



ΤΕΧΝΟΛΟΓΙΚΟ  
ΕΚΠΑΙΔΕΥΤΙΚΟ  
ΙΔΡΥΜΑ  
ΤΕΙ ΗΠΕΙΡΟΥ

# Παπανικολάου Γεώργιος

Σχολή: Τεχνολογικών Εφαρμογών, ΤΕΙ Ηπείρου

Τμήμα: Μηχανικών Πληροφορικής Τ. Ε.

Επιβλέπων Καθηγητής: Τσούλος Ιωάννης

## ΕΦΑΡΜΟΓΗ ANDROID ΓΙΑ ΠΡΟΓΝΩΣΗ ΚΑΙΡΟΥ ΣΤΗΝ ΗΠΕΙΡΟ ΜΕ ΧΡΗΣΗ ΤΟΥ MIT App Inventor

---

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

**Τόπος:**

.....

**Ημερομηνία:**

.....

**ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ**

1.

.....

2.

.....

3.

.....

# Περιεχόμενα

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ .....	2
ΕΥΧΑΡΙΣΤΙΕΣ.....	3
ΠΕΡΙΛΗΨΗ .....	4
<b>ΚΕΦΑΛΑΙΟ 1ο, ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ ANDROID .....</b>	<b>5</b>
1.1 Ιστορική αναδρομή Android .....	5
1.2.1 Google Play Store.....	7
1.2.2 Μερίδιο Χρήσης.....	8
<b>1.3 Παρόμοια λειτουργικά συστήματα .....</b>	<b>8</b>
<b>1.4 Αρχιτεκτονική του Android.....</b>	<b>9</b>
<b>1.5 Εκδόσεις Android.....</b>	<b>10</b>
<b>ΚΕΦΑΛΑΙΟ 2ο, ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ .....</b>	<b>14</b>
<b>2.1 Πακέτο Ανάπτυξης Λογισμικού (SDK)- Android.....</b>	<b>14</b>
<b>2.2 Eclipse IDE.....</b>	<b>14</b>
<b>2.3 MIT App-Inventor.....</b>	<b>15</b>
<b>2.4 Hyper Next Android Creator .....</b>	<b>15</b>
<b>2.5 Windows phone SDK .....</b>	<b>15</b>
<b>2.6 iOS SDK-Apple.....</b>	<b>15</b>
<b>ΚΕΦΑΛΑΙΟ 3ο, MIT APP INVENTOR .....</b>	<b>16</b>
<b>3.1 Τι είναι το App Inventor .....</b>	<b>16</b>
<b>3.2 Ανάπτυξη εφαρμογών.....</b>	<b>17</b>
<b>3.3 Απαιτήσεις Συστήματος.....</b>	<b>20</b>
<b>ΚΕΦΑΛΑΙΟ 4ο WUNDERGROUND.....</b>	<b>21</b>
<b>4.1 Γενικά .....</b>	<b>21</b>
<b>4.2 Υπηρεσίες .....</b>	<b>21</b>
<b>4.3 Βασικό μενού διαχείρισης.....</b>	<b>23</b>
<b>ΚΕΦΑΛΑΙΟ 5ο, ΑΠΑΙΤΟΥΜΕΝΕΣ ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ .....</b>	<b>24</b>
<b>5.1 JAVA .....</b>	<b>24</b>
<b>5.2 XML.....</b>	<b>24</b>
<b>5.3 JSON.....</b>	<b>25</b>
<b>ΚΕΦΑΛΑΙΟ 6ο, ΑΝΑΛΥΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....</b>	<b>26</b>
6.1 Brand Name .....	26
6.2 Εισαγωγή στην υλοποίηση της εφαρμογής.....	26
6.3 Κατασκευή της εφαρμογής .....	27
<b>ΚΕΦΑΛΑΙΟ 9<sup>ο</sup> ΣΥΜΠΕΡΑΣΜΑΤΑ .....</b>	<b>46</b>
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ .....</b>	<b>47</b>

## Πίνακας εικόνων

Εικόνα 1 Λογότυπο Android .....	5
Εικόνα 2 Google Nexus One.....	6
Εικόνα 3 Google Play Store .....	7
Εικόνα 4 Στατιστικά στοιχεία εκδόσεων-Απρίλιος 2015 .....	8
Εικόνα 5 Αρχιτεκτονική Android .....	9
Εικόνα 6 Εκδόσεις Android.....	11
Εικόνα 7 Λογότυπο Eclipse.....	14
Εικόνα 8 Δομή του Designer.....	17
Εικόνα 9 Δομή Blocks Editor-πλακίδια .....	19
Εικόνα 10 QR code.....	20
Εικόνα 11 Μενού διαχείρισης WU-API .....	22
Εικόνα 12 XML vs JSON.....	25
Εικόνα 13 Λογότυπο Εφαρμογής.....	26
Εικόνα 14 Αρχική οθόνη εφαρμογής.....	28
Εικόνα 15 Μετονομασία ετικέτας.....	29
Εικόνα 16 Οθόνη EpirusWeatherApp.....	30
Εικόνα 17 Components-βασικές επιλογές .....	30
Εικόνα 18 Εμφάνιση και επιλογές οθόνης search.....	31
Εικόνα 19 Οθόνη τριήμερης πρόγνωσης.....	32
Εικόνα 20 Βασικές ρυθμίσεις οθόνης 3 <sup>ης</sup> πρόγνωσης.....	33
Εικόνα 21 Screen 1 blocks editor .....	33
Εικόνα 22 Μεταβλητές και api.....	34
Εικόνα 23 EpirusWeatherApp initialize blocks .....	35
Εικόνα 24 XML κωδικοποίηση δεδομένων.....	35
Εικόνα 25 XML response.....	36
Εικόνα 26 Autocomplete και XML.....	37
Εικόνα 27 Εμφάνιση δεδομένων απο XML.....	37
Εικόνα 28 Μήνυμα λάθους σύνδεσης δικτύου.....	38
Εικόνα 29 Api forecast και λίστα πόλεων.....	39
Εικόνα 30 Διαδικασία getWeatherData.....	40
Εικόνα 31 Επιλογή πόλης απο λίστα .....	40
Εικόνα 32 Κωδικοποίηση Json και εμφάνιση δεδομένων.....	41
Εικόνα 33 Επιλογή στοιχείων απο το Json.....	42
Εικόνα 34 Εικονίδιο καιρού για την νύχτα.....	42
Εικόνα 35 Εμφάνιση κατάστασης καιρού.....	42
Εικόνα 36 Τελική εμφάνιση εφαρμογής.....	43
Εικόνα 37 Μενού επιλογής πόλης .....	44
Εικόνα 38 Τριήμερη πρόγνωση καιρού .....	44
Εικόνα 39 Αναζήτηση πόλης.....	45

## Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου κ. Τσούλο Ιωάννη για την δυνατότητα που μου έδωσε να ασχοληθώ με την παρούσα πτυχιακή εργασία και την βοήθεια που μου παρείχε σε τυχόν απορίες μου, την οικογένεια μου και τους φίλους για την απεριόριστη στήριξη τους. Τέλος τους καθηγητές μου για τις γνώσεις που μου παρείχαν όλο αυτό το διάστημα.

*«Δηλώνω υπεύθυνα ότι το παρόν κείμενο αποτελεί προϊόν προσωπικής μελέτης και εργασίας και πως όλες οι πηγές που χρησιμοποιήθηκαν για τη συγγραφή της δηλώνονται σαφώς είτε στις παραπομπές είτε στη βιβλιογραφία. Γνωρίζω πως η λογοκλοπή αποτελεί σοβαρότατο παράπτωμα και είμαι ενήμερος για την επέλευση των νομίμων συνεπειών»*

## Περίληψη

Η παρούσα πτυχιακή εργασία έχει ως θέμα την υλοποίηση εφαρμογής πρόγνωσης καιρού για την Ήπειρο, για συσκευές με Android με την χρήση του προγραμματιστικού περιβάλλοντος Mit App Inventor. Ο τίτλος της εφαρμογής είναι **Epirus Weather APP**.

Το προγραμματιστικό περιβάλλον MIT App Inventor είναι δωρεάν εργαλείο για εκμάθηση και κατασκευή εφαρμογών Android. Το κέρδος απ' τη δημιουργία της εφαρμογής αυτής είναι, ότι θα μπορούμε να βλέπουμε τον καιρό για την Ήπειρο και συγκεκριμένα για τους τέσσερις νομούς (Άρτα, Πρέβεζα, Ηγουμενίτσα, Ιωάννινα) από την Android συσκευή μας, με χρήση του διαδικτύου. Στην ουσία με την εγκατάσταση της εφαρμογής θα έχουμε μια πλήρη ενημέρωση για τις καιρικές συνθήκες στην Ήπειρο με συνεχής ενημέρωση των δεδομένων. Επίσης θα δίνεται η δυνατότητα για αναζήτηση οποιασδήποτε πόλη της Ελλάδος. Τα δεδομένα εμφανίζονται με χρήση της υπηρεσίας <http://wunderground.com>, όπου παρέχει δωρεάν API για προγραμματιστές και την χρήση XML, JSON αρχείων.

Στο πρώτο κεφάλαιο παρουσιάζονται οι εισαγωγικές έννοιες για το Android αλλά και βασικά χαρακτηριστικά του λειτουργικού. Στη συνέχεια και στο δεύτερο κεφάλαιο παρουσιάζονται εργαλεία για τον προγραμματισμό Android εφαρμογών και τα βασικά στοιχεία. Στο τρίτο κεφάλαιο παρουσιάζεται το προγραμματιστικό περιβάλλον MIT App Inventor με το οποίο υλοποιήθηκε η εφαρμογή. Στο τέταρτο κεφάλαιο αναφέρεται ο πάροχος Wunderground που μέσω αυτού υλοποιήθηκε η εφαρμογή και χρησιμοποιήθηκαν τα δεδομένα που παρέχει. Παρακάτω στο πέμπτο και έκτο κεφάλαιο παρουσιάζονται αναλυτικά ο τρόπος υλοποίησης της εφαρμογής, καθώς και εγχειρίδιο χρήσης αυτής. Τέλος παρέχονται πληροφορίες εξέλιξης της εφαρμογής και βελτιστοποίησης της, καθώς και συμπεράσματα από την εκπόνηση της εργασίας.

## Κεφάλαιο 1. Λειτουργικό Σύστημα Android

### 1.1 Ιστορική αναδρομή Android

Το Android είναι ένα λειτουργικό σύστημα ανοιχτού κώδικα, το οποίο τρέχει τον πυρήνα του λειτουργικού Linux, για φορητές συσκευές όπως smartphones και tablets και εμφανίστηκε για πρώτη φορά το 2003. Το λειτουργικό αυτό, αναπτύχθηκε από τη Google από το 2005 και μετά το 2007 η εξέλιξή του συνεχίστηκε από την Open Handset Alliance. Η τελευταία αποτελεί μία τεράστια ομάδα 48 μελών που περιλαμβάνει κατασκευαστές συσκευών, εταιρίες παραγωγής λογισμικού, παροχής υπηρεσιών κινητής τηλεφωνίας και εμπορικές εταιρίες. Οι γλώσσες προγραμματισμού του Android είναι οι C (για τον πυρήνα), C++ και Java. Την παρούσα στιγμή, το Android θεωρείται η πιο διαδεδομένη πλατφόρμα μεταξύ των προγραμματιστών με το 75% από αυτούς να προγραμματίζουν εφαρμογές γι' αυτό. Στην OHA συμμετέχουν σημαντικές εταιρίες όπως οι Intel, Vodafone, Motorola, Samsung, Sony Ericsson, Ebay, Google, Nvidia, Texas Instruments, Htc, Toshiba και Huawei[1],[2],[7].



Εικόνα 1 Λογότυπο Android

Τον Ιούλιο του 2005, η Google εξαγόρασε την Android Inc μια μικρή εταιρεία με έδρα το Palo Alto στην California των ΗΠΑ. Εκείνη την εποχή ελάχιστα ήταν γνωστά για τις λειτουργίες της Android Inc, εκτός του ότι ανέπτυσαν λογισμικό για κινητά τηλέφωνα. Αυτή ήταν η αρχή της φημολογίας περί σχεδίων της Google για να διεισδύσει στην αγορά κινητής τηλεφωνίας. Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance. Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android

υπό τους όρους της Apache License, μιας συνθήκης άδειας λογισμικού. Πηγαίος κώδικας του Android παρέχεται δωρεάν στο διαδίκτυο, μιας και όπως αναφέρθηκε προηγουμένως πρόκειται για λογισμικό ανοικτού κώδικα. Αυτό σημαίνει πως οποιοσδήποτε προγραμματιστής ή εταιρία, μπορεί να επέμβει στον κώδικα και να τον προσαρμόσει στις δικές του ανάγκες και επιθυμίες.

Η Google παρουσίασε το πρώτο smartphone της με ονομασία Nexus One που χρησιμοποιεί το λειτουργικό σύστημα Android. Η συσκευή κατασκευάστηκε από την HTC Corporation και έγινε διαθέσιμη στις 5 Ιανουαρίου 2010



Εικόνα 2 Google Nexus One

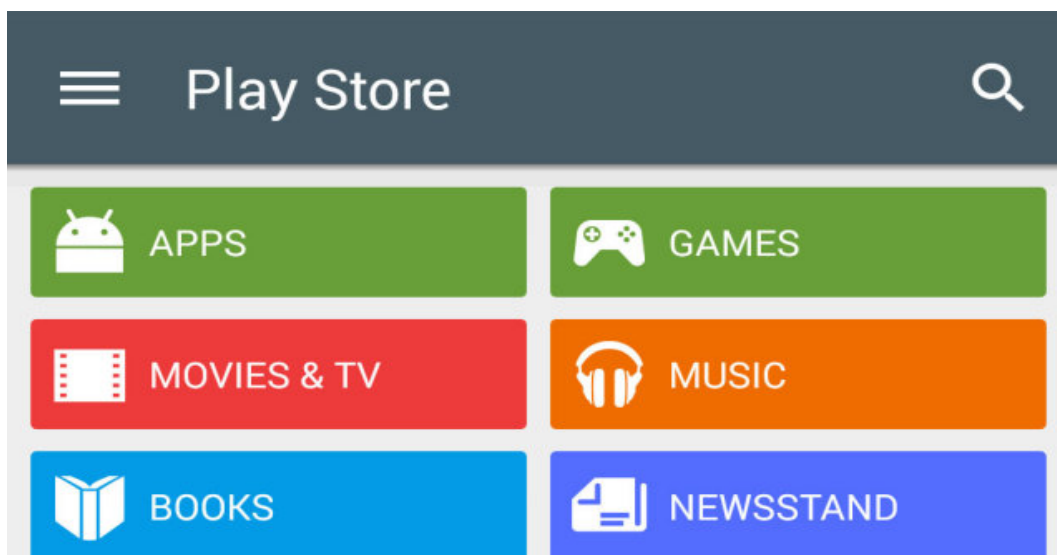
## 1.2 Εφαρμογές Android

Το Android υποστηρίζεται από μια μεγάλη κοινότητα προγραμματιστών, των οποίων οι εφαρμογές ξεπερνούν σήμερα το 1.000.000 ενώ μέχρι τώρα έχουν γίνει πάνω από 25 δις λήψεις από το Play Store, που αποτελεί και το βασικό ηλεκτρονικό κατάστημα των εφαρμογών. Οι προγραμματιστές επεκτείνουν τη λειτουργικότητα των συσκευών. Το App Inventor, που αναφέρεται η συγκεκριμένη πτυχιακή είναι ένα γραφικό περιβάλλον που παρέχει τη δυνατότητα δημιουργίας Android εφαρμογών ακόμα και σε αρχάριους προγραμματιστές ώστε να μπορεί κάποιος να τις κατεβάσει από το ηλεκτρονικό κατάστημα Play Store, όπως και από άλλες ιστοσελίδες και να τα εγκαταστήσει στον tablet ή το κινητό του[2],[3].



### 1.2.1 Google Play Store

Το Play Store είναι το ηλεκτρονικό κατάστημα που δημιουργήθηκε από την Google για συσκευές Android. Υπάρχει ως εφαρμογή διαθέσιμη και εγκαταστημένη σε όλες της συσκευές Android. Σε αυτό υπάρχει τεράστια γκάμα από εφαρμογές, παιχνίδια, βιβλία, μουσική, βίντεο και άλλα για να επιλέξει ο χρήστης να κατεβάσει είτε δωρεάν είτε επί πληρωμή.



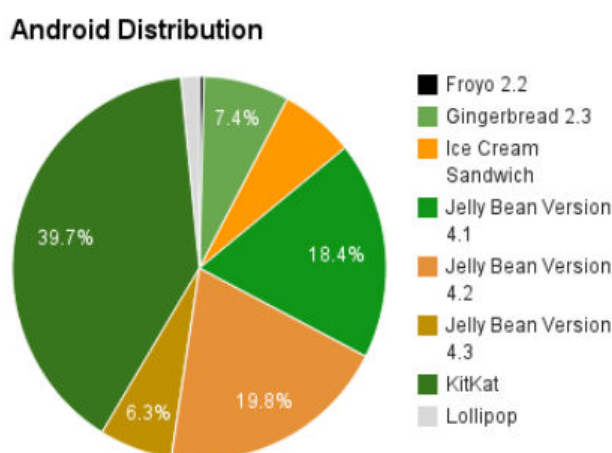
Εικόνα 3 Google Play Store

Μόνο οι εφαρμογές που συμμορφώνονται με τις απαιτήσεις συμβατότητας της Google μπορούν να εισέλθουν σε αυτό. Περιέχει φίλτρα που παρουσιάζουν μόνο τις εφαρμογές που είναι συμβατές με τη συσκευή του χρήστη έτσι ώστε να αποφευχθούν προβλήματα συμβατότητας. Επίσης οι προγραμματιστές των εφαρμογών έχουν την δυνατότητα να περιορίζουν τις εφαρμογές που έχουν δημιουργήσει σε συγκεκριμένες χώρες για επαγγελματικούς κυρίως λόγους.

Βέβαια στον χώρο των «έξυπνων κινητών τηλεφώνων» το Android δεν είναι το μόνο που υπάρχει. Αντίστοιχα η εταιρία Apple με το λειτουργικό σύστημα iOS και το ηλεκτρονικό κατάστημα Apple App Store είναι ο βασικός αντίπαλος της Google στο χώρο των λειτουργικών συστημάτων για κινητά τηλέφωνα.

### 1.2.2 Μερίδιο Χρήσης

Από τον Μάιο του 2013 άλλαξε ο τρόπος λήψης των στατιστικών στοιχείων, περιλαμβάνοντας πλέον μόνο τις συσκευές που συνδέονται ενεργά με το Google Play, αντί στο παρασκήνιο[4]. Έτσι σχεδόν κάθε μήνα η Google δίνει στην δημοσιότητα τα ποσοστά χρήσης για τις εκδόσεις που κυκλοφορούν. Το μεγαλύτερο μερίδιο της αγοράς κατέχει η έκδοση KitKat 4.4. Η πρόσφατη και τελευταία έκδοση Android 5.0 ονομάζεται Lollipop. Παρακάτω εμφανίζεται το μερίδιο χρήσης κατά την τελευταία μέτρηση τον Απρίλιο 2015



Εικόνα 4 Στατιστικά στοιχεία εκδόσεων-Απρίλιος 2015

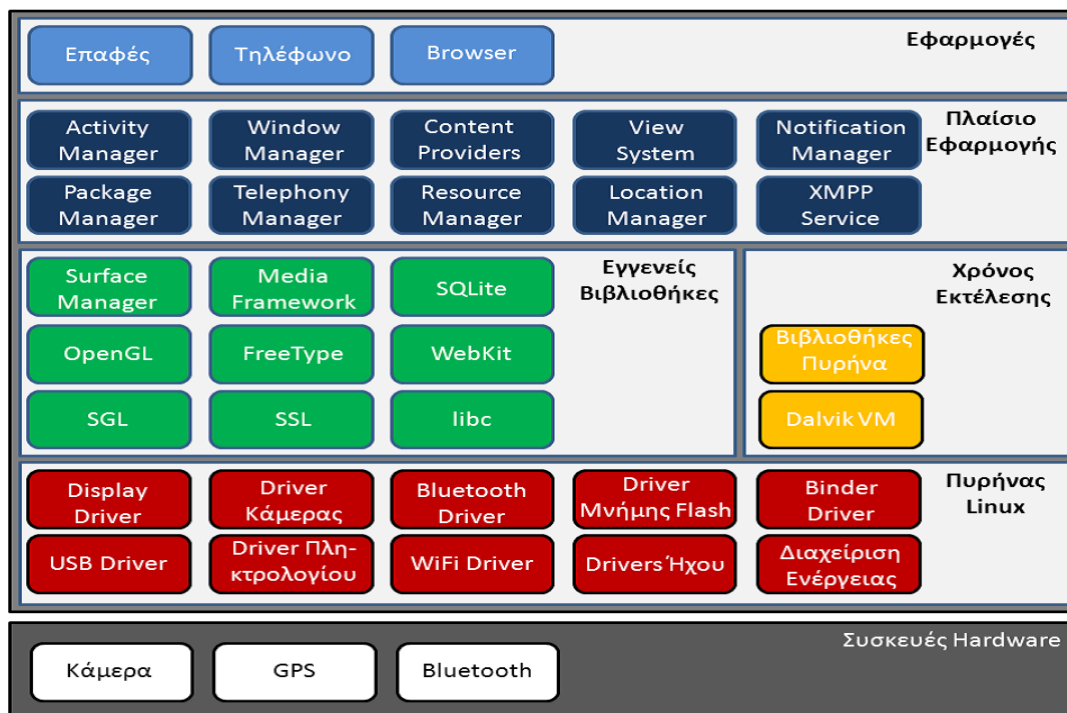
### 1.3 Παρόμοια λειτουργικά συστήματα

Παρακάτω παρουσιάζονται βασικά λειτουργικά συστήματα στον χώρο των κινητών συσκευών που υπάρχουν σήμερα. Αξίζει να αναφερθεί ότι το μερίδιο χρήσης μεταξύ των παρακάτω είναι συντριπτικά υπέρ του Android έναντι του iOS.

- **iOS** : Λειτουργικό της εταιρίας Apple. Είναι σήμερα ο μεγάλος ανταγωνιστής του Android και της Google. Πρώτη κυκλοφορία το 2007.
- **Windows phone** : Αναπτύχθηκε από τη Microsoft. Η πρώτη έκδοση, εμφανίστηκε το 2004 με την ονομασία Photon.
- **Symbian**
- **Tizen** : Λειτουργικό της Samsung, Πρώτη κυκλοφορία 2015

## 1.4 Αρχιτεκτονική του Android

Η αρχιτεκτονική του λειτουργικού συστήματος χωρίζεται σε τέσσερα επίπεδα. Το κατώτερο από αυτά, είναι ο πυρήνας Linux. Στο δεύτερο επίπεδο, βρίσκονται οι βιβλιοθήκες μαζί με το Android Runtime, στο τρίτο επίπεδο το Application Framework και στο τέταρτο οι εφαρμογές [5],[7].



Εικόνα 5 Αρχιτεκτονική Android

Όπως βλέπουμε χρησιμοποιεί τέσσερα επίπεδα. Κάθε επίπεδο στην αρχιτεκτονική αυτή, χρησιμοποιεί τις υπηρεσίες που του προσφέρονται από τα πιο κάτω επίπεδα.

Αναλυτικότερα, στο χαμηλότερο επίπεδο βρίσκεται ο πυρήνας Linux. Ο πυρήνας που χρησιμοποιείται από τη Google έχει τροποποιηθεί κατάλληλα από το βασικό πυρήνα που χρησιμοποιείται στους Η/Υ. Πρακτικά, ο kernel (που είναι γραμμένος κυρίως σε γλώσσα C), μεσολαβεί ανάμεσα στο software και το hardware παρέχοντας τους κατάλληλους οδηγούς (drivers). Όπως φαίνεται και στο παραπάνω σχήμα, οι οδηγοί σχετίζονται με τη

χρήση και τη διαχείριση του υλικού. Το υλικό αυτό είναι: απεικόνιση, κάμερα, Bluetooth, μνήμη flash, USB, πληκτρολόγιο, Wi-Fi, ήχος, διαχείριση ενέργειας

Στο δεύτερο επίπεδο της στοίβας έχουμε τις βιβλιοθήκες του Android. Αυτές ουσιαστικά αποτελούν τα APIs που είναι διαθέσιμα στους προγραμματιστές για την ανάπτυξη των εφαρμογών. Όλες αυτές είναι γραμμένες στην γλώσσα προγραμματισμού C και C++ και μεταγλωττίστηκαν για την συγκεκριμένη αρχιτεκτονική υλικού που χρησιμοποιείται από τηλέφωνα με λειτουργικό Android. Στο τρίτο επίπεδο έχουμε το Android Runtime που αποτελείται από δύο στοιχεία, το Dalvik Virtual Machine και τις βιβλιοθήκες του πυρήνα. Το Dalvik είναι το λογισμικό που τρέχει τις εφαρμογές σε μια συσκευή. Ο κώδικά του είναι γραμμένος σε Java.

Τέλος έχουμε το Application Framework όπου περιλαμβάνει όλες τις εφαρμογές, είτε αυτές που υπάρχουν μαζί με το λειτουργικό, είτε αυτές που εγκαθιστά ο χρήστης. Όλες οι εφαρμογές, γράφονται σε γλώσσα Java.

Βασικές βιβλιοθήκες είναι :

- 1) Surface Manager - για την δημιουργία 2D και 3D παραθύρων και γραφικών
- 2) Media Framework - για την υποστήριξη, αναπαραγωγή και εγγραφή πολλών δημοφιλών μέσων ήχου και εικόνας, όπως: MPEG4, H.264, MP3, AAC, AMR, JPG και PNG.
- 3) android.app - Ένα πακέτο υψηλού επιπέδου που παρέχει πρόσβαση στο μοντέλο εφαρμογών.
- 4) SQLite- που αποτελεί μια κοινόχρηστη σχεσιακή βάση δεδομένων.

### **1.5 Εκδόσεις Android**

Από την έναρξη κυκλοφορίας του λειτουργικού μέχρι και σήμερα το Android έχει περάσει από διάφορες εκδόσεις με στόχο κάθε φορά τη βελτίωση της απόδοσης και της εμφάνισης του. Παρακάτω παρουσιάζονται αναλυτικά οι εκδόσεις και οι νέες τεχνολογίες[2][4][6]. Στην επόμενη εικόνα φαίνονται όλες οι εκδόσεις μέχρι σήμερα, με την τελευταία να είναι η Lollipop 5.0.



Εικόνα 6 Εκδόσεις Android

- **Android 1.5 Cupcake**

Η έκδοση “Cupcake”, είναι η πρώτη επίσημη ονομασία βασισμένη στο Linux Kernel 2.6.27 και παρουσιάστηκε στις 30 Μαΐου του 2009. Υποστηρίζει νέες λειτουργίες και ανανεώσεις στην διεπιφάνεια χρήστη (User Interface) σε σχέση με παλιότερα λογισμικά που είχαν αναπτυχθεί.

- **Android 1.6 Donut**

Η έκδοση “Donut”, βασισμένη στο Linux Kernel 2.6.29, παρουσιάστηκε στις 15 Σεπτεμβρίου του 2009 και προσφέρει βελτιωμένη αρχική οθόνη και ταχύτερη απόκριση σε σχέση με την προηγούμενη έκδοση.

- **Android 2.0/2.1 Eclair**

Η έκδοση “Eclair”, βασισμένη και αυτή στον Linux Kernel 2.6.29, παρουσιάστηκε στις 26 Οκτωβρίου του 2009, ενώ τον Ιανουάριο του 2010 επανεκδόθηκε σε Android 2.1 Eclair (MR1). Κύρια χαρακτηριστικά της είναι ότι έχει ενσωματωθεί η υποστήριξη φλας για την κάμερα η οποία έχει πλέον και ψηφιακό zoom αλλά και η λειτουργία Bluetooth 2.1 για γρηγορότερη μεταφορά δεδομένων.

- **Android 2.2 Froyo**

Η έκδοση “Froyo”, συντόμευση της φράσης “Frozen Yogurt” (παγωμένο γιαούρτι) είναι βασισμένη στο Linux Kernel 2.6.32 και παρουσιάστηκε στις 20 Μαΐου του 2010. Προσφέρει επιπλέον την δυνατότητα εγκατάστασης εφαρμογών στην κάρτα μνήμης και την μεταφορά τους εκεί από τη μνήμη του τηλεφώνου. Ακόμη το τηλέφωνο μπορεί να μετατραπεί σε WiFi hotspot.

- **Android 2.3 Gingerbread**

Η έκδοση “Gingerbread”, βασισμένη στο Linux Kernel 2.6.35.7, παρουσιάστηκε στις 6 Δεκεμβρίου του 2010, ενώ τον Φεβρουάριο του 2011 επανεκδόθηκε σε Android 2.3.3. Στην έκδοση αυτή ενσωματώθηκε το πρωτόκολλο SIP για κλήσεις μέσω VoIP και υπάρχει η δυνατότητα για Αντιγραφή-Επικόλληση σε όλο το σύστημα και όχι μόνο στην ίδια εφαρμογή, κάτι που δεν υπήρχε στις προηγούμενες εκδόσεις.

- **Android 3.0 Honeycomb**

Η έκδοση “Honeycomb”, βασισμένη στο Linux Kernel 2.6.36, παρουσιάστηκε στις 9 Μαΐου του 2011, με την ιδιαιτερότητα ότι προοριζόταν αποκλειστικά για tablets με το πρώτο να είναι το Motorola Xoom.

- **Android 4.0 Ice Cream Sandwich**

Η έκδοση “Ice Cream Sandwich”, βασισμένη στο Linux Kernel 3.0.1, παρουσιάστηκε στις 19 Οκτωβρίου του 2011. Για άλλη μια φορά έχει βελτιωθεί η ταχύτητα και η απόδοση του συστήματος. Έτσι έχουμε βελτίωση της ασφάλειας του συστήματος με την προσθήκη αναγνώρισης προσώπου για να ξεκλειδώσει η συσκευή και η δυνατότητα ο χρήστης να τερματίσει εφαρμογές οι οποίες τρέχουν στο background.

- **Android 4.1, 4.2, 4.3 Jelly Bean**

Η ανάγκη για αναβάθμιση του λειτουργικού συστήματος οδήγησε αρχικά στην δημιουργία της έκδοσης Jellybean 4.1 που κυκλοφόρησε τον Ιούνιο του 2012 βασισμένο στον πυρήνα Linux Kernel 3.0.31. Στόχος ήταν η βελτίωση της λειτουργικότητας και της απόδοσης της διεπιφάνειας χρήστη.

