



**ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ**

**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ**

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**Ανάπτυξη Συστήματος επιτήρησης χώρου με  
χρήση αισθητήρων, ασυρμάτων καμερών και  
φορητών προσωπικών συσκευών**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Γεώργιος Λαδής**

**ΑΜ: 021997**

**Επιβλέπων:**

**Δρ. Χατζημίσιος Περικλής**

**Επίκουρος Καθηγητής Τμήματος Πληροφορικής Α.Τ.Ε.Ι.Θ.**

**Θεσσαλονίκη, Σεπτέμβριος 2011**

## Πρόλογος

Η παρούσα πτυχιακή εργασία δημιουργήθηκε στα πλαίσια του προπτυχιακού προγράμματος σπουδών του Τμήματος Πληροφορικής του Α.Τ.Ε.Ι. Θεσσαλονίκης. Η πτυχιακή εργασία έχει τίτλο *«Ανάπτυξη Συστήματος επιτήρησης χώρου με χρήση αισθητήρων, ασυρμάτων καμερών και φορητών προσωπικών συσκευών»* και ο επιβλέπων καθηγητής ήταν ο Δρ. Περικλής Χατζημίσιος, Επίκουρος Καθηγητής του Τμήματος Πληροφορικής του Α.Τ.Ε.Ι.Θ. και εκπονήθηκε σύμφωνα με τον κανονισμό πτυχιακών εργασιών του Τμήματος.

## Περίληψη

Ο κύριος σκοπός της παρούσης πτυχιακής εργασίας είναι η αξιολόγηση της υπάρχουσας τεχνολογίας έξυπνων τηλεφώνων και των εμπορικών προοπτικών τους και στην συνέχεια η ανάπτυξη εφαρμογής που θα μπορεί ανταποκρίνεται σε πραγματικές ανάγκες και απαιτήσεις μίας πραγματικής επιχείρησης, της Ελληνικής Αεροπορικής Βιομηχανίας (Ε.Α.Β.).

Πιο συγκεκριμένα, η παρούσα πτυχιακή ασχολείται με τον προγραμματισμό έξυπνων κινητών συσκευών (κυρίως smartphones και tablets) που βασίζονται στο λειτουργικό σύστημα Android. Επίσης, μετά από κατάλληλη μελέτη όλων των παραμέτρων που πρέπει να ληφθούν υπόψη, έχει αναπτυχθεί εφαρμογή που καλύπτει τις ανάγκες ασφάλειας μίας «κρίσιμης» εγκατάστασης και συμπληρώνει τις υπάρχουσες υποδομές αυτής.

Το πρώτο μέρος της πτυχιακής περιέχει μία λεπτομερή ανάλυση των έξυπνων τηλεφώνων και ειδικότερα των λειτουργικών συστημάτων και της εξέλιξης τους. Επίσης, γίνεται λεπτομερή ανάλυση της τεχνολογίας, των στοιχείων αγοράς που έχουν διαμορφωθεί και οι προβλέψεις διαφόρων ειδών αξιολόγησης τεχνολογίας. Στο δεύτερο μέρος της πτυχιακής παρουσιάζεται η εφαρμογή που έχει επιλεγεί και ειδικότερα το περιβάλλον ανάπτυξης eclipse και svn που χρησιμοποιήθηκε, η ανάλυση των προδιαγραφών και απαιτήσεων που έθεσε η Ε.Α.Β., η σχεδίαση της εφαρμογής, η υλοποίηση, η χρήση καθώς και οι δοκιμές που έγιναν και τα συμπεράσματα που εξήχθησαν.

Η εφαρμογή που έχει αναπτυχθεί καλύπτει τα ακόλουθα πεδία:

1. Γεωγραφικό προσδιορισμό της θέσης της συσκευής και αποτύπωση γειτονικών αισθητήρων. Οι αισθητήρες που χρησιμοποιήθηκαν είναι κάμερες και μέσω του διαδραστικού περιβάλλοντος της εφαρμογής, ο

χρήστης μπορεί να επιλέγει οποιαδήποτε από αυτές τις κάμερες και να χειρίζεται την έξοδό τους.

2. Σύμβαση με κεντρικό εξυπηρετητή όπου ο χρήστης μπορεί να καταχωρεί ή να ανακαλεί διάφορα στοιχεία, μέσω και φορμών που θα αναπτυχθούν, π.χ. σχετικά με αντικείμενα που παρατηρήθηκαν μέσω κάποιου αισθητήρα.

Επιπλέον, υποστηρίζονται ασύρματες συνδέσεις ώστε να διευκολύνεται η κινητικότητα του χρήστη και η κάλυψη μιας πολύ μεγάλης περιοχής. Η εφαρμογή καλύπτει θέματα διεπαφής με το περιβάλλον καθώς και της διασύνδεσης με το υπόλοιπο δίκτυο, επιδεικνύοντας τη χρησιμότητα τέτοιας γεφύρωσης. Πολύ σημαντικό είναι ότι προσφέρει ένα αποτελεσματικό αλλά και εύχρηστο διαδραστικό περιβάλλον. Η εφαρμογή έχει πολλές δυνατότητες αξιοποίησης, όπως από φύλακες εγκαταστάσεων οι οποίοι ενώ περιπολούν θα μπορούν να έχουν άμεση πρόσβαση μέσω της εφαρμογής (και των καμερών ασφαλείας) σε οποιοδήποτε σημείο της περιοχής επιτήρησης. Επίσης, η εφαρμογή παρέχει την δυνατότητα ελέγχου και καταγραφής συμβάντων (event recording), όπως για παράδειγμα την είσοδο στις εγκαταστάσεις ενός υπαλλήλου ή άλλου προσώπου.

Η εφαρμογή επικεντρώθηκε σε σύστημα ασφαλείας όπου οι φύλακες περιπολούν μεταφέροντας μία ειδική συσκευή. Η εφαρμογή αναπτύχθηκε στο λειτουργικό σύστημα Android και λαμβάνεται βίντεο σε πραγματικό χρόνο από κάμερες εγκατεστημένες στη περιοχή επιτήρησης. Στην περίπτωση που οι κάμερες είναι τύπου Pan-Tilt-Zoom (PTZ) η εφαρμογή παρέχει και δυνατότητες τηλεχειρισμού. Η επιλογή της κάμερας γίνεται μέσα από διαδραστικό περιβάλλον όπου απεικονίζονται σε χάρτη οι θέσεις των καμερών και η θέση του χρήστη (σε αυτή την περίπτωση, απαραίτητη προϋπόθεση είναι η κινητή συσκευή να διαθέτει GPS).

Επιπλέον, η εφαρμογή επιτρέπει την αναφορά συμβάντος με συμπλήρωση έτοιμων φορμών από τον χρήστη και η αποστολή της πληροφορίας μπορεί να γίνει είτε με e-mail είτε με SMS.

Για την πειραματική ανάπτυξη, η επικοινωνία με τις γύρω συσκευές γίνεται μέσω δικτύου Wi-Fi και χρησιμοποιήθηκαν συσκευές διαφορετικού τύπου (Google Nexus με οθόνη 3,7 ιντσών, Dell Strike με οθόνη 5 ιντσών και Samsung Galaxy

Tab με οθόνη 7 ιντσών) στις οποίες προσαρμόστηκε η εφαρμογή μετά από κατάλληλη προετοιμασία. Επιπλέον, χρησιμοποιήθηκαν κάμερες διαφορετικού τύπου (σταθερή κάμερα μάρκας Axis, PTZ κάμερα PT7137 Vivotek, PTZ κάμερα PZ7152 Vivotek) και η εφαρμογή συνδέθηκε με όλες αυτές. Το σύστημα δοκιμάστηκε επιτυχώς και για το σενάριο της ταυτόχρονης χρήσης όλων των καμερών από όλες τις συσκευές. Επίσης η εφαρμογή διασυνδέθηκε και δοκιμάστηκε επιτυχώς και με συστήματα διοίκησης και ελέγχου καθώς και με ασύρματα και δορυφορικά δίκτυα της Ελληνικής Αεροπορικής Βιομηχανίας.

## Abstract

The main scope of the current thesis is to assess the existing technology of Smartphones and their commercial prospective as well as to develop an application based on real requirements of a demanding business environment such as the Hellenic Aerospace Industry (HAI).

The current thesis deals with the programming of smart mobile devices (smartphones and tablets) based on an Android OS. Following a complete study of all the parameters that had to be taken into account, an application was developed covering and complementing the security requirements of a critical infrastructure.

The first part of the thesis covers a detailed analysis of Smartphones, their Operating Systems and their development. An analysis is presented about the technologies, the market data as well as the future predictions as provided by international assessment firms.

The second part of the thesis covers the development environments SDK, eclipse and svn, which were used, the specifications and the requirement analysis as an input from HAI, the design and the development of the application, its usage and testing and finally the conclusions and suggestions for future development.

The application that has developed covers the following areas:

1. Geographical positioning of devices and display of neighboring sensors. The sensors used are cameras and through an interactive application environment, the user can select any of them and control its functions.
2. Connecting to a central server where the user can enter or withdraw various items through forms that were developed, eg on events observed through a sensor.

Wireless connections are supported in order to facilitate the movement of the

user. The application covers interfaces with the environment and connections to a wider network, demonstrating the usefulness of such a bridge offering in the same time a friendly interactive environment. The application could be used, e.g. by plant security guards who can observe the output from security cameras, any time and anywhere while patrolling, and will also be able to record events, e.g. entering of another employee in the premises.

In particular, the implementation is focused on security systems, where guards who patrol are carrying a special device. The application is developed in the Android operating system and receives real-time video from cameras located in the surrounding area and may be controllable (in case the cameras have PTZ-Pan-Tilt-Zoom functionality). The selection of the camera is done through an interactive environment where both the positions of cameras and the user location is displayed (the latter is possible, if the mobile device has a GPS). Moreover, the application allows reporting the incidents by completing available forms. The information can be sent by email or SMS.

For the experimental development, devices are communicating over a WiFi network. Different types of devices were used (Google Nexus 3.7-inch screen, Dell Strike screen 5-inch Samsung Galaxy Tab with 7 inch screen) and the implementation is adapted to all. Different types of cameras were used (fixed camera Axis, PTZ Camera PT7137 Vivotek, PTZ Camera PZ7152 Vivotek) and the application is connected to all of them. The system was tested successfully with simultaneous use of all cameras from all devices, while still interconnected and tested successfully with command and control systems and wireless and satellite networks of the Hellenic Aerospace Industry.

## Ευχαριστίες

Για την εκπόνηση αυτής της πτυχιακής εργασίας συνεργάστηκα με τον επιβλέποντα καθηγητή μου Δρ. Περικλή Χατζημίσιο, με τον Δρ. Θεόδωρο Ζαχαριάδη, Επίκουρο Καθηγητή του ΤΕΙ Χαλκίδας καθώς και με μηχανικούς και στελέχη της Ελληνικής Αεροπορικής Βιομηχανίας οι οποίοι μου προσέφεραν αμέριστη συμπαράσταση και βοήθεια στην εκτέλεση του έργου μου διαθέτοντας τα απαραίτητα μέσα για να πετύχω ένα ικανοποιητικό αποτέλεσμα.

Ιδιαίτερα θα ήθελα να ευχαριστήσω:

- Τον επιβλέποντα μου Δρ. Περικλή Χατζημίσιο για την καθοδήγησή του στην διάρκεια εκπόνησης της εργασίας και την ευκαιρία που μου έδωσε να ασχοληθώ με την πτυχιακή αυτή εργασία κατά την εκπόνηση της πρακτικής μου άσκησης στην Ε.Α.Β.
- Τον Δρ. Θεόδωρο Ζαχαριάδη για την επίβλεψη της εργασίας μου.
- Όλο το προσωπικό της Διεύθυνσης Ηλεκτρονικών Συστημάτων και Λογισμικού της Ε.Α.Β. για την συμπαράστασή τους και ιδιαίτερα τον συνεργάτη μου μηχανικό κ. Αγγελή Αγγελή και τον προϊστάμενο του Τομέα Δορυφορικών Εφαρμογών κ. Αθανάσιο Πουλακίδα για την καθοδήγησή τους κατά την εκπόνηση της πτυχιακής μου εργασίας .



# Πίνακας περιεχομένων

Πρόλογος .....	2
Περίληψη .....	3
Abstract .....	6
Ευχαριστίες .....	8
Πίνακας περιεχομένων .....	9
Λίστα εικόνων .....	11
Κεφάλαιο 1:Εισαγωγή.....	12
1.1 Εισαγωγή.....	12
1.2 Σκοπός πτυχιακής εργασίας .....	12
1.3 Δομή πτυχιακής εργασίας .....	13
Κεφάλαιο 2 :Γενικά στοιχεία έξυπνων τηλεφώνων και λειτουργικών συστημάτων. 15	
2.1 Εισαγωγή.....	15
2.2 Εισαγωγή στην τεχνολογία .....	15
2.3 Ανάλυση λειτουργικών συστημάτων για έξυπνα τηλέφωνα .....	18
2.4 Πως χρησιμοποιούνται τα έξυπνα τηλέφωνα.....	23
2.5 Εφαρμογές για πολλές πλατφόρμες.....	25
2.6 Smartphones – έξυπνα τηλέφωνα .....	26
2.7 Στατιστικά στοιχεία αγοράς .....	29
2.8 Λειτουργικό Σύστημα ANDROID .....	33
Κεφάλαιο 3 :Περιβάλλον Ανάπτυξης.....	38
3.1 Εισαγωγή.....	38
3.2 Περιβάλλον ανάπτυξης Eclipse.....	38
3.3 Svn – Subversion .....	46
3.4 Οδηγίες εγκατάστασης και χρήσης των εργαλείων ανάπτυξης sdk eclipse svn 49	

---

<b>Κεφάλαιο 4 :Η εφαρμογή .....</b>	<b>56</b>
<b>4.1 Εισαγωγή.....</b>	<b>56</b>
<b>4.2 Προδιαγραφές.....</b>	<b>56</b>
<b>4.2.1. Περίπτωση Χρήσης ΠΧ: Επιτήρηση χώρου εταιρίας. ....</b>	<b>56</b>
<b>4.2.2. Επιλογή ANDROID έναντι άλλων λειτουργικών συστημάτων .....</b>	<b>58</b>
<b>4.3 Σχεδίαση.....</b>	<b>59</b>
<b>4.3.1 Γενική δομή .....</b>	<b>59</b>
<b>4.3.2 Αρχιτεκτονική υλικού .....</b>	<b>60</b>
<b>4.3.3 Βασικές κλάσεις.....</b>	<b>61</b>
<b>4.4 Υλοποίηση.....</b>	<b>62</b>
<b>4.5 Περιγραφή λειτουργίας της εφαρμογής .....</b>	<b>71</b>
<b>4.5.1 Οδηγίες εγκατάστασης εφαρμογής .....</b>	<b>71</b>
<b>4.5.2. Χρήση εφαρμογής .....</b>	<b>72</b>
<b>4.6 Δοκιμές. ....</b>	<b>76</b>
<b>Κεφάλαιο 5 :Συμπεράσματα-Νέες Επεκτάσεις .....</b>	<b>80</b>
<b>5.1 Συμπεράσματα.....</b>	<b>80</b>
<b>5.2 Επεκτάσεις .....</b>	<b>80</b>
<b>Βιβλιογραφία.....</b>	<b>82</b>
<b>Παραπομπές.....</b>	<b>82</b>
<b>Γενική βιβλιογραφία.....</b>	<b>83</b>
<b>Παράρτημα (κώδικας) .....</b>	<b>84</b>

## Λίστα εικόνων

Εικόνα 1 - Οι 11 πρώτες (“hot”) τεχνολογίες της δεκαετίας.....	16
Εικόνα 2 - Περιβάλλον ανάπτυξης εφαρμογών σε Smartphone .....	21
Εικόνα 3 - Χαρακτηριστικά λειτουργικών συστημάτων .....	22
Εικόνα 4 - Κατηγορίες εφαρμογών Smartphone .....	24
Εικόνα 5 - Δημοφιλείς εφαρμογές .....	24
Εικόνα 6 - Εξέλιξη έξυπνων τηλεφώνων.....	28
Εικόνα 7 - Στατιστικά στοιχεία πωλήσεων λειτουργικών συστημάτων.....	30
Εικόνα 8 - Στατιστικά στοιχεία πωλήσεων Smartphones .....	31
Εικόνα 9 - Πωλήσεις Smartphone versus PC .....	31
Εικόνα 10 - Πωλήσεις Android versus iPhone.....	32
Εικόνα 11 - Διαφορετικές εκδόσεις Android .....	35
Εικόνα 12 - Αρχιτεκτονική Android.....	37
Εικόνα 13 - Αρχιτεκτονική subversion (svn) .....	47
Εικόνα 14 - Το πρόβλημα των συνεργατικών αναπτύξεων .....	49
Εικόνα 15 - Βασική αρχιτεκτονική δομή εφαρμογής .....	59
Εικόνα 16 - Αρχιτεκτονική υλικού εφαρμογής .....	60
Εικόνα 17 - Διάγραμμα κλάσεων.....	61
Εικόνα 18 - Διάταξη δοκιμών .....	77
Εικόνα 19 - Δοκιμές σε σύστημα επιτήρησης συνόρων .....	78
Εικόνα 20 - Δοκιμές με δορυφορικό δίκτυο επικοινωνιών.....	79

# Κεφάλαιο 1: Εισαγωγή

## 1.1 Εισαγωγή

Η αυξανόμενη ισχύς και ευχρηστία των κινητών συσκευών κάθε είδους και η πορεία προς ένα διαδίκτυο των αντικειμένων (Internet of Things), έχουν δημιουργήσει μία πολλά υποσχόμενη αγορά λογισμικού για κινητές συσκευές (smartphones, tablets κλπ.). Πέρα από την αυξανόμενη ανάληψη εργασιών που συνηθίζονται σε περιβάλλον desktop, συχνά οι κινητές συσκευές αλληλεπιδρούν έντονα με το περιβάλλον τους.

Οι συσκευές αυτές χρησιμοποιούνται σήμερα για να καλύπτουν τις ανάγκες επικοινωνιών των κατόχων τους, καθώς και για πρόσβαση στο διαδίκτυο και σε δίκτυα κοινωνικής δικτύωσης.

Η ευρεία διάδοση και η μελλοντική θέση των κινητών συσκευών στην αγορά δημιουργεί την ευκαιρία για χρήση τους σε επαγγελματικές εφαρμογές, όπου εκτιμάται ότι στο μέλλον θα αποτελέσουν την κύρια υπολογιστική μονάδα του κάθε χρήστη, η οποία θα του εξασφαλίζει τις επικοινωνίες του αλλά και πρόσβαση στον επαγγελματικό χώρο του.

Η Ελληνική Αεροπορική Βιομηχανία αναπτύσσει μεταξύ άλλων συστήματα Ασφάλειας και αξιόπιστα δίκτυα επικοινωνιών. Η εξέλιξη της τεχνολογίας αλλά και οι δυνατότητες και προοπτικές που υπόσχονται τα έξυπνα τηλέφωνα, ώθησαν την Ε.Α.Β. να στηρίξει την ανάπτυξη μίας εφαρμογής που μελλοντικά θα μπορεί να την αξιοποιήσει σε συστήματα που προωθεί στην αγορά.

Συνοψίζοντας, στα πλαίσια που αναφέρθηκαν προηγουμένως ανατέθηκε η συγκεκριμένη πτυχιακή εργασία.

## 1.2 Σκοπός πτυχιακής εργασίας

Η πτυχιακή εργασία έχει αρχικά σκοπό να αναλύσει την υπάρχουσα τεχνολογία των έξυπνων φορητών συσκευών και των λειτουργικών συστημάτων τους και στην

συνέχεια να εφαρμόσει την τεχνολογία αυτή με την ανάπτυξη μίας εφαρμογής που θα καλύπτει τις ανάγκες ασφάλειας μίας «κρίσιμης» εγκατάστασης. Για τον σκοπό αυτό, οι χρήστες του συστήματος (βάσει των αναγκών τους) προσδιόρισαν τις απαιτήσεις και λειτουργικές προδιαγραφές της εφαρμογής. Τέλος, η πτυχιακή εργασία περιλαμβάνει επίσης την ανάλυση των απαιτήσεων, την σχεδίαση καθώς και η υλοποίηση της εφαρμογής.

## 1.3 Δομή πτυχιακής εργασίας

Η δομή της πτυχιακής εργασίας αναλύεται ως εξής:

### Κεφάλαιο 1: Εισαγωγή

- Παρουσιάζεται η περίληψη, ο σκοπός και η δομή της πτυχιακής εργασίας.

### Κεφάλαιο 2: Γενικά στοιχεία έξυπνων τηλεφώνων και λειτουργικών συστημάτων.

- Παρουσιάζεται η ανάλυση της τεχνολογίας των έξυπνων τηλεφώνων και των λειτουργικών τους συστημάτων με ιδιαίτερη έμφαση στο λειτουργικό σύστημα Android. Επίσης, παρουσιάζονται στατιστικά στοιχεία αγοράς καθώς και προβλέψεις για μελλοντικές πωλήσεις.

### Κεφάλαιο 3: Περιβάλλον ανάπτυξης

- Παρουσιάζεται το περιβάλλον ανάπτυξης Eclipse καθώς και το εργαλείο Subversion που χρησιμοποιήθηκε. Επιπλέον, αναλύονται οι δυνατότητες η δομή και η λειτουργία τους. Τέλος, δίνονται οδηγίες για την εγκατάσταση και χρήση τους.

### Κεφάλαιο 4: Η εφαρμογή

- Αναλύονται οι προδιαγραφές της εφαρμογής, παρουσιάζεται η σχεδίαση και η υλοποίηση καθώς και οι δοκιμές που έγιναν. Επίσης, δίνονται οδηγίες για εγκατάσταση και χρήση της εφαρμογής.

## **Κεφάλαιο 5:** Συμπεράσματα – Νέες επεκτάσεις

- Παρουσιάζονται τα συμπεράσματα και προτείνονται μελλοντικές επεκτάσεις και βελτιώσεις της εφαρμογής

Τέλος, παρουσιάζεται η βιβλιογραφία και οι πηγές που χρησιμοποιήθηκαν καθώς και ο κώδικας της εφαρμογής (στο παράρτημα).

# Κεφάλαιο 2 : Γενικά στοιχεία

## έξυπνων τηλεφώνων και λειτουργικών

### συστημάτων

## 2.1 Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζεται η τεχνολογία των έξυπνων φορητών συσκευών και η εξέλιξή τους. Αναλύονται τα λειτουργικά συστήματα που χρησιμοποιούνται σε αυτές τις συσκευές και οι σημερινές τους χρήσεις. Επίσης, παρουσιάζονται στατιστικά στοιχεία αγοράς και οι προβλέψεις που κάνουν οίκοι αξιολόγησης τεχνολογίας για την μελλοντική επικράτηση των συσκευών αυτών στην παγκόσμια αγορά. Τέλος, παρουσιάζεται το λειτουργικό σύστημα Android.


## 2.2 Εισαγωγή στην τεχνολογία



Σύμφωνα με το επιστημονικό περιοδικό Spectrum του IEEE, τα έξυπνα τηλέφωνα ή υπολογιστές τσέπης χαρακτηρίστηκαν σαν η πρώτη σε αποδοχή τεχνολογία της δεκαετίας.

Αναλυτικότερα οι 11 πρώτες (“hot”) τεχνολογίες της δεκαετίας παρουσιάζονται στον πίνακα που ακολουθεί<sup>1</sup>.

<sup>1</sup> <http://spectrum.ieee.org/at-work/innovation/top-11-technologies-of-the-decade>

 <p><b>NO. 1 Smartphones: The Pocketable PC</b> Is your phone smarter than a fifth grader?</p>	 <p><b>NO. 2 Social Networking: Friended</b> Bandwidth, digital cameras, and a hunger for connectedness have created a virtual dinner party</p>	 <p><b>NO. 3 Voice Over IP: Setting Phone Service Free</b> How Ma Bell's cash cow became a free software app</p>
 <p><b>NO. 4 LED Lighting: Blue + Yellow = White</b> Giving LEDs the blues was the key to replacing the incandescent bulb</p>	 <p><b>NO. 5 Multicore CPUs: Processor Proliferation</b> From multicore to many-core to hard-to-describe-in-a-single-word cores</p>	 <p><b>NO. 6 Cloud Computing: It's Always Sunny in the Cloud</b> Cloud computing puts your desktop wherever you want it</p>
 <p><b>NO. 7 Drone Aircraft: How the Drones Got Their Stingers</b> Unmanned aerial vehicles come of age</p>	 <p><b>NO. 8 Planetary Rovers: Are We Alone?</b> Planetary rovers attempt to answer the most profound question in science</p>	 <p><b>NO. 9 Flexible AC Transmission: The FACTS Machine</b> Flexible power electronics will make the smart grid smart</p>
 <p><b>NO. 10 Digital Photography: The Power of Pixels</b> Digital photography changed not only how we take pictures but also how we communicate</p>	 <p><b>NO. 11 Class-D Audio: The Power and the Glory</b> A quiet revolution is transforming audio electronics</p>	 <p><b>Next-to-the-Best Technologies of 2000-2010</b> These innovations just barely missed the cut for our Top 11 list</p>

**Εικόνα 1 - Οι 11 πρώτες (“hot”) τεχνολογίες της δεκαετίας**



Ειδικότερα σύμφωνα με το ίδιο επιστημονικό περιοδικό<sup>2</sup> τα έξυπνα τηλέφωνα (ή υπολογιστές της τσέπης, όπως συχνά ονομάζονται), είναι θαύματα της τεχνολογίας, που περιέχουν περισσότερες λειτουργίες από έναν μέσο υπολογιστή, ενώ έχουν γίνει ο κυριότερος οδηγός καινοτομίας τόσο για υλικό όσο και για λογισμικό.

Ίσως εδώ είναι χρήσιμο να αναφέρουμε ότι πέρασαν 50 χρόνια ώστε να μειωθεί το μέγεθος του υπολογιστή από ένα δωμάτιο σε μία βαλίτσα και μόνο μία δεκαετία για να μειωθεί στο μέγεθος ενός πορτοφολιού.

Στα έξυπνα τηλέφωνα ενσωματώθηκαν οι λειτουργίες ξεχωριστών συσκευών μουσικής, πλοήγησης, παιχνιδιών, ανάγνωσης ηλεκτρονικών βιβλίων, κλπ ενώ συχνά εκτελούν λειτουργίες ασύρματης δικτύωσης αντικαθιστώντας ασύρματους δρομολογητές και μόντεμ. Σύντομα αναμένεται να αποτελέσουν και το ηλεκτρονικό πορτοφόλι προσφέροντας μεταξύ άλλων και υπηρεσίες πιστωτικών καρτών.

Γενικώς, τα έξυπνα τηλέφωνα επιτρέπουν στους χρήστες να παραμένουν συνδεδεμένοι με την πληροφορία οποιαδήποτε στιγμή και σε οποιοδήποτε σημείο, ενώ αξίζει να σημειωθεί ότι αλλάζουν την συμπεριφορά των χρηστών. Ενώ με το κινητό τηλέφωνο οι χρήστες ξόδευαν τον χρόνο τους μιλώντας ή στέλνοντας μηνύματα τώρα οι βασικές επικοινωνίες έχουν περάσει σε δεύτερο ρόλο και ο περισσότερος χρόνος αφιερώνεται σε εφαρμογές, διαδικτυακά παιχνίδια και αποστολή μηνυμάτων μέσω ηλεκτρονικού ταχυδρομείου. Αυτό συμβαίνει για τρεις κυρίως λόγους:

1. **Το υλικό.** Οι οθόνες επαφής υψηλής ανάλυσης, οι πολυπύρνοι επεξεργαστές με ταχύτητες GHz, οι μικροσκοπικοί φακοί καμερών, οι ενσωματωμένοι αισθητήρες, οι κεραίες επικοινωνιών, κλπ προσφέρουν δυνατότητες πολύ διαφορετικές από αυτές των προκατόχων τους, δηλαδή των κινητών τηλεφώνων.
2. **Το λογισμικό.** Το λειτουργικό σύστημα που υποστηρίζει εφαρμογές σε αυτές τις συσκευές αποτελεί το θεμέλιο των εφαρμογών. Όλοι σχεδόν οι

---

<sup>2</sup> <http://spectrum.ieee.org/telecom/wireless/smartphones-the-pocketable-pc>

κατασκευαστές υποστηρίζουν την ανάπτυξη λογισμικού από τρίτους (software from third parties) έτσι ώστε να προσφέρουν πολύ περισσότερες εφαρμογές από αυτές που θα μπορούσαν να αναπτύξουν μόνοι τους. Οι εφαρμογές αυτές στέλνουν και δέχονται δεδομένα από το διαδίκτυο, κυρίως λόγω του διαθέσιμου εύρους ζώνης.

3. **Το εύρος ζώνης** που σήμερα είναι διαθέσιμο (κυρίως «ασύρματα») και η δυνατότητα αυτόματης μετάπτωσης από WiFi σε 3G δίκτυα (και αντιστρόφως).

Οι μελλοντικές δυνατότητες είναι επίσης απεριόριστες, αφού λόγω των ενσωματωμένων αισθητήρων, μπορούν να αποτελέσουν ένα τεράστιο δίκτυο συλλογής πληροφοριών κάθε είδους.

## 2.3 Ανάλυση λειτουργικών συστημάτων για έξυπνα τηλέφωνα

Η σύγχρονη τεχνολογία επιτρέπει σε λειτουργίες που έτρεχαν σε υπολογιστές γραφείου, να εκτελούνται σε φορητές υπολογιστικές συσκευές όπως τα έξυπνα τηλέφωνα. Η εξέλιξη των έξυπνων τηλεφώνων και οι χαμηλές τους τιμές έχουν σαν αποτέλεσμα την αύξηση της κίνησης του Διαδικτύου μέσω και αυτών των συσκευών. Για να αντιμετωπιστεί η απαίτηση της αγοράς, αναπτύχθηκαν πολλά λειτουργικά συστήματα για εφαρμογή σε κινητές συσκευές. Αυτά τα λειτουργικά συστήματα προσφέρονται σαν πλατφόρμα ανάπτυξης νέων καινοτόμων εφαρμογών<sup>3</sup>.

Τα βασικά λειτουργικά συστήματα αναπτύχθηκαν από εταιρίες που ανταγωνίζονται, όπως η Google, Microsoft, Apple, Symbian, Palm κλπ. Κανένα από τα προαναφερθέντα λειτουργικά συστήματα δεν ανταποκρίνεται απόλυτα στις ανάγκες όλων των χρηστών.

Τα βασικά αυτά λειτουργικά συστήματα είναι:

---

3 Comprehensive Analysis of Smartphone OS Capabilities and Performance. Michael Wei plus others, 20 April 2009

1. **ANDROID** – ανακοινώθηκε το 2007 από την Open Handset Alliance και το πρώτο εμπορικό προϊόν παρουσιάστηκε τον Σεπτέμβριο του 2008. Το σύγχρονο αυτό λειτουργικό σύστημα είναι κτισμένο πάνω σε ένα Linux kernel που προσφέρει την διασύνδεση και τον έλεγχο του υλικού. Όλες οι εφαρμογές του Android είναι αναπτυγμένες σε java (την πιο δημοφιλή γλώσσα για εφαρμογές που τρέχουν σε διαφορετικές πλατφόρμες). Περισσότερη ανάλυση ακολουθεί στο Κεφάλαιο 2.
2. **iPhone OS** – είναι παράγωγο του Darwin\_POSSIX που ανέπτυξε η Apple Inc. Το λειτουργικό αυτό σύστημα τρέχει μόνο σε συσκευές Apple και γνώρισε μεγάλη αποδοχή κυρίως λόγω της καινοτομίας που είχε στις διεπαφές χρήστη καθώς και στις πολλές εφαρμογές που αναπτύχθηκαν από τρίτους. Ένα κεντρικό ηλεκτρονικό κατάστημα συγκεντρώνει και διαθέτει σε όλους τους χρήστες εφαρμογές, επεκτάσεις και αναβαθμίσεις. Συμπληρωματικά, επειδή το υλικό και λειτουργικό σύστημα είναι του ίδιου κατασκευαστή, επιτυγχάνεται πλήρης βελτιστοποίηση των δυνατοτήτων. Διαθέτει ένα πλήρες περιβάλλον ανάπτυξης SDK ενώ έχει και μία σειρά από περιορισμούς όπως είναι η προμήθεια του υλικού και λογισμικού από τον ίδιο κατασκευαστή, η διάθεση επικοινωνιών και ιντερνέτ μέσω συγκεκριμένων παρόχων και η περιορισμένη λειτουργικότητα σε multitasking.
3. **SYMBIAN** – σχεδιάστηκε αποκλειστικά για χρήση σε κινητές συσκευές με περιορισμένη απαίτηση μνήμης και χαμηλή κατανάλωση. Είναι ένα ανοικτό λειτουργικό σύστημα που επιτρέπει την ανάπτυξη εφαρμογών από τρίτους και ανεξάρτητα από τον κατασκευαστή του υλικού. Διαθέτει ένα ανεπτυγμένο API σε C++ που επιτρέπει πρόσβαση σε διάφορες υπηρεσίες όπως τηλεφωνία και τρέχει σε πολλές, διαφορετικού τύπου, συσκευές. Υποστηρίζει αρχιτεκτονικές εξυπηρετητή-πελάτη, ενώ όλες οι λειτουργίες συστήματος τρέχουν απευθείας από ROM και μόνο οι εφαρμογές χρηστών τρέχουν από RAM.

4. **WINDOWS MOBILE** – το λειτουργικό σύστημα βασίζεται στο Windows CE και αρχικά έτρεχε σε Pocket PC ενώ αργότερα χρησιμοποιήθηκε και σε έξυπνα τηλέφωνα. Η ανάπτυξη του λογισμικού βασίζεται σε Visual C++ και αξιοποιεί τις δυνατότητες του .net. ενώ υποστηρίζει 256 επίπεδα προτεραιοτήτων για νήματα και μέχρι 32 διεργασίες. Το WINDOWS Phone 7 ανακοινώθηκε πρόσφατα και διαδέχεται το προηγούμενο λειτουργικό σύστημα. Έχει αναπτυχθεί ειδικά για κινητές συσκευές και σε συνεργασία με την NOKIA θα χρησιμοποιηθεί σαν το λειτουργικό σύστημα της τελευταίας για τα έξυπνα τηλέφωνα.
5. **PALM OS** - ιδιόκτητο λειτουργικό σύστημα που σχεδιάστηκε από την Palm Inc. Και αρχικά είχε εφαρμογή σε συσκευές PDA PALM που κυριαρχούσαν στην αγορά. Τα τελευταία χρόνια η αγορά της PALM έπεσε σημαντικά κυρίως λόγω των περιορισμών του λειτουργικού συστήματος. Χρησιμοποιήθηκε από διάφορους κατασκευαστές έξυπνων τηλεφώνων.
6. **BLACKBERRY** – είναι το πιο δημοφιλές λειτουργικό σύστημα που υποστηρίζει ηλεκτρονικό ταχυδρομείο για όσους ταξιδεύουν. Έχει την δυνατότητα να ολοκληρώνει διάφορα συστήματα ηλεκτρονικού ταχυδρομείου σε κινητά τηλέφωνα. Το λειτουργικό σύστημα υποστηρίζει και άλλες εφαρμογές όπως πρόσβαση στο ιντερνέτ και διαθέτει SDK Java για ανάπτυξη εφαρμογών.

Το περιβάλλον ανάπτυξης εφαρμογών για τα προαναφερόμενα λειτουργικά συστήματα παρουσιάζεται στον πίνακα που ακολουθεί.

Mobile OS	ANDROID	IPHONE	SYMBIAN	WINDOWS MOBILE	PALM
SDK Available	Yes	Yes	Yes	Yes	Yes
SDK Language	Android (JAVA derivative)	COCOA (Objective C)	C++, OPL, Python, Visual Basic, Simkin, and Perl – together with the Java ME and PersonalJava flavors of Java.	.NET framework (Visual C++, C#, etc)	C/C++, Visual Basic, Visual Basic.Net, C#, Java, Pascal, Lua, Forth
IDE Options	Eclipse, Other Java Development Environments	XCode	CodeWarrior VistaMax Eclipse, NS Basic, Borland, Xcode	MS Visual Studio	CodeWarrior, Eclipse, PRC-Tools, Palm OS Developer Suite, PocketC Architect, AppForge Crossfire, Handheld Basic, Pendragon /Satellite Forms, PocketStudio

**Εικόνα 2 - Περιβάλλον ανάπτυξης εφαρμογών σε Smartphone**

Και τα χαρακτηριστικά των λειτουργικών συστημάτων παρουσιάζονται στον πίνακα που ακολουθεί.

Mobile OS	ANDROID	IPHONE	SYMBIAN	WINDOWS MOBILE	PALM
<b>3rd Party Multitasking</b>	Yes	No	Yes	Yes	No
<b>Thread Priority Levels</b>	40	N/A	12	256	2+
<b>Kernel Type</b>	Linux-based (v2.6)	XNU Kernel	EKA2 nanokernel	WinCE	AMX RTOS Kadak Kernel
<b>System Call Handling</b>	User thread handles system calls with altered permissions	N/A	EKA2 is multi-threaded, allowing low-priority kernel preemption	Triggers an exception handled at the kernel level	Single-threaded Event-Driven Loop/Queue
<b>Multi-task Scheduling</b>	Every application runs as single separated process. Multi-task scheduling is done by Linux	-Only one 3rd party application can run at a time -Applications all have the same priority	Provides five threads: Null (idles CPU, defrags RAM), Supervisor, DFC 0 (runs DFCs), DFC 1 (runs timer queue), and Timer	Kernel keeps priority list for all threads, Allows real-time interrupt re-prioritizing, Scheduling allowed within threads, Kernel methods for synchronization	Multi-task scheduling is only available to the OS; 3rd party apps must be called from the User Interface Application Shell (UIAS) task thread
<b>Memory Management</b>	Memory management is done by Linux. In addition, Android uses more compact Dalvik EXecutable (dex) files as the bytecode format instead of conventional format to run applications with smaller memory use.	Memory for objects are freed based on reference counting. When the reference counter is 0 for a specific object, that object's memory is freed.	Provides four models: Direct (no MMU), Moving (similar to EKA1), Multiple (ASICs with tagged caches), and Emulator (Symbian OS emulator on Windows), Performs low-level memory management, context switching, and inter-process data transfers	User access to memory states for custom memory management; System auto-handles memory low conditions for hibernation and active states through hibernate, close, and out-of-memory messages and forced automatic shutdowns	No fragmentation allowed; Utilizes "Relocatable", "Locked Relocatable", and "Non-Relocatable" designations for memory blocks, allowing memory to be re-organized when allocating new chunks of memory

**Εικόνα 3 - Χαρακτηριστικά λειτουργικών συστημάτων**

## 2.4 Πως χρησιμοποιούνται τα έξυπνα τηλέφωνα

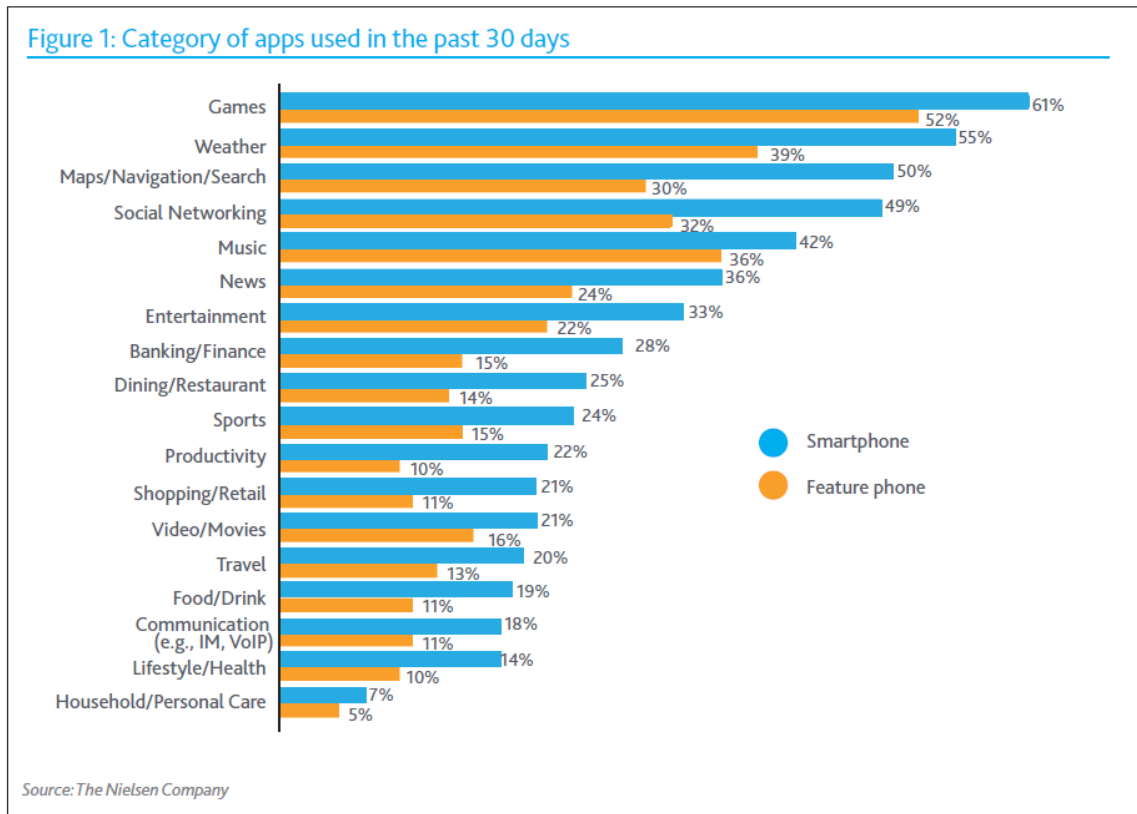
Σύμφωνα με έρευνα<sup>4</sup> της εταιρίας Nielsen (Η Nielsen Company είναι μία εταιρία μάρκετινγκ με έδρα στην Νέα Υόρκη) οι εφαρμογές για έξυπνα τηλέφωνα παρουσιάζουν σημαντική ανάπτυξη και αποδοχή. Πάνω από το 60% των χρηστών έξυπνων τηλεφώνων δηλώνουν ότι έχουν κατεβάσει κάποια εφαρμογή τον τελευταίο μήνα, εκτός από αυτές που έχει το τηλέφωνό τους.

Ειδικά σε περιπτώσεις όπου απαιτούνται μεγαλύτερη υπολογιστική ισχύς, μεγαλύτερες οθόνες και λειτουργικές διεπαφές χρήστη, όπως για παράδειγμα σε εφαρμογές χαρτών και πλοήγησης, τότε η χρήση των έξυπνων τηλεφώνων υπερτερεί των απλών κινητών. Το ίδιο βέβαια συμβαίνει και σε εφαρμογές όπου απαιτείται συνεχής διασύνδεση με το διαδίκτυο όπως είναι οι εφαρμογές κοινωνικής δικτύωσης.

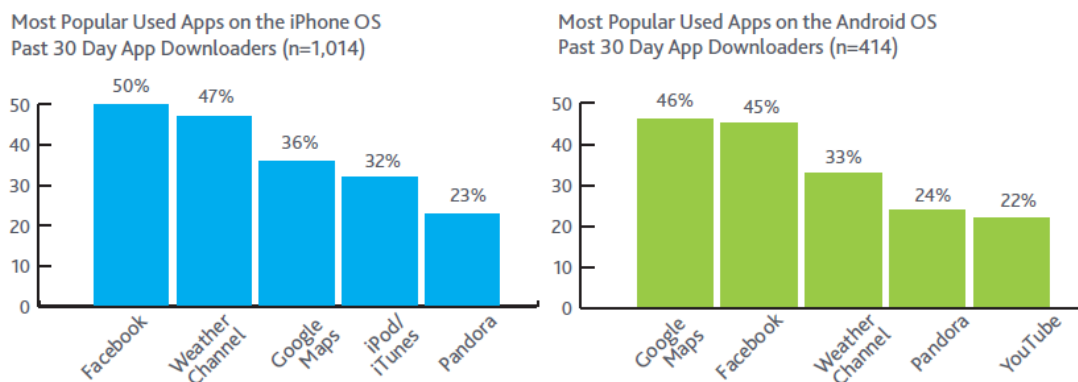
Στο διάγραμμα που ακολουθεί φαίνεται το πώς οι χρήστες χρησιμοποιούν τα έξυπνα τηλέφωνα τους.

---

4 The State Of Mobile Apps. Nielsen, September 2010



**Εικόνα 4 - Κατηγορίες εφαρμογών Smartphone**



**Εικόνα 5 - Δημοφιλείς εφαρμογές**

Η έκθεση καταλήγει ότι η αγορά των εφαρμογών είναι πολύ νέα και υπάρχει αρκετός χώρος για νεοεισερχόμενη καινοτομία και για νέες κατηγορίες εφαρμογών, που ακόμα δεν μπορούμε να φανταστούμε.



Μία άλλη μελέτη ασχολείται με τον τρόπο που οι χρήστες αλληλεπιδρούν με τα έξυπνα τηλέφωνα<sup>5</sup>. Ο στόχος της μελέτης είναι να προτείνει βελτιώσεις στις διεπαφές του χρήστη και στην εξοικονόμηση ενέργειας που είναι ο μεγαλύτερος περιοριστικός παράγοντας σε κινητές εφαρμογές. Για παράδειγμα, αν η αλληλεπίδραση με την εφαρμογή είναι πολύ συχνή τότε η συνεχής αδρανοποίηση και ενεργοποίηση του τηλεφώνου είναι αντιπαραγωγική ενώ η διατήρηση της εφαρμογής στην μνήμη μπορεί να μειώσει σημαντικά τους χρόνους απόκρισης.

Από την μελέτη προκύπτει ότι, επειδή υπάρχει μεγάλη διαφορά στην χρήση των έξυπνων τηλεφώνων, ο πιο αποτελεσματικός τρόπος είναι να αναπτυχθούν μηχανισμοί που θα μαθαίνουν το προφίλ του κάθε χρήστη και θα προσαρμόζουν την εφαρμογή στην συγκεκριμένη συμπεριφορά.

## 2.5 Εφαρμογές για πολλές πλατφόρμες

Οι πολλές διαφορετικές πλατφόρμες που είναι σήμερα διαθέσιμες αποτελούν πρόκληση για τους σχεδιαστές εφαρμογών<sup>6,7</sup>.

Για τους προγραμματιστές των κλασικών επιτραπέζιων και φορητών υπολογιστών τα πράγματα είναι πολύ εύκολα. Οι περισσότεροι μπορούν να φτιάχνουν εφαρμογές για το 90% της αγοράς. Δεν ισχύει όμως το ίδιο για τους προγραμματιστές εφαρμογών σε έξυπνα τηλέφωνα.

Η αγορά μοιράζεται μεταξύ Android, iPhone, Blackberry και Symbian παρουσιάζοντας ένα ετερογενές περιβάλλον. Τα έξυπνα τηλέφωνα, οι ασύρματες ευρυζωνικές επικοινωνίες και οι δικτυακές εφαρμογές «νέφους» παρουσιάζουν πολύ μεγάλες ευκαιρίες για ανάπτυξη εφαρμογών. Όμως η πληθώρα και πολυπλοκότητα των χαρακτηριστικών που τα έξυπνα τηλέφωνα διαθέτουν δημιουργεί μεγάλες δυσκολίες στους σχεδιαστές των εφαρμογών.

---

5 Diversity in Smartphone Usage. Falaki, Mahajan, Kandula, ACM June 2010

6 Cross-Platform Smartphone Apps Still Difficult. Daniel Dem, IEEE Spectrum, June 2010

7 <http://spectrum.ieee.org/geek-life/tools-toys/crossplatform-smartphone-apps-still-difficult>

Ένα πρόγραμμα περιήγησης ιστού μπορεί να έχει βελτιστοποιηθεί για χρήση σε κάποιο συγκεκριμένο έξυπνο τηλέφωνο αλλά να μην έχει την ίδια απόδοση σε κάποιο άλλο ή ακόμα και σε άλλο μοντέλο του ίδιου κατασκευαστή. Αυτό συμβαίνει διότι η υπολογιστική ισχύς, η διαθέσιμη μνήμη και οι ενσωματωμένες λειτουργίες διαφέρουν από συσκευή σε συσκευή. Μπορεί να ελαχιστοποιηθεί αυτή η διαφορά αν ο σχεδιαστής μεταφέρει κάποιες λειτουργίες σε κεντρικό server σε βάρος βέβαια της ταχύτητας.

Μία εφαρμογή που έχει αναπτυχθεί σε Blackberry δεν τρέχει σε iPhone. Η μεταφορά μιας εφαρμογής από το ένα περιβάλλον στο άλλο μπορεί να σημαίνει την συγγραφή του κώδικα της εφαρμογής από την αρχή, ιδιαίτερα αφού σχεδόν το κάθε σύστημα χρησιμοποιεί διαφορετική γλώσσα προγραμματισμού.

Ο Chris Chodnicki (CTO μιας εταιρίας τεχνολογίας στην Βαλτιμόρη) υποστηρίζει ότι το να γνωρίζει, σε επίπεδο ειδικού, κάποιος πάνω από δύο γλώσσες για κινητές εφαρμογές αποτελεί άθλο.

Η αγορά αντιμετωπίζει την ανάπτυξη εφαρμογών για διαφορετικές πλατφόρμες με ειδικά εργαλεία όπως είναι τα Appcelerator Titanium, Phonegap, QuickConnect, Rhomobile κλπ. Τα περισσότερα από αυτά χρησιμοποιούν ένα μίγμα από HTML, CSS και JavaScript σε συνδυασμό με τοπικούς κώδικες wrapper για την πρόσβαση στις κάμερες, GPS, accelerometers κλπ.

## 2.6 Smartphones – έξυπνα τηλέφωνα

Τα Smartphones είναι κινητά τηλέφωνα τα οποία προσφέρουν προηγμένες δυνατότητες υπολογισμού και συνδεσιμότητας από ένα σύγχρονο κινητό τηλέφωνο. Τα Smartphones μπορούν να θεωρηθούν ως φορητοί υπολογιστές, οι οποίοι ενσωματώνονται μέσα σε ένα κινητό τηλέφωνο. Ενώ όμως τα περισσότερα σύγχρονα τηλέφωνα είναι σε θέση να τρέξουν εφαρμογές που βασίζονται σε πλατφόρμες όπως Java ME, ένα Smartphone συνήθως επιτρέπει στο χρήστη να εγκαταστήσει και να εκτελέσει πιο προηγμένες εφαρμογές. Ένα Smartphone

μπορεί να θεωρηθεί ως προσωπικός υπολογιστής τσέπης (Personal Pocket Computer - PPC) με τις λειτουργίες των κινητών τηλεφώνων, επειδή αυτές οι συσκευές κυρίως χαρακτηρίζονται ως υπολογιστές, αν και είναι πολύ μικρότεροι από έναν επιτραπέζιο υπολογιστή (Desktop Computer). Επιπλέον οι PPC είναι πιο προσωπικοί από τους επιτραπέζιους υπολογιστές.

Σε αναφορά του επιστημονικού περιοδικού IEEE Spectrum<sup>8</sup> παρουσιάζονται 8 ορόσημα που προσδιορίζουν την εξέλιξη των κινητών τηλεφώνων, όπως φαίνεται στην παρακάτω εικόνα:

### HOW PHONES GOT SMART: EIGHT MILESTONES

#### IBM SIMON PERSONAL COMMUNICATOR [1993]

IBM was ahead of the curve back in 1993. The Simon was a touch-screen phone with a calendar, address book, calculator, and even the capability to send and receive e-mails and faxes. It was impressive, but also bulky, and came with a hefty price tag: US \$899.



#### NOKIA 9000 COMMUNICATOR [1996]

If you wanted a smartphone in the late 1990s, your choice was pretty much limited to the pricey Communicator. The device was like a personal digital assistant and a mobile phone sandwiched together. (It's rumored that an early prototype was actually a Hewlett-Packard PDA connected to a Nokia phone by a hinge.)

#### ERICSSON R380 [2000]

The R380 was one of the first phones actually marketed as a "smartphone." It was the first commercial phone to run the Symbian OS. Symbian has long been the most common smartphone operating system, but its market share has rapidly declined since it began facing competition from Apple's iOS and Google's Android.



#### RIM BLACKBERRY 6210 [2003]

Research in Motion's early communicators offered two-way text paging, but the company soon realized that push e-mail service was its killer app. RIM developed a QWERTY keyboard for quick thumb typing and BlackBerry Enterprise Servers to tap into existing e-mail infrastructure. In addition to e-mail and basic Web browsing, the 6200 series were the first BlackBerries that were also fully integrated phones (earlier models required a headset).



#### SONY ERICSSON P800 [2002]

The P800 came with several new features that we consider standard for smartphones today: It could play MP3s, came with a camera, and featured a color touch screen (although it supported only 4 096 colors).



#### PALM TREO 600 [2003]

The Treo combined some of the best features of the Palm Pilot PDAs with a mobile phone shape, creating a smartphone that had appeal beyond business users. With its Palm OS, the Treo supported lots of third-party apps. Palm OS also gave the Treo integrated functions, like the ability to dial directly from the contact list and check the calendar while on the phone.



8 <http://spectrum.ieee.org/image/1752995>



### Εικόνα 6 - Εξέλιξη έξυπνων τηλεφώνων

- Το 2000 η Ericsson έβγαλε το πρώτο Smartphone με λειτουργικό Symbian OS και το 2002 το P800 ήταν το πρώτο Smartphone με κάμερα .
- Το 2003 η Research in Motion (RIM) Blackberry με την σειρά 6200 έδωσε στην αγορά πλήρως ολοκληρωμένα τηλέφωνα με δυνατότητες ηλεκτρονικού ταχυδρομείου και περιήγηση ιστού.
- Το 2003 η PALM αξιοποίησε την τεχνογνωσία για PDA δίνοντας στην αγορά το TREO 600 με λειτουργικό σύστημα Palm OS και απευθύνθηκε σε χρήστες πέρα των επιχειρήσεων. Το έξυπνο αυτό τηλέφωνο είχε την δυνατότητα ολοκληρωμένων εφαρμογών.
- Το 2007 η Apple παρουσίασε το iPhone που αμέσως σάρωσε την αγορά προσφέροντας οθόνη αφής και πρόγραμμα περιήγησης ιστού, το mobile safari, που κατέκτησε τους χρήστες.
- Το 2010 η HTC με το EVO4G έδωσε στην αγορά το καλύτερο Android έξυπνο τηλέφωνο με δυνατότητες ,μεταξύ άλλων, να αποτελεί επικοινωνιακό κόμβο (hotspot) σε δίκτυα WiMAX.

Μια γενική παρατήρηση είναι ότι τα έξυπνα τηλέφωνα εξελίσσονται σε υπολογιστές και σίγουρα κάποια ημέρα ο κύριος υπολογιστής που θα

χρησιμοποιούμε στο γραφείο μας θα είναι στην τσέπη μας και μάλιστα θα διαθέτει όλες τις λειτουργίες του τηλεφώνου μας<sup>9</sup>.

Η αρχιτεκτονική του έξυπνου τηλεφώνου μπορεί να χωριστεί σε δύο ενότητες, στον επεξεργαστή επικοινωνιών και στον επεξεργαστή εφαρμογών. Ενώ στα κινητά τηλέφωνα ο επεξεργαστής εφαρμογών είναι ένας χαμηλού επιπέδου μικροελεγκτής, στην περίπτωση των έξυπνων τηλεφώνων είναι μία πολύ ισχυρή μηχανή επεξεργασίας. Σήμερα η εξέλιξη της ενσωματωμένης τεχνολογίας επιτρέπει την τοποθέτηση και των δύο αυτών επεξεργαστών στο ίδιο chip με την αρχιτεκτονική του system-on-chip και μάλιστα έχει ήδη αρχίσει η εμφάνιση πολυπύρηνων επεξεργαστών σε έξυπνα τηλέφωνα. Ήδη από το 2007 οι πωλήσεις έξυπνων τηλεφώνων στην Αμερική ξεπέρασαν τις πωλήσεις των φορητών υπολογιστών.

## 2.7 Στατιστικά στοιχεία αγοράς

Σε μία έκθεση του email marketing report <sup>10</sup> παρουσιάζονται αναλύσεις διάφορων οίκων αξιολόγησης αγοράς και τεχνολογίας που αφορούν τα έξυπνα τηλέφωνα.

Σύμφωνα με την Nielsen , το 38% των χρηστών κινητών τηλεφώνων στην Αμερική είχαν έξυπνο τηλέφωνο (Μάιος 2011) ενώ στο τέλος του 2011 το ποσοστό αυτό θα ξεπεράσει το 50%.

Η IMS Research αναμένει πωλήσεις 420 εκατομμυρίων έξυπνων τηλεφώνων το 2011 σε παγκόσμια κλίμακα, αντιπροσωπεύοντας το 28% της αγοράς, ενώ προβλέπουν ότι το 2016 οι πωλήσεις θα φτάσουν το 1 δις συσκευές , δηλαδή θα καλύπτουν το 50% της αγοράς. Τα ίδια περίπου στοιχεία προβλέπει η IDC και η Gartner.

Η Gartner προβλέπει ότι το 2011 στην Αμερική οι πωλήσεις των έξυπνων τηλεφώνων θα αποτελούν την υψηλότερη κατηγορία πωλήσεων ηλεκτρονικών συσκευών. Οι πωλήσεις το 2010 σε έξυπνα τηλέφωνα ανήλθαν σε 297 εκατομμύρια συσκευές σε παγκόσμια κλίμακα, από το σύνολο του 1,6 δις κινητών

<sup>9</sup> When cell phones become computers, IEEE pervasive computing, April-June 2009

<sup>10</sup> Smartphone statistics and market share. Mark Brow, August 2011

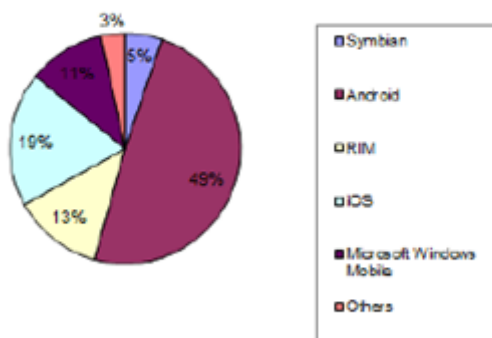


τηλεφώνων που πουλήθηκαν εκείνη την χρονιά, παρουσιάζοντας άνοδο 72% σε σχέση με το 2009.

Η Coda research consultancy προβλέπει πωλήσεις 2,5 δις έξυπνων τηλεφώνων την περίοδο 2010-2015 ενώ η πρόσβαση στο ίντερνέτ μέσω αυτών των συσκευών θα αυξηθεί 50 φορές μέσα στο ίδιο διάστημα.

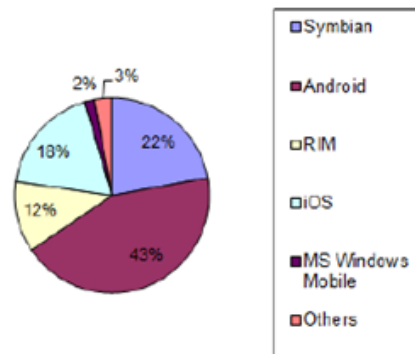
Στην παρακάτω εικόνα παρουσιάζεται μια ανάλυση της Gartner σε παγκόσμια κλίμακα για το 2ο τρίμηνο του 2011 όπου δείχνει ότι το λειτουργικό σύστημα ANDROID κυριαρχεί στην αγορά.

In 2012, they expect this domination to continue, largely at the expense of the phased-out Symbian system, predicting:



Source: Gartner (April 2011)

Gartner's analysis of global Q2 2011 smartphone sales shows the Android operating system dominating market share (rounded to nearest percentage point):

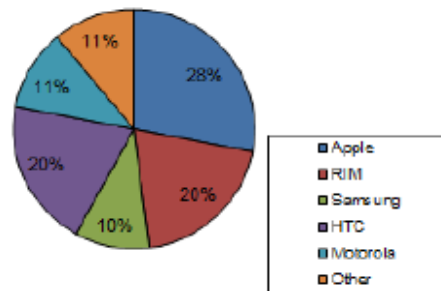


Source: Gartner (August 2011)

### Εικόνα 7 - Στατιστικά στοιχεία πωλήσεων λειτουργικών συστημάτων

Ενώ το 2012 το ANDROID θα εξακολουθεί να κυριαρχεί (50% της αγοράς) λόγω της απόσυρσης του Symbian και της αντικατάστασής του με το Windows Phone 7 σύμφωνα με την Gartner.

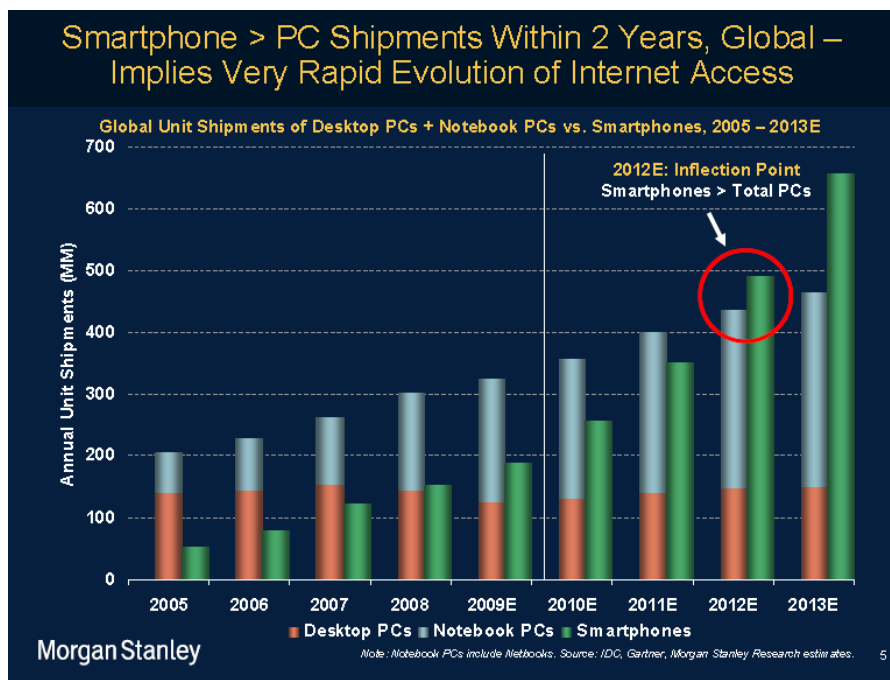
Η Nielsen παρουσιάζει στοιχεία για το 2011 όπου δείχνει ότι η Apple πουλάει τα περισσότερα έξυπνα τηλέφωνα στην αγορά.



Source: Nielsen

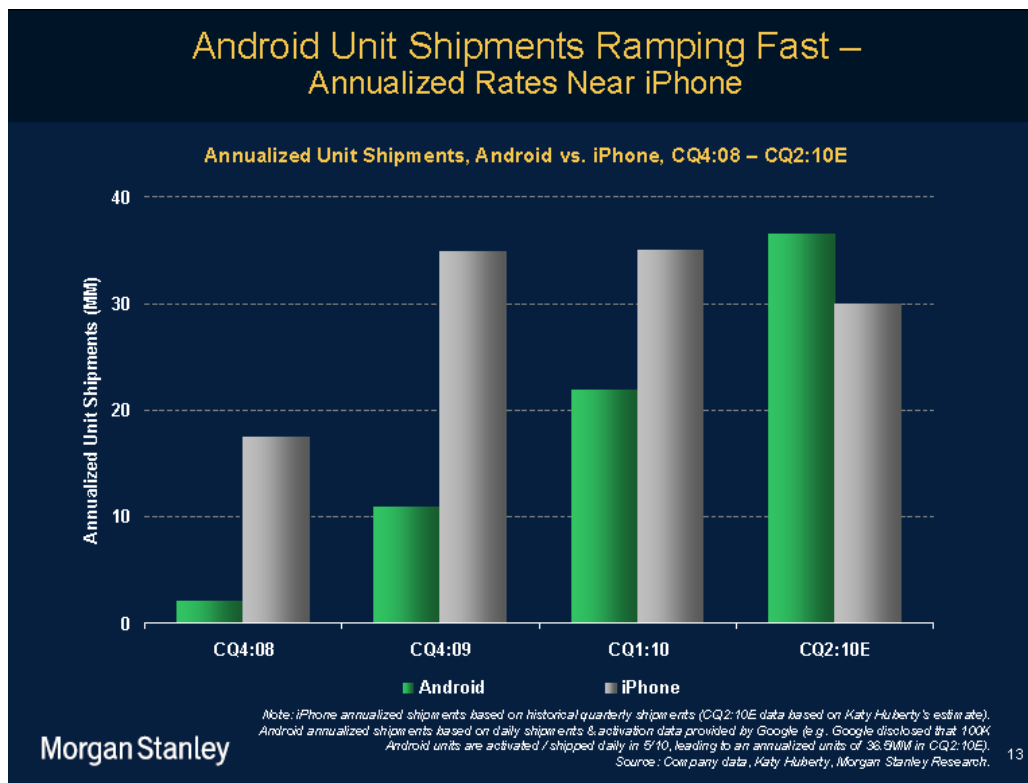
### Εικόνα 8 - Στατιστικά στοιχεία πωλήσεων Smartphones

Η Morgan Stanley<sup>11</sup> παρουσιάζει ανάλυση με την οποία φαίνεται ότι το 2012 οι πωλήσεις έξυπνων τηλεφώνων θα ξεπεράσουν τις πωλήσεις των υπολογιστών (επιτραπέζιων και φορητών), ενώ ήδη από το 2010 το λειτουργικό σύστημα ANDROID ξεπέρασε σε πωλήσεις το iPhone της Apple.



### Εικόνα 9 - Πωλήσεις Smartphone versus PC

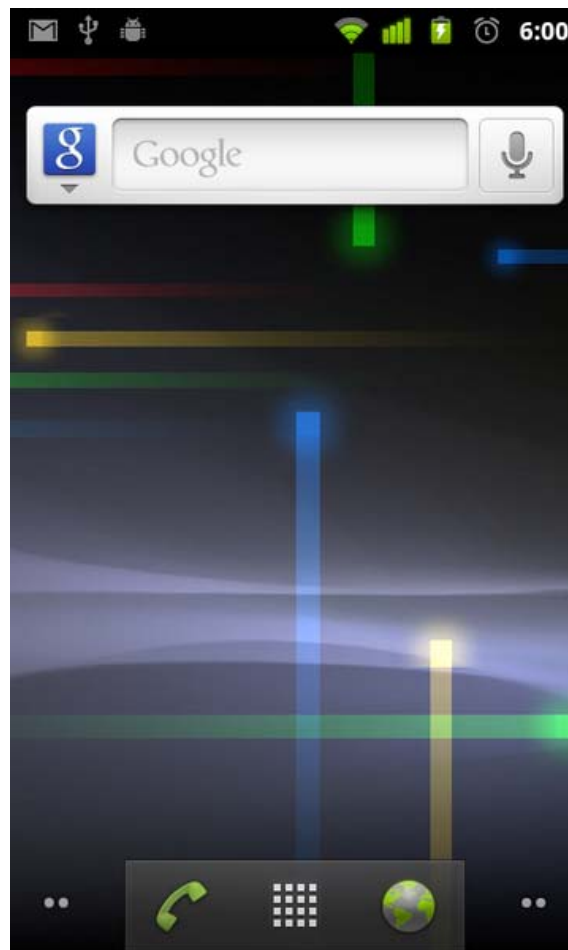
11 Internet trends CM summit New York. Morgan Stanley, 7 June 2010



Εικόνα 10 - Πωλήσεις Android versus iphone



## 2.8 Λειτουργικό Σύστημα ANDROID



Το Android<sup>12</sup> είναι ένα ανοικτό λογισμικό σε μορφή στοίβας που περιλαμβάνει το λειτουργικό σύστημα, μεσολογισμικό (middleware) και εφαρμογές για κινητές συσκευές μαζί με ένα σύνολο API βιβλιοθηκών για την ανάπτυξη εφαρμογών.

<sup>12</sup> [http://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))

## Ιστορικό

Η Android Inc. ιδρύθηκε στην Καλιφόρνια (Πάλο Άλτο) των ΗΠΑ το 2003 με σκοπό να αναπτύξει «...περισσότερο έξυπνες φορητές συσκευές που αναγνωρίζουν την θέση και τις προτιμήσεις του ιδιοκτήτη τους» όπως δήλωσε ο Andy Rubin, συνιδρυτής της εταιρίας.

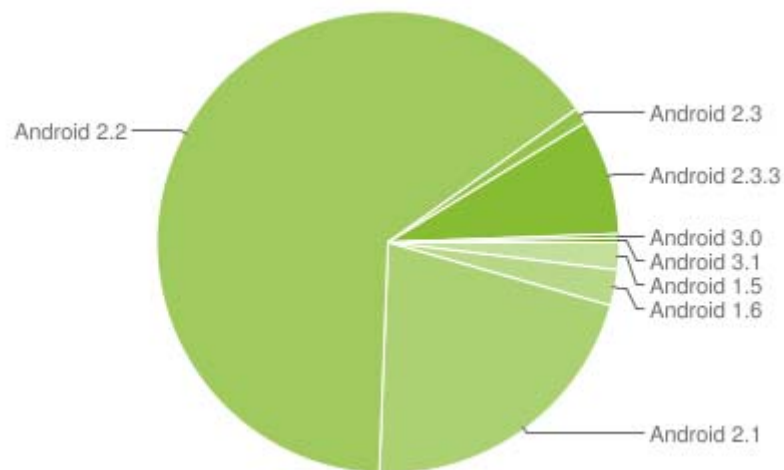
Το 2005 η Google, αναγνωρίζοντας τη δυναμική της εταιρίας, της ομάδας των ιδρυτών αλλά και της αγοράς των κινητών τηλεφώνων, αγόρασε την Android Inc. και ανέθεσε στην ίδια ομάδα να αναπτύξει μία πλατφόρμα για κινητές συσκευές την οποία και διέθεσε στην αγορά.

Στις 5 Νοεμβρίου 2007 ανακοινώθηκε η δημιουργία του Open Handset Alliance που ήταν μία κοινοπραξία των Broadcom Corporation, Google, HTC, Intel, LG, Marvell Technology Group, Motorola, Nvidia, Qualcomm, Samsung, Electronics, SprintNextel, T-Mobile και Texas Instruments με σκοπό να αναπτύξουν ανοικτά πρότυπα για κινητές συσκευές. Την ίδια ημέρα η κοινοπραξία ανακοίνωσε και το πρώτο της προϊόν το Android SDK βασισμένο σε Linux Kernel.

Στις 9 Δεκεμβρίου 2008 η κοινοπραξία δέχθηκε 14 νέα μέλη (ARM Holdings, Atheros Communications, Asustek Computer Inc, Garmin Ltd, Huawei Technologies, PacketVideo, Softbank, Sony Ericsson, Toshiba Corp, and Vodafone Group Plc).

Από τις 21 Οκτωβρίου 2008 το Android προσφέρεται στην αγορά με άδεια ελεύθερου λογισμικού και δημοσιευμένο τον πηγαίο κώδικα. Παρόλα αυτά η Google πιστοποιεί κάθε κινητή συσκευή ώστε να είναι συμβατή και να μπορεί να χρησιμοποιήσει το εμπορικό όνομα android και να έχει πρόσβαση στο Android market.

Η κατανομή των διαφόρων εκδόσεων που σήμερα κυκλοφορούν στην αγορά φαίνεται στο διάγραμμα που ακολουθεί.



**Εικόνα 11 - Διαφορετικές εκδόσεις Android**

Version	API level	Distribution
3.x.x <i>Honeycomb</i>	11	0.6%
2.3.x <i>Gingerbread</i>	10	9.2%
2.2.x <i>Froyo</i>	8	64.6%
2.0.x/2.1.x <i>Eclair</i>	7	21.2%
1.6 <i>Donut</i>	4	2.5%
1.5 <i>Cupcake</i>	3	1.9%

Η πιο δημοφιλής έκδοση είναι η 2.2 ενώ έχει ανακοινωθεί η νεώτερη έκδοση 3.1 σε API επίπεδο 12 <http://developer.Android.com/sdk/android-3.1.html>



Και αναμένεται η έκδοση 3.5 (ice cream sandwich) στο τέταρτο τρίμηνο του 2011.

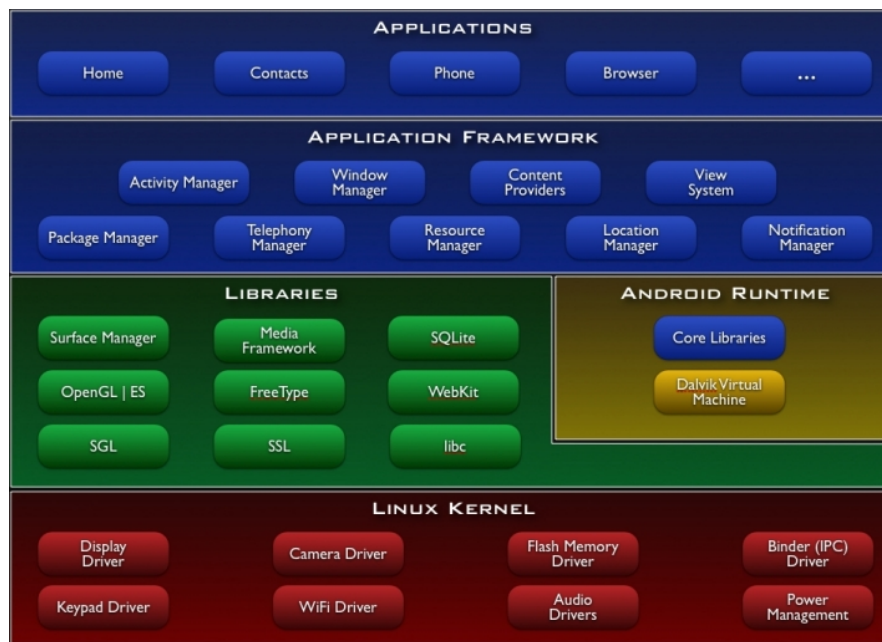
### **Δυνατότητες**

Το Android SDK προσφέρει νέες δυνατότητες για εφαρμογές, προσφέροντας ένα ανοικτό περιβάλλον ανάπτυξης χτισμένο στο ανοικτό πυρήνα του LINUX και με χρήση της γλώσσας προγραμματισμού JAVA . Η πρόσβαση στο υλικό είναι διαθέσιμη σε όλες τις εφαρμογές μέσω των API βιβλιοθηκών και η αλληλεπίδραση των εφαρμογών , όταν ελέγχονται προσεκτικά , υποστηρίζονται πλήρως. Οι εφαρμογές από τρίτους και οι αρχικές Android εφαρμογές είναι γραμμένες με τις ίδιες API βιβλιοθήκες και εκτελούνται στο ίδιο χρόνο. Οι χρήστες μπορούν να σβήσουν οποιαδήποτε αρχική εφαρμογή και να την αντικαταστήσουν με μια δικιά τους, όπως την κύρια οθόνη και τον τρόπο που γίνονται οι κλήσεις. Το περιβάλλον εφαρμογών επιτρέπει την επαναχρησιμοποίηση και αντικατάσταση συστατικών του λογισμικού. Διαθέτει εικονική μηχανή Dalvik βελτιστοποιημένη για κινητές συσκευές καθώς και ενσωματωμένο browser ανοικτού κώδικα. Υποστηρίζει SQLite για αποθήκευση δεδομένων, διάφορους τύπους format πολυμέσων όπως MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF . Υποστηρίζει επικοινωνίες GSM, bluetooth, EDGE, 3G, WiFi, καθώς και GPS, πυξίδα, επιταχυνσιόμετρα κλπ. Υποστηρίζει streaming media (RTP/RTSP, HTML progressive download, Adobe flash streaming, HTTP Dynamic streaming

Η Αρχιτεκτονική του Android<sup>13</sup> φαίνεται στο διάγραμμα που ακολουθεί:

---

<sup>13</sup> <http://developer.android.com/index.html>



**Εικόνα 12 - Αρχιτεκτονική Android**

Το Android περιλαμβάνει κάποιες βασικές εφαρμογές γραμμένες σε γλώσσα java ενώ το ανοικτό περιβάλλον εφαρμογών επιτρέπει την ανάπτυξη καινοτόμων εφαρμογών που εκμεταλλεύονται όλες τις δυνατότητες του υλικού. Παρέχεται πρόσβαση στο ίδιο API των βασικών εφαρμογών και η δυνατότητα τροποποίησης και επαναχρησιμοποίησης του κώδικα. Οι βιβλιοθήκες είναι προσβάσιμες μέσω του περιβάλλοντος των εφαρμογών και περιλαμβάνουν System C library, 3D libraries, Media Libraries, SGL, SQLite, Android browser, κλπ. Η κάθε εφαρμογή τρέχει στην δική της διεργασία και με το δικό της στιγμιότυπο της εικονικής μηχανής. Η μηχανή αυτή εξαρτάται από το kernel του linux για την διαχείριση της μνήμης και των οδηγών. Το kernel προσφέρει και ένα επίπεδο αφαίρεσης μεταξύ του υλικού και της στοίβας του λογισμικού.

Οι εφαρμογές προσφέρονται στους χρήστες δωρεάν ή με κάποια (μικρή συνήθως) αμοιβή προς όσους τις έχουν αναπτύξει ή εμπορεύονται μέσα από το Android Market – Αγορά Android.<sup>14 15 16</sup>

<sup>14</sup> [http://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)#cite\\_note-i.2Fo\\_2011\\_stats-14](http://en.wikipedia.org/wiki/Android_(operating_system)#cite_note-i.2Fo_2011_stats-14)

<sup>15</sup> <http://www.letsgomobile.org/en/7301/android-smartphone/>

<sup>16</sup> [http://www.engr.sjsu.edu/ges/media/pdf/mse\\_prj\\_rpts/spring2010](http://www.engr.sjsu.edu/ges/media/pdf/mse_prj_rpts/spring2010)

# Κεφάλαιο 3 : Περιβάλλον Ανάπτυξης

## 3.1 Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζεται το περιβάλλον ανάπτυξης που χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής.

Το περιβάλλον αυτό αποτελείται από το eclipse με την προσθήκη του Android SDK (Software Development Kit) για την ανάπτυξη του λογισμικού καθώς και το subversion για να υποστηρίξει την διαχείριση των αλλαγών που γίνοντουσαν κατά την φάση της ανάπτυξης.

Τέλος, στο κεφάλαιο αυτό παρέχονται οδηγίες για την εγκατάσταση και παραμετροποίηση των παραπάνω εργαλείων.

## 3.2 Περιβάλλον ανάπτυξης Eclipse

Το eclipse με την προσθήκη του Android SDK παρέχει την δυνατότητα να αναπτύξουμε εφαρμογές με τη γλώσσα προγραμματισμού java, οι οποίες λειτουργούν σε smartphones που υποστηρίζουν το λογισμικό Android.

Για να ξεκινήσουμε να γράφουμε μια εφαρμογή πάμε στο File → New , επιλέγουμε το Android Project και συμπληρώνουμε αυτά που μας ζητάει. Για να είμαστε όμως σίγουροι ότι η εφαρμογή μας θα τρέξει σε όλες της συσκευές Android πρέπει να επιλέξουμε ένα Android version που να υποστηρίζεται από τις περισσότερες συσκευές. Στην εφαρμογή μου επειδή χρειαζόμαστε να βλέπουμε το χάρτη της Google επιλέγουμε το Google APIs. Όταν τελειώσουμε θα μας εμφανίσει στο Package explorer την δομή της εφαρμογής μας, που αποτελείται από συγκεκριμένους φακέλους :

1. **src** περιέχει το package. Εκεί υπάρχει και η κλάση μας, αλλά και όποια άλλη κλάση χρειαστούμε θα τοποθετηθεί εκεί μέσα

2. **gen** έχει μόνο το `R.java` , το οποίο περιέχει τα `id` από ό,τι χρησιμοποιήσουμε για να αναπτύξουμε το γραφικό περιβάλλον της εφαρμογής. Δεν την τροποποιούμε εμείς, αλλά φτιάχνεται και ενημερώνεται από μόνη της αν υπάρξουν αλλαγές

**To res(resources)** περιέχει:

2.1. καταλόγους `drawable` (`hdpi`,`ldpi`,`mdpi`) όπου αποθηκεύουμε τις εικόνες που θέλουμε να έχει η εφαρμογή, σε 3 διαφορετικά μεγέθη (`hdpi`: 64x64, `mdpi`: 48x48, `ldpi`: 32x32) για να μπορεί να επιλεγεί το κατάλληλο μέγεθος ανάλογα με τον τύπο της συσκευής

2.2. Ο κατάλογος `layout` ορίζει το περιβάλλον του χρήστη της εφαρμογής μας σε μορφή `xml`. Περιέχει από την αρχή ένα `xml` αρχείο που μπορούμε να το δούμε σε 2 διαφορετικές μορφές, γραφική αναπαράσταση του `xml` και στη κλασική του μορφή με τα `tags`. Στη γραφική αναπαράσταση φαίνεται σαν μια οθόνη που στα αριστερά της, έχει τα διάφορα `views` και `layout` που μπορούμε να τοποθετήσουμε πάνω της και στα δεξιά της έχει, σε δένδρική μορφή τα στοιχεία που έχουμε βάλει. Επιλέγοντας ένα στοιχείο, ανοίγει μια καρτέλα με τις διάφορες ιδιότητες που έχει το συγκεκριμένο στοιχείο. Με αυτό τον τρόπο ενημερώνεται το `xml` και δεν χρειάζεται να γράψεις το κάθε `tag` με τις ιδιότητες που θέλεις να έχεις.

Γενικώς όταν θέλουμε σε κάποια ιδιότητα να βάλουμε στοιχεία που βρίσκονται στους καταλόγους `drawable` και στο κατάλογο `values` τότε, πατάμε στο κελί που είναι διπλά στην ιδιότητα και στη άκρη του κελιού εμφανίζεται ένα κουμπί. Το πατάμε και μας ανοίγει ένα παράθυρο, ανοίγοντας τη λίστα που θέλουμε επιλέγουμε το στοιχείο που χρειαζόμαστε για την ιδιότητά μας

Στην εφαρμογή μου έχουμε δημιουργήσει 5 `xml` αρχεία:

A. το `main_screen.xml` περιέχει :

- A.1. ένα γενικό `linear_layout` με ιδιότητες `Background`, το χρώμα που έχουμε ορίσει στο `strings.xml` με όνομα `tag bodyColor`. Το `Orientation` είναι `vertical` ώστε τα αντικείμενα που θα τοποθετήσουμε μέσα στο `layout` να είναι κάθετα, `Layout height/width` είναι `fill_parent` για να καλύπτει όλη την οθόνη του κινητού/ tablet η εφαρμογή.
- A.2. Ένα `linear_layout` με `Background` το `titleColor`, `Orientation horizontal`, `Padding 5dp`. Το `layout` περιέχει ένα `ImageView` με `background` το σήμα της Ε.Α.Β. που το έχουμε φυλάξει στο `drawable` φάκελο και ένα `TextViewHeader` με `Text color` το `titleTextColor`, και `Text` από το αρχείο `String` το `tag String` με το όνομα που θέλουμε να βάλουμε σαν κείμενο.
- A.3. Ένα δευτερο `layout` με `Orientation vertical` και `Gravity center` με 4 `Buttons` με `Drawable left`, την εικόνα που θέλουμε και `Text` το `String tag` που θέλουμε και `Layout width fill_parent`
- B. `map_view` περιέχει εκτός από το κύριο `layout` και ένα `mapView` της Google

```
<com.google.android.maps.MapView  
  
    android:id="@+id/mapView"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent"  
    android:enabled="true"  
    android:clickable="true"  
    android:apiKey="0RaDw7OKA3PehzjBNr2g_YMONnABd4RlHq6kC-g" />
```

Για να μπορέσουμε να δούμε το χάρτη πρέπει να έχουμε ένα API key από την Google. Για να το πάρουμε χρειάζεται να βρούμε το MD5 fingerprint ακολουθώντας τις οδηγίες <http://code.google.com/intl/el-GR/android/add-ons/google-apis/mapkey.html>

- C. Στο `form xml`, το κύριο `layout` και το `layout` με το `imageView` και το `textView` είναι τα ίδια με το `main_screen`. Μετά ακολουθούν μια σειρά από `layout` που χρησιμεύουν για την ομαδοποίηση των `textView` που χρησιμεύουν σαν τίτλος για τα `EditText` και για τα `Spinner(drop-down list)`. Για να φτιαχτεί ένα



Spinner πρέπει να ορίσουμε στο κατάλογο values ένα νέο xml . Ο κώδικας του είναι ο ακόλουθος:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

<string-array name="gender">
  <item>@string/Gender_Male_String</item>
  <item>@string/Gender_Female_String</item>
</string-array>

<string-array name="armed">
  <item>@string/Armed_Yes_String</item>
  <item>@string/Armed_No_String</item>
</string-array>

<string-array name="alertLevel">
  <item>@string/Level_Red_String</item>
  <item>@string/Level_Yellow_String</item>
  <item>@string/Level_Green_String</item>
</string-array>

<string-array name="eventCharact">
  <item>@string/Event_Observation_String</item>
  <item>@string/Event_Sabotage_String</item>
  <item>@string/Event_Theft_String</item>
</string-array>

<string-array name="place">
  <item>@string/Place_InHAI_String</item>
  <item>@string/Place_Perimeter_String</item>
  <item>@string/Place_Outside_String</item>
</string-array>

</resources>
```

Στο αρχείο strings.xml ορίζουμε επίσης και το περιεχόμενο του Spinner με το tag String . Όταν κάνουμε αυτή την διαδικασία πάμε στο form.xml και στις ιδιότητα Entries του Spinner βάζουμε από το αρχείο arrays.xml το όνομα του tag string-array που θέλουμε.

- D. Το camera\_view περιέχει 6 κουμπιά και ανάμεσά τους ένα VideoView το οποίο χρησιμεύει για να μπορούμε να δούμε της κάμερες ενώ τα κουμπιά είναι χρήσιμα για τον τηλεχειρισμό των καμερών από την συσκευή μας (δεν τηλεχειρίζονται όλες οι κάμερες) . Το κάθε κουμπί έχει 3 καταστάσεις (normal, focused και pressed) που η κάθε μια έχει το δικό της εικονίδιο. Τα κουμπιά που υπάρχουν στο Android έχουν ήδη τις καταστάσεις και δεν χρειάζεται να επέμβουμε. Αν όμως θέλουμε να βάλουμε στην εφαρμογή μας δικά μας κουμπιά ή ένα δικό μας background στο κουμπί τότε πρέπει με την βοήθεια ενός σχεδιαστικού εργαλείου να σχεδιάσουμε και τις 3 καταστάσεις και στην συνέχεια να γράψουμε ένα xml αρχείο το οποίο περιέχει ένα selector tag που μας επιτρέπει να παρέχουμε στα κουμπιά τις 3 καταστάσεις, ο συνδυασμός των state\_focused και state\_pressed δημιουργούν τις καταστάσεις:

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_focused="true" android:state_pressed="false"
android:drawable="@drawable/btn_bottom_arrow_selected" />

    <item android:state_focused="false" android:state_pressed="true"
android:drawable="@drawable/btn_bottom_arrow_pressed" />

    <item android:state_focused="false" android:state_pressed="false"
android:drawable="@drawable/btn_bottom_arrow_default" />
/>
</selector>
```

Τα σχεδιασμένα εικονίδια μαζί με τα xml μπαίνουν στο κατάλογο drawable

Επίσης, υπάρχει και ένας άλλος τρόπος να σχεδιάσουμε κουμπιά. Αντί να χρησιμοποιήσουμε κάποιο σχεδιαστικό εργαλείο φτιάχνουμε 3 xml αρχεία για

κάθε μία κατάσταση με τις ιδιότητες που θέλουμε να έχει το κάθε κουμπί και ένα ακόμα xml αρχείο όπως και πριν.

Για να μπορέσουμε τώρα να βάλουμε τις καταστάσεις στο κουμπί το μόνο που έχουμε να κάνουμε είναι στην ιδιότητα background να βάλουμε το xml που ορίζει και τις 3 καταστάσεις.

E. Το axis\_webview xml μας επιτρέπει να δούμε την κάμερα μέσω ενός browser και περιέχει μόνο ένα webView.

Αυτό το xml το αναπτύξαμε επειδή δεν μπορούσαμε να δούμε την AXIS κάμερα μέσω videoView και η AXIS κάμερα που έχουμε στην E.A.B. δεν διαθέτει δυνατότητες τηλεχειρισμού. Το VideoView επίσης καθυστερεί να δείξει την εικόνα για λίγα δευτερόλεπτα από την στιγμή που θα δώσουμε το σήμα να κινηθεί.

**2.3.** Στο κατάλογο anim υπάρχουν 2 xml αρχεία τα οποία χρησιμεύουν για το animation στην αρχή της εφαρμογής. Το animTeam καθορίζει τι θα γίνεται στο animation. Με το alpha tag καθορίζουμε ποσό ορατή/άορατη θα είναι μια εικόνα και ποσό θα κάνει να εμφανιστεί. Το επόμενο tag είναι το scale το οποίο θα μεγεθύνει την εικόνα . Οι δυο πρώτες ιδιότητες του alpha είναι για το πόσο άορατο θα είναι στην αρχή και πόσο ορατό στο τέλος(0 για άορατο, 1 για ορατό), η τρίτη ιδιότητα είναι η διάρκεια που θα πάρει να γίνει ορατό(σε milliseconds). Οι τέσσερις πρώτες ιδιότητες καθορίζουν πόσο θα είναι η αρχική εικόνα και πόσο θα μεγαλώσει στη συνέχεια. Η επόμενη ιδιότητα είναι η διάρκεια (σε milliseconds)αυτού του animation και πόση θα είναι η καθυστέρηση μέχρι να αρχίσει(Pro-android σελ 209-211). Το δεύτερο xml αρχείο διευκρινίζει ότι η κίνηση θα πρέπει να προχωρήσει σε αντιστροφή, και ότι η κίνηση για κάθε είδος πρέπει να αρχίζει με 30% καθυστέρηση σε σχέση με τη συνολική διάρκεια του animation.

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
```

```
android:interpolator="@android:anim/accelerate_interpolator">
```

```
<alpha  
android:fromAlpha="0.0"  
android:toAlpha="1.0"  
android:startOffset="0"  
android:duration="1000" />
```

```
<scale  
android:fromXScale="0.0"  
android:toXScale="0.7"  
android:fromYScale="0.0"  
android:toYScale="0.7"  
android:pivotX="50%"  
android:pivotY="50%"  
android:startOffset="1000"  
android:duration="2500"  
    android:fillBefore="true" />
```

```
<layoutAnimation  
xmlns:android="http://schemas.android.com/apk/res/android"  
android:delay="30%"  
android:animationOrder="reverse"  
    android:animation="@anim/animteam"
```

- 3.** Ο κατάλογος values περιέχει το String.xml που χρησιμεύει για να αποθηκεύουμε της ετικέτες και τα χρώματα που έχουμε στην εφαρμογή μας. Αυτό χρειάζεται ώστε να είναι εύκολη η αλλαγή, αλλιώς θα έπρεπε να πάμε σε κάθε στοιχείο των οθονών και να τροποποιούμε την αντίστοιχη ιδιότητα. Αφού το κάνουμε αυτό γυρνάμε στο layout και συνδέουμε την ιδιότητα που θέλουμε με το συγκεκριμένο tag και αυτό προσφέρεται σε 2 διαφορετικές μορφές. Η μια, έχει μια μπάρα επιλογής τον tag και από κάτω συμπληρώνουμε το όνομα του συγκεκριμένου tag που θα το έχει σαν τιμή. Επίσης ενημερώνει αυτόματα και την κανονική μορφή του xml.

```
<?xml version="1.0" encoding="utf-8"?>  
<resources>
```

```
<string name="Header_String_HAI">HAI surveillance system</string>
<string name="Cam_String">Camera Live View</string>
<string name="Application_String">HAI Forms</string>
<string name="SendMail_String">Mail Application</string>
<string name="Signal_Alert_String">Immediate Alert</string>
```

4. Αν θέλουμε να τρέχει η εφαρμογή μας και στα Ελληνικά πρέπει να προσθέσουμε έναν κατάλογο values-el που θα βάλουμε μέσα ένα String xml που θα περιέχει τις ετικέτες στα Ελληνικά και τα χρώματα που θέλουμε να αλλάζουν ανάλογα με την γλώσσα.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="Header_String_HAI">Σύστημα Επιτήρησης EAB</string>
  <string name="Cam_String">Ζωντανή Εικόνα Κάμερας</string>
  <string name="Application_String">Αναφορά Συμβάντος</string>
  <string name="SendMail_String">Αποστολή e-mail</string>
  <string name="Signal_Alert_String">Άμεσος Συναγερμος</string>
```

5. Κάθε project Android περιλαμβάνει ένα manifest αρχείο, το Android Manifest.xml. Το manifest, σου επιτρέπει να ορίσεις τη δομή και τα μεταδεδομένα της εφαρμογής σου, τα συστατικά ( components )και τις απαιτήσεις. Περιλαμβάνει κόμβους για κάθε ένα από τα συστατικά (Activities,Services,...), χρησιμοποιεί Intent Filters και Permissions, προσδιορίζοντας πως θα αλληλεπιδρούν μεταξύ τους και με άλλες εφαρμογές. Επίσης προσφέρει χαρακτηριστικά (attributes) για να προσδιορίσει τα μεταδεδομένα της εφαρμογής (όπως το εικονίδιο της και το θέμα της (theme). Meier, R. (2010),Professional Android 2 Application Development, Wiley Publishing, Inc

### 3.3 Svn – Subversion <sup>17</sup>

Για την ανάπτυξη της εφαρμογής, επιλέχθηκε η χρήση του subversion, προκειμένου να εξασφαλισθεί η παρακολούθηση των αλλαγών και να διευκολυνθεί η συνεργασία με τους επιβλέποντες και συνεργάτες.

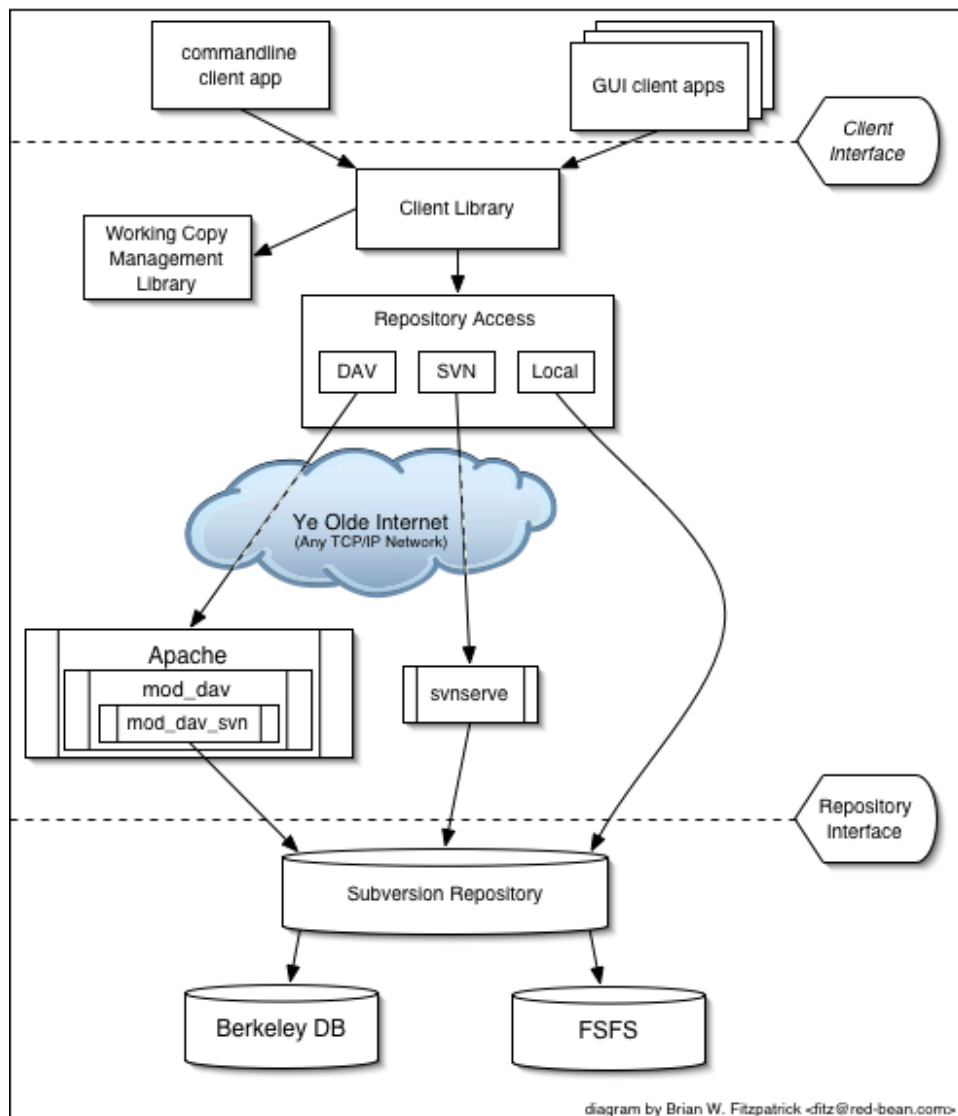
Το Subversion(svn) είναι ένα σύστημα παρακολούθησης αλλαγών ανοικτού κώδικα. Επιτρέπει την διαχείριση αρχείων και αλλαγών που γίνονται στην διάρκεια ανάπτυξης μιας εφαρμογής και την ανάκτηση προηγούμενων εκδόσεων αν χρειασθεί. Με αυτό τον τρόπο, βοηθάει στην συνεργασία πολλών πάνω σε μία εφαρμογή, αφού κάθε φορά δημιουργείται μία ξεχωριστή έκδοση του αρχείου (version) που αποθηκεύεται σε server ενώ όλα τα αρχεία είναι διαθέσιμα στους χρήστες.

Το SVN σε ανοικτό κώδικα, δημιουργήθηκε το 2001, σαν αντικατάσταση του concurrent version system(CVS) από την CollabNet([www.collab.net](http://www.collab.net)) επιλύοντας πολλά από τα προβλήματα που προϋπήρχαν. Το 2010 η CollabNet σε συνεργασία με άλλους που ανέπτυσαν παρόμοια συστήματα ολοκλήρωσε το svn στο Apache SoftwareFoundation (ASF) με αποτέλεσμα να υιοθετηθεί από όλους σχεδόν τους χρήστες που ανέπτυσαν εφαρμογές σε ανοικτό κώδικα.

Η βασική αρχιτεκτονική του svn παρουσιάζεται στο διάγραμμα που ακολουθεί:

---

<sup>17</sup> Version Control with Subversion. 2011 Ben Collins-Sussman



**Εικόνα 13 - Αρχιτεκτονική subversion (svn)**

Στην μία άκρη διατηρούνται όλα τα δεδομένα με διαφορετικές εκδόσεις και στο άλλο άκρο τρέχει το subversion client program που διαχειρίζεται τα αντίγραφα των αρχείων σε κάθε χρήστη/πελάτη.

Τα βασικά συστατικά στοιχεία του svn είναι:

### **Svn**

Το πρόγραμμα που τρέχει στον πελάτη

**Svnversion**

Το πρόγραμμα που αναφέρει την κατάσταση κάθε αρχείου που βρίσκεται σε επεξεργασία

**Svnlook**

Εργαλείο που επιθεωρεί άμεσα την αποθήκη των αρχείων

**Svnadmin**

Εργαλείο για δημιουργία αποθήκης εκδόσεων και διαχείρισης αυτών

**Mod dav svn**

Plug-in για apache http server που επιτρέπει την πρόσβαση στην αποθήκη εκδόσεων μέσα από δίκτυο υπολογιστών

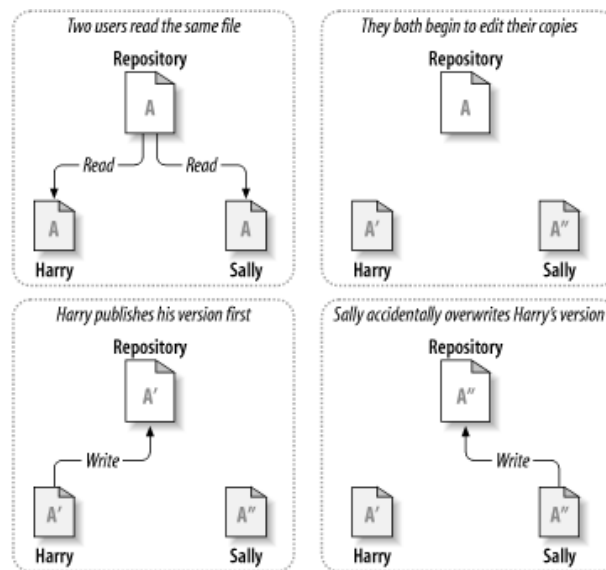
**Svnsync**

Πρόγραμμα που επιτρέπει τον συγχρονισμό των εκδόσεων που βρίσκονται σε κάθε πελάτη.

Η βασική λειτουργία του svn είναι:

***Τι θέλουμε να αποφύγουμε***





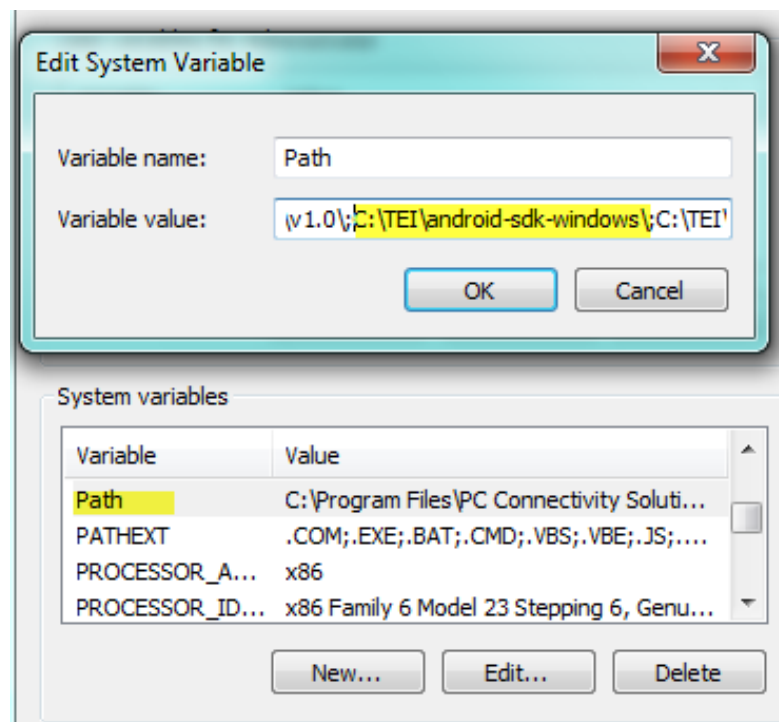
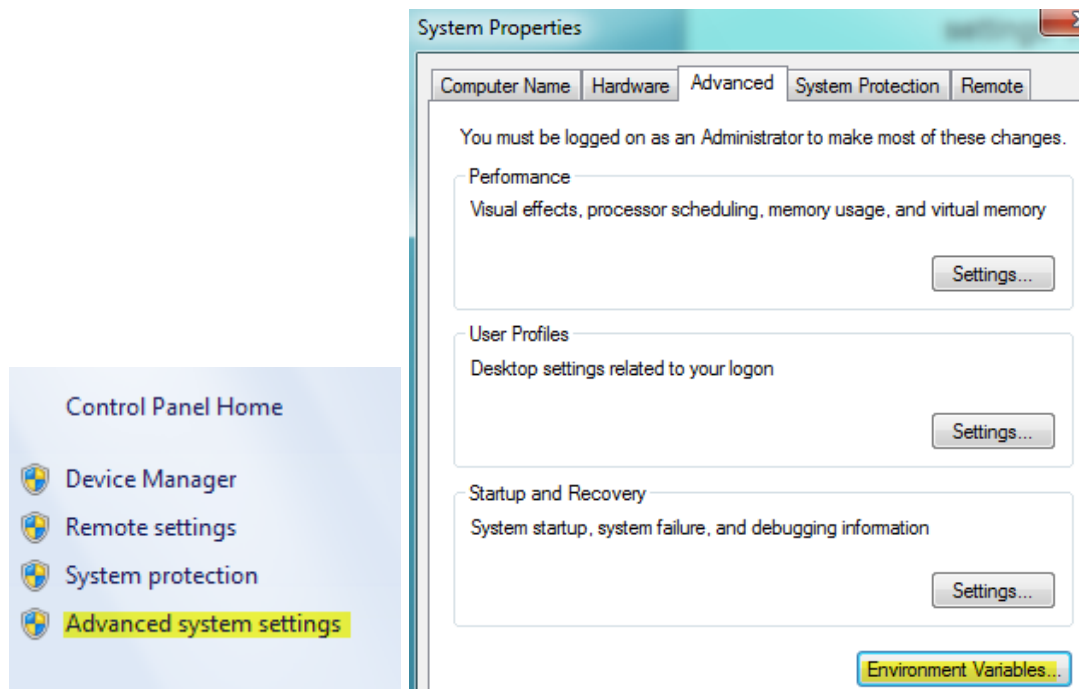
**Εικόνα 14 - Το πρόβλημα των συνεργατικών αναπτύξεων**

### ***Πως το επιτυγχάνουμε***

Δημιουργούνται αντίγραφα εργασίας στο περιβάλλον του πελάτη. Αφού δοκιμασθεί ο κώδικας τότε μπορεί ο πελάτης να δημοσιεύσει το αρχείο ώστε αυτό, στην έκδοση που θα πάρει, να είναι διαθέσιμο σε άλλους. Προσφέρεται και η επιλογή να συμπυκτούν εκδόσεις του ίδιου αρχείου με κατάλληλη αρίθμηση της έκδοσης.

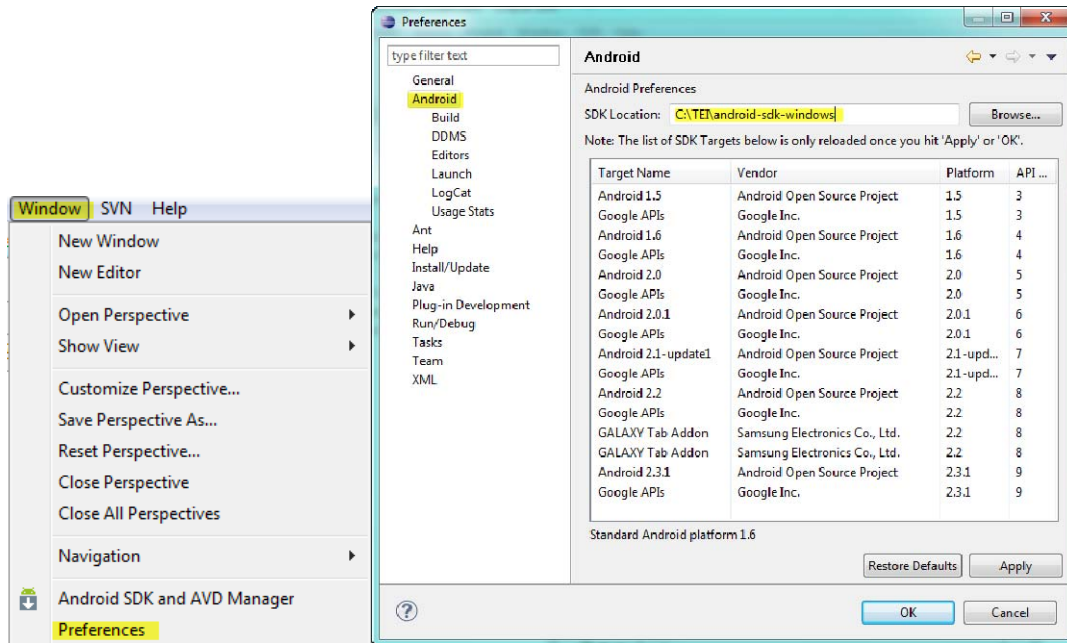
## **3.4 Οδηγίες εγκατάστασης και χρήσης των εργαλείων ανάπτυξης sdk eclipse svn**

1. Από το site <http://developer.android.com/sdk/index.html> κατεβάζουμε το Android sdk ανάλογα με το λειτουργικό μας.
2. Το αποσυμπιέζουμε και τρέχουμε το sdk setup. Όταν τελειώσει πρέπει να το προσθέσουμε στο path(δεξί κλικ στο computer → properties, επιλέγουμε το advanced system settings → environment variables και στο system variables είναι το path. Με διπλό κλικ και προσθέτουμε το path του sdk)

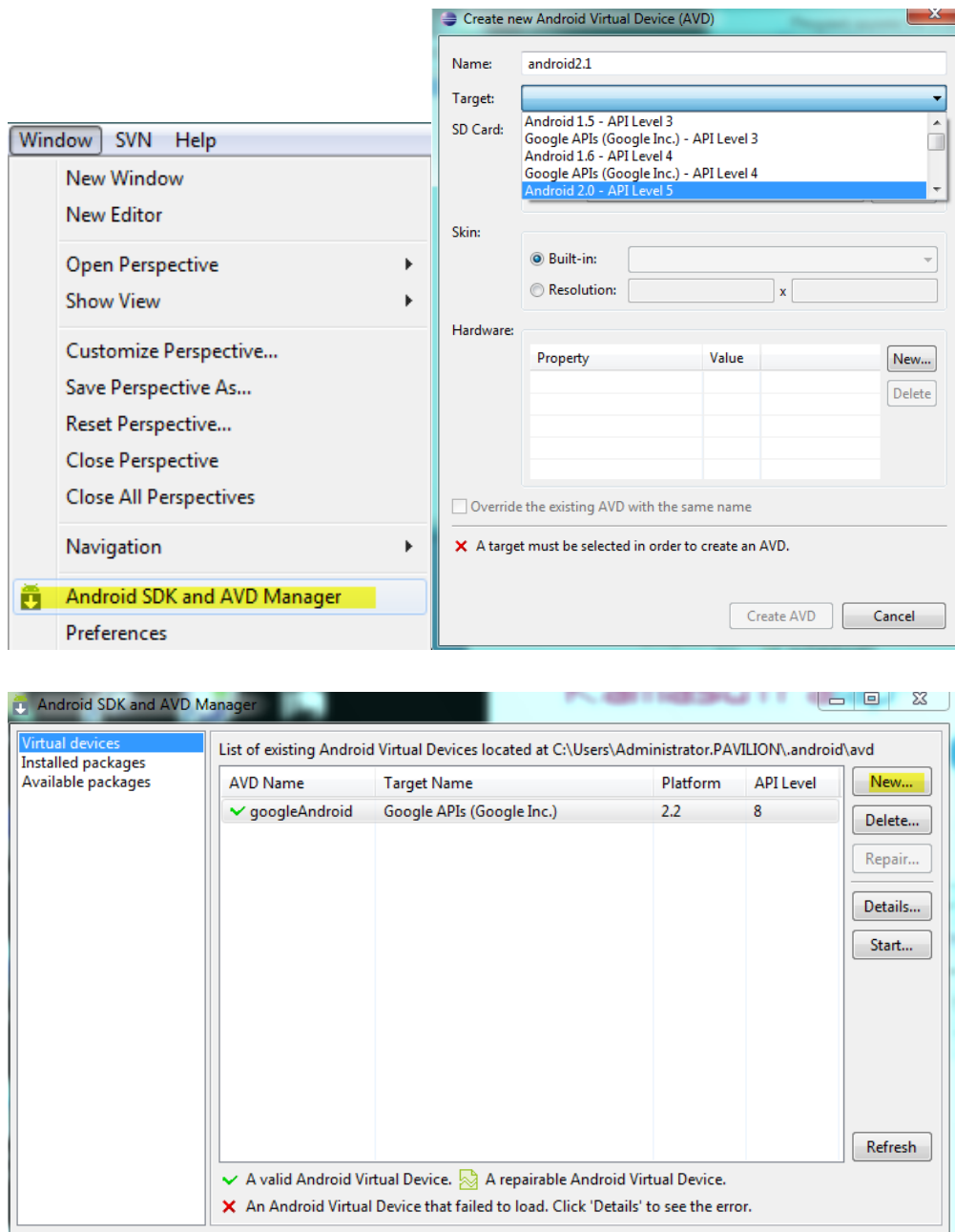


3. Ανοίγουμε το eclipse πάμε στο help → install new software ,add → όνομα και αντιγράφουμε την διεύθυνση του ADT plugin: <https://dl-ssl.google.com/android/eclipse/> , επιλέγουμε τα αρχεία και μετά next.

Windows → preferences, android και βάζουμε το path του android.



Για να μπορέσουμε να τρέξουμε τις εφαρμογές μας πρέπει να έχουμε έναν emulator. Τον οποίο θα πρέπει να τον ορίσουμε εμείς : Windows → android sdk and AVD Manager στο παράθυρο που θα ανοίξει επιλέγουμε από την λίστα το virtual Devices και μετά New και συμπληρώνουμε τις ρυθμίσεις που θέλουμε να έχει ο emulator. Όπως το όνομα του emulator και με το target επιλέγουμε σε ποια έκδοση θα αναφέρεται:



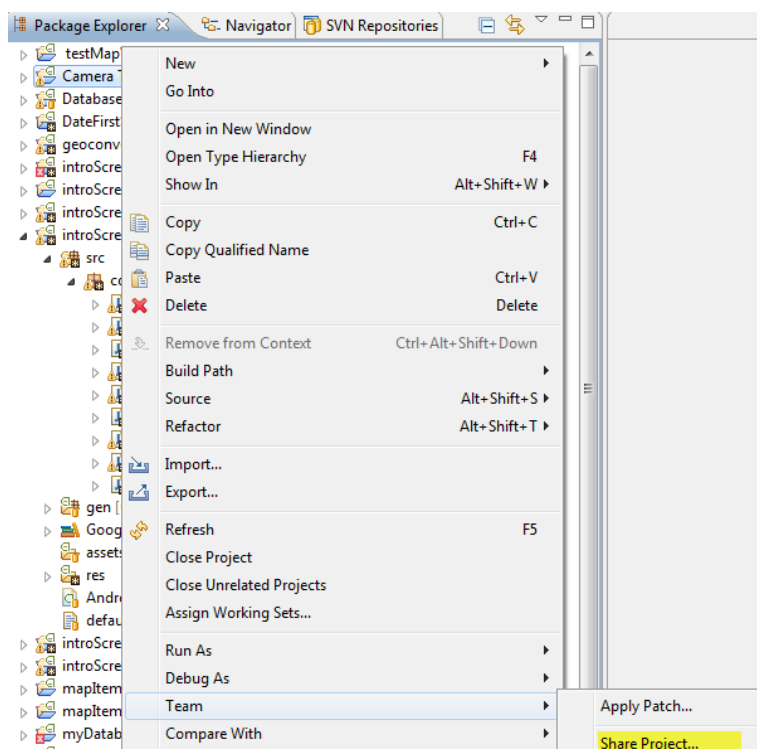
Εγκατάσταση SVN :

1. Σαν πρώτο βήμα πρέπει να κατεβάσουμε μια έκδοση subclipse.
2. Στη συνέχεια χρειαζόμαστε ένα κοινό χώρο που να φυλάγονται τα projects όπως ένας server.

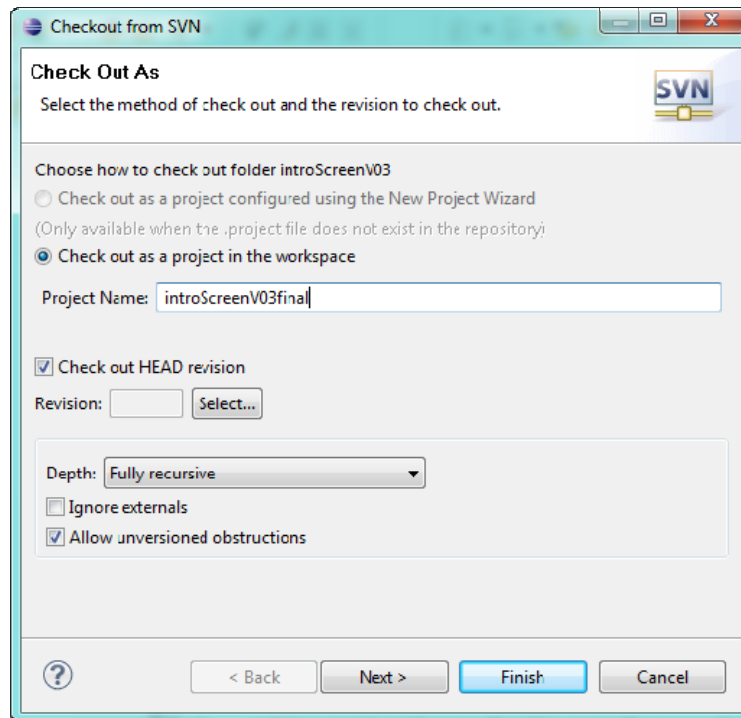
3. Πάμε στο help->install new software μετά στο add...και εγκαθίσταται το plug-in στο eclipse.

Χρήση SVN:

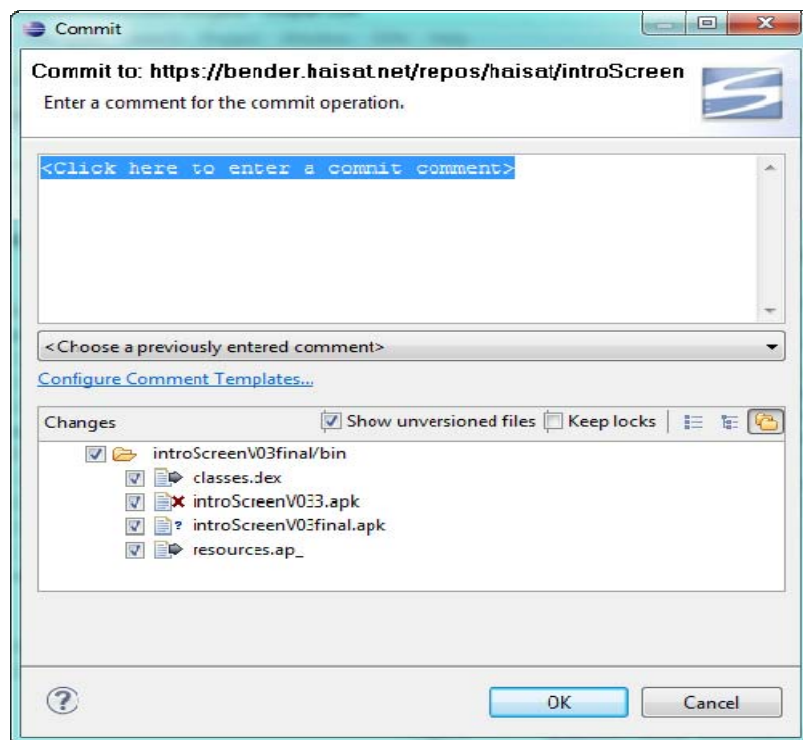
4. Αφού ολοκληρωθεί η εγκατάσταση επιλέγουμε από τη καρτέλα package explorer το project που θέλουμε να διαμοιράσουμε και με δεξί κλικ στο project και στην συνέχεια team->Share Project όπου το ανεβάζουμε στο server.



5. **Κατέβασμα του project από το server** - Πάμε στην καρτέλα svn repository και με δεξί κλικ στο project και στην συνέχεια checkout.



6. Αφού το έχουμε δουλέψει το project και έχουμε κάνει κάποιες αλλαγές τότε μπορούμε να ξαναεβάσουμε το project στον server. Στην καρτέλα package και με δεξί κλικ στο project επιλέγουμε Team->commit όπου μας εμφανίζει μια καρτέλα που μας επιτρέπει να προσθέσουμε σχόλια με τις αλλαγές που έχουμε κάνει.



# Κεφάλαιο 4 : Η εφαρμογή

## 4.1 Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζονται οι προδιαγραφές του συστήματος με την μορφή της περίπτωσης χρήσης. Επιπλέον, παρουσιάζεται η σχεδίαση και ανάπτυξη της εφαρμογής καθώς και οι οδηγίες εγκατάστασης και χρήσης. Τέλος, παρουσιάζονται οι δοκιμές που έγιναν.

## 4.2 Προδιαγραφές

### 4.2.1. Περίπτωση Χρήσης ΠΧ: Επιτήρηση χώρου εταιρίας.

**Κύριος συμμετέχων:** Φύλακας

**Εμπλεκόμενοι και Ενδιαφέροντα:**

- Φύλακας: Επιτήρηση κλειστού και ανοιχτού χώρου με περιπολία, καταγραφή συμβάντων και αναφορά.
- Υπεύθυνος ασφαλείας: Η γρηγορότερη δυνατή ενημέρωση όταν υπάρχει συμβάν, για ενέργειες.

**Προϋποθέσεις:** Ο φύλακας να έχει μαζί του tablet συσκευή ή smartphone η οποία θα είναι συνδεδεμένη στο δίκτυο της εταιρίας

**Εγγύηση Επιτυχίας:** Εντοπισμός κινδύνου και ειδοποίηση της ασφαλείας

**Κύριο Σενάριο Επιτυχίας (ή Βασική Ροή):**

1. Ο φύλακας εκτελεί περιπολία στο χώρο.
2. Ο φύλακας παρακολουθεί το χώρο μέσω ενός χάρτη που υπάρχει στην συσκευή.



3. Ο φύλακας βλέπει τους συναδέλφους του μέσω του στίγματος της συσκευής τους.
4. Ο φύλακας βλέπει τη θέση των καμερών στο χάρτη.
5. Ο φύλακας επιλέγοντας μια κάμερα μπορεί να βλέπει τι καταγράφει η κάμερα εκείνη τη στιγμή και σε πραγματικό χρόνο και να την ελέγχει.
6. Ο φύλακας ακόμα μπορεί να συμπληρώνει μια φόρμα με τα στοιχεία του ατόμου/αντικειμένου/συμβάντος που εντόπισε.
7. Το σύστημα καταγράφει τα στοιχεία για μελλοντική χρήση. (ΒΔ)
8. Ο φύλακας μπορεί να στέλνει sms/mail σε προκαθορισμένο αριθμό και με προκαθορισμένο κείμενο. Ο φύλακας μπορεί να προσθέσει και κείμενο. Ο υπεύθυνος ασφαλείας λαμβάνει το sms/mail και εκτελεί αναλόγως.
9. Ο Φύλακας επίσης για άμεση ενημέρωση μπορεί να στείλει ένα σύντομο sms σε προκαθορισμένο αριθμό και επιλέγοντας το τι θα περιέχει το μήνυμα από μια λίστα με προκαθορισμένα γεγονότα.

#### **Επεκτάσεις (ή Εναλλακτικές Ροές):**

Ο φύλακας μπορεί να στέλνει αναφορά ότι όλα είναι εντάξει (OK)

#### **Ειδικές απαιτήσεις:**

- Η συσκευή θα διαθέτει οθόνη αφής για την διαχείριση του γραφικού περιβάλλοντος της εφαρμογής (GUI).
- Υπάρχει εξυπηρετητής (server) που θα έχει web interface και θα καταγράφει το συμβάντα.
- Η συσκευή θα τρέχει Android OS.
- Η εφαρμογή θα μπορεί να τρέξει σε συσκευές διαφορετικού τύπου (tablet,laptop,Smartphone).
- Η συσκευή απεικονίζει το χάρτη.
- Η συσκευή διαθέτη συνδεσιμότητα μέσω Wi-Fi εναλλακτικά GPRS.
- Το δίκτυο Wi-Fi της εταιρίας καλύπτει την περιοχή περιπολίας.

**Συχνότητα εμφάνισης:** Ο φύλακας μπορεί να παρακολουθεί τις κάμερες συχνά ενώ τα συμβάντα συμβαίνουν σπάνια.

## 4.2.2. Επιλογή ANDROID έναντι άλλων λειτουργικών συστημάτων

Από την ανάλυση που παρουσιάστηκε στο γενικό μέρος της πτυχιακής εργασίας προκύπτουν οι παρακάτω λόγοι επιλογής:

1. Μεριδίο αγοράς – το Android κυριαρχεί έναντι των άλλων συστημάτων
2. Διαθεσιμότητα πολλών τύπων και χαμηλού κόστους συσκευών – η ανοικτή αρχιτεκτονική επιτρέπει την χρήση του λειτουργικού συστήματος σε συσκευές πολλών κατασκευαστών.
3. Πλατφόρμα ανοικτού λογισμικού. Διαθεσιμότητα πηγαίου κώδικα (AOSP) <http://source.android.com/about/index.html>
4. Ισχυρό υπόβαθρο – είναι μέλος του open handset alliance .
5. Χρήση γλώσσας JAVA – πολλοί προγραμματιστές είναι σήμερα εξοικειωμένοι με την JAVA που έχει την δυνατότητα να τρέχει σε κάθε συσκευή Android.
6. Ο δημιουργός της εφαρμογής μπορεί να διανείμει την εφαρμογή του χωρίς περιορισμούς.
7. Δωρεάν εργαλεία ανάπτυξης.

## 4.3 Σχεδίαση

### 4.3.1 Γενική δομή

Για την ικανοποίηση των προδιαγραφών, η εφαρμογή του κινητού κόμβου διαχειρίζεται αισθητήρες δεδομένων (κάμερες) μέσω IP δικτύου (τυπικά WiFi). Επιπλέον, διαχειρίζεται πληροφορίες (θέση καμερών και φρουρού) σε χάρτη (με χρήση Google maps). Η θέση του φρουρού βρίσκεται αυτόματα μέσω GPS. Διασυνδέεται ακόμα με τοπική βάση δεδομένων και επικοινωνεί με άλλους χρήστες μέσω SMS και email.

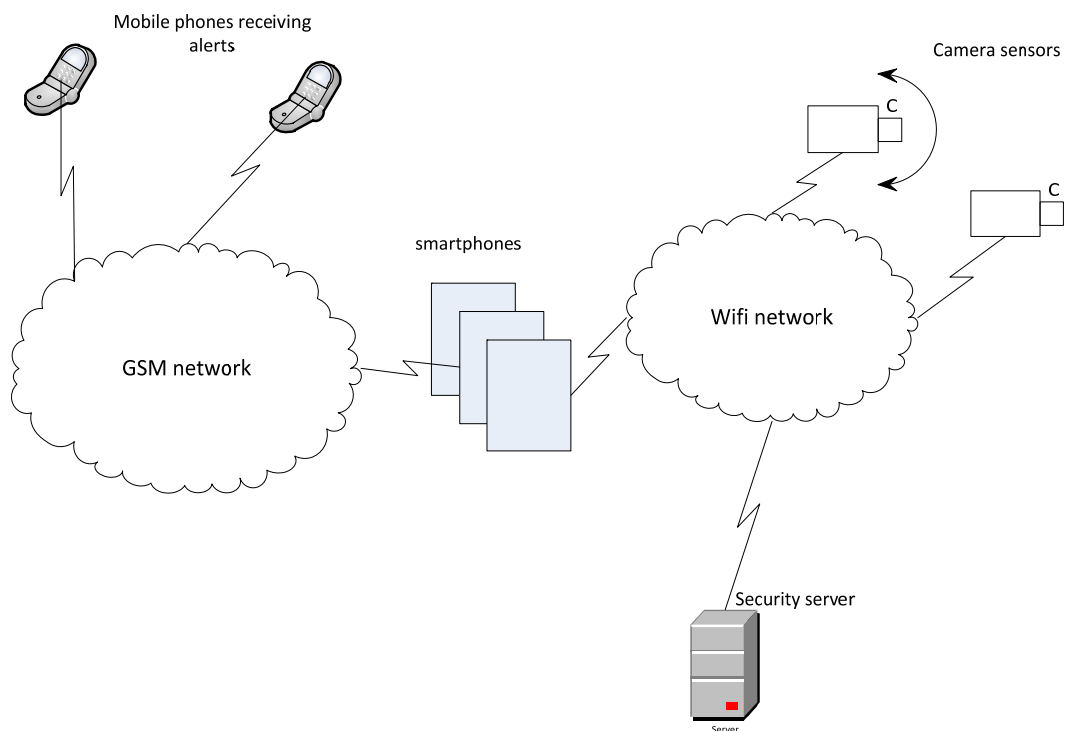


Εικόνα 15 - Βασική αρχιτεκτονική δομή εφαρμογής

## 4.3.2 Αρχιτεκτονική υλικού

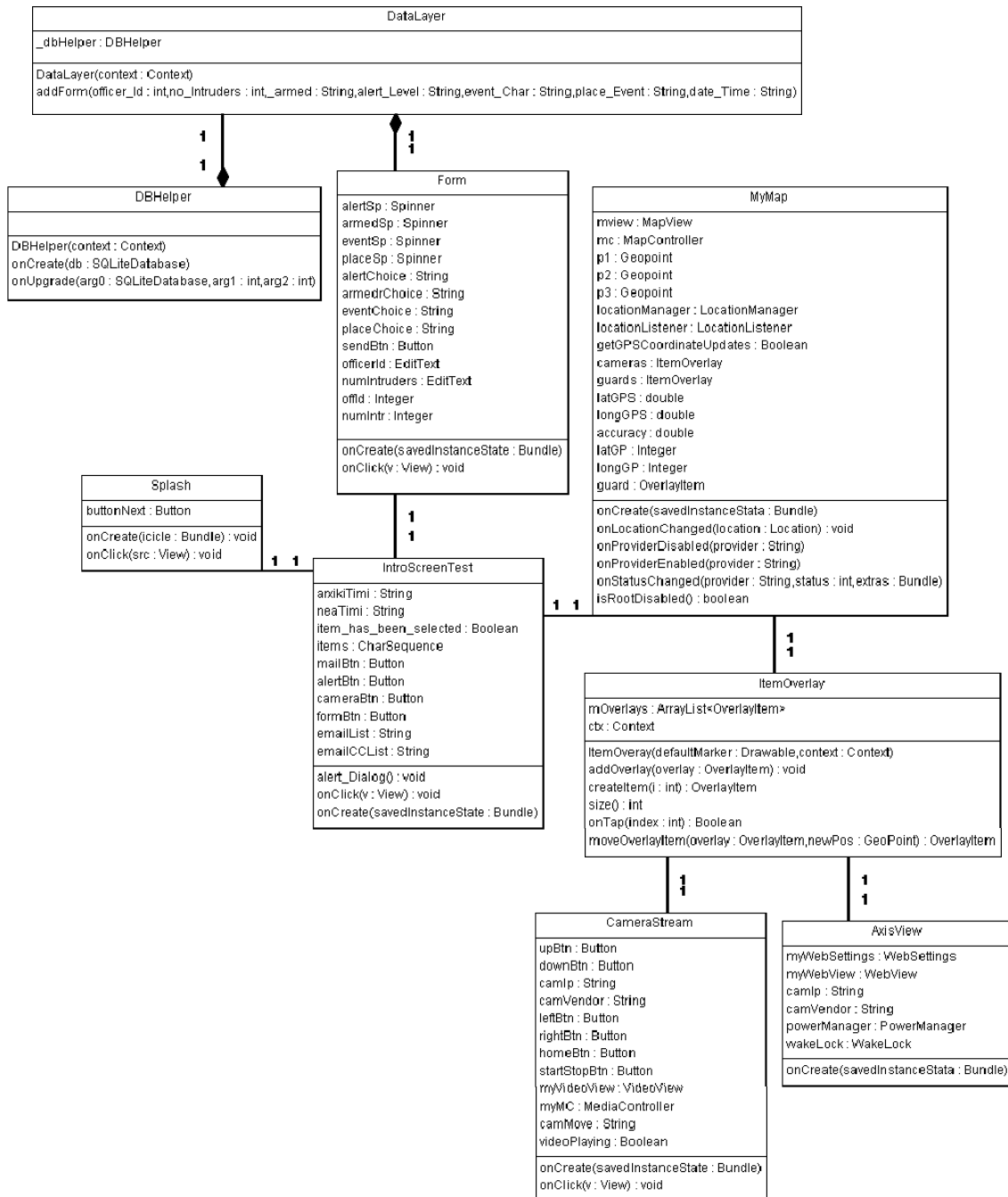
Μία σειρά από έξυπνα τηλέφωνα μέσω ενός ασύρματου δικτύου wifi επικοινωνούν, λαμβάνουν εικόνα και τηλεχειρίζονται κάμερες που επιτηρούν κάποιο χώρο. Στοιχεία μηνυμάτων αποθηκεύονται σε έναν εξυπηρετητή. Τα τηλέφωνα συνδέονται σε ένα GSM δίκτυο το οποίο χρησιμοποιούν για να στέλνουν μηνύματα στους υπεύθυνους που έχουν ορισθεί.

### HARDWARE ARCHITECTURE



Εικόνα 16 - Αρχιτεκτονική υλικού εφαρμογής

### 4.3.3 Βασικές κλάσεις



Εικόνα 17 - Διάγραμμα κλάσεων

Η κλάση Splash μας εμφανίζει ένα animation με ένα κουμπί, το οποίο όταν πατηθεί μας συνδέει με την κλάση introScreenTest μέσω intent.

Η introScreenTest είναι η κύρια κλάση, που περιέχει τέσσερα κουμπιά που συνδέονται με τις κλάσεις MyMap, Form και με έτοιμες εφαρμογές όπως το email . Η εφαρμογή μας εμφανίζει ένα παράθυρο για να στέλνουμε email, χωρίς να χρειάζεται να φτιάξουμε από τη αρχή μια εφαρμογή για e-mail .

Η MyMap εμφανίζει το χάρτη της google σε μια συγκεκριμένη περιοχή και τοποθετεί σε αυτόν τα εικονίδια με τις κάμερες και το εικονίδιο του φρουρού. Επίσης συνδέεται με σύνθεση με την κλάση ItemOverlay.

Η κλάση ItemOverlay δημιουργεί τις μεθόδους για τα αντικείμενα που υπάρχουν στην MyMap και επίσης συνδέεται με τις κλάσεις CameraStream και AxisView.

Η CameraStream εμφανίζει το παράθυρο για τον έλεγχο των καμερών.

Η AxisView εμφανίζει ένα παράθυρο ενός browser όπου εκεί θα εμφανιστεί η εικόνα της κάμερας.

Η Form περιέχει πεδία για συμπλήρωση και drop-down list . Με το πάτημα του κουμπιού συνδέεται με την κλάση DataLayer

Η κλάση DataLayer γεμίζει έναν πίνακα βάσης δεδομένων με τα στοιχεία της κλάσης που έχει δημιουργηθεί στην κλάση DBHelper

## 4.4 Υλοποίηση

Κάθε κλάση περιέχει την μέθοδο onCreate() η οποία καλείται όταν ξεκινάει η εφαρμογή. Οι αρχικοποιήσεις πρέπει να δηλώνονται μέσα στην μέθοδο ως εξής: setContentView(R.layout.main\_screen), το οποίο ορίζει ποιο layout θα χρησιμοποιηθεί στην εφαρμογή και τα διάφορα findViewById(R.id.Button01) που βρίσκει το id των views, τα οποία έχουν δηλωθεί στο συγκεκριμένο layout.

Η IntroScreenTest είναι η κύρια κλάση μας που εμφανίζει την κεντρική οθόνη, ορίζει τα κουμπιά και βρίσκει το id τους για να τα συνδέσει με το γεγονός (event) OnClick μέσω ενός switch που ανάλογα ποιο κουμπί έχει πατηθεί κάνει και την αντίστοιχη ενέργεια.

Επίσης, υπάρχει και μια μέθοδος (`alert_dialog()`) που χρησιμεύει για να ορίσουμε ένα `dialog box`, όπως τον τίτλο του, τι εικονίδιο θα περιέχει, τον αριθμό των κουμπιών και τι θα κάνει το καθένα, καθώς επίσης και τι θα υπάρχει στο σώμα του `dialog`. Στη περίπτωση μας θέλουμε στο σώμα να έχουμε μια ομάδα από `radio buttons` με κάποιες τιμές (`fire`, `reinforcement`, `Intruder`, `false alarm`). Πατώντας μια από τις επιλογές η τιμή της θα αποθηκεύεται σε μία μεταβλητή για να μπορούμε να την χρησιμοποιήσουμε στη συνέχεια της μεθόδου. Υπάρχουν 3 είδη κουμπιών που μπορεί να περιέχει ένα `dialog box` (`positive`, `negative`, `neutral`). Στην εφαρμογή μου χρησιμοποίησα το `Positive` για να ορίσω το κουμπί `OK` που περιέχει ένα αντικείμενο της κλάσης `intent` βάζοντας του της κατάλληλες παραμέτρους. Ειδικότερα, η πρώτη παράμετρος είναι για να του πούμε την ενέργεια που θα κάνει (Το `ACTION_SENDTO` είναι όταν θέλουμε το `intent` να καλέσει την αρχική εφαρμογή για αποστολή `sms`) και η δεύτερη παράμετρος αφορά τον αριθμό του παραλήπτη (`implicit intent`). Το `Neutral` είναι το `cancel`. Επίσης υπάρχει η δυνατότητα στο σώμα του `dialog` να έχει `radio buttons` ή `checkboxes`. Το `setSingleChoiceItem()` παίρνει τρεις παραμέτρους: Η πρώτη παράμετρος είναι ένας πίνακας με τις τιμές που θα έχουν τα `radio Buttons`, η δεύτερη παράμετρος προσδιορίζει αν θα είναι κάποια τιμή προεπιλεγμένη. Το `-1` δηλώνει καμία τιμή. Επειδή οι τιμές είναι σε πίνακα, αν θέλουμε να είναι κάποια τιμή προεπιλεγμένη, τότε θα ξεκινήσουμε από το `0` για την πρώτη τιμή, `1` για την δεύτερη και `n-1` για την τελευταία.

Για να μπορέσουμε να καλέσουμε μια άλλη κλάση, η οποία θα εμφανίσει το `layout` της στην οθόνη, έχουμε το αντικείμενο `intent` με 2 παραμέτρους, η πρώτη είναι η τρέχουσα εφαρμογή και η δεύτερη είναι η `Activity` κλάση που θέλουμε να εκτελέσουμε (`explicit intent`). Φυσικά για να μην μας βγάλει σφάλμα πρέπει να πάμε να ορίσουμε την κλάση και στο `Manifest`: 

```
<activity android:name=".myMapV2"></activity>
```

Η κλάση **MyMapV2** ορίζει γεωγραφικά σημεία. Κάποια από αυτά αναπαριστούν τις κάμερες σε συγκεκριμένες θέσεις μέσω συντεταγμένων `latitude` και `longitude` και κάποια άλλα τον φύλακα που βρίσκεται σε κίνηση όταν υπάρχει σήμα `GPS`.

```
p1 = new GeoPoint(38326062, 23588784);
OverlayItem overlayitem = new OverlayItem(p1, "CAMERAS", "Camera1");

guard = new OverlayItem(new GeoPoint(38327686, 23586059), "HAI",
"Guard1");
```

Βρίσκοντας το id του mapView μας δίνει κάποια επιπλέον στοιχεία που θα έχει ο χάρτης:

```
mview = (MapView) findViewById(R.id.mapView);

mview.setBuiltInZoomControls(true); προσθέτει στο χάρτη κουμπια για zoom
in και zoom out
mview.setSatellite(true); δορυφορική μορφή του χάρτη
mc = mview.getController();
mc.setZoom(18); πόσο zoom θα έχει αρχικά ο χάρτης
mc.animateTo(p1); σε ποιο σημείο θα είναι κεντραρισμένος
```

Επιπλέον επειδή θέλουμε οι κάμερες και ο φύλακας να εμφανίζονται διαφορετικά πάνω στο χάρτη

```
Drawable drawable = this.getResources().getDrawable(R.drawable.camera);
Drawable drawable2 =this.getResources().getDrawable(R.drawable.security);
cameras = new ItemOverlay(drawable, this);
guards = new ItemOverlay(drawable2, this);
```

και προσθέτουμε το εικονίδιο στο αντίστοιχο overlayItem και μετά σε μια λίστα που περιέχει όλα τα items.

Επειδή θέλουμε πάνω στον χάρτη να φαίνεται η θέση της συσκευής (άρα και του φύλακα)και η θέση να αλλάζει καθώς κινούμαστε, θα χρειαστούμε να πάρουμε τα στίγμα της συσκευής με GPS (σε latitude και σε longitude). Για να μπορέσουμε να τα χρησιμοποιήσουμε πρέπει να τα μετατρέψουμε από double σε integer. Παρακάτω φαίνεται ο τρόπος μετατροπής:

```
latGPS = location.getLatitude();//σε double morfh px 38.326062, 23.588784
longGPS = location.getLongitude();

latGPS = (location.getLatitude())*1E06;//metatroph se integer gia na mpei
```



```
longGPS = location.getLongitude() * 1E06; //sto geopoint
latGP = (int) latGPS;
longGP = (int) longGPS;
```

για να φαίνεται η αλλαγή της θέσης στο χάρτη πρέπει να καλέσουμε την παρακάτω μέθοδο:

```
guard = guards.moveOverlayItem(guard, new GeoPoint(latGP, longGP));
mview.invalidate();
```

η οποία διαγράφει την προηγούμενη θέση της συσκευής και τοποθετεί το εικονίδιο στην θέση που έχει λάβει από το GPS.

Η κλάση **ItemOverlay** προσθέτει τα εικονίδια που εμφανίζονται πάνω στο χάρτη και επιτρέπει το πάτημα των εικονιδίων. Στην εφαρμογή θέλουμε, πατώντας ένα εικονίδιο της κάμερας, να μπορούμε να παρακολουθούμε την αντίστοιχη κάμερα. Για να μπορέσουμε να πατήσουμε τα εικονίδια πρέπει να υλοποιήσουμε την μέθοδο **protected boolean** onTap(int index). Το όρισμα index είναι η θέση που έχει μπει το αντικείμενο σε μια λίστα (0 το πρώτο αντικείμενο ,1 το δεύτερο). Με την βοήθεια του switch καλούμε ένα intent και του προσθέτουμε κάποια επιπλέον δεδομένα όπως το IP των καμερών ,το μοντέλο τους και τον τρόπο που επιτρέπει κάθε κάμερα να την τηλεχειριστούμε.

```
camInt=new Intent (ctx, CameraStream.class);
camInt.putExtra("CAM_IP", "192.168.2.85");
camInt.putExtra("CAM_VENDOR", "VIVOTEK1");
camInt.putExtra("CAM_MOVE", "camctrl.cgi");
```

Εκτός από τις κάμερες έχουμε βάλει και τον φρουρό να απεικονίζεται πάνω στο χάρτη. Όταν πατήσουμε το εικονίδιο του φρουρού τότε, επειδή έχουμε βάλει και αυτόν στην θέση μηδέν στην λίστα, δεν θέλουμε όταν τον επιλέξουμε να μας ανοίξει το παράθυρο με την κάμερα. Για αυτό τον λόγο, τοποθετούμε ένα if στο πρώτο case του switch όπου ελέγχουμε τον τίτλο που έχουμε δώσει στο Intent, ώστε να μπορέσουμε να διαχωρίσουμε τις λειτουργίες.

Επίσης υλοποιεί την μέθοδο **public** OverlayItem moveOverlayItem(OverlayItem overlay, GeoPoint newPos) η οποία φτιάχνει ένα αντικείμενο OverlayItem, αφαιρεί το αντικείμενο που θα της δώσουμε σαν παράμετρο όταν την καλέσουμε και προσθέτει το καινούργιο αντικείμενο με σκοπό να εξαφανίζεται η προηγούμενη θέση του αντικειμένου πάνω στον χάρτη και να εμφανίζεται το αντικείμενο στην νέα του θέση.

Μέσω της κλάσης **CameraStream** μπορούμε να χειριστούμε τις κάμερες. Περιέχει τις μεταβλητές των 6 κουμπιών και το VideoView που θα χρησιμεύσουν για να πάρουμε τα id των στοιχείων του xml camera\_view και επιπλέον έναν controller , ο οποίος εμφανίζει τα κουμπιά play/stop στην μπάρα του VideoView. Για να μπορέσουμε να τηλεχειριστούμε τις κάμερες χρειάζεται να χρησιμοποιήσουμε τα επιπλέον στοιχεία του intent που είχαμε προσθέσει στην μέθοδο onTap της κλάσης ItemOverlay.

```
Bundle extras=getIntent().getExtras();
    if(extras!=null){
        camIp=extras.getString("CAM_IP");
        camVendor=extras.getString("CAM_VENDOR");
        camMove=extras.getString("CAM_MOVE");
    }
```

Στην συνέχεια στην μέθοδο onClick()πρώτα ελέγχουμε με try και catch αν έχουν φτιαχτεί σωστά τα URL της κίνησης των καμερών και αφού τα ελέγξουμε δημιουργούμε μια σύνδεση για κάθε ένα URL.Το switch μας βοηθάει να προσδιορίσουμε ανάλογα με το id του κουμπιού που πατήθηκε ποια περίπτωση θα εκτελεστεί. Όταν πατήσουμε το κουμπί που έχει ορισθεί για να αρχίσουμε να βλέπουμε την κάμερα, ελέγχουμε μια Boolean μεταβλητή για να μπορούμε να σταματάμε το video και να αλλάζουμε το κείμενο του κουμπιού. Στη συνέχεια φορτώνουμε το URI της AXIS κάμερας, ξεκινάμε το video, αλλάζουμε την τιμή της μεταβλητής από false σε true και το κείμενο του κουμπιού σε STOP.

```

case R.id.BtnStartStop:
    if (videoPlaying==false) {
        Uri
videoUri=Uri.parse ("rtsp://" + camIp + "/live.sdp");
        myVideoView.setVideoURI (videoUri);
        myVideoView.start ();
        videoPlaying=true;
        startStopBtn.setText (R.string.stopCam);
    }
    else{
        myVideoView.stopPlayback ();
        videoPlaying=false;
        startStopBtn.setText (R.string.startCam);
    }
    break;

```

Οι περιπτώσεις που αφορούν την κίνηση είναι ακριβώς οι ίδιες με μοναδική αλλαγή το URL που περνάμε κάθε φορά.

Η Κλάση **AxisView** χρησιμεύει για να μπορούμε να δούμε την κάμερα τύπου Axis. Καταρχήν μπορούμε να βάλουμε ένα progress bar για να μπορούμε να βλέπουμε ποσό κάνει να φορτώσει το URL της κάμερας. Επίσης, επειδή θα φορτώνεται σε webview, για να μην «κολαέι», πρέπει να πούμε στην οθόνη να παραμένει ανοιχτή. Στην συνέχεια του δίνουμε την εντολή ο προσανατολισμός (orientation) της οθόνης να είναι οριζόντιος.

```

this.getWindow().requestFeature(Window.FEATURE_PROGRESS);

setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);

powerManager = (PowerManager)
getSystemService(Context.POWER_SERVICE);

wakeLock = powerManager.newWakeLock(
    PowerManager.SCREEN_BRIGHT_WAKE_LOCK,
    "geoconverter app wake lock");

wakeLock.acquire();

```

Θα χρειαστούμε και κάποια settings να ρυθμίσουμε για να μπορούμε να δούμε το video και επιπλέον ενεργοποίηση του flash και δυνατότητα υποστήριξης zoom. Επίσης, παρατηρήσαμε με τις δοκιμές ότι φόρτωνε πολύ η μνήμη cache για αυτό το λόγο προσθέσαμε

```
myWebSettings.setCacheMode(WebSettings.LOAD_NO_CACHE);
```

Για να μπορέσουμε να δούμε το URL πρέπει να το φορτώσουμε στο webView

```
myWebView.loadUrl("http://" + camIp + "/axis-cgi/mjpg/video.swf");
```

Στην συνέχεια ορίσαμε το client και τον τρόπο που θα κάνει update το progress bar:

```
final Activity MyActivity = this;
myWebView.setWebChromeClient(new WebChromeClient() {
    public void onProgressChanged(WebView view, int
progress) {
        // Make the bar disappear after URL is loaded,
and changes
        // string to Loading...
        MyActivity.setTitle("Loading...");
        MyActivity.setProgress(progress * 100); // Make
the bar

        // disappear after URL

        // is loaded

        // Return the app name after finish loading
        if (progress == 100)
            MyActivity.setTitle(R.string.app_name);
    }
});
```

Η κλάση **DBHelper** κληρονομεί από την κλάση `SQLiteOpenHelper`, η οποία μας βοηθάει να διαχειριστούμε την δημιουργία της βάσης μας. Ο δομητής ορίζει μέσω του δομητή της `SQLiteOpenHelper` το όνομα της βάσης.

Η μέθοδος `onCreate` δημιουργεί τον πίνακα και ορίζει τα πεδία που θα περιέχει ως εξής:

```
public void onCreate(SQLiteDatabase db) {
    String createSql="CREATE TABLE formData "+(eventId INTEGER
PRIMARY KEY AUTOINCREMENT NOT NULL , "+
    "officerId INTEGER, "+noIntruders INTEGER, "+armed TEXT,
"+"alertLevel TEXT, "+eventChar TEXT, "+placeEvent TEXT, "+
    "timeStamp DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
"+"latitude REAL , "+longitude REAL);";
    db.execSQL(createSql);
}
```

Η επόμενη κλάση είναι η **DataLayer**. Υλοποιεί μια μέθοδο την `addform`, η οποία, όταν την καλέσουμε με τις παραμέτρους που της έχουμε, θα μας εισάγει τα δεδομένα στον πίνακα μας ως εξής:

```
public void addForm(int officer_Id,int no_Intruders,String _armed,String
alert_Level
                ,String event_Char,String place_Event,String
date_Time,double _latitude,double _longitude ){
    SQLiteDatabase db=_dbHelper.getWritableDatabase();
    try{
        ContentValues values =new ContentValues();

        values.put("officerId",officer_Id);
        values.put("noIntruders",no_Intruders);
        values.put("armed",_armed);
        values.put("alertLevel",alert_Level);
        values.put("eventChar",event_Char);
        values.put("placeEvent",place_Event);
        values.put("latitude",_latitude );
        values.put("longitude",_longitude);
        values.put("timeStamp",date_Time);

        db.insert("formData",null,values);

    }
}
```

```
        finally{
            if (db!=null)
                db.close();
        }
    }
```

Ο σκοπός της κλάσης **Form** είναι να γεμίσει τον πίνακα της βάσης δεδομένων με στοιχεία ανάλογα με τις επιλογές της φόρμας και πατώντας το κουμπί αποστολή φόρμας. Για να μπορέσουμε να το υλοποιήσουμε πρέπει να πάρουμε τα id από τα views του form.xml με σκοπό να χρησιμοποιήσουμε τις τιμές που θα εισάγουμε ή θα επιλέξουμε από τα spinner. Επιπλέον θέλουμε να αποθηκεύσουμε στην βάση και την ώρα που συμπληρώσαμε την φόρμα .

```
alertChoice=alertSp.getSelectedItem().toString();
armedChoice=armedSp.getSelectedItem().toString();
eventChoice=eventSp.getSelectedItem().toString();
placeChoice=placeSp.getSelectedItem().toString();
offId=new Integer(officerId.getText().toString());
numIntr=new Integer(numIntruders.getText().toString());

SimpleDateFormat dateFormat=new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
Date date=new Date();
```

Επειδή η μέθοδος για να εισάγουμε τα στοιχεία στον πίνακα είναι σε άλλη κλάση , πρέπει να ορίσουμε ένα αντικείμενο και τότε να καλέσουμε την μέθοδο addForm, βάζοντας ως παραμέτρους τις αντίστοιχες μεταβλητές , όπως φαίνεται στο παρακάτω κομμάτι κώδικα:

```
DataLayer dl=new DataLayer(getBaseContext());
dl.addForm(offId,numIntr,armedChoice,alertChoice
,eventChoice,placeChoice,dateFormat.format(date),44.0,70.0);
```

Η κλάση **Splash** ορίζει μια μεταβλητή buttonNext τύπου Button, ότι δεν θα εμφανίζεται ο τίτλος της εφαρμογής και το layout της εφαρμογής θα καταλαμβάνει

όλη την οθόνη του smartphone. Στην συνέχεια συνδέει την κλάση με το συγκεκριμένο layout, βρίσκει το id του κουμπιού και με την μέθοδο onClick συνδέει μέσω του αντικειμένου Intent την κλάση με την κλάση IntroScreenTest.

## 4.5 Περιγραφή λειτουργίας της εφαρμογής

### 4.5.1 Οδηγίες εγκατάστασης εφαρμογής

Αν θέλουμε να δοκιμάσουμε την εφαρμογή μας σε συσκευές που υποστηρίζουν το λειτουργικό Android:

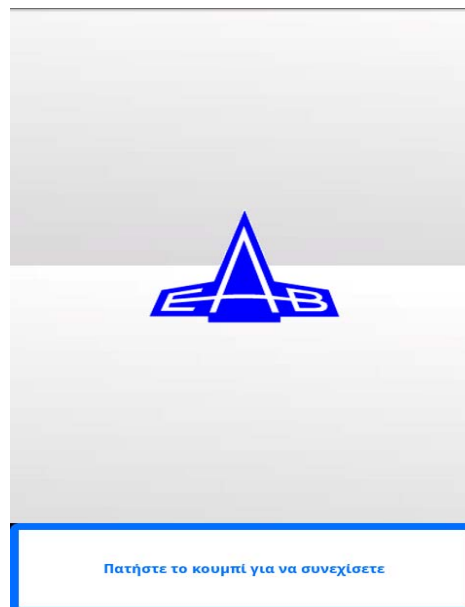
1. Κατεβάζουμε από την εφαρμογή Market που υπάρχει ήδη στη συσκευή ένα flashplayer,
2. Συνδέουμε την συσκευή στον υπολογιστή με το καλώδιο usb που περιέχεται στη συσκευασία της και
3. Τέλος πρέπει να βρούμε drivers ανάλογα με την συσκευή.

Για να εγκαταστήσουμε τους drivers, πάμε στο control panel και ανοίγουμε το device manager. Θα μας εμφανίσει το μοντέλο, οπότε κάνουμε δεξί κλικ και μετά επιλέγουμε Update Driver Software ... στη συνέχεια browse ... . Για συσκευές της Google οι drivers βρίσκονται στον κατάλογο android-sdk windows\usb\_driver ή απλά τους εγκαθιστούμε. Επιπλέον τις συσκευές της google πρέπει να τις ρυθμίσουμε για usb debugging, το οποίο βρίσκεται στις ρυθμίσεις του κινητού -> applications -> development και το επιλέγουμε. Χωρίς αυτήν την επιλογή η συσκευή δεν θα μπορέσει να αναγνωριστεί από το eclipse όταν πάμε να φορτώσουμε οποιαδήποτε εφαρμογή.

Για να τρέξει το eclipse για εφαρμογές Android πρέπει να εγκαταστήσουμε την τελευταία έκδοση του java SE/EE

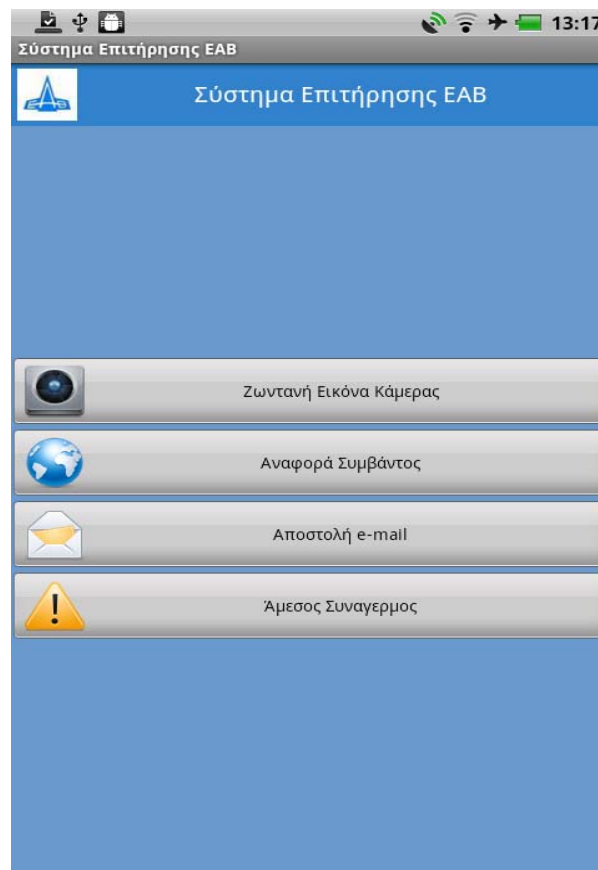
## 4.5.2. Χρήση εφαρμογής

Για να ξεκινήσουμε την εφαρμογή μας πάμε στο μενού με τις εφαρμογές και πατάμε το εικονίδιο της. Επίσης κρατώντας πατημένο το εικονίδιο μας το εμφανίζει στην κεντρική οθόνη της συσκευής, που μπορούμε να το τοποθετήσουμε σε όποια από τις οθόνες θέλουμε και να δημιουργήσουμε ένα εικονίδιο συντόμευσης (shortcut) της εφαρμογής. Μόλις επιλέξουμε την εφαρμογή (HAI surveillance system ή Σύστημα επιτήρησης E.A.B.) ξεκινάει ένα animation με το σήμα της E.A.B. και ένα κουμπί για να συνεχίσεις με την κύρια οθόνη της εφαρμογής. Πατώντας το κουμπί μας εμφανίζει την κύρια οθόνη της εφαρμογής.

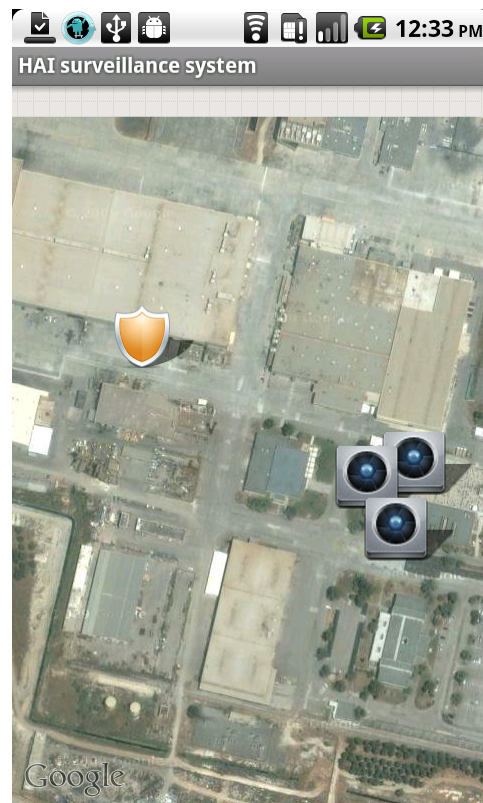


Η κύρια οθόνη αποτελείται από μια επικεφαλίδα με το σήμα της E.A.B. και τον τίτλο της εφαρμογής. Από κάτω ακριβώς βρίσκονται τέσσερα κουμπιά (Ζωντανή Εικόνα Κάμερας, Αναφορά Συμβάντος, Αποστολή e-mail, Άμεσος Συναγερμός). Πατώντας το κάθε κουμπί μας οδηγεί στην αντίστοιχη οθόνη.

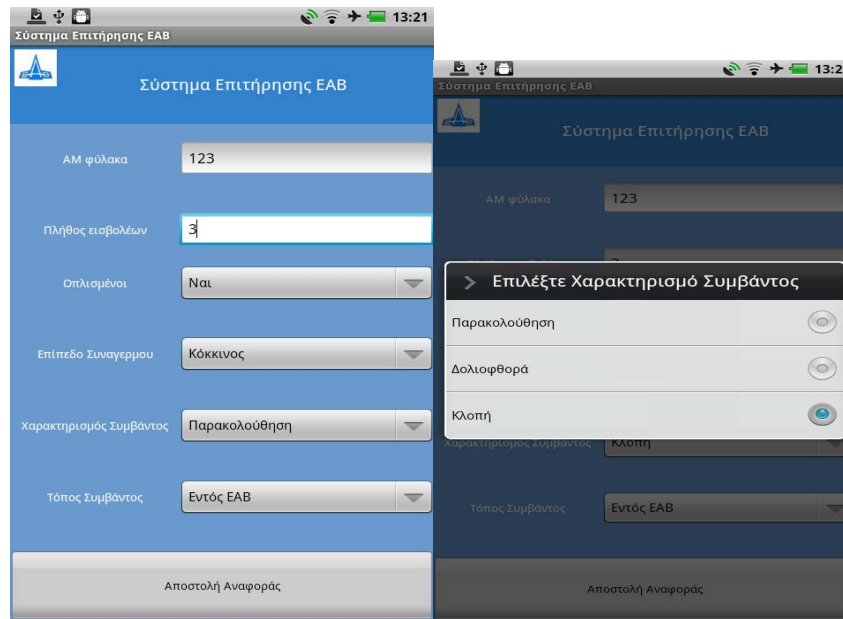




Το κουμπί Ζωντανή Εικόνα Κάμερας μας εμφανίζει τον χάρτη της Ε.Α.Β. μέσω του Google maps και πάνω σε αυτόν τοποθετημένες τις κάμερες και έναν φρουρό. Πατώντας πάνω στο εικονίδιο της κάμερας μας μεταφέρει σε μια άλλη οθόνη, που μπορούμε να δούμε και να τηλεχειριστούμε τις κάμερες. Για να ξεκινήσουμε να βλέπουμε την κάμερα πατάμε το κουμπί Start, με τα κουμπιά βελάκια μετακινούμε την κάμερα πάνω- κάτω –αριστερά –δεξιά. Με το κουμπί σπίτι η κάμερα επανέρχεται στην αρχική της θέση. Ενώ το εικονίδιο του φρουρού μόλις βρει σήμα το GPS θα μετακινηθεί από την αρχική του θέση στην καινούργια θέση και όσο μετακινείται με την συσκευή θα αλλάζει και η θέση του σημείου.

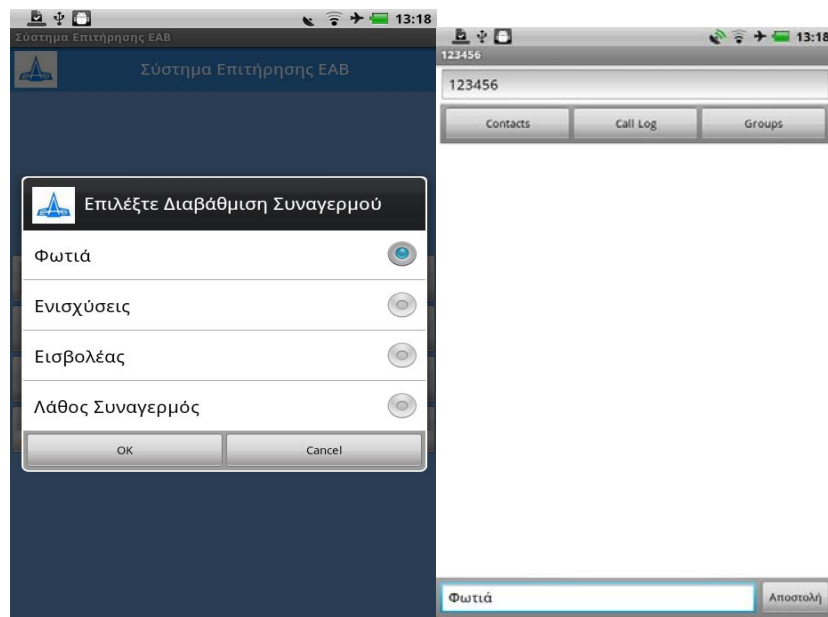


Η Αναφορά Συμβάντος περιέχει μια φόρμα για επιλογή και συμπλήρωση διαφόρων πεδίων. Τα δύο πρώτα πεδία είναι για να συμπληρώσουμε τον ΑΜ του φύλακα και τον αριθμό των εισβολών. Τα υπόλοιπα πεδία, πατώντας τα, μας εμφανίζουν ένα παράθυρο για να διαλέξουμε την περίπτωση που θέλουμε. Τέλος, πατώντας το κουμπί Αποστολή Αναφοράς, αποθηκεύει τα στοιχεία σε μια βάση δεδομένων που βρίσκεται αποθηκευμένη στο κινητό.



Με την Αποστολή e-mail μπορούμε να συντάξουμε e-mail και να το στείλουμε σε συγκεκριμένο παραλήπτη και κοινοποίηση.

Με τον Άμεσο Συναγερμό μπορούμε να στείλουμε ένα SMS με συγκεκριμένο κείμενο που έχουμε επιλέξει από ένα dialog box σε κάποιο προεπιλεγμένο νούμερο κινητού τηλεφώνου.



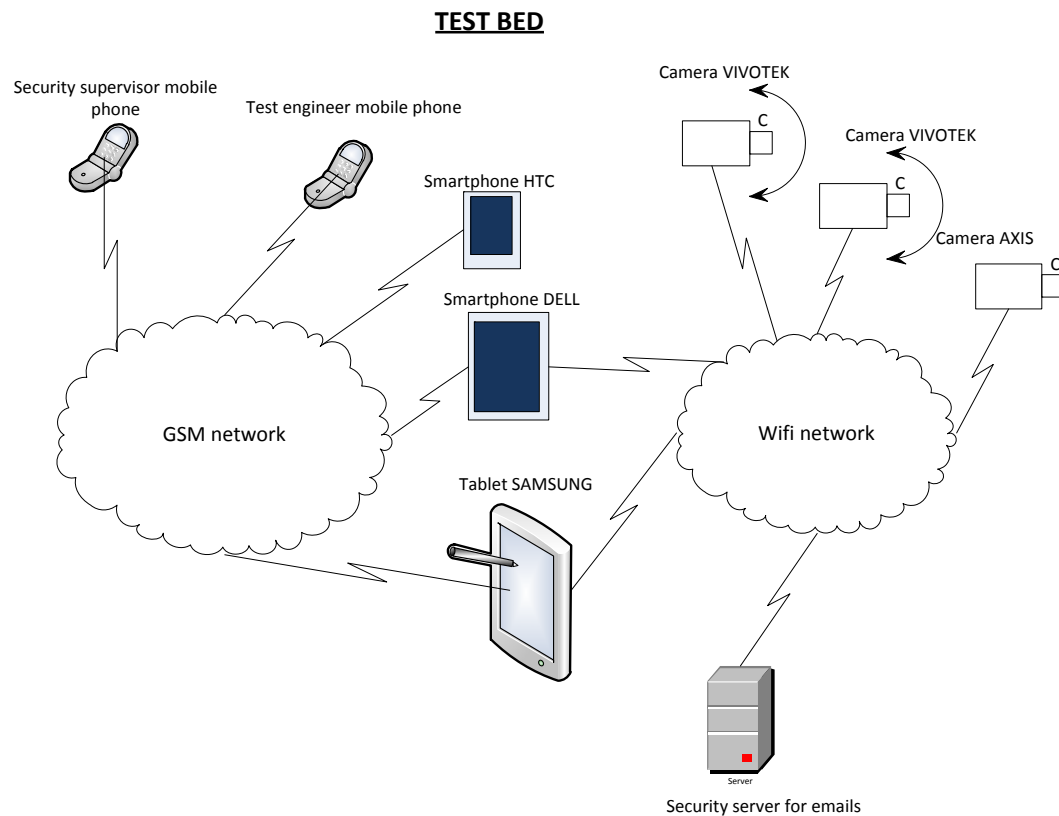
## 4.6 Δοκιμές.

Η εφαρμογή δοκιμάστηκε σε τρία διαφορετικά Smartphones που υποστηρίζουν λογισμικό Android, όπου το καθένα έχει διαφορετικό μέγεθος οθόνης και διαφορετική ανάλυση οθόνης. Το Google Nexus One με μέγεθος οθόνης 3.7 ίντσες και ανάλυση 800 x 480 , Dell Streak 5 ίντσες 800 x 400, Samsung Galaxy tab με 7 ίντσες και ανάλυση 1024 x 600.

Τρεις διαφορετικές IP ,PTZ (Pan Tilt Zoom) κάμερες από τις οποίες οι 2 ήταν VIVOTEK PT7137,PZ7152 και υποστηρίζουν τηλεκίνηση και μια AXIS χωρίς τηλεκίνηση.

Η εφαρμογή υποστηρίζει ασύρματο δίκτυο Wi-Fi για την παρακολούθηση και τον τηλεχειρισμό των καμερών και την αποστολή e-mail καθώς και GSM για την αποστολή SMS. Η εφαρμογή αναπτύχθηκε σε συνεργασία με επιβλέποντα μηχανικό της E.A.B..

Στο διάγραμμα που ακολουθεί παρουσιάζεται η διάταξη που αναπτύχθηκε και με την οποία έγιναν οι δοκιμές.



**Εικόνα 18 - Διάταξη δοκιμών**

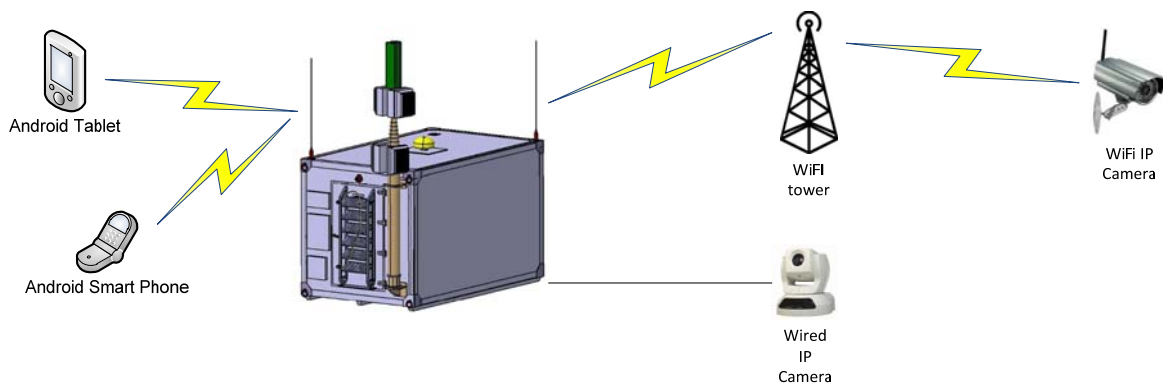
Σε πρώτο στάδιο, το σύστημα παρουσιάστηκε στο προσωπικό ασφάλειας της Ε.Α.Β., όπου σχεδιάστηκαν οι οθόνες και προσδιορίστηκε η επιχειρησιακή λογική σύμφωνα με τις απαιτήσεις των χρηστών.

Η εφαρμογή δοκιμάστηκε σε διάφορα στάδια, πρώτα από την τεχνική ομάδα της Ε.Α.Β. ώστε να ενσωματωθούν τεχνικές βελτιώσεις, στην συνέχεια δόθηκε σε προσωπικό της ασφάλειας που δοκίμασαν διάφορες συσκευές και έδωσαν παρατηρήσεις κυρίως επιχειρησιακού χαρακτήρα.

Τον Ιανουάριο του 2011 επισκέφθηκαν την Ε.Α.Β. εκπρόσωποι του Υπουργείου Εθνικής Άμυνας από τα διάφορα επιτελεία (ΓΕΕΘΑ, ΓΕΣ, ΓΕΝ, ΓΕΑ) όπου παρουσιάστηκε μεταξύ άλλων και η συγκεκριμένη εφαρμογή. Στην παρουσίαση που έκανα συνδεθήκαμε με κάμερες που επιτηρούν την περίφραξη της Ε.Α.Β. και οι αξιωματικοί είχαν την ευκαιρία να χειριστούν την εφαρμογή.

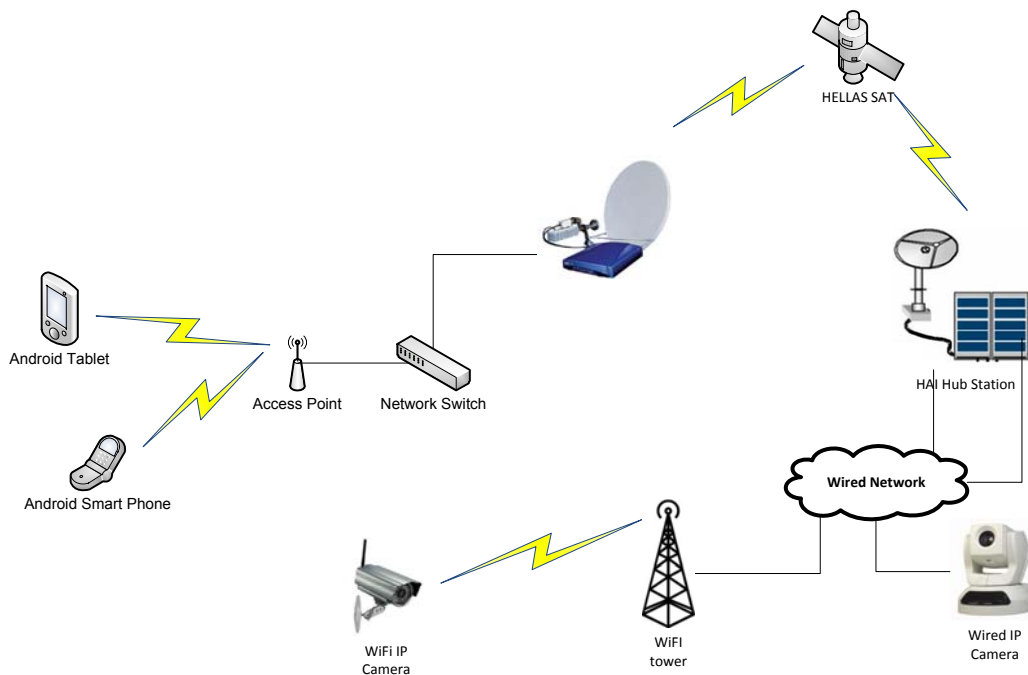
Το ίδιο διάστημα επισκέφθηκαν την Ε.Α.Β. εκπρόσωποι από εταιρίες με τις οποίες συνεργάζεται η Ε.Α.Β., όπως η THALES, στους οποίους επίσης παρουσίασα την εφαρμογή και είχαν την ευκαιρία να την χειριστούν μόνοι τους .

Στις αρχές Φεβρουαρίου 2011 επισκέφθηκε την Ε.Α.Β. κλιμάκιο της Ελληνικής Αστυνομίας και του Κέντρου Μελετών της Αστυνομίας. Η εφαρμογή παρουσιάσθηκε σε συνδυασμό με το έργο TALOS που εκτελεί η Ε.Α.Β.. Το TALOS ασχολείται με την επιτήρηση συνόρων σε μεγάλη έκταση και συγκεκριμένα η Ε.Α.Β. αναπτύσσει το σύστημα διοίκησης και ελέγχου σε μεταφερόμενο κλωβό. Ο κλωβός συνδέεται με κάμερες και μη επανδρωμένα οχήματα και επιτηρεί μία περιοχή συνόρων. Ο κλωβός έχει και δικές του ασύρματες κάμερες για να προστατεύεται από παρείσδυση. Οι κάμερες αυτές συνδέθηκαν με το σύστημα που ανέπτυξα και επέτρεπαν στους χειριστές να απομακρύνονται από τον κλωβό έχοντας την επιτήρηση του χώρου στο έξυπνο τηλέφωνό τους.



**Εικόνα 19 - Δοκιμές σε σύστημα επιτήρησης συνόρων**

Τέλος, τον Απρίλιο του 2011 η εφαρμογή παρουσιάστηκε στην έκθεση ασφαλείας EXPOSEC 2011 στο Ζάππειο Μέγαρο όπου συνδέθηκε μέσω δορυφόρου με κάμερα στην περίμετρο της Ε.Α.Β. στην Τανάγρα και επέτρεπε την επιτήρηση και τηλεχειρισμό της κάμερας από το Ζάππειο. Την δορυφορική σύνδεση έκαναν οι μηχανικοί της Ε.Α.Β..



**Εικόνα 20 - Δοκιμές με δορυφορικό δίκτυο επικοινωνιών**

Τα σχόλια και οι παρατηρήσεις από τις προαναφερθείσες επιδείξεις και δοκιμές αξιολογήθηκαν και ενσωματώθηκαν στην εφαρμογή (στο μέτρο του δυνατού) αλλά και θα αποτελέσουν προτάσεις για μελλοντική αναβάθμιση.

Σε γενικές γραμμές, τα σχόλια που πήρα ήταν πάρα πολύ θετικά και η εφαρμογή θεωρήθηκε χρήσιμη και κυρίως ότι προσφέρει πολλές δυνατότητες εκμεταλλευόμενη μια σειρά σύγχρονων τεχνολογικών χαρακτηριστικών .

# **Κεφάλαιο 5 : Συμπεράσματα-Νέες**

## **Επεκτάσεις**

### **5.1 Συμπεράσματα**

Από την πτυχιακή αυτή εργασία καθώς και από την μελέτη των αποτελεσμάτων μετά την εφαρμογή της στο περιβάλλον της E.A.B. προκύπτει ότι η χρήση των έξυπνων τηλεφώνων που θα βρίσκονται σε ευρεία εφαρμογή σε μερικά χρόνια θα μπορεί να στραφεί και σε προσφορά υπηρεσιών σε επαγγελματικούς χρήστες.

Η χρήση των συσκευών αυτών θα βρίσκεται στο επίκεντρο της ανάπτυξης ολοκληρωμένων ηλεκτρονικών εφαρμογών επιτρέποντας μεγαλύτερη φορητότητα και άμεση διασύνδεση και ολοκλήρωση με το περιβάλλον.

Προσωπικά, κατά την διάρκεια της πρακτικής μου άσκησης, είχα την ευκαιρία να εργασθώ σε έναν χώρο που συνδυάζει την έρευνα με την ανάπτυξη προϊόντων. Συνεργάστηκα με έμπειρους μηχανικούς και χρησιμοποίησα επαγγελματικά εργαλεία για ανάπτυξη εφαρμογών, ακλουθώντας τις σχεδιαστικές πειθαρχίες της ομάδας. Μου δόθηκε η ευκαιρία να συνδέσω την εφαρμογή που ανέπτυξα με ώριμα συστήματα της E.A.B.. Τέλος, είχα την ικανοποίηση να παρουσιάσω την εργασία μου σε στελέχη της E.A.B. καθώς και σε εκπροσώπους Υπουργείων και εταιριών που επισκέφτηκαν τον χώρο των εργαστηρίων της E.A.B..

### **5.2 Επεκτάσεις**

Πέρα από τα πλαίσια αυτής της πτυχιακής, η εφαρμογή θα μπορούσε να επεκταθεί στο μέλλον με διάφορους τρόπους:

- Να απεικονίζει και τη θέση άλλων χρηστών στην περιοχή.



- Να συγχρονίζει την τοπική βάση δεδομένων της συσκευής με την κεντρική και αυτή των άλλων χρηστών. Με αυτόν τον τρόπο θα μπορεί να ενημερώνεται και να ενημερώνει για νέες πληροφορίες.
- Να επιτρέπει τη διασύνδεση ακόμα κι αν ο χρήστης είναι εκτός εμβέλειας Wi-Fi δικτύου, με τεχνικές αναμετάδοσης και ad-hoc δικτύωσης μέσω άλλων χρηστών.
- Να δοκιμαστεί με κινητά δίκτυα και πρωτόκολλα επόμενης γενιάς (WiMAX, LTE, IEEE 802.11p) για κάλυψη μεγάλων περιοχών και κινητών χρηστών με μεγάλη ταχύτητα.
- Να υποστηρίζει την αυτόματη μετάπτωση από το ένα δίκτυο δεδομένων σε άλλο, όταν η απόδοση/κάλυψη του πρώτου δεν είναι καλή. Απαιτεί και προσαρμογή της αναμετάδοσης του βίντεο στις ταχύτητες του δεύτερου δικτύου.
- Να δομηθεί ώστε να δέχεται εύκολα επεκτάσεις plugin για προσαρμογή σε ιδιαίτερα συστήματα και συσκευές ασφαλείας.
- Να προσαρμοστεί ώστε να τρέχει αποδοτικά σε συσκευές multicore. Να αξιοποιήσει τις υπολογιστικές δυνατότητες των επερχόμενων multicore συσκευών, ώστε να κάνει τοπικά επεξεργασία του βίντεο και να παράγει σχετικές ειδοποιήσεις και δυνατότητες (π.χ. ανίχνευση κίνησης, ανίχνευση και παρακολούθηση αντικειμένων).
- Να υλοποιηθεί και σε άλλες ανταγωνιστικές πλατφόρμες πλην Android (π.χ. iPhone, Windows Phone 7). Να υλοποιηθεί και σε περιβάλλον PC, ώστε να μπορεί ο χρήστης να την εκτελεί και σε φορητό υπολογιστή (laptop).

## Βιβλιογραφία

### Παραπομπές

1. <http://spectrum.ieee.org/at-work/innovation/top-11-technologies-of-the-decade>
2. <http://spectrum.ieee.org/telecom/wireless/smartphones-the-pocketable-pc>
3. Comprehensive Analysis of SmartPhone OS Capabilities and Performance. Michael Wei plus others, 20 April 2009
4. The State Of Mobile Apps. Nielsen, September 2010
5. Diversity in Smartphone Usage. Falaki, Mahajan, Kandula, ACM June 2010
6. Cross-Platform Smartphone Apps Still Difficult. Daniel Dern, IEEE Spectrum, June 2010
7. <http://spectrum.ieee.org/geek-life/tools-toys/crossplatform-smartphone-apps-still-difficult>
8. <http://spectrum.ieee.org/image/1752995>
9. When cell phones become computers, IEEE pervasive computing, April-June 2009
10. Smartphone statistics and market share. Mark Brow, August 2011
11. Internet trends CM summit New York. Morgan Stanley, 7 June 2010
12. [http://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))
13. <http://developer.android.com/index.html>
14. [http://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)#cite\\_note-i.2Fo\\_2011\\_stats-14](http://en.wikipedia.org/wiki/Android_(operating_system)#cite_note-i.2Fo_2011_stats-14)
15. <http://www.letsgomobile.org/en/7301/android-smartphone/>
16. [http://www.engr.sjsu.edu/ges/media/pdf/mse\\_prj\\_rpts/spring2010](http://www.engr.sjsu.edu/ges/media/pdf/mse_prj_rpts/spring2010)
17. Version Control with Subversion. 2011 Ben Collins-Sussman

## Γενική βιβλιογραφία

1. Retro Meier, "*Professional Android Application Development*", Wrox.
2. Retro Meier, "*Professional Android 2 Application Development*", Wrox.
3. Sayed Y. Hashimi and Satya Komatineni , "*Pro Android*", Apress, 2009.
4. Jerome DiMarzio , "*Android a programmer's guide*", Mc Graw Hill, 2008.
5. Mark L. Murphy, "*Beginning Android*", Apress.
6. Chris Haseman, "*Android Essential*", Apress.
7. Ed Burnette, "*Hello, Android Introducing Google's Mobile Development Platform*", ATTICA, 2009.

# Παράρτημα (κώδικας)

## Splash Class

```
package com.IntroScrn;

import android.app.Activity;
import android.content.Intent;
//import android.media.MediaPlayer;
import android.os.Bundle;

import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.view.View.OnClickListener;

import android.widget.Button;

/**
 * Η κλάση εμφανίζει ένα animation
 *
 * Λαδής Γεώργιος
 *
 */
public class Splash extends Activity implements OnClickListener {

    // =====
    // Fields
    // =====
    Button buttonNext;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // *****
        // fullscreