστημα, το NFC (Near Field Communication) και η ύπαρξη πολλαπλών καμερών.

1.2.6 Android 3.0 Honeycomb



Εικόνα 1.6: Android Honeycomb

Εμφανίστηκε το Μάιο του 2011 και προοριζόταν αποκλειστικά για tablets. Υπάρχει η δυνατότητα υποστήριξης διπύρηνων και τετραπύρηνων επεξεργαστών. Επίσης έχουμε την εμφάνιση του κουμπιού recent apps που προσφέρει στο χρήστη την εύκολη προσπέλαση από τη μια εφαρμογή στην άλλη. Τέλος έχουμε την δυνατότητα υποστήριξης 3D γραφικών, Private Browsing και δυνατότητα προβολής εικόνων σε πλήρη οθόνη.

1.2.7 Android 4.0 Ice Cream Sandwich



Εικόνα 1.7: Android Icecream Sandwich

Εμφανίστηκε τον Οκτώβριο του 2011 με νέο User Interface πιο αποδοτικό που εισάγει νέα εικονικά κουμπιά αντικαθιστώντας τα αφής ή φυσικά κουμπιά.

Έχουμε επαναπροσαρμογή των εφαρμογών Gmail και Google Chrome και νέα προσαρμόσιμη οθόνη κλειδώματος. Τέλος έχουμε δυνατότητα μέτρησης δεδομένων ιντερνετ και δυνατότητα αναγνώρισης προσώπου.

1.2.8 Android 4.1/4.2/4.3 Jellybean



Εικόνα 1.8: Android Jellybean

Παρουσιάστηκε τον Ιούλιο του 2012 και έχουμε την αναβάθμιση του Android Market σε Google Play Store. Η μεγαλύτερη καινοτομία της έκδοσης ήταν ο ψηφιακός βοηθός Google Now ο οποίος μπορεί να μας παρέχει πληροφορίες και απαντήσεις για τα καθημερινά μας ενδιαφέροντα.

1.2.9 Android 4.4 KitKat



Εικόνα 1.9: Android KitKat

Παρουσιάστηκε τον Οκτώβριο του 2013. Σε αυτή την έκδοση είχαμε μεγάλη βελτίωση του User Interface σε τέτοιο βαθμό ώστε να τρέχει απροβλημάτιστα και σε συσκευές χαμηλών δυνατοτήτων.

1.2.10 Android 5.0/5.1 Lollipop



Εικόνες 1.10: Android Lollipop

Σε αυτή την έκδοση έχουμε το νέα επίπεδο λειτουργικό με εμφάνιση με περισσότερα χρώματα και πολλά εφε κίνησης

1.2.11 Android 6.0 Marshmallow

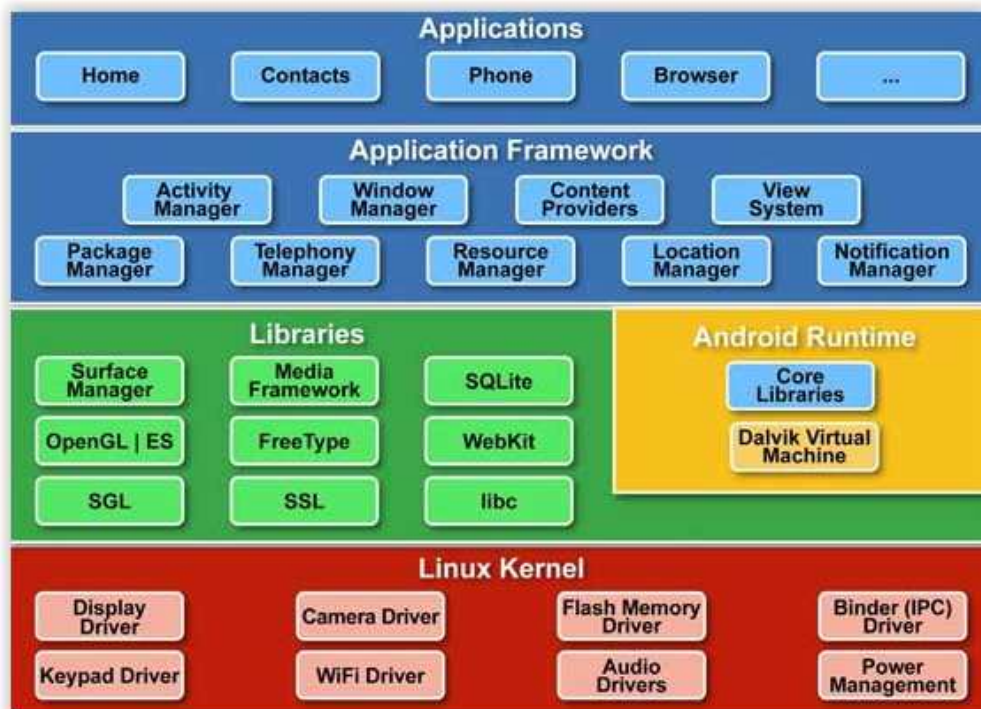


Εικόνα 1.11; Android Marshmallow

Στην έκδοση αυτή έχουμε τη δυνατότητα αναγνώρισης δακτυλικού αποτυπώματος για το ξεκλείδωμα της οθόνης ή ακόμα για την πραγματοποίηση πληρωμών.

1.3 Αρχιτεκτονική του Android

Το Android είναι μια στοίβα απο διεργασίες και υπηρεσίες που επικοινωνούν με τις εφαρμογές γνωστές και ως middleware, τις κύριες εφαρμογές γνωστές και ως core, λειτουργικό σύστημα.



Εικόνα 1.12: Αρχιτεκτονική Android 1

Η αρχιτεκτονική του λειτουργικού συστήματος αποτελείται από 5 βασικά επίπεδα.

- Τον πυρήνα Linux (Linux Kernel)
- Τις βιβλιοθήκες (Libraries)
- Την εικονική μηχανή Dalvik
- Τον χρόνο εκτέλεσης (Android Runtime)
- Το πλαίσιο εφαρμογής (Application Framework)

1.4 Εσωτερικό Εφαρμογής Android

Το εσωτερικό μιας εφαρμογής Android αποτελείται από τους εξής φακέλους μέσα στους οποίους είναι δομημένα πολλά αρχεία

- AndroidManifest.xml
- Φάκελος src
- Φάκελος res

1.4.1 AndroidManifest.xml

Μέσα σε αυτό το xml αρχείο είναι καταχωρημένες οι σημαντικότερες πληροφορίες της εφαρμογής για χρήση από το λειτουργικό σύστημα. Κάποιες από αυτές είναι:

- Το όνομα που φένεται στο χρήστη
- Το όνομα πακέτου της εφαρμογής
- Οι άδειες χρήσης της εφαρμογής
- Όλες οι δραστηριότητες, πάροχοι περιεχομένου, υπηρεσίες, κλπ, που περιέχει και χρησιμοποιεί η εφαρμογή.

1.4.2 Φάκελος src

Στο φάκελο αυτό περιέχονται όλα αρχεία κλάσης της Java όλων των Activities, Services κλπ. και αποτελεί ταυτόχρονα τον μοναδικό φάκελο στον οποίο αποθηκεύονται τα αρχεία του κώδικά μας.

1.4.3 Φάκελος res

Περιέχει όλα τα αρχεία εικόνας, κειμένου, layout, κλπ τα οποία χρησιμοποιούνται από τις Activities που βρίσκονται στον φάκελο src ταξινομημένα σε υποφακέλους ανάλογα με το είδος τους.

1.5 Ασφάλεια στο Android.

Κάθε εφαρμογή που εγκαθίσταται στη συσκευή λειτουργεί στη δική της εικονική μηχανή και αυτό είναι πλαίσιο ασφαλείας της εφαρμογής. Το Android είναι ένα λειτουργικό σύστημα το οποίο αντιμετωπίζει την κάθε εφαρμογή σαν ξεχωριστό χρήστη. Το σύστημα παρέχει ένα μοναδικό αριθμό ID ο οποίος είναι άγνωστος

στην εφαρμογή και καθώς αναθέτει συγκεκριμένες άδειες χρήσης στα αρχεία της κάθε εφαρμογής, μόνο η εφαρμογή με το σωστό ID μπορεί να έχει πρόσβαση σε αυτά.

Κάθε εφαρμογή τρέχει στην δική της εικονική μηχανή απομονωμένη από τις υπόλοιπες εφαρμογές μόλις της ζητηθεί από το σύστημα και κλείνει είτε επειδή δεν χρησιμοποιείται πλέον, είτε επειδή το σύστημα θέλει να ελευθερώσει τους πόρους της μνήμης για χρήση από άλλη εφαρμογή.

2. ANDROID STUDIO

2.1 Γενικά το Android Studio

Το Android Studio αποτελεί ένα προγραμματιστικό περιβάλλον για την ανάπτυξη εφαρμογών της πλατφόρμας Android. Η πρώτη έκδοση κυκλοφόρησε τον Δεκέμβριο του 2014 και είναι διαθέσιμο για Windows, Mac OS X και Linux

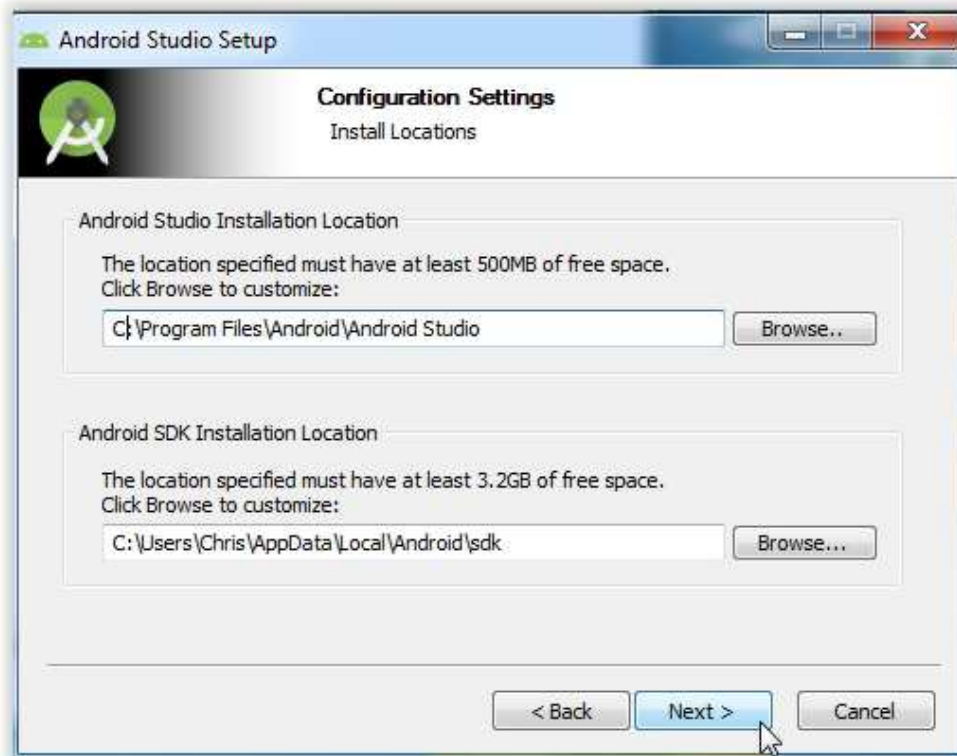
2.2 Εγκατάσταση του Android Studio.

Κάνουμε λήψη της εφαρμογής Android Studio ανάλογα με το λειτουργικό σύστημα που διαθέτουμε από την παρακάτω διεύθυνση <https://developer.android.com/studio/index.html>
Εκτελούμε το αρχείο εγκατάστασης androidstudio-bundle.exe και ξεκινάμε τη διαδικασία εγκατάστασης. Μέσα από διάφορες οθόνες διαμορφώνουμε την εγκατάσταση ώστε να καλυφθούν τις απαιτήσεις μας όπως φέεται στις εικόνες παρακάτω.



Εικόνα 2.1: Καλωσόρισμα στον Οδηγό Εγκατάστασης

Στην παρακάτω εικόνα επιλέγουμε που θα γίνει η εγκατάσταση του προγράμματος



Εικόνα 2.2: Επιλογή τοποθεσίας εγκατάστασης

Αφού ολοκληρώσουμε σωστά όλα τα βήματα τότε βλέπουμε την παρακάτω εικόνα.



Εικόνα 2.3: Ολοκλήρωση εγκατάστασης

2.3 Android Standar Development Kit (SDK)

Το Android SDK αποτελεί το βασικό εργαλείο ανάπτυξης εφαρμογών και περιλαμβάνει τις βιβλιοθήκες Java, τον emulator για τη δοκιμαστική εκτέλεση, τον compiler που παράγει τον κώδικα και διάφορες βιβλιοθήκες.

2.4 Αποσφαλμάτωση (Debugging)

Το Android Studio παρέχει κάποια εργαλεία αποσφαλμάτωσης της εφαρμογής. Το λειτουργικό σύστημα καθορίζει ποιούς πόρους θα χρησιμοποιήσει η εφαρμογή και για αυτή την διαδικασία χρησιμοποιεί το Logcat που καταγράφει το οτιδήποτε συμβαίνει στην εφαρμογή και στους πόρους του συστήματος που χρησιμοποιεί, καθώς επίσης ενημερώνει τον προγραμματιστή για ότι προκύψει όπως η μνήμη που χρησιμοποιεί η εφαρμογή, τυχόν συντακτικά και λογικά λάθη στον κώδικα, σφάλματα στο μητρώο, παραβίαση αδειών, παραβίαση δεδομένων της εφαρμογής και άλλα. Επίσης ένα άλλο σημαντικό εργαλείο που χρησιμοποιούν οι προγραμματιστές είναι το DDMS το οποίο παρέχει την δυνατότητα

τεσταρίσματος της εφαρμογής σε εικονικές συσκευές που δεν έχουν πρόσβαση σε ζωντανά δεδομένα. Αν πούμε ότι για παράδειγμα φτιάχνουμε μια εφαρμογή ανταλλαγής μηνυμάτων κειμένου και στατικών εικόνων πριν την δοκιμάσουμε σε κανονική συσκευή και μας επιφέρει πρόσθετες χρεώσεις μόνο για τις δοκιμές μπορούμε να στέλνουμε fake notifications ή και μηνύματα στην εικονική συσκευή με στόχο την βελτίωση. Το ίδιο γίνεται και με την γεωγραφική θέση, κλήση αριθμού, προβολή website, προβολή στατιστικών heap memory allocation, thread memory size, network statistics και άλλα.

2.5 Android Virtual Device Manager(AVD)

Ένα σημαντικό εργαλείο για τους προγραμματιστές είναι οι εικονικές συσκευές. Ο AVD μας παρέχει όλα τα απαραίτητα αρχεία για την δημιουργία μιας εικονικής συσκευής με σκοπό να τεστάρουμε μια εφαρμογή. Για αυτό είναι απαραίτητο να έχουμε κατεβάσει τα απαραίτητα αρχεία, να έχουμε εγκαταστήσει την απαιτούμενη έκδοση του API και να έχουμε επιλέξει τις κατάλληλες ρυθμίσεις συσκευής για την καλύτερη επίδοση της εφαρμογής.

2.6 Δημοσίευση Εφαρμογής.

Αφού γίνει στην εφαρμογή το απαραίτητο debugging και τεσταριστεί σε εικονικές συσκευές αν λειτουργεί σωστά τότε προχωράμε στην δημοσίευσή της. Για να δημοσιεύσουμε την εφαρμογή στο Play Store της Google θα πρέπει αρχικά να εγγραφούμε σαν προγραμματιστές. Στη συνέχεια κάνοντας μια περιγραφή και παρέχοντας ορισμένα screenshots η εφαρμογή μας δημοσιεύεται και είναι διαθέσιμη για download στους άλλους χρήστες

3. Απαιτήσεις Εφαρμογής

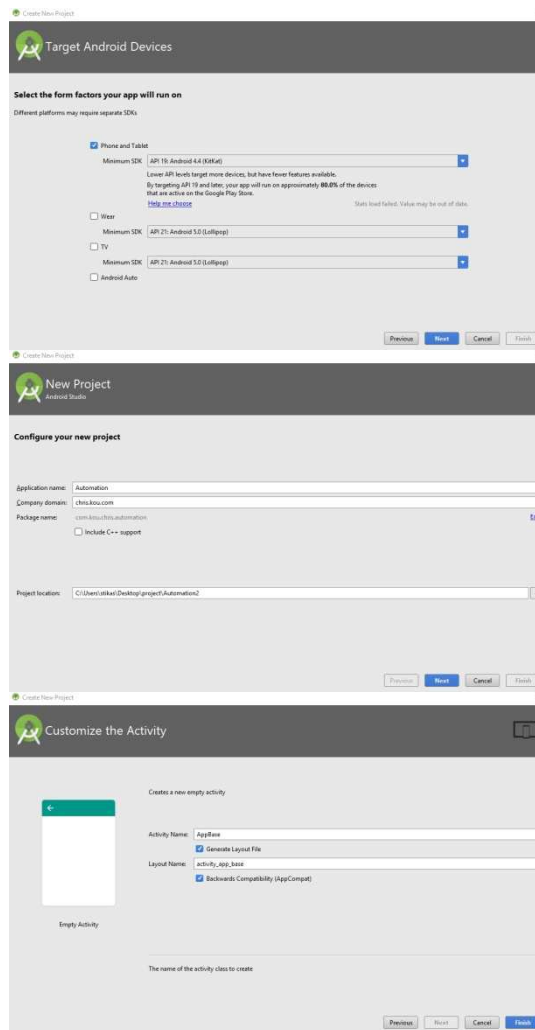
Η εργασία έχει ως σκοπό την δημιουργία μιας εφαρμογής για την πρόσβαση στις ανακοινώσεις και πληροφορίες του Τμήματος Μηχανικών Αυτοματισμού του ΑΤΕΙ ΠΕΙΡΑΙΑ καθώς και κάποια εργαλεία για τους φοιτητές. Ο σχεδιασμός της εφαρμογής πρέπει να γίνει με τέτοιο τρόπο ώστε ο χρήστης να εντοπίζει εύκολα που βρίσκεται το αντικείμενο που θέλει να χρησιμοποιήσει.

Η εφαρμογή θα παρέχει πρόσβαση στις ανακοινώσεις του τμήματος μέσω του RSS Feed. Επίσης θα παρέχει πληροφορίες για τα μαθήματα και το εκπαιδευτικό προσωπικό καθώς και στοιχεία επικοινωνίας του τμήματος. Επιπλέον θα υπάρχει δυνατότητα άμεσης πρόσβασης στην ηλεκτρονική γραμματεία. Τέλος θα παρέχονται δύο χρήσιμα εργαλεία για τους χρήστες της εφαρμογής. Το πρώτο θα δίνει τη δυνατότητα να αποθηκεύει ο χρήστης το εβδομαδιαίο πρόγραμμά της χολής ανα ημέρα. Το δεύτερο θα δίνει την δυνατότητα στο χρήστη να κρατάει και να αποθηκεύσει εύκολα κάποια σημαντική σημείωση.

4. Υλοποίηση της εφαρμογής Automation

4.1 Δημιουργία νέας εφαρμογής στο Android Studio

Επιλέγουμε τη δημιουργία νέου project. Στην πρώτη οθόνη δίνουμε το όνομα της εφαρμογής , το όνομα της «εταιρείας» μας καθώς και που θα αποθηκευτεί το project. Στην επόμενη οθόνη επιλέγουμε το minimum SDK που λειτουργεί η εφαρμογή. Εμείς θα επιλέξουμε το API 19 Android 4.4 (KitKat) καθώς μας εξασφαλίζει ότι η εφαρμογή θα λειτουργεί στο 80% των συσκευών χωρίς να υπάρχουν ασυμβατότητες μεταξύ διαφορετικών εκδόσεων του λειτουργικού. Τέλος στην τελευταία οθόνη επιλέγουμε το όνομα της κεντρικής μας activity καθώς και το όνομα του layout στο οποίο αντιστοιχεί αυτή.



Εικόνα 4.1: Τα τρία στάδια δημιουργίας νέου project

4.2 Υλοποίηση Κεντρικής Οθόνης

Θέλουμε η εφαρμογή μας να δίνει πρόσβαση στις υπόλοιπες δυνατότητες μέσω μερικών κουμπιών τα οποία θα είναι οργανωμένα σε πλέγμα. Η χρήση της GridView κρίθηκε μονόδρομος καθώς έτσι εξασφαλίζουμε σωστή εμφάνιση σε όλα τα μεγέθη οθόνης. Δημιουργούμε ένα LinearLayout και μέσα σε αυτό τοποθετούμε το GridView χρησιμοποιώντας τον παρακάτω κώδικα.

```
<GridView
    android:layout_width="match_parent"
    android:numColumns="2"
    android:id="@+id/grid"
    android:background="#e7e7e7"
```

```

        android:columnWidth="150dp"

        android:gravity="center"
        android:verticalSpacing="10dp"
        android:horizontalSpacing="10dp"
        android:stretchMode="columnWidth"
        android:layout_height="match_parent">

</GridView>

```

Το gravity δηλώνει τη τοποθέτηση του αντικειμένου του κάθε κελιού μέσα στο layout. Το numColumns δηλώνει τον αριθμό των στηλών που θα έχουμε.

Θα πρέπει όμως να δημιουργήσουμε και το xml αρχείο του GridView. Έτσι δημιουργούμε ένα νέο LinearLayout με όνομα gridView και σε αυτό θα συμπεριλάβουμε ένα ImageView για την εικόνα και ένα TextView για το κείμενο. Έχουμε λοιπόν τον εξής κώδικα:

```
<LinearLayout
```

```

xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="350dp"
    android:layout_height="200dp"
    android:background="#ffffff"
    android:elevation="5dp"
    android:orientation="vertical"
    android:padding="15dp">

```

```
<ImageView
```

```

    android:id="@+id/imageHolder"
    android:layout_width="140dp"
    android:layout_height="70dp"
    android:layout_gravity="center"
    android:layout_weight="0.21"
    android:src="@drawable/ic_notes" />

```

```
<TextView
```

```

    android:id="@+id/namePlacer"
    android:layout_width="wrap_content"
    android:layout_height="35dp"
    android:layout_gravity="center"
    android:gravity="center"
    android:text="TITLE"
    android:textColor="#000000"
    android:textSize="14dp"
    android:textStyle="bold" />

```

```
</LinearLayout
```

Ένα σημαντικό στοιχείο που πρέπει να λάβουμε υπόψη μας είναι πως προβάλεται η GridView σε κάθε προσανατολισμό της συσκευής. Σε κάθε προσανατολισμό έχουμε διαφορετικό χώρο να χρησιμοποιήσουμε. Δημιουργούμε λοιπόν δυο ίδια base_layout με διαφορετικό προσανατολισμό, το ένα κάθετο (portrait) και το άλλο οριζόντιο (landscape). Σε αυτό με τον οριζόντιο προσανατολισμό δηλώνουμε numColumns=3 καθώς έτσι εκμεταλευόμαστε καλύτερα το διαθέσιμο χώρο.

Στην κεντρική μας activity (AppBase.class) δηλώνουμε τις τιμές που μπορεί να πάρει το TextView του GridView.

Έτσι έχουμε :

```
gridView = (GridView) findViewById(R.id.grid);
basicFields.add("ΣΠΟΥΔΕΣ");
basicFields.add("ΠΡΟΓΡΑΜΜΑ");
basicFields.add("NOTES");
basicFields.add("ΗΛ. ΓΡΑΜΜΑΤΕΙΑ");
basicFields.add("ΑΝΑΚΟΙΝΩΣΕΙΣ");
basicFields.add("ΕΠΙΚΟΙΝΩΝΙΑ");
adapter = new gridViewAdapter(this, basicFields);
gridView.setAdapter(adapter);
```

Δημιουργούμε μια νέα class με όνομα gridViewAdapter και σε αυτή αναλογα με το τι έχουμε στο TextView τοποθετεί την αντίστοιχη εικόνα για κουμπί και καλεί την αντίστοιχη κλάση όταν πατήσουμε το κουμπί,

Έχουμε για παράδειγμα:

```
TextView textView =
    (TextView) v.findViewById(R.id.namePlacer);

ImageView imageView =
    (ImageView) v.findViewById(R.id.imageHolder);

if(names.get(position).toString().equals("ΣΠΟΥΔΕΣ"))
{
    imageView.setImageResource(R.drawable.ic_spoudes);

    v.setOnClickListener(new View.OnClickListener() {

        @Override

        public void onClick(View v) {
```



```

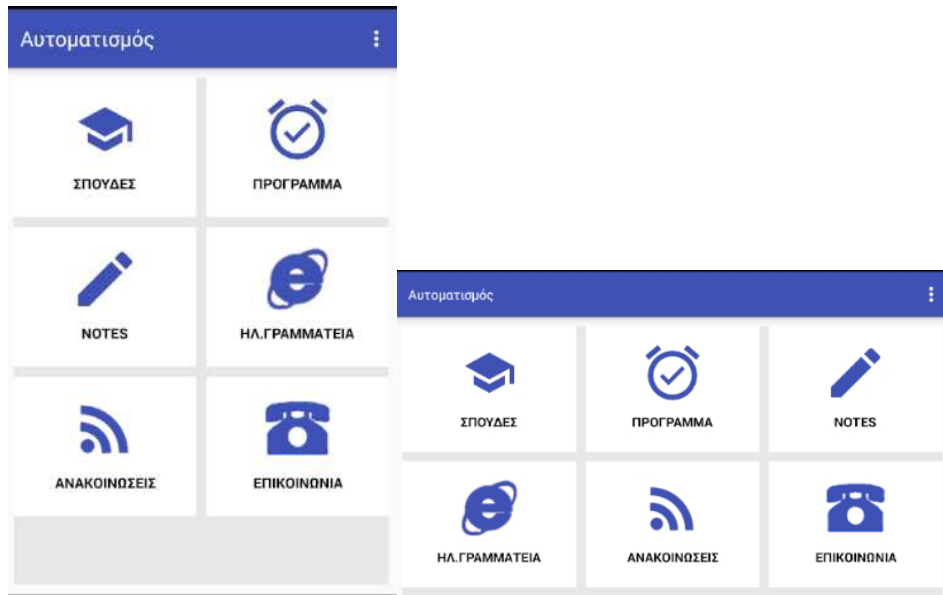
Intent launchIntent = new Intent(activity,
Spoudes.class);

        activity.startActivity(launchIntent);
    }

```

Με τον ίδιο τρόπο λειτουργούμε σε όλες τις περιπτώσεις.

Έτσι έχουμε το εξής αποτέλεσμα για κεντρική οθόνη:



Εικόνα 4.2 Κάθετη και Οριζόντια απεικόνιση της κεντρικής οθόνης

4.3 Υλοποίηση Ανακοινώσεων

Δημιουργούμε ένα νέο Layout με όνομα `anakoynoseis.xml` και μέσα σε αυτό τοποθετούμε μια `Listview`

```
<ListView
    android:id="@+id/listView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
```

Στη συνέχεια δημιουργούμε μια νέα κλάση με όνομα `Anakoynoseis.class`.

Σε αυτή με τη χρήση ενός parser παίρνουμε απο το feed των ανακοινώσεων τον τίτλο και το link της κάθε μιας ανακοίνωσης.

Έχουμε λοιπόν

```
protected ArrayAdapter doInBackground(Object[] params)
{
    headlines = new ArrayList();
    links = new ArrayList();
    try
    {
        URL url = new
URL("http://feeds.feedburner.com/skai/aqOL?format=xml");
        XmlPullParserFactory factory =
XmlPullParserFactory.newInstance();
        factory.setNamespaceAware(false);
        XmlPullParser xpp = factory.newPullParser();

        // We will get the XML from an input stream
        xpp.setInput(getInputStream(url), "UTF_8");
        boolean insideItem = false;

        // Returns the type of current event: START_TAG,
END_TAG, etc..
        int eventType = xpp.getEventType();
        while (eventType != XmlPullParser.END_DOCUMENT)
        {
            if (eventType == XmlPullParser.START_TAG)
            {
                if (xpp.getName().equalsIgnoreCase("item"))
                {
                    insideItem = true;
                }
            }
        }
    }
}
```

```

        else if
(xpp.getName().equalsIgnoreCase("title"))
        {
            if (insideItem)
                headlines.add(xpp.nextText());
//extract the headline
        }
        else if
(xpp.getName().equalsIgnoreCase("link"))
        {
            if (insideItem)
                links.add(xpp.nextText()); //extract
the link of article
        }

    }
    else if(eventType==XmlPullParser.END_TAG &&
xpp.getName().equalsIgnoreCase("item"))
    {
        insideItem=false;
    }
    eventType = xpp.next(); //move to next element
}

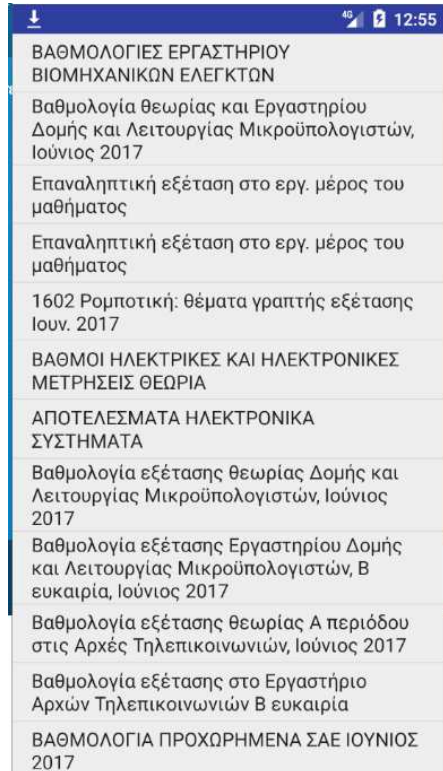
```

Μετά τοποθετούμε όλους τους τίτλους σε έναν ArrayAdapter ως εξής:

```

protected void onPostExecute(ArrayAdapter adapter)
{
    adapter = new ArrayAdapter(Anakoinoseis.this,
android.R.layout.simple_list_item_1, headlines);
    setListAdapter(adapter);
}

```



Εικόνα 4.3 Οθόνη Ανακοινώσεων

Τέλος ανάλογα με ποιόν τίτλο κλικάρουμε θέλουμε να μας ανοίγει στον browser του κινητού το αντίστοιχο link και χρησιμοποιούμε τον εξής κώδικα

```
protected void onItemClick(ListView l, View v, int
position, long id)
{
    Uri uri = Uri.parse((links.get(position)).toString());
    Intent intent = new Intent(Intent.ACTION_VIEW, uri);

    startActivity(intent);
}
```

4.4 Υλοποίηση Ηλ.Γραμματείας

Δημιουργούμε ένα layout τυπου webview με όνομα egram.xml

```
<WebView
xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/webview1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
/>
```

Δημιουργούμε μια class με όνομα Egrammateia.class

Και μέσω αυτής προβάλουμε το link της ηλεκτρονικής γραμματείας στο webview

```
public class Egrammateia extends Activity {
    private WebView mWebView1;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.egram);
        mWebView1 = (WebView) findViewById(R.id.webview1);
        WebSettings webSettings = mWebView1.getSettings();
        webSettings.setJavaScriptEnabled(true);

        mWebView1.loadUrl("https://sso.teipir.gr/login?TARGET=http%3a%2
f%2fsecr.teipir.gr%2fsecr3w%2fdefault.aspx");
        mWebView1.setWebViewClient(new EgrammateiaClient());
    }
    private class EgrammateiaClient extends WebViewClient {
        @Override
        public boolean shouldOverrideUrlLoading(WebView
webview1, String url) {
            webview1.loadUrl(url);
            return true;
        }
    }

    public boolean onKeyDown(int keyCode, KeyEvent event)
    {
        if ((keyCode == KeyEvent.KEYCODE_BACK) &&
mWebView1.canGoBack())
        {
            mWebView1.goBack();
            return true;
        }
        return super.onKeyDown(keyCode, event);
    }
}
```

4.5 Υλοποίηση Προγράμματος

Δημιουργούμε ένα νέο LinearLayout με όνομα programma.xml και τοποθετούμε ένα TabHost

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

    <TabHost
        android:id="@android:id/tabhost"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

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

            <TabWidget
                android:id="@android:id/tabs"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:layout_weight="0"
                />

            <FrameLayout
                android:id="@android:id/tabcontent"
                android:layout_width="fill_parent"
                android:layout_height="0dip"
                android:layout_weight="1"
                />

        </LinearLayout>
    </TabHost>
</LinearLayout>
```

Στη συνέχεια δημιουργούμε μια κλάση με όνομα ProgrammaActivity.class
Σε αυτή δημιουργούμε Tabs για κάθε μέρα ως εξής.

```
TabHost tabHost = (TabHost) findViewById(android.R.id.tabhost);
TabHost.TabSpec spec;
Intent intent;

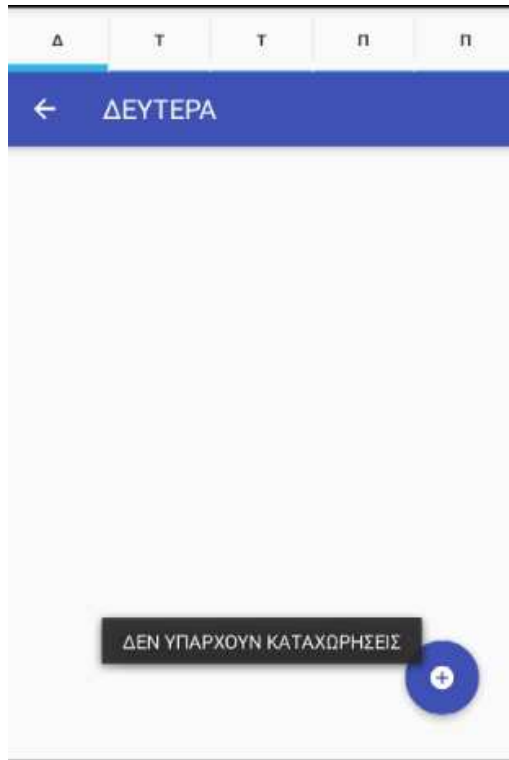
spec = tabHost.newTabSpec("ΔΕΥΤΕΡΑ");
spec.setIndicator("Δ");
```

```

intent = new Intent(this, DeuteraActivity.class);
spec.setContent(intent);
tabHost.addTab(spec);

```

Για κάθε tab ημέρα δημιουργούμε μια διαφορετική κλάση πχ DeuteraActivity.class καθώς και το κοινό για όλα τα tabs Layout, mera.xml όπου στο οποίο έχουμε μια ListView για να προβάλουμε το αντίστοιχο πρόγραμμα κάθε μέρας.



Εικόνα 4.4 Εμφάνιση προγράμματος

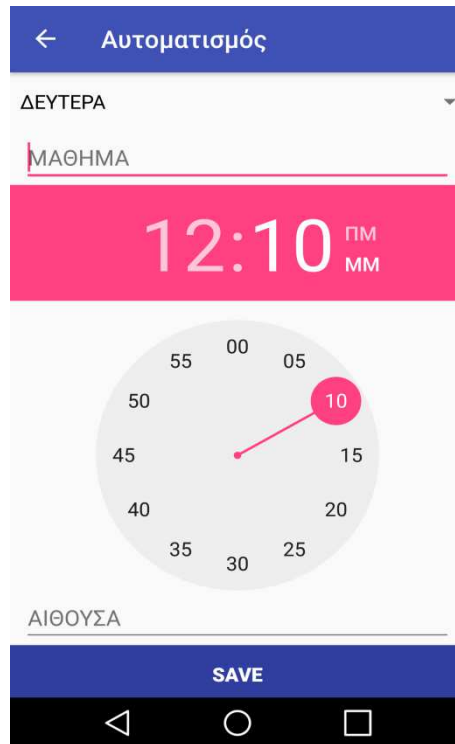
Σε κάθε class της μέρας έχουμε ένα κουμπί

```

fab = (FloatingActionButton) findViewById(R.id.fab_sch);
assert fab != null;
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent launchIntent = new Intent(getBaseContext(),
        make_schedule.class);
        startActivity(launchIntent);
    }
});

```

Το οποίο πατώντας το φορτώνουμε την κλάση make_schedule μέσω της οποίας δημιουργούμε το πρόγραμμα



Εικόνα 4.5 Οθόνη δημιουργίας νέου στοιχείου στο πρόγραμμα

4.6 Υλοποίηση Επικοινωνίας

Δημιουργούμε ένα LinearLayout με όνομα epikoinonia.xml και μέσα σε αυτό τοποθετούμε ένα TextView

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="@string/epikoinonia"
        android:textColor="@android:color/black"
        android:autoLink="all"
        android:textSize="16dp"
        android:textStyle="normal" />
</LinearLayout>
```

Με την ιδιότητα `autolink="all"` κάνουμε τα διαφορα επιμέρους στοιχεία clickable. Δηλαδή ένα νούμερο τηλεφώνου μπορούμε να το πατήσουμε και να μας το ανοίξει στην εφαρμογή τηλεφώνου ωστε να το καλέσουμε, ένα email όταν το πατήσουμε να ανοίξει αυτόματα η υπηρεσία ηλεκτρονικού ταχυδρομίου που χρησιμοποιούμε ωστε να στείλουμε mail σε αυτο και τέλος ένα link όταν το πατήσουμε να μας το ανοίξει αυτόματα στον browser που έχουμε προεπιλέξει.

Δημιουργούμε και την αντίστοιχη κλάση με όνομα `epikoinonia.class` στην οποία απλά δηλώνουμε σαν `contentview` το αντίστοιχο `layout`

```
public class epikoinonia extends Activity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.epikoinonia);  
    }  
}
```

4.7 Δήλωση των Activities στο AndroidManifest

Όλες οι activities της εφαρμογής (καθώς και οι ιδιότητές τους) πρέπει να δηλώνονται στο αρχείο `AndroidManifest`

Αρχικά στο `Android Manifest` δηλώνουμε τις άδειες χρήσης:

```
<uses-permission android:name="android.permission.INTERNET" />
```

Με αυτό δίνουμε άδεια στην εφαρμογή να έχει πρόσβαση στο `Internet`.

Στη συνέχεια δηλώνουμε την κεντρική δραστηριότητα;

```
<activity  
    android:name="com.chriskou.automation.AppBase"  
    android:label="@string/app_name">  
    <intent-filter>  
        <action android:name="android.intent.action.MAIN" />  
  
        <category  
android:name="android.intent.category.LAUNCHER" />  
    </intent-filter>  
</activity>
```

Έτσι δηλώνουμε ότι η activity `AppBase` είναι η κεντρική activity όταν τρέχουμε την εφαρμογή. Με την παράμετρο `intent-filter` δηλώνουμε ότι θέλουμε να πηγαίνουμε από την μια δραστηριότητα στην άλλη μέσω `Intent`.

Για τις υπόλοιπες activities δηλώνουμε και ποιά είναι η parent activity αυτής
Για παράδειγμα έχουμε:

```
<activity  
    android:name="com.chriskou.automation.ProgrammaActivity"  
    android:parentActivityName="com.chriskou.automation.AppBase" />
```

Σε αυτό το παράδειγμα δηλώνουμε την activity ProgramActivity και έχει για parent την AppBase activity.

Στη περίπτωση της activity Egrammateia η οποία θέλουμε να εμφανίζεται οριζόντια (landscape) δηλώνουμε και αυτή την ιδιότητα ως εξής:

```
<activity  
android:name="com.chriskou.automation.Egrammateia"  
android:screenOrientation="landscape"  
android:parentActivityName="com.chriskou.automation.AppBase"  
>
```

5. Μελλοντική εξέλιξη

Σαν μελλοντική εξέλιξη θα μπορούσαμε να ενσωματώσουμε notification όποτε γινόταν ανάρτηση μιας νέας ανακοίνωσης. Επίσης να υπήρχε ειδοποίηση λίγο πριν την έναρξη ενός μαθήματος όπως το έχουμε δηλώσει στο πρόγραμμά μας. Επιπρόσθετα θα μπορούσε να προσθεθεί και κάποιο άλλο εργαλείο όπως ένας μετατροπέας μονάδων. Τέλος θα μπορούσαμε να δώσουμε τη δυνατότητα να μπορούμε να μεταφέρουμε την εφαρμογή στην εξωτερική μονάδα αποθήκευσης (sd card).

ΠΗΓΕΣ

- J. Friesen, 2010, “Learn Java for Android Development”, Apress
- M. Murphy, 2011, "Android Programming Tutorials, 3rd Edition", CommonsWare
- J. Morris, 2011, “Android User Interface Development Beginner's Guide”, Pakt Publishing
- C. Hasenan, 2008, “Android Essentials”, Firstpress
- Android για προγραμματιστές, συλλογικό έργο Deitel Paul J., Deitel Harvey M., Deitel Abbey, Morgano Michael, εκδόσεις Γκιούρδας.
- J. Steele, 2010, “The Android Developer's Cookbook”, Addison & Wesley
- R. Meier, 2010, “Professional Android 2 Application Development”, Wrox
- <https://developer.android.com/index.html>
- <https://stackoverflow.com/>
- <https://android.stackexchange.com/>
- <https://thenewboston.com/>
- <https://www.youtube.com/?gl=GR>
- <http://www.allaboutandroid.gr/?p=6362>
- <https://www.androidpit.com>