- **Android 4.4 KitKat**

Παρόλο που αρχικά η Google σκόπευε να ονομάσει το Android 4.4 Lime Pie, τελικά αποφάσισε το KitKat. Ο λόγος που τελικά επέλεξαν το KitKat ήταν γιατί είναι ένα από τα αγαπημένα σνακ των προγραμματιστών της Android στο γραφείο, διατηρώντας τη μακρόχρονη παράδοση της εταιρείας να ονομάζει κάθε έκδοση του λειτουργικού συστήματος της με ονόματα γλυκών (Icecream Sandwich, Jellybean κλπ).

- **Android 5.0 Lollipop**

Στα τέλη του 2014 ανακοινώθηκε το νέο και τελευταίο λειτουργικό με ονομασία Lollipop. Στην νέα έκδοση βλέπουμε νέα χαρακτηριστικά όπως η μεγαλύτερη διάρκεια μπαταρίας, καλύτερη ασφάλεια δεδομένων αλλά και πιο προσιτό προς το χρήστη γραφικό περιβάλλον. Επίσης σημαντική είναι και η ταχύτητα των διεργασιών σε πολλαπλές εφαρμογές που παραμένουν ανοιχτές.

## Κεφάλαιο 2. Εργαλεία ανάπτυξης εφαρμογών

### 2.1 Πακέτο Ανάπτυξης Λογισμικού (SDK)- Android

Η γλώσσα προγραμματισμού Java είναι κατά κύριο λόγο η γλώσσα που χρησιμοποιείται για την ανάπτυξη εφαρμογών. Το επίσημο ολοκληρωμένο περιβάλλον ανάπτυξης του Android είναι μια τροποποιημένη έκδοση του γνωστού περιβάλλοντος Eclipse, το οποίο περιλαμβάνει την τελευταία έκδοση του Android SDK. Το Android SDK είναι το βασικότερο εργαλείο για τους προγραμματιστές οι οποίοι αναπτύσσουν εφαρμογές για το Android. Περιλαμβάνει βιβλιοθήκες (API Libraries), μηχανισμό αποσφαλμάτωσης (debugger) καθώς και τον εικονικό εξομοιωτή συσκευής Android ο οποίος λειτουργεί όπως κάθε συσκευή Android[8][9].

### 2.2 Eclipse IDE

Το Eclipse IDE (Integrated Development Environment) είναι μια πλατφόρμα ανάπτυξης κώδικα που υποστηρίζει πολλές γλώσσες προγραμματισμού. Το Eclipse παρέχει τη δυνατότητα, εκτός από τη ανάπτυξη του κώδικα, την εκτέλεση του για δοκιμαστικούς σκοπούς (testing) αλλά και την αποσφαλμάτωσή του (debugging). Ένα σημαντικό χαρακτηριστικό που καθιστά το Eclipse απαραίτητο εργαλείο για την ανάπτυξη εφαρμογών για το Android είναι ότι υποστηρίζει πλήρως το Android SDK (Software Development Kit). Θεωρείται η καλύτερη και πιο ολοκληρωμένη πλατφόρμα ανάπτυξης εφαρμογών για συσκευές Android[8][9].



Εικόνα 7 Λογότυπο Eclipse



### 2.3 MIT App-Inventor

Η υλοποίηση της παρούσας εφαρμογής πραγματοποιήθηκε με την χρήση του App Inventor καθώς δεν χρειάζεται κάποιος να έχει ιδιαίτερες γνώσεις προγραμματισμού για την ανάπτυξη μιας εφαρμογής. Αυτή είναι και η βασική διαφορά ανάμεσα στο SDK και το App Inventor. Κάνει χρήση Java βιβλιοθηκών, για την δημιουργία των εικονικών blocks, ενώ ο μεταφραστής της γλώσσας των visual blocks (η οποία ενσωματώθηκε και ρυθμίστηκε ειδικά για το Android) χρησιμοποιεί την γλώσσα Kawa Language Framework[9].

### 2.4 Hyper Next Android Creator

Με το Hyper Next δεν υπάρχει καμία ανάγκη για την εκμάθηση java ή Android SDK και είναι πολύ πιο εύκολο στην χρήση από το Eclipse. Το περιβάλλον του Hyper Next έχει μόνο ένα παράθυρο σχεδίασης και μια απλή γραμμή εργαλείων. Έλεγχοι, όπως για παράδειγμα, κουμπιά, μπορούν να τοποθετηθούν γρήγορα και η επεξεργασία τους γίνεται με την χρήση του Script Editor[9].

### 2.5 Windows phone SDK

Για το Windows Phone υπάρχει το πακέτο λογισμικού Windows Phone SDK το οποίο περιλαμβάνει το ομώνυμο SDK, το πρόγραμμα Visual Studio Express ως IDE και κάνει χρήση οποιασδήποτε γλώσσας προγραμματισμού υποστηρίζεται από το πλαίσιο λογισμικού .NET, που χρησιμοποιείται από την Microsoft[9].

### 2.6 iOS SDK-Apple

Για το iOS της εταιρίας Apple, αντίπαλο δέος της Google, υπάρχει το πακέτο λογισμικού «Εργαλεία Προγραμματιστή iOS» (iOS Developer Toolset) το οποίο περιλαμβάνει το iOS SDK, το πρόγραμμα XCode ως IDE και κάνει χρήση της γλώσσας προγραμματισμού Objective-C (υπερσύνολο της C). Στο Xcode βρίσκεται ουσιαστικά όλος ο κώδικας της εφαρμογής[5][9].

## Κεφάλαιο 3. MIT App Inventor



### 3.1 Τι είναι το App Inventor

Το App Inventor είναι μια web εφαρμογή ανοικτού κώδικα γραμμένη σε γλώσσα Java, που δημιουργήθηκε από τη Google. Χρησιμοποιείται για την υλοποίηση εφαρμογών για φορητές συσκευές με περιβάλλον Android. Στη συνέχεια το λογισμικό πέρασε στο MIT το οποίο το υποστηρίζει μέχρι σήμερα. Το περιβάλλον είναι πολύ εύχρηστο και επιτρέπει ακόμα και αρχάριους χρήστες να δημιουργήσουν τις δικές τους εφαρμογές, χωρίς να χρειάζεται να γράψουν κώδικα. Η τρέχουσα έκδοση της εφαρμογής είναι η 2 και έγινε διαθέσιμη το Δεκέμβριο του 2013[7],[10].

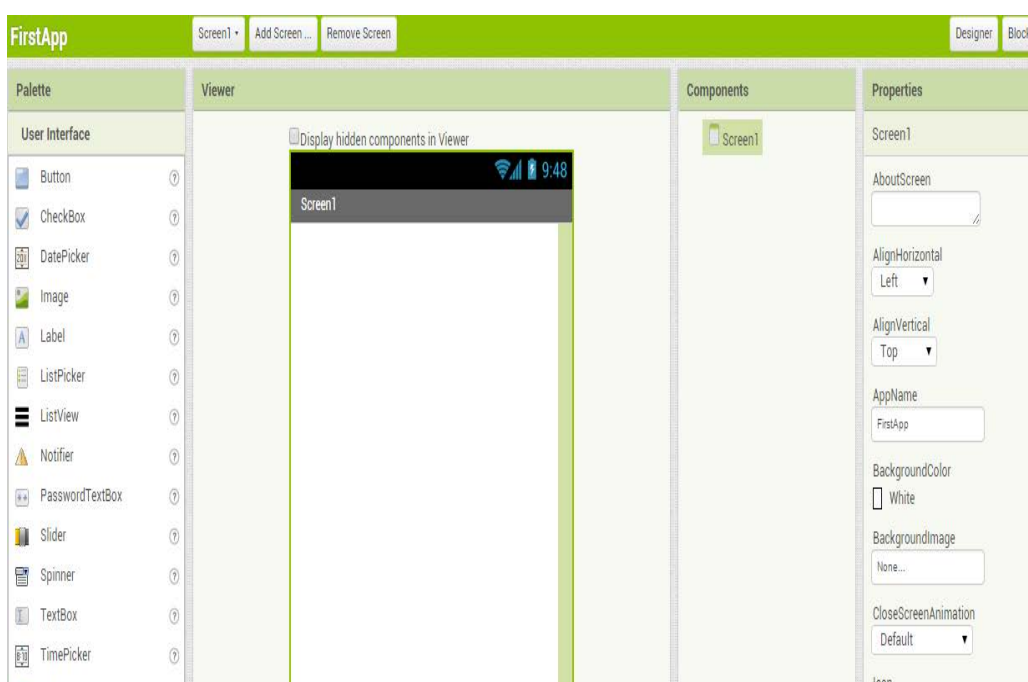
Πιο συγκεκριμένα για την ανάπτυξη εφαρμογών με το App Inventor είναι απαραίτητη η χρήση ενός φυλλομετρητή (web browser), σύνδεση στο Internet και ένα τηλέφωνο με Android. Αποτελεί έξυπνη ιδέα αφού δίνει την ευκαιρία σε όποιον ενδιαφέρεται αλλά δεν έχει τις απαραίτητες γνώσεις, όπως γλώσσα C και Java, να φτιάξει μια εφαρμογή για Android. Υπάρχουν οδηγίες χρήσης της εφαρμογής, το περιβάλλον στο οποίο η εισαγωγή γίνεται μόνο αν ο χρήστης διαθέτει λογαριασμό της Google, μαθήματα εκμάθησης, βιβλιοθήκες που παρέχουν οδηγίες, έγγραφα, και αναφορές, εκπαιδευτικά εγχειρίδια (για εκπαιδευτικούς) και forum στο οποίο ο χρήστης μπορεί να ανταλλάξει ιδέες και να λύσει προβλήματα που εμφανίζονται στις εφαρμογές του. Το App Inventor χρησιμοποιείται ήδη για εκπαιδευτικούς σκοπούς σε Σχολεία και Πανεπιστήμια για την υλοποίηση εφαρμογών.

### 3.2 Ανάπτυξη εφαρμογών

Το γραφικό περιβάλλον προγραμματισμού του App Inventor [5][7][10] αποτελείται από τον Designer, τον Blocks Editor, και τον Emulator:

#### 1. Designer

Στον Designer γίνεται η σχεδίαση της εφαρμογής, όπου επιλέγονται τα βασικά δομικά στοιχεία της εφαρμογής που θα δημιουργηθεί, όπως κουμπιά, ήχος, πεδία κειμένου, εικόνες κλπ. Για την δημιουργία νέου project επιλέγουμε απ το μενού **My projects/New** και επιλέγουμε το επιθυμητό όνομα της εφαρμογής που θα δημιουργήσουμε.



Εικόνα 8 Δομή του Designer

#### ➤ Βασικές επιλογές και χαρακτηριστικά

Το μενού της εφαρμογής, περιλαμβάνει ένα γραφικό περιβάλλον που χωρίζεται σε τρία τμήματα. Στο κεντρικό τμήμα, κυριαρχεί η (υποθετική) οθόνη του smartphone (**Viewer**). Στα αριστερά, υπάρχουν οι κατηγορίες με τα αντικείμενα που μπορεί να προσθέσει στην οθόνη (**Palette**), με τη χρήση της μεθόδου drag and drop και στα δεξιά υπάρχουν οι

καρτέλες ιδιοτήτων και συστατικών όπου ο χρήστης μπορεί να ρυθμίσει το κάθε αντικείμενο, που έχει προσθέσει, ξεχωριστά (**Components**).

Η πραγματική συμπεριφορά των Components καθορίζεται στον Blocks Editor. Η λογική που πρέπει να ακολουθήσει ο χρήστης, είναι να φτιάξει πρώτα το γραφικό περιβάλλον της εφαρμογής του, επιλέγοντας τα χαρακτηριστικά που θέλει και στη συνέχεια να μεταβεί από το χώρο σχεδίασεως στο χώρο των μπλοκ όπου εκεί θα δηλώσει μεταβλητές και λειτουργίες (**Blocks Editor**).

Στο πάνω μέρος της σελίδας υπάρχει ένα μενού από drop down τα οποία παρέχουν επιπρόσθετες λειτουργίες. Τα μενού αυτά είναι Project από το οποίο δημιουργείται νέο project, εισαγωγή project από τον υπολογιστή του χρήστη, διαγραφή και αποθήκευση. Το μενού Connect επιτρέπει τη δοκιμή της εφαρμογής είτε σε προσομοιωτή, είτε σε φορητή συσκευή. Το μενού Build παρέχει στο χρήστη την εφαρμογή σε δύο μορφές. Η πρώτη δείχνει ένα κώδικα QR ώστε ο χρήστης να κατεβάσει την εφαρμογή στη συσκευή (άμεσα) και η δεύτερη παράγει το apk ώστε να μεταφερθεί στη συσκευή έμμεσα, π.χ. μέσω καλωδίου. Το My Projects εμφανίζει μια λίστα με όλα τα projects που έχει δημιουργήσει ο χρήστης και εμφανίζει ημερομηνίες δημιουργίας και τελευταίας τροποποίησης.

Το αριστερό τμήμα, που ονομάζεται παλέτα (Pallete), χωρίζεται στις εξής κατηγορίες. Διεπαφή χρήστη, Διάταξη, Μέσα, Σχεδίαση και Σχεδιοκίνηση, Αισθητήρες, Αποθήκευση, Συνδεσιμότητα, Social. Το καθένα από αυτά περιλαμβάνει διάφορα αντικείμενα, όπως φαίνεται και στον πίνακα που ακολουθεί.

Διεπαφή χρήστη	Πλήκτρο, checkbox, ρολόι, εικόνα, ετικέτα, λίστα, ειδοποίηση, πεδίο password, πεδίο κειμένου, κύλιση οριζόντιου άξονα, πρόσβαση σε περιβάλλον web
Διάταξη	Οριζόντια, κάθετη, πίνακας
Μέσα	Φωτογραφική μηχανή, λήψη βίντεο, ηχογράφηση, αναπαραγωγή, αναγνώριση ήχου, μετατροπή κειμένου σε ήχο
Σχεδίαση και Σχεδιοκίνηση	Sprite, Περιβάλλον σχεδίασης και κίνησης των sprites και ball (αντικείμενο που έχει περιορισμούς σε σχέση με το sprite)
Αισθητήρες	Επιταχυνσιόμετρο, εντοπιστής θέσης και σαρωτής barcode

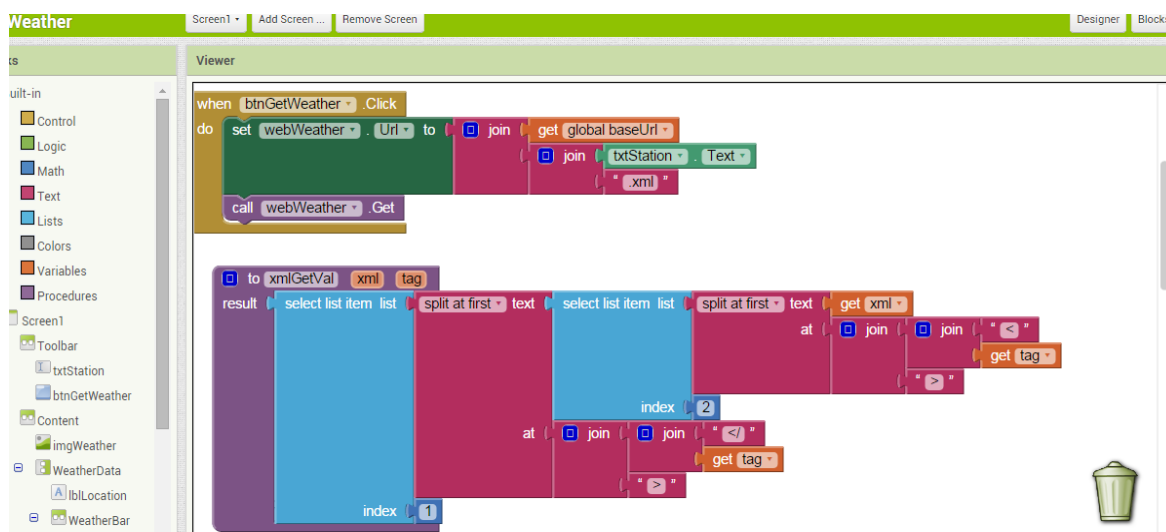
Αποθήκευση	TinyDB, TinyWebDB, πρόκειται για τις βάσεις δεδομένων που μπορούν να αποθηκευτούν δεδομένα και πληροφορίες που αφορούν την εφαρμογή
Συνδεσιμότητα	Web,εκκίνηση δραστηριότητας, Bluetooth πελάτης ή διακομιστής
Social	Επιλογή επαφής, email, τηλεφωνική κλήση, γραπτά μηνύματα

Πίνακας 1 Βασικό μενού Designer-Palette

## 2. Blocks Editor

Στον App Inventor Blocks Editor, λαμβάνει χώρα ο καθορισμός – προγραμματισμός της συμπεριφοράς των βασικών δομικών στοιχείων της εφαρμογής. Εδώ ο προγραμματισμός συμβαίνει με δόμηση των προγραμμάτων με οπτικό τρόπο, ταιριάζοντας πλακίδια μαζί όπως τα κομμάτια ενός πάζλ. Ο χρήστης σχεδιάζει το περιβάλλον της κάθε οθόνης όπως επιθυμεί με τη χρήση των συστατικών που αναφέρθηκαν παραπάνω. Στη συνέχεια, για την κάθε μια οθόνη μεταβαίνει στην καρτέλα των Blocks, όπως φαίνεται και στην εικόνα 9.

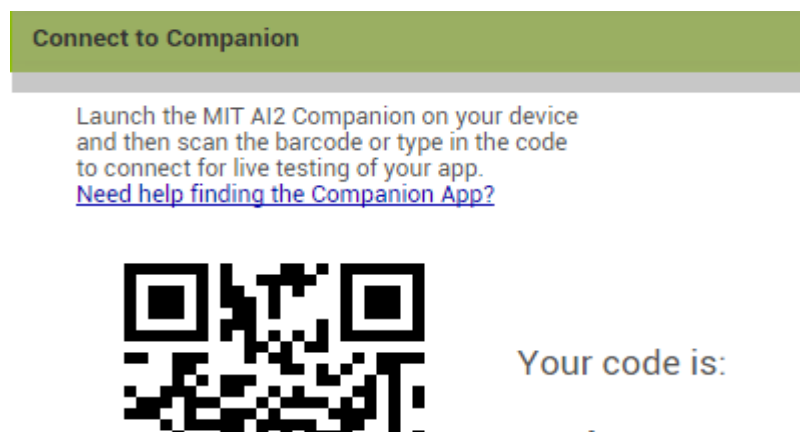
Ο Blocks editor χρησιμοποιεί την Open Blocks Java library μια βιβλιοθήκη της Java που επιτρέπει την χρησιμοποίηση εικονικών δομικών στοιχείων (visual component blocks). Ο compiler που μετατρέπει την Visual Block Language για την εφαρμογή στο Android χρησιμοποιεί την Kawa, μια γλώσσα προγραμματισμού βασισμένη στην γλώσσα προγραμματισμού Scheme[5][7][10].



Εικόνα 9 Δομή Blocks Editor-πλακίδια

### 3. Emulator

Στον App Inventor Emulator, όπου εμφανίζεται η εφαρμογή που δημιουργείται από τον προγραμματιστή στον εξομοιωτή τηλεφώνου βήμα-προς-βήμα, όπως προστίθενται τα κομμάτια σε αυτή, ώστε να είναι δυνατός ο έλεγχος της εφαρμογής καθώς αυτή χτίζεται. Με την ολοκλήρωση της εφαρμογής, υπάρχει η δυνατότητα πακεταρίσματος(apk) της εφαρμογής έτσι ώστε να είναι δυνατή η εγκατάσταση της σε Android συσκευές. Ο χρήστης βλέπει την εκτέλεση της εφαρμογής του και σε πραγματικό χρόνο μπορεί να επέμβει στον κώδικα και να κάνει αλλαγές. Οι αλλαγές περνούν αυτόματα στη συσκευή. Για να λειτουργήσει αυτή η μέθοδος, πρέπει στη συσκευή να εγκατασταθεί το MIT AI2 Companion App που είναι διαθέσιμο στο Google Play. Αμέσως μετά πραγματοποιείται η σύνδεση μεταξύ του App Inventor και της συσκευής μέσω κώδικα QR ή με την εισαγωγή στη συσκευή ενός κωδικού που παρέχει το App Inventor[10].



Εικόνα 10 QR code

#### 3.3 Απαιτήσεις Συστήματος

- Λειτουργικό σύστημα: Mac OS X 10.5, Windows XP, Ubuntu 8, Debian 5 και νεότερες εκδόσεις.
- Browser: Mozilla Firefox 3.6, Apple Safari 5.0, Google Chrome 4.0, Microsoft Internet Explorer 7 και νεότερες εκδόσεις
- Σύνδεση στο Διαδίκτυο, Java 7 και νεότερες εκδόσεις [10].

## Κεφάλαιο 4ο Wunderground



### 4.1 Γενικά

Η υπηρεσία Wunderground παρέχει δωρεάν πληροφορίες καιρού για οποιαδήποτε ημερομηνία σε όλο τον κόσμο. Παρέχει τόσο ιστοσελίδα για την πρόγνωση του καιρού όσο και εφαρμογή στο Play Store. Δημόσια βγήκε στο διαδίκτυο το 2003 και σήμερα θεωρείται από τις καλύτερες υπηρεσίες καιρού παγκοσμίως. Τα δεδομένα παρέχονται από 100.000 και πάνω τοπικούς σταθμούς πρόγνωσης καιρικών φαινομένων. Επιπλέον για προγραμματιστές παρέχεται δωρεάν API για οποιαδήποτε χρήση, όπως και στην παρούσα εργασία, σε μορφή XML και JSON[11].

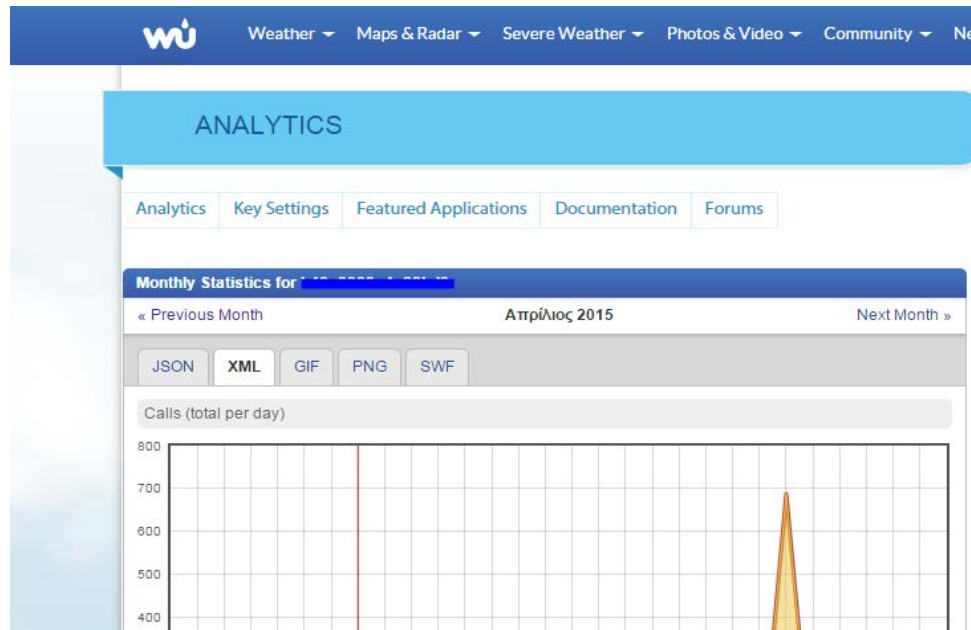
### 4.2 Υπηρεσίες

Όπως προαναφέραμε οι υπηρεσίες της Wunderground παρέχονται δωρεάν. Παρόλα αυτά για την χρησιμοποίηση κάποιων λειτουργιών όπως και στην παρούσα εργασία ήταν απαραίτητη η εγγραφή μας στην σελίδα.

Αρχικά επισκεφθήκαμε την ιστοσελίδα <http://wunderground.com>. Στην συνέχεια από το μενού επιλέξαμε την ενότητα “more” και ακολούθως την επιλογή “Weather Api for Developers”, όπου θα μας ζητηθεί να κάνουμε εγγραφή πατώντας την επιλογή “Sign Up”. Με την εγγραφή μας θα παραχθεί αυτόματα ένα προσωπικό **API KEY**, το οποίο και χρησιμοποιούμε για την υλοποίηση της εφαρμογής.

Εφόσον έχουμε ολοκληρώσει την εγγραφή βάζοντας το επιθυμητό Όνομα χρήστη (username) και Κωδικό (password) πλέον έχουμε μπροστά μας το βασικό μενού

διαχείρισης των υπηρεσιών API για προγραμματιστές όπως φαίνεται στην παρακάτω εικόνα (Εικόνα 11).



Εικόνα 11 Μενού διαχείρισης WU-API

### Βασικές υπηρεσίες

- Διαδραστικός χάρτης με την πρόγνωση καιρού (WunderMap)
- 10ήμερη απεικόνιση μέσω δορυφόρων των καιρικών συνθηκών
- Προειδοποιήσεις για επικίνδυνα καιρικά φαινόμενα όπως τυφώνες, πυρκαγιές και καταιγίδες
- 100.000 προσωπικούς μετεωρολογικούς σταθμούς παγκοσμίως
- API για προγραμματιστές σε 80 γλώσσες, με χρήση XML, JSON δεδομένων
- Ζωντανή απεικόνιση με φωτογραφίες και βίντεο της περιοχής(Webcam)
- Ιστορικά και στατιστικά στοιχεία



### 4.3 Βασικό μενού διαχείρισης

- **Analytics:** Εμφανίζει στατιστικά στοιχεία ανά μήνα με βάση το προσωπικό API Key και την χρήση που έχει γίνει στις υποστηριζόμενες μορφές JSON, XML, GIF, PNG, SWF.
- **Key Settings:** Παρέχει πληροφορίες και ρυθμίσεις για το προσωπικό API Key
- **Featured Applications:** Εμφανίζει υπηρεσίες και εφαρμογές που χρησιμοποιούν το API του Wunderground.
- **Documentation:** Παρέχονται όλες οι δυνατότητες, οι ρυθμίσεις και οι λειτουργίες του API με παραδείγματα για την κατανόηση τους. Στην ουσία παρέχονται τα δεδομένα που μπορεί να χρησιμοποιήσει κανείς στην εφαρμογή του. Στην παρούσα εργασία ο τρόπος υλοποίησης της εφαρμογής έγινε με συνδυασμό λειτουργιών από την συγκεκριμένη ενότητα. Στο επόμενο κεφάλαιο αναλύεται διεξοδικώς ο τρόπος δημιουργίας της εφαρμογής “EpirusWeatherApp”.
- **Forums:** Κοινότητα υποστήριξης μελών για απορίες και προβλήματα χρήσης της υπηρεσίας.

### 4.4 Άλλοι πάροχοι

Εκτός από το Wunderground για την πρόγνωση καιρού υπάρχουν επιπλέον χιλιάδες πάροχοι, μετεωρολογικοί σταθμοί και υπηρεσίες παγκοσμίως. Γι αυτό και οι καιρικές συνθήκες μπορεί να διαφέρουν μεταξύ αυτών. Παρακάτω παρουσιάζουμε ενδεικτικά σημαντικούς πάροχους στην Ελλάδα και τον υπόλοιπο κόσμο. Οι περισσότεροι από αυτούς παρέχουν API για προγραμματιστές και εφαρμογές για κινητές συσκευές.

1. [www.accuweather.com](http://www.accuweather.com)
2. [www.weather.com](http://www.weather.com)
3. <https://weather.yahoo.com>
4. [www.meteo.gr](http://www.meteo.gr)
5. [www.okairos.gr](http://www.okairos.gr)
6. [www.emy.gr](http://www.emy.gr) (Εθνική Μετεωρολογική υπηρεσία)

## Κεφάλαιο 5. Απαιτούμενες γλώσσες προγραμματισμού

### 5.1 JAVA

Η Java είναι μία αντικειμενοστραφής γλώσσα προγραμματισμού που σχεδιάστηκε από την εταιρία πληροφορικής Sun Microsystems. Ένα από τα βασικά χαρακτηριστικά της Java έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα. Η γλώσσα προγραμματισμού Java είναι κατά κύριο λόγο η γλώσσα που χρησιμοποιείται για την ανάπτυξη εφαρμογών για το Android. Στην παρούσα εργασία η Java βρίσκεται “πίσω” από τον Blocks Editor στον APP Inventor κατά την ένωση και προγραμματισμό των πλακιδίων [14].

### 5.2 XML

Σε ένα κόσμο όπου οι πληροφορίες παρέχονται μέσω του παγκόσμιου διαδικτύου, τα έγγραφα πρέπει να είναι εύκολα προσβάσιμα, μεταφερόμενα και ευέλικτα. Πρέπει επίσης να είναι ανεξάρτητα οποιουδήποτε συστήματος και περιεχομένου.

Η XML σχεδιάστηκε ώστε να δώσει στα έγγραφα ένα μεγαλύτερο επίπεδο προσαρμοστικότητας στο στυλ και στη δομή από αυτό που υπήρχε παλαιότερα στην HTML. Στην πραγματικότητα, η XML είναι μια markup γλώσσα για έγγραφα που περιέχουν δομημένες πληροφορίες, δηλαδή ένας μηχανισμός που καθορίζει δομές σε ένα έγγραφο. Οι δομημένες πληροφορίες περιλαμβάνουν περιεχόμενο και κάποιες διευκρινίσεις για το ρόλο που παίζει το περιεχόμενο και σχεδόν όλα τα έγγραφα έχουν την ίδια δομή. Στην XML οι χρήστες έχουν τη δυνατότητα να καθορίζουν τα tags και τις δομημένες σχέσεις μεταξύ τους[12][13].

Ένα λογισμικό μοντέλο που καλείται επεξεργαστής XML χρησιμοποιείται να διαβάζει XML έγγραφα και παρέχει πρόσβαση στο περιεχόμενο και τη δομή τους. Υποτίθεται ότι ο επεξεργαστής XML λειτουργεί εκ μέρους ενός άλλου μοντέλου που καλείται application (εφαρμογή). Αυτή η προδιαγραφή περιγράφει την απαιτούμενη συμπεριφορά του επεξεργαστή και συγκεκριμένα πως θα πρέπει να διαβάζει τα XML δεδομένα και ποιές

πληροφορίες πρέπει να παρέχει στην εφαρμογή. Η υπηρεσία Wunderground παρέχει τα δεδομένα μέσω XML, τρόπος με τον οποίο έγινε και η παρούσα εφαρμογή κατά την υλοποίηση της όπως θα δούμε στον επόμενο κεφάλαιο.

### 5.3 JSON

Το JSON (JavaScript Object Notation) είναι ένα ελαφρύ πρότυπο ανταλλαγής δεδομένων. Είναι εύκολο για τους ανθρώπους να το διαβάσουν και να το γράψουν και για τις μηχανές να το αναλύσουν και να το παράγουν. Είναι βασισμένο πάνω σε ένα υποσύνολο της γλώσσας προγραμματισμού JavaScript[15]. Το JSON είναι ένα πρότυπο κειμένου το οποίο είναι τελείως ανεξάρτητο από γλώσσες προγραμματισμού αλλά χρησιμοποιεί πρακτικές οι οποίες είναι γνωστές στους προγραμματιστές της οικογένειας προγραμματισμού C, συμπεριλαμβανομένων των C, C++, C#, Java, JavaScript, Perl, Python, και πολλών άλλων. Αυτές οι ιδιότητες κάνουν το JSON μια ιδανική γλώσσα προγραμματισμού ανταλλαγής δεδομένων. Παρακάτω (εικόνα 12) φαίνεται η διαφορά στην μορφή xml και json.

```
JSON
{
  "siblings": [
    {"firstName": "Anna", "lastName": "Clayton"},
    {"lastName": "Alex", "lastName": "Clayton"}
  ]
}

XML
<siblings>
<sibling>
<firstName>Anna</firstName>
<lastName>Clayton</lastName>
</sibling>
<sibling>
<firstName>Alex</firstName>
<lastName>Clayton</lastName>
</sibling>
</siblings>
```

Εικόνα 12 XML vs JSON

## Κεφάλαιο 6. Ανάλυση της εφαρμογής

### 6.1 Brand Name

Για τη δημιουργία, την κατανόηση και την καλύτερη λειτουργία του έργου της παρούσας πτυχιακής, έπρεπε να δημιουργηθεί ένα όνομα της εφαρμογής. Το όνομα που δόθηκε είναι EpirusWeatherApp και το λογότυπο εμφανίζεται παρακάτω (εικόνα 13). Σχεδιάστηκε με την εφαρμογή σχεδίασης CorelDraw X6 Trial.



Εικόνα 13 Λογότυπο Εφαρμογής

### 6.2 Εισαγωγή στην υλοποίηση της εφαρμογής

Η υλοποίηση της εφαρμογής αυτής έχει ως στόχο την πρόγνωση καιρού με χρήση του Ίντερνετ από μια Android συσκευή μέσω της υπηρεσίας Wunderground [10],[16]. Το πρόγραμμα που χρησιμοποιήθηκε για την κατασκευή της εφαρμογής ήταν το MIT App Inventor. Ο χρήστης θα μπορεί να βλέπει τις τωρινές καιρικές συνθήκες αλλά και για το επόμενο τριήμερο. Αρχικά μπαίνουμε στην σελίδα του App Inventor (<http://appinventor.mit.edu/explore/>) και πατώντας create μπορούμε να ξεκινήσουμε μια νέα εφαρμογή. Απαραίτητο για την σύνδεση, και την χρήση της πλατφόρμας είναι η ύπαρξη λογαριασμού Gmail. Εφόσον έχουμε συνδεθεί, βρισκόμαστε στην αρχική σελίδα όπου πατώντας **Start New Project** δημιουργούμε μια νέα εφαρμογή, όπως φαίνεται και στην εικόνα 8. Με τον τρόπο αυτό βρισκόμαστε στον «Designer». Στην αριστερή πλευρά

βλέπουμε τα βασικά δομικά στοιχεία που μπορούμε να χρησιμοποιήσουμε (user interface). Από εκεί με «drag n' drop» μπορούμε να εισάγουμε το κάθε στοιχείο. Αν θέλουμε να χρησιμοποιήσουμε ένα κουμπί (button), στην εφαρμογή που θα κάνει κάτι μόλις το πατήσουμε, τότε εισάγουμε το button και μπορούμε να του αλλάξουμε τις ιδιότητες στην δεξιά μεριά της σελίδας (properties).

Όπως βλέπουμε μπορούμε να εισάγουμε ένα κουμπί μια εικόνα (image), ήχο (sound) στην καρτέλα media, ένα κείμενο (textbox), ακόμα και μία βάση δεδομένων (tinydb) από την καρτέλα storage για να αποθηκεύσει τα στοιχεία που θέλουμε. Πρέπει να σημειωθεί ότι ο «Designer» είναι μόνο για να βάλουμε τα δομικά στοιχεία της εφαρμογής και αποτελεί το ένα από τα δύο μέρη για την ολοκλήρωση της. Απομένει να ενώσουμε κατάλληλα όλα αυτά τα στοιχεία στον «Blocks Editor» για να είναι έτοιμη προς δοκιμή η εφαρμογή μας.

### 6.3 Κατασκευή της εφαρμογής

Αρχικά για την δημιουργία της εφαρμογής δημιουργήσαμε τέσσερις οθόνες, ώστε να εναλλασσόμαστε μεταξύ αυτών και να μπορούμε σε κάθε μία να εισάγουμε και κάτι νέο. Όλες οι ενέργειες γίνονται με σύρσιμο «drag n' drop» των στοιχείων στον καμβά[18][19]. Η καθεμία εξυπηρετεί και συγκεκριμένο σκοπό στην εφαρμογή.

- **Screen 1** (Οθόνη έναρξης με το λογότυπο)
- **EpirusWeatherApp** («Ζωντανές» καιρικές συνθήκες και των τεσσάρων πόλεων)
- **Arta3daysForecast** (3μερη πρόγνωση για την Άρτα, Ιωάννινα, Ηγουμενίτσα, Πρέβεζα)
- **Search** (Αναζήτηση οποιασδήποτε πόλη της Ελλάδας)

#### 6.3.1. Designer

Στον Designer όπως προαναφέραμε γίνεται ο σχεδιασμός της εφαρμογής. Παρακάτω αναλύονται όλες οι οθόνες που αναφέρθηκαν προηγουμένως και ο τρόπος υλοποίησής τους κάθε μιας ξεχωριστά.

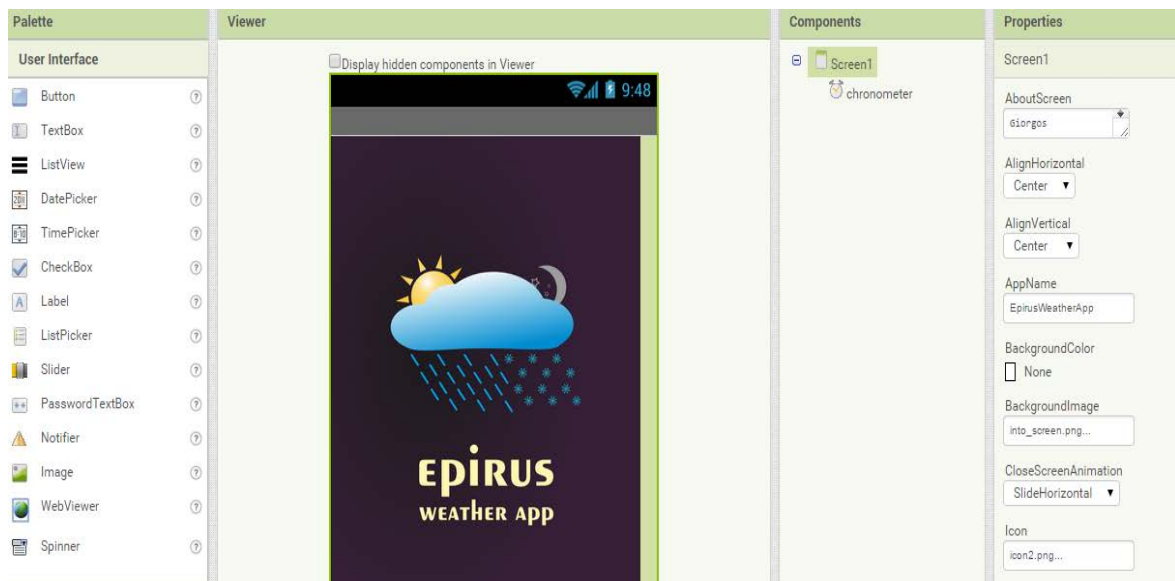
- **Screen 1**

Η Screen 1 λειτουργεί ως η αρχική οθόνη με την έναρξη της εφαρμογής. Εισάγαμε μια εικόνα ως λογότυπο, και ένα χρονοδιακόπτη για την μεταφορά στην επόμενη σελίδα αυτόματα.

**User interface** → **image**

**Properties** → **BackgroundImage** → **λογότυπο EpirusWeatherApp**

**Sensors** → **clock**

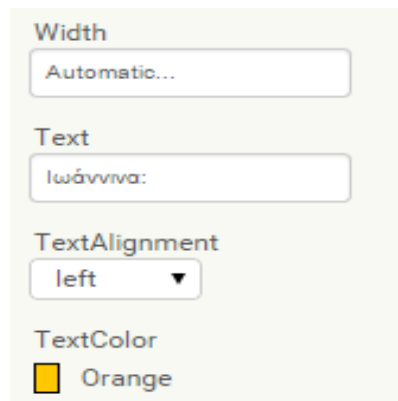


Εικόνα 14 Αρχική οθόνη εφαρμογής

- **Screen\_EpirusWeatherApp**

Η οθόνη αυτή εμφανίζει τον καιρό και για τις τέσσερις πόλεις της Ηπείρου. Συγκεκριμένα εμφανίζει θερμοκρασία, κατάσταση, εικόνα και υγρασία για τις πόλεις Ιωάννινα, Άρτα, Πρέβεζα και Ηγουμενίτσα. Επιπλέον εμφανίζει «drop-down» μενού (**τριήμερο**) για την επιλογή πόλης και εμφάνιση 3μερης πρόγνωσης. Τέλος έχει κουμπί αναζήτησης για οποιαδήποτε πόλη της Ελλάδας.

**User interface** → **Label x4** → **properties** → **text** μετονομασία σε Ιωάννινα, Άρτα, Ηγ/νιτσα, Πρέβεζα



Εικόνα 15 Μετονομασία ετικέτας

**User interface**→ **image** x4

**User interface**→ **Listpicker**→ **Properties**→ **text** (τριήμερο)

**User interface**→ **Button** → **Components** → μετονομασία σε Bn\_Search

**Layout**→ **table, horizontal, vertical**→ τακτοποίηση εικονιδίων

**Connectivity**→ **web**→ **Components**→ μετονομασία σε WebAutocomplete, WebConditions

**Social**→ **sharing**

**User interface**→ **Notifier**

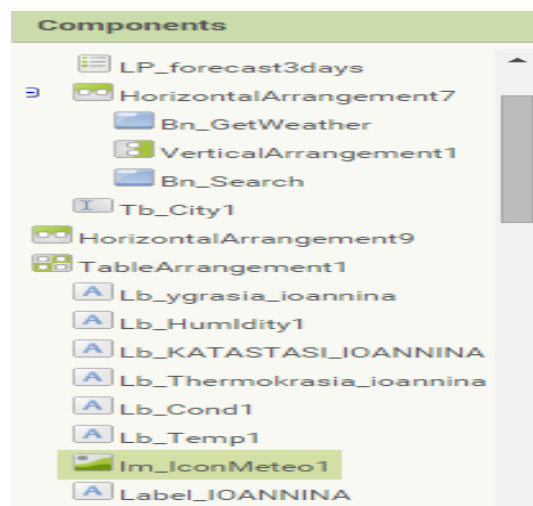
Πιο αναλυτικά, στην οθόνη αυτή θέλουμε να εμφανίζεται ο καιρός που υπάρχει **τόρα** και στις τέσσερις πόλεις. Δημιουργήθηκαν τέσσερις ετικέτες για την κάθε μία πόλη ξεχωριστά και στην κάθε μία πόλη επιπλέον τρεις ετικέτες με στόχο την εμφάνιση της θερμοκρασίας, κατάσταση, υγρασία όπως φαίνεται και στην εικόνα 16. Ακόμη εισάγαμε τέσσερις εικόνες ώστε να εμφανίζεται και το αντίστοιχο εικονίδιο καιρού, πχ αν έχει ήλιο, αν βρέχει κλπ.

Στη συνέχεια δημιουργήσαμε μια λίστα (listpicker) με την ονομασία **τριήμερο** και την εκχώρηση των τεσσάρων πόλεων, ώστε επιλέγοντας μια πόλη να εμφανίζεται η τριήμερη πρόγνωση του καιρού. Επιπλέον εισάγαμε και ένα κουμπί αναζήτησης με την αντίστοιχη εικόνα ώστε ο χρήστης να ανακατευθύνεται σε νέα οθόνη, ώστε να αναζητήσει μια πόλη και να εμφανίσει τον καιρό. Τα δεδομένα της εφαρμογής, οι καιρικές συνθήκες, μπορεί να στέλνονται με χρήση του sharing(e-mail, viber, Bluetooth κλπ). Τέλος εισάγαμε την επιλογή web όπου με αυτόν τον τρόπο γίνεται η σύνδεση μεταξύ της εφαρμογής, του διαδικτύου και του wunderground. Ο τρόπος που ενώνονται όλα τα παραπάνω γίνεται στον designer. Παρακάτω (εικόνες 16, 17) φαίνονται μερικά από τα πεδία που έχουμε

εισάγει στην οθόνη αυτή. Για την λειτουργία της εφαρμογής απαραίτητη είναι η χρήση του διαδικτύου. Αν δεν υπάρχει σύνδεση θα εμφανιστεί σφάλμα με το αντίστοιχο μήνυμα κάνοντας χρήση του notifier.



Εικόνα 16 Οθόνη EpirusWeatherApp



Εικόνα 17 Components-βασικές επιλογές



- **Screen\_Search**

Σκοπός της οθόνης αυτής είναι η αναζήτηση οποιασδήποτε πόλη της Ελλάδας και την εμφάνιση των καιρικών φαινομένων.

**User interface**→**button** x2

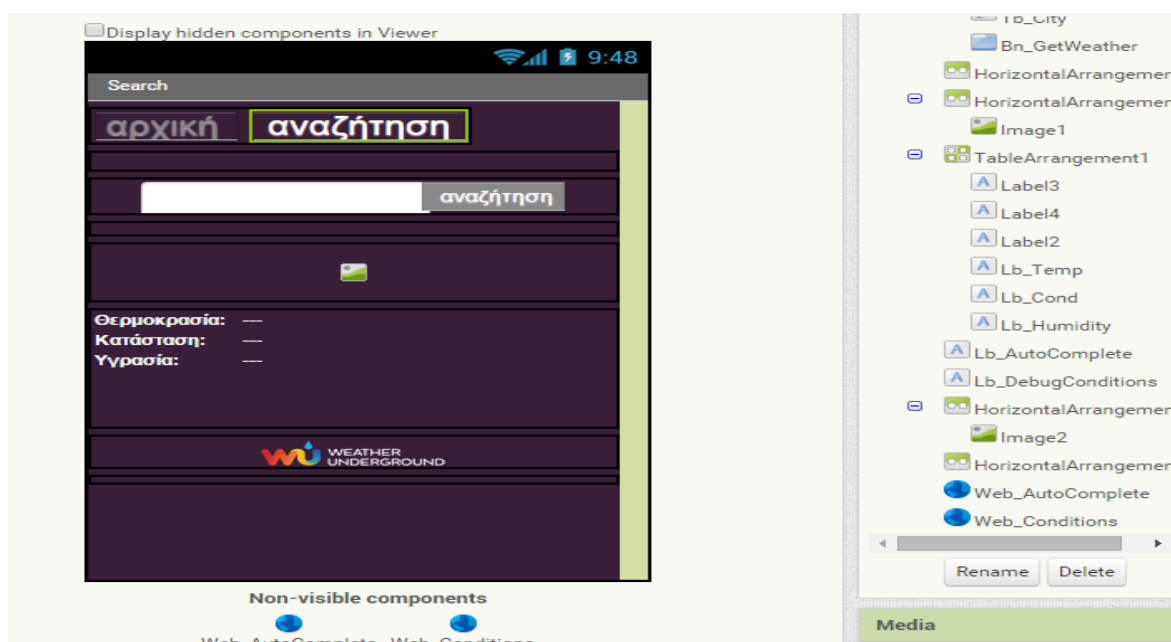
**User interface**→**textbox** (πεδίο κειμένου αναζήτησης)

**User interface**→ **Label** x4 → **text** μετονομασία σε θερμοκρασία, κατάσταση, υγρασία

**User interface**→**image**

**Connectivity**→**web**→**Components**→μετονομασία σε WebAutocomplete, WebConditions

Στην οθόνη αυτή δημιουργήσαμε 2 κουμπιά, αρχική, αναζήτηση(getweather) με στόχο πατώντας το κουμπί αρχική να μεταφερόμαστε στην οθόνη EpirusWeatherApp και πατώντας την επιλογή αναζήτηση η εφαρμογή να ψάχνει να βρει τα δεδομένα από την πόλη που έχει πληκτρολογήσει ο χρήστης στο πεδίου κειμένου. Επίσης δώσαμε τίτλο σε τρεις ετικέτες ώστε να εμφανίζεται η θερμοκρασία, η κατάσταση και η υγρασία της πόλης μαζί με το εικονίδιο. Τέλος εισάγαμε και εδώ το web όπου με αυτόν τον τρόπο γίνεται η σύνδεση μεταξύ της εφαρμογής, του διαδικτύου και του wunderground. Η συνολική εμφάνιση και οι επιλογές φαίνεται στην παρακάτω εικόνα.



Εικόνα 18 Εμφάνιση και επιλογές οθόνης search

- **Screen 3daysforecast**

Η οθόνη αυτή δημιουργήθηκε για την τριήμερη εμφάνιση των καιρικών συνθηκών στην πόλη που θα επιλέξει ο χρήστης [16][19][20]. Τα στοιχεία που χρησιμοποιήσαμε είναι τα ίδια με αυτά όπως και στις προηγούμενες οθόνες. Εμφανίζεται η ελάχιστη και η μέγιστη θερμοκρασία για την κάθε ημέρα καθώς και αντίστοιχο εικονίδιο. Όπως και στις προηγούμενες οθόνες και εδώ θα εμφανιστεί μήνυμα λάθους αν δεν υπάρχει σύνδεση σε κάποιο δίκτυο.

**User interface**→ **Label** (ημέρα, νύχτα 1,2,3,4)

**User interface**→ **image** x8

**Connectivity**→**web**

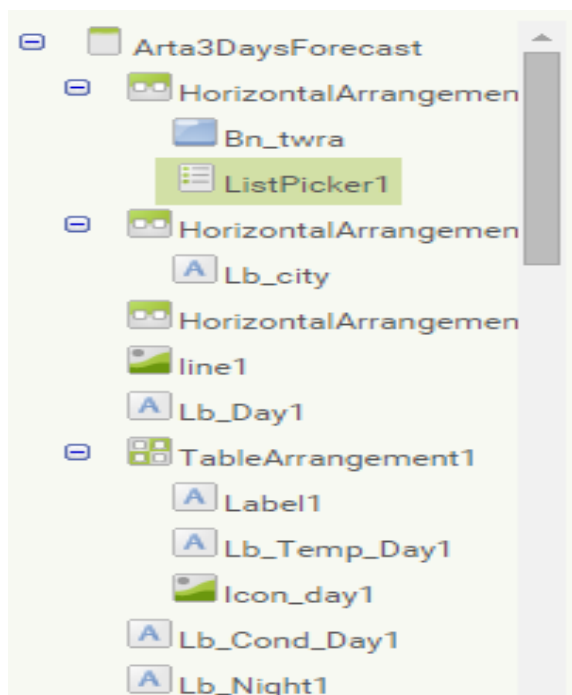
**User interface**→**button**→μετονομασία σε button\_twra

**User interface**→**listpicker**

**User interface**→**notifier**



Εικόνα 19 Οθόνη τριήμερης πρόγνωσης



Εικόνα 20 Βασικές ρυθμίσεις οθόνης 3<sup>ης</sup> πρόγνωσης

### 6.3.2. Blocks Editor

Στον Blocks Editor όπως προαναφέρθηκε και σε προηγούμενο κεφάλαιο γίνεται ο προγραμματισμός όλων αυτών των στοιχείων που έχουμε εισάγει στον Designer. Έτσι για την κάθε οθόνη ενώθηκαν κατάλληλα όλα τα υλικά για την ολοκλήρωση της εφαρμογής. Παρακάτω παρουσιάζεται αναλυτικά η κάθε οθόνη και ο τρόπος προγραμματισμού[10][17].

- **Screen1**

Στην οθόνη αυτήν που είναι και οθόνη έναρξης ο κώδικας της εφαρμογής είναι μικρός και απλός, αφού το μόνο που υπάρχει είναι ένα χρονόμετρο για μετάβαση στην επόμενη σελίδα αυτόματα, χωρίς να πατήσουμε κάτι. Στην ουσία αυτό που ζητάμε από τον κώδικα είναι «όταν το χρονόμετρο τελειώσει, άνοιξε την οθόνη με το όνομα EpirusWeatherApp».



Εικόνα 21 Screen 1 blocks editor

- **Screen\_EpirusWeatherApp**

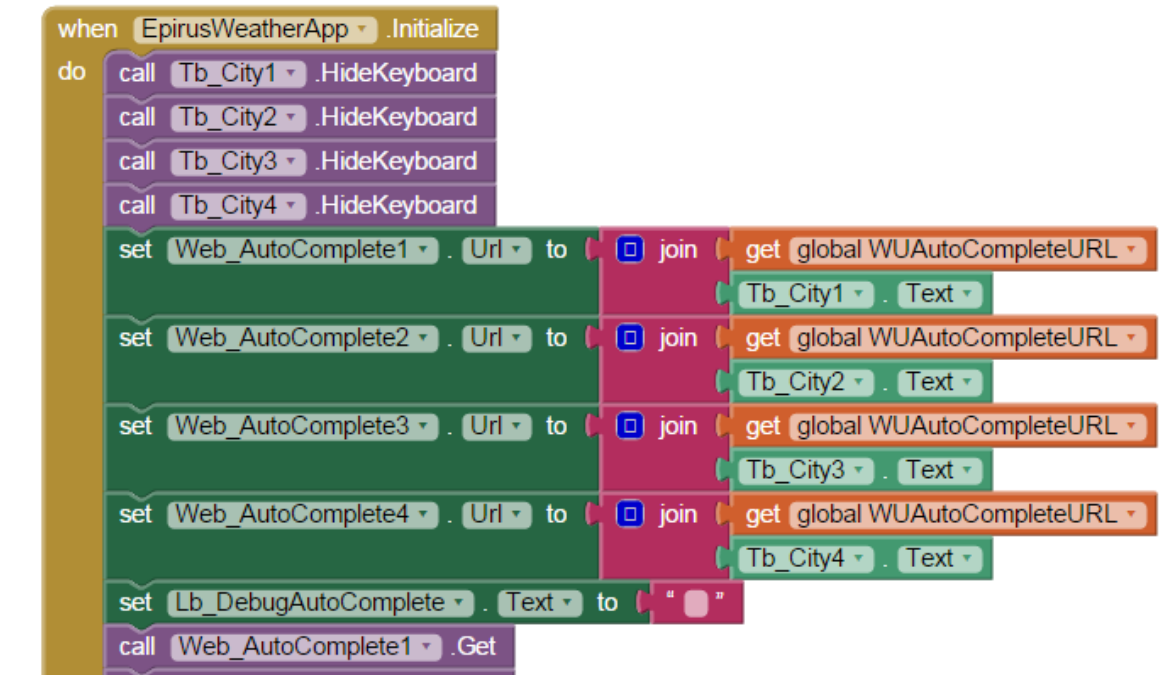
Η οθόνη αυτή είναι μία από τις δύο κύριες οθόνες της εφαρμογής. Εμφανίζει τον καιρό που υπάρχει τώρα και στις τέσσερις πόλεις[18][19][20]. Παρακάτω δείχνεται πως υλοποιήθηκε η μία πόλη, αφού για τις υπόλοιπες τρεις ακολουθήθηκε ακριβώς η ίδια διαδικασία. Ο κώδικας εδώ είναι πολύπλοκος και έχει αρκετές ιδιότητες.

```
initialize global WUAutoCompleteURL to " http://autocomplete.wunderground.com/aq?c=GR&format=xml&query="
initialize global WUApiURL to " http://api.wunderground.com/api/b40e3909c4c83bd9/conditions/lang:GR/pws:0 "
initialize global l_city to " not found "
```

Εικόνα 22 Μεταβλητές και api

Δημιουργήθηκαν τρεις μεταβλητές με σκοπό την επικοινωνία της εφαρμογής με την υπηρεσία wunderground και το api που παρέχει για προγραμματιστές. Αρχική προϋπόθεση είναι η εγγραφή στην σελίδα της wu για να μας παρέχει έναν κωδικό api (b40e39...) όπως φαίνεται παραπάνω (εικόνα 22). Το WUAutoCompleteUrl παρέχει αυτοματοποιημένα δεδομένα καιρού σε μορφή xml για εύρεση καιρού στην Ελλάδα(GR). Το WUApiURL μας δείχνει σε ποια ενότητα απευθυνόμαστε από τις επιλογές που έχει το wu και πως συντάσσεται η κάθε μία. Στο κομμάτι αυτό της εφαρμογής χρησιμοποιήσαμε το conditions που παρέχει τωρινά, live δεδομένα καιρού στα ελληνικά.

Για την κατανόηση αρκεί να επισκευθούμε την σελίδα <http://www.wunderground.com/weather/api/d/docs?d=data/conditions>.



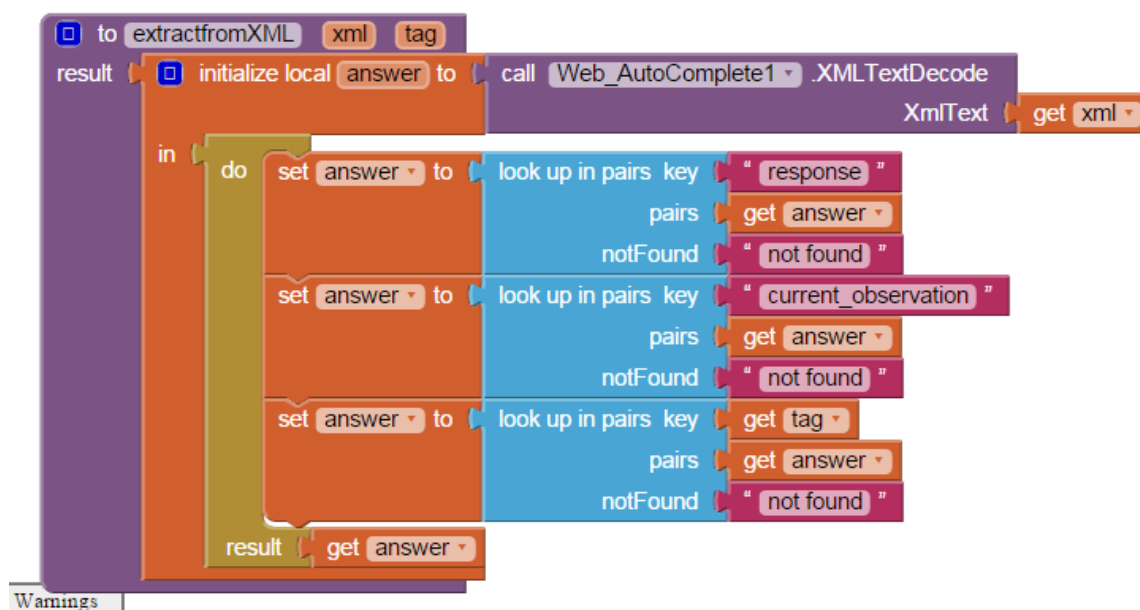
Εικόνα 23 EpirusWeatherApp initialize blocks

Για ευκολότερη πρόσβαση και άμεση ανταπόκριση χωρίς τη χρήση κάποιου κουμπιού καλούμε τη συνθήκη «when screenname.initialize» που μόλις ανοίγει η εκάστοτε «screen» αυτόματα εκτελεί τις εντολές που έχουν δοθεί. Καλεί τις πόλεις τις ενώνει με το URL και φέρνει τα αποτελέσματα σε xml για την κάθε πόλη ξεχωριστά.



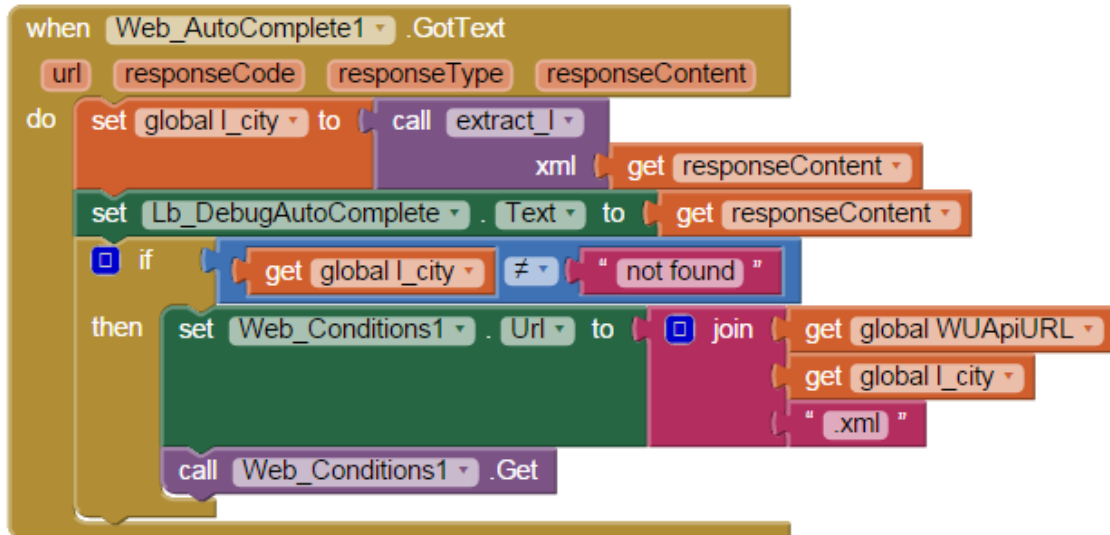
Εικόνα 24 XML κωδικοποίηση δεδομένων

Με το **URL Web\_AutoComplete** επιβεβαιώνει την πόλη που του δόθηκε και επιστρέφει ένα αρχείο xml με πληροφορίες όπως ο κωδικός «weather id», γεωγραφικές συντεταγμένες, πλήρη ονομασία, τοπική ώρα κλπ. Δεν χρειάζεται να εμφανίσουμε αυτές της πληροφορίες κάπου στην εφαρμογή μας αλλά χρειάζεται να “τρέξει” για να επικυρωθεί η σωστή πόλη και να αντλήσει πληροφορίες κατάστασης καιρού που επικρατούν εκείνη τη στιγμή για τη συγκεκριμένη πόλη (εικόνα 24).



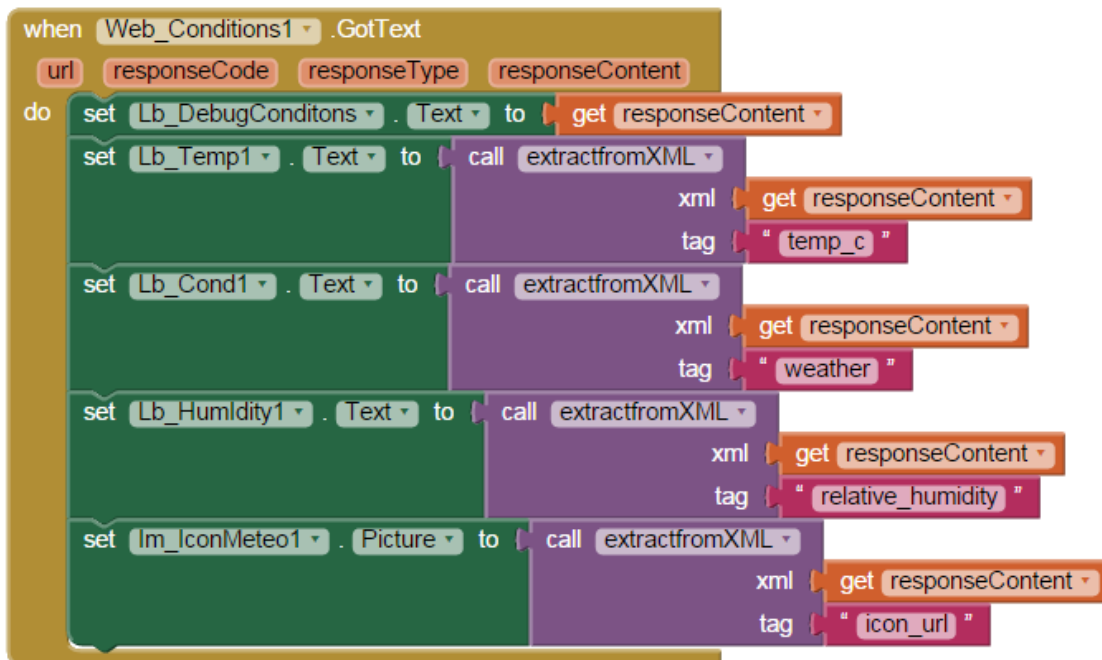
Εικόνα 25 XML response

Η παραπάνω διεργασία είναι αυτή που μας δίνει τη δυνατότητα να διαβάσουμε το xml με την τρέχουσα κατάσταση καιρού και να αντλούμε δεδομένα στη συνέχεια ζητώντας μόνο τα «tags» του xml. Το current observation είναι βασικό στοιχείο του xml και μέσα σε αυτό βρίσκονται τα δεδομένα που αντλούνται από την εφαρμογή



Εικόνα 26 Autocomplete και XML

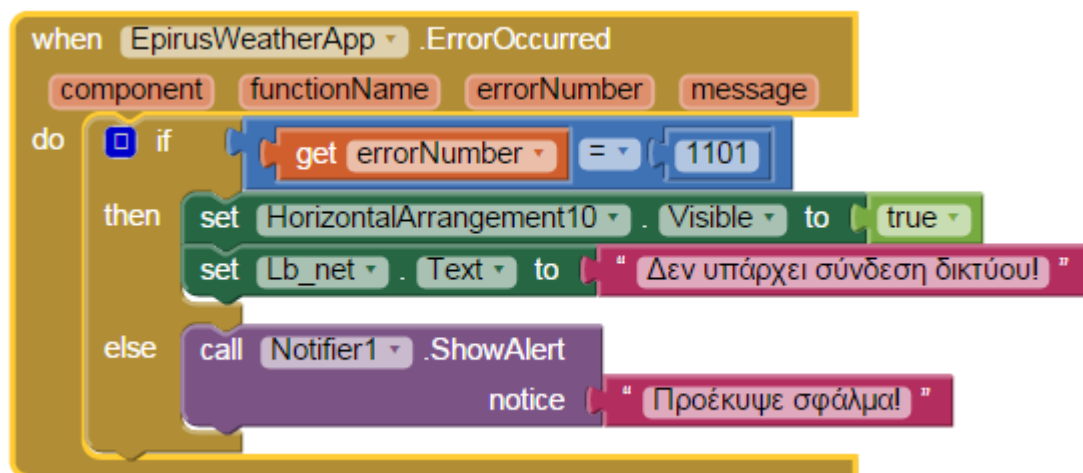
Αφού επιβεβαιωθεί με το Web AutoComplete η πόλη που ζητήσαμε ότι υπάρχει στη συνέχεια καλεί με το Web Conditions ένα συγκεκριμένο URL που μας παρέχει το Wunderground και τοποθετώντας την πόλη στη σωστή θέση στο URL επιστρέφει ένα αρχείο xml με την τρέχουσα κατάσταση του καιρού για την πόλη που έγινε η αναζήτηση.



Εικόνα 27 Εμφάνιση δεδομένων απο XML

Στην συνέχεια επιλέγουμε σε ποια πεδία και πού θα εμφανιστούν τα δεδομένα. Συγκεκριμένα από το xml αρχείο επιλέγουμε να εμφανιστούν τα πεδία **temp\_c** (θερμοκρασία), **weather**(κατάσταση), **relative\_humidity**(υγρασία) και **icon\_url**(εικονίδιο καιρού), όπως φαίνεται και στην εικόνα 27. Τα δεδομένα αυτά τα βρίσκουμε στην σελίδα της wunderground και μπορούμε να επιλέξουμε προς εμφάνιση όποια από αυτά θέλουμε που βρίσκονται στην ενότητα response fields του condition. Παρακάτω δίνεται η διεύθυνση της σελίδας.

<http://www.wunderground.com/weather/api/d/docs?d=data/conditions&MR=1>



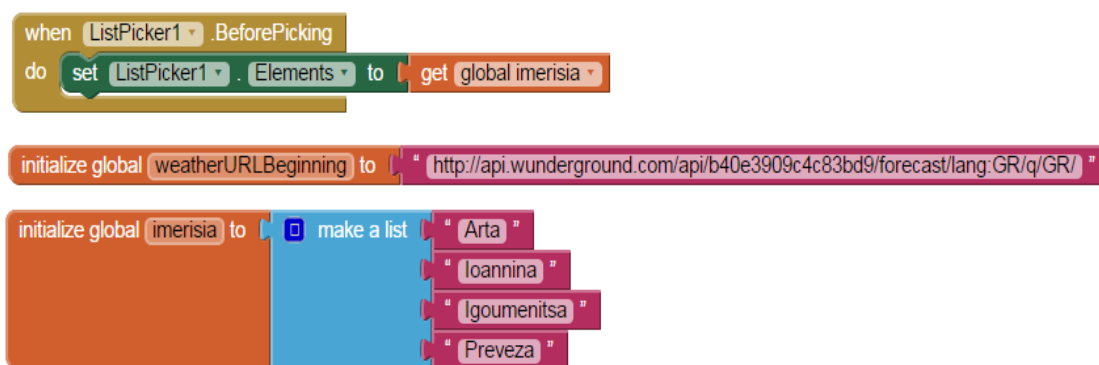
Εικόνα 28 Μήνυμα λάθους σύνδεσης δικτύου

Αν κατά της χρήση της εφαρμογής δεν υπάρχει σύνδεση σε δίκτυο το app inventor εμφανίζει μήνυμα σφάλματος «error code 1101». Το error αυτό μετετράπει σε μήνυμα που θα εμφανίζεται στην οθόνη και θα αναφέρει «Δεν υπάρχει σύνδεση δικτύου» όπως φαίνεται και παραπάνω (εικόνα 28).



- **Screen 3DaysForecast**

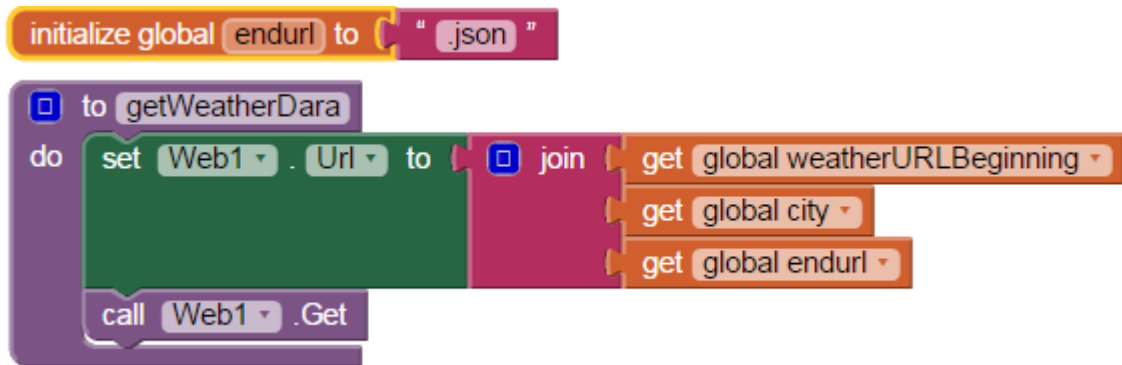
Για την εμφάνιση των δεδομένων χρησιμοποιήσαμε την ενότητα **forecast** από τις υπηρεσίες της wu. Τα δεδομένα που αντλούνται από την εφαρμογή που έχει δημιουργηθεί είναι σε μορφή JSON. Από το αρχείο JSON που παρέχει όλες τις πληροφορίες για τον καιρό επιλέξαμε τα στοιχεία που θέλουμε να εμφανίσουμε. Λειτουργεί σαν λίστα και κάθε φορά ζητάς ένα στοιχείο από την λίστα προς εμφάνιση. Για την διαχείριση των δεδομένων της κάθε πόλης δημιουργήσαμε μια λίστα (listpicker) για την κάθε πόλη (εικόνα 29).



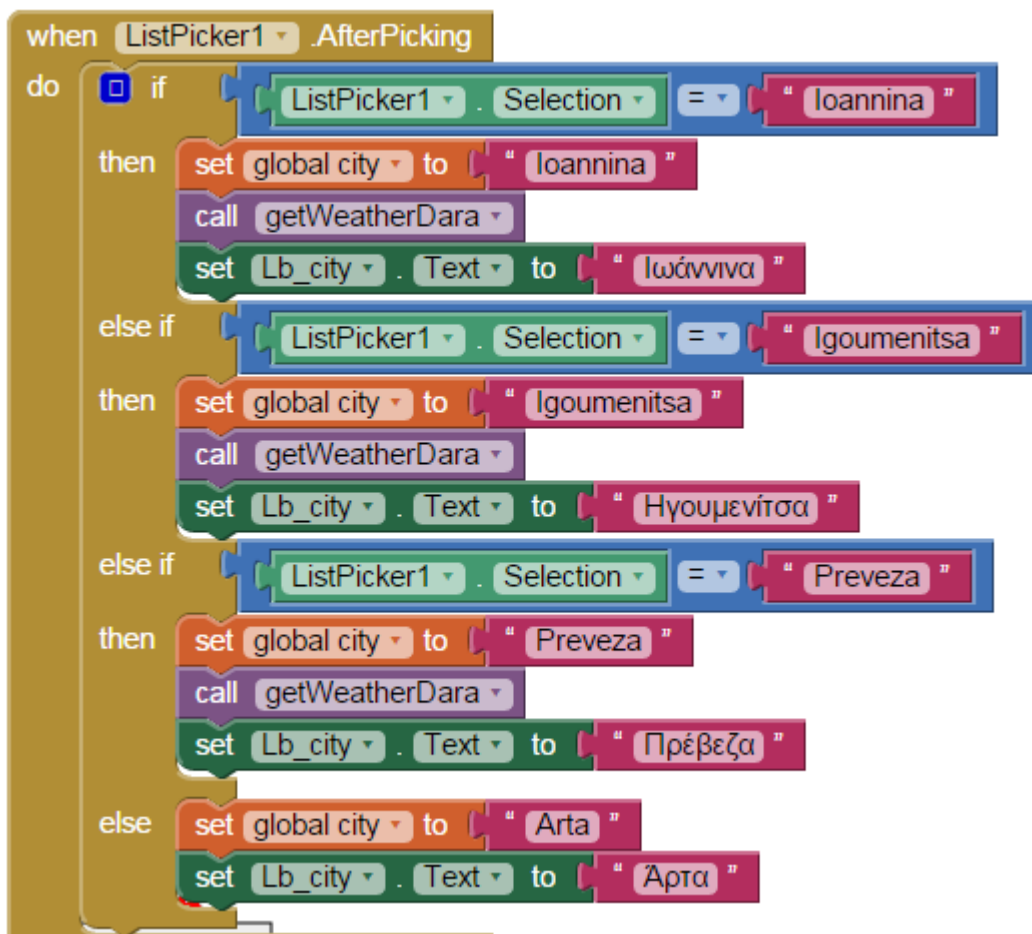
Εικόνα 29 Api forecast και λίστα πόλεων

Στη συνέχεια αφού δημιουργήσαμε την λίστα, ενώσαμε τα κατάλληλα πεδία ώστε εάν επιλεγθεί ένα στοιχείο της λίστας, δηλαδή μια πόλη, αυτόματα να εμφανιστούν οι καιρικές συνθήκες γι αυτήν την πόλη (εικόνα 29). Για την εμφάνιση των δεδομένων από το αρχείο JSON δημιουργήσαμε την διαδικασία **getWeatherData** (εικόνα 30), όπου παίρνει τα δεδομένα από το wu, και τα εμφανίζει σε κάθε πόλη.

Το **endurl** δηλώνει ότι όλο το link που αναφέρεται παραπάνω στην μεταβλητή weatherURLBeginning, θα ενσωματωθεί στην εφαρμογή με κατάληξη .json. Πλέον απομένει μόνο να πάρουμε τα δεδομένα που θέλουμε από το αρχείο και να τα εμφανίσουμε στα πεδία για την κάθε πόλη. Αυτό θα γίνει με την εντολή web1, που θα συνδέσει την εφαρμογή με το διαδίκτυο.

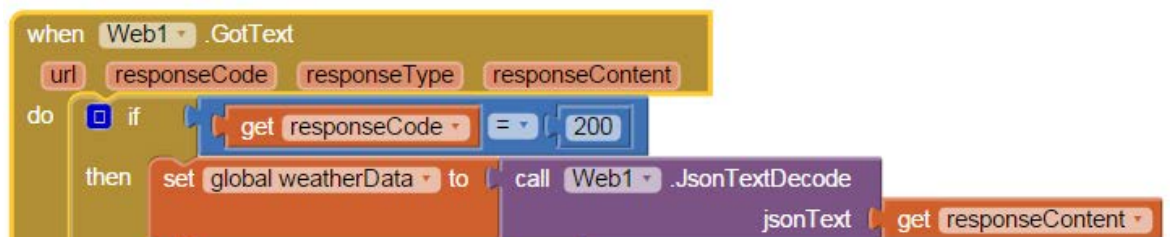


Εικόνα 30 Διαδικασία getWeatherData



Εικόνα 31 Επιλογή πόλης απο λίστα

Η εντολή **web1.getText** δηλώνει ότι όταν ζητήσουμε τα δεδομένα καιρού από το αρχείο JSON, θα κωδικοποιήσει το αρχείο με το **JsonTextDecode** και θα φέρει τα αποτελέσματα στην εφαρμογή. Το αρχείο αυτό παρέχει όλες τις σημερινές καιρικές συνθήκες για μια πόλη, αλλά και τριήμερη πρόγνωση. Για τις ανάγκες της εφαρμογής χρησιμοποιήσαμε μόνο τα στοιχεία για την θερμοκρασία, την κατάσταση, την υγρασία και το εικονίδιο καιρού (εικόνα 32).



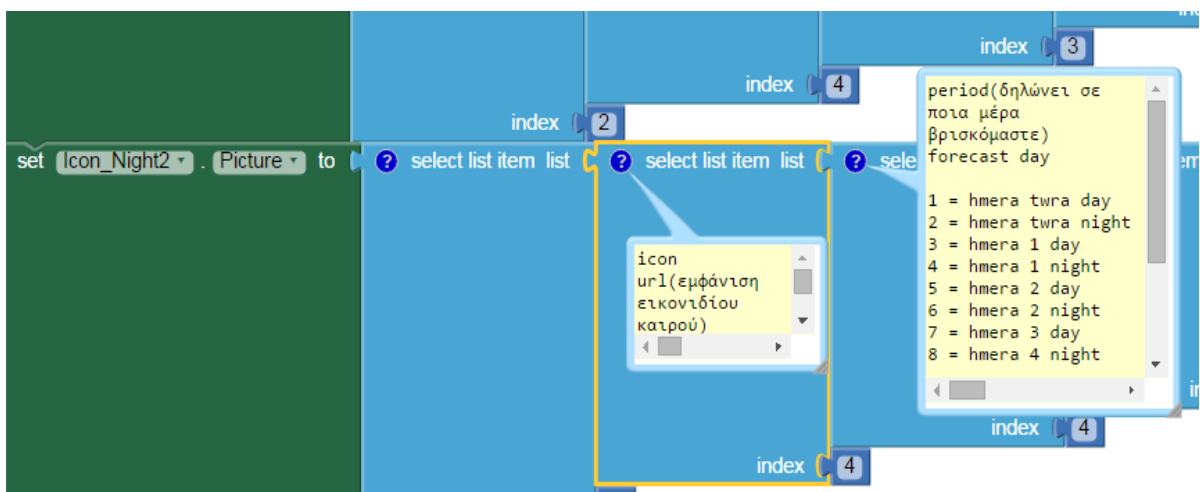
Εικόνα 32 Κωδικοποίηση Json και εμφάνιση δεδομένων

Με την ετικέτα **Lb\_Temp\_Day1.Text** εμφανίζουμε την μέγιστη θερμοκρασία για την σημερινή ημερομηνία και συγκεκριμένα για την ημέρα. Αντίστοιχα έχουμε **Lb\_Temp\_Nigth** για την εμφάνιση της ελάχιστης θερμοκρασίας την νύχτα. Με την εντολή **select list item list** επιλέγουμε από το αρχείο την θέση 5 της λίστας, στην συνέχεια την 2, και τέλος την 1. Πρόκειται ουσιαστικά για μία μεγάλη λίστα, με υπολίστες, όπου κάθε φορά πρέπει να αναζητήσεις στη λίστα, μία άλλη λίστα, μέχρι να φτάσεις στο στοιχείο θερμοκρασία της λίστας (εικόνα 33). Αντίστοιχα έχουμε δημιουργήσει την **Lb\_Temp\_Day2.Text** όπου εμφανίζει την μέγιστη θερμοκρασία για την αυριανή μέρα, μέχρι και **Lb\_Temp\_Day4.Text** όπου θα εμφανίσει την τελευταία μέρα του τριήμερου.

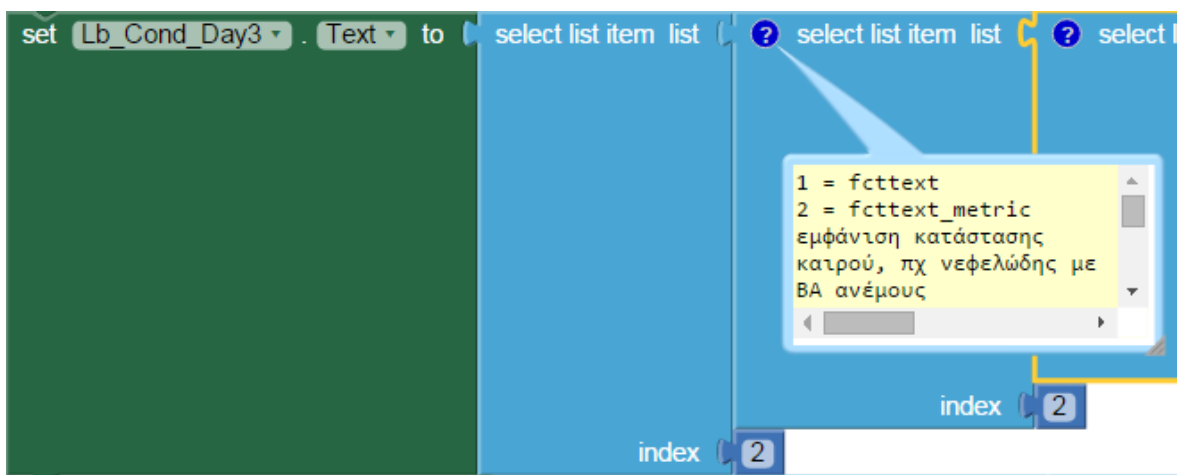
Επίσης έχουμε δημιουργήσει για την κατάσταση και το εικονίδιο καιρού αντίστοιχες επιλογές. Οι επιλογές **icon\_url**, **period**, **fcctext** βρίσκονται μέσα στο αρχείο-λίστα JSON και από κει επιλέγουμε και τα νούμερα στο index, ανάλογα με την θέση που βρίσκονται.



Εικόνα 33 Επιλογή στοιχείων απο το Json



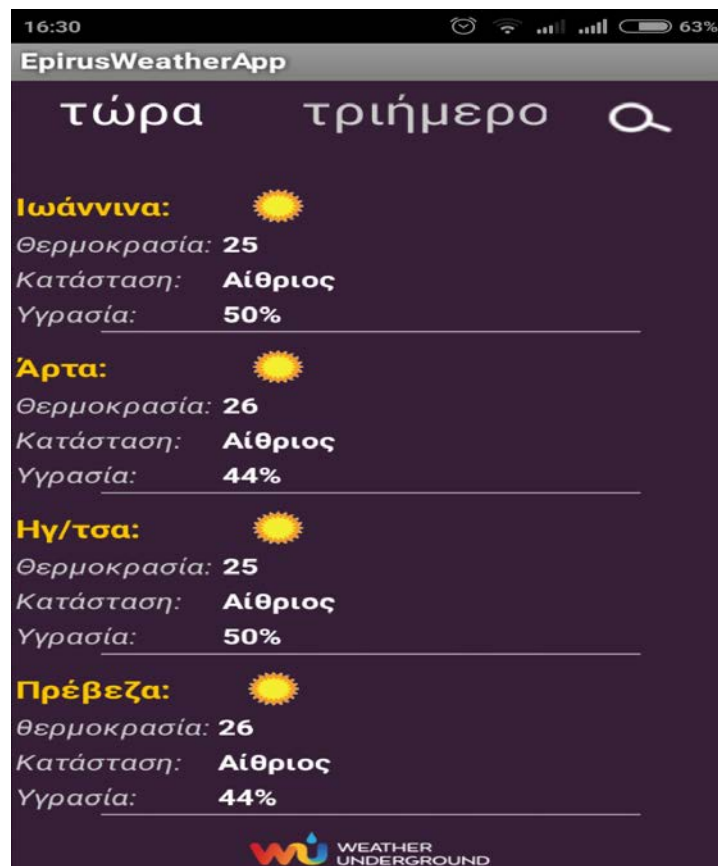
Εικόνα 34 Εικονίδιο καιρού για την νύχτα



Εικόνα 35 Εμφάνιση κατάστασης καιρού

## Κεφάλαιο 7. Εγχειρίδιο χρήσης

**Βήμα 1:** Ανοίγουμε την εφαρμογή EpirusWeatherApp. Αρχικά θα εμφανιστεί το λογότυπο της εφαρμογής και στην συνέχεια αυτόματα θα εμφανιστεί ο καιρός για τις τέσσερις πόλεις που έχουμε επιλέξει (εικόνα 36). Βρισκόμαστε στην επιλογή **τώρα**.



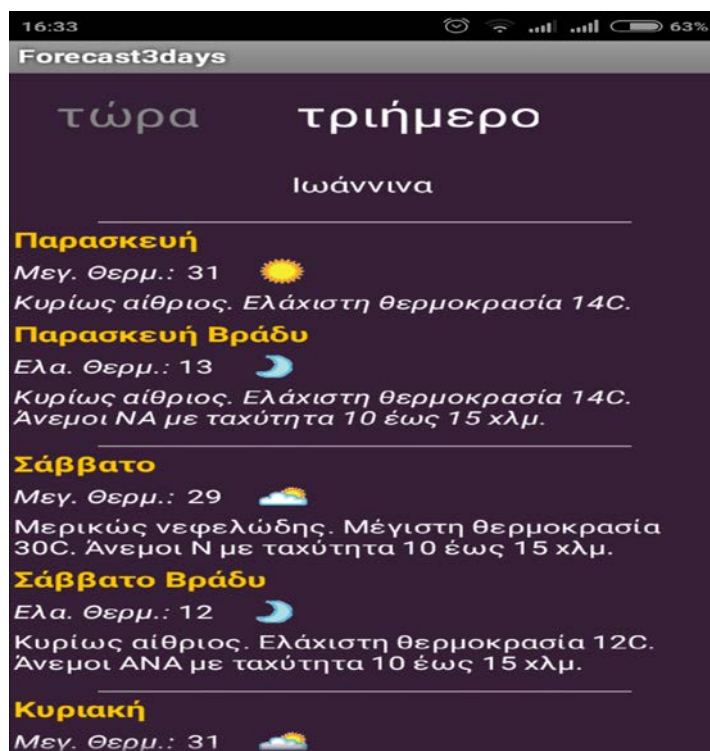
Εικόνα 36 Τελική εμφάνιση εφαρμογής

**Βήμα 2:** Επιλέγουμε το κουμπί **τριήμερο**. Θα μας εμφανίσει μενού με τις τέσσερις πόλεις.



Εικόνα 37 Μενού επιλογής πόλης

**Βήμα 3:** Εφόσον επιλέξαμε πόλη, αυτόματα θα εμφανιστεί ο καιρός για τις επόμενες τρεις ημέρες στην συγκεκριμένη πόλη.



Εικόνα 38 Τριήμερη πρόγνωση καιρού

**Βήμα 4:** Πατάμε την επιλογή **τώρα** και επιλέγουμε την αναζήτηση πόλης από το αντίστοιχο εικονίδιο. Θα μας μεταφέρει στην οθόνη ώστε να πληκτρολογήσουμε την πόλη που επιθυμούμε για να δούμε τον καιρό. Αφορά μόνο πόλεις την Ελλάδα. Αν η πόλη δεν υπάρχει θα εμφανιστεί αντίστοιχο μήνυμα λάθους που θα προτρέπει να κλείσουμε την εφαρμογή (εικόνα 39).



Εικόνα 39 Αναζήτηση πόλης

## Κεφάλαιο 9<sup>ο</sup> Συμπεράσματα

Στην παρούσα πτυχιακή εργασία παρουσιάζεται η υλοποίηση μιας Android εφαρμογής για την πρόγνωση καιρού στην Ήπειρο, με την ονομασία EpirusWeatherApp. Η εφαρμογή επιπλέον έχει την δυνατότητα πρόγνωσης για όλη την Ελλάδα με αντίστοιχο πεδίο αναζήτησης. Το κόστος λειτουργίας είναι μηδενικό και δεν απαιτεί εξειδικευμένες γνώσεις πληροφορικής για τη σωστή χρήση. Τέλος κατά την ολοκλήρωση της εργασίας, θα χαρακτηρίζαμε ως μελλοντικά σχέδια για εξέλιξη της εφαρμογής τα παρακάτω.

1. Ενσωμάτωση GPS και εύρεση τοποθεσίας για εμφάνιση του καιρού
2. Αποθήκευση πόλης για γρήγορη και εύκολη πρόγνωση καιρού
3. Εμφάνιση καιρού για ολόκληρο τον κόσμο



## Βιβλιογραφία

1. <http://datalabs.edu.gr:81/Forum/default.aspx?g=posts&t=409>
2. Διπλωματική Εργασία: «Υλοποίηση εφαρμογής κινητού τηλεφώνου με χρήση της πλατφόρμας Google Android», του Ευθύμιου Δούκα φοιτητή του Τμήματος Τεχνολογικών Εφαρμογών της Σχολής Τεχνολογίας Πληροφορικής και Τηλεπικοινωνιών του ΑΤΕΙ Λάρισας.
3. Διπλωματική Εργασία: «Κινητή εφαρμογή με επίγνωση θέσης για εύρεση σημείων ενδιαφέροντος», του Χρήστου Δάνα φοιτητή του Τμήματος Επιστήμης και Τεχνολογίας Υπολογιστών ΑΕΙ Πελοποννήσου.
4. <http://socialcompare.com/en/comparison/android-versionscomparison>
5. Διπλωματική Εργασία: «Mobile Interfaces and Interaction Technologies», του Παλάσκα Αλέξανδρου φοιτητή του Τμήματος Πληροφορικής Υπολογιστών, Αριστοτέλειο Θεσσαλονίκης
6. <http://www.android.com/history/>
7. Διπλωματική Εργασία: «Ανάλυση και χρήση εργαλείων για προγραμματισμό εφαρμογών σε Android» του Δασκαλάκη Ελευθέριου, φοιτητή του Τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων της Σχολής Τεχνολογικών Εφαρμογών του ΑΤΕΙ Κρήτης.
8. <http://developer.android.com/sdk/index.html>
9. Διπλωματική Εργασία: «Ανάπτυξη Android εφαρμογής με το MIT Inventor» του Ζαχαρία Κατσαράκη, φοιτητή του Τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων της Σχολής Τεχνολογικών Εφαρμογών του ΑΤΕΙ Κρήτης.
10. <http://www.appinventor.mit.edu>
11. <http://wunderground.com/>
12. Parallel Distributed Processing Laboratory (PDP Lab), Department of Applied Informatics, University of Macedonia Thessaloniki, Greece

13. Πανεπιστημιακές σημειώσεις «Η γλώσσα σήμανσης xml» Πανεπιστήμιο Πειραιώς. Διαθέσιμο σε [http://www.fme.aegean.gr/sites/default/files/dsampsom\\_xml\\_lectures-notes-dec2003.pdf](http://www.fme.aegean.gr/sites/default/files/dsampsom_xml_lectures-notes-dec2003.pdf)
14. Πανεπιστημιακές σημειώσεις «Εισαγωγή στην Java» Τμήμα Πληροφορικής και Τηλεπικοινωνιών, Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών.
15. <http://json.org/>
16. Προγραμματισμός σε App Inventor». Βασίλης Βασιλάκης, Γιώργος Χατζηνικολάκης.. Σύλλογος Εκπαιδευτικών Πληροφορικής Χίου, 2014
17. <http://www.coderdojotrabane.com/wp-content/uploads/2014/05/App-Inventor-Tutorial-10.pdf>
18. <http://www.appinventorblocks.com/appinventor-tutorials-tips/appinventor-lists>
19. <http://puravidaapps.com/tutorials.php>
20. <http://schools.tdsb.on.ca/westhill/business/ICS4U/tinydb/TinyDBandListPicker.htm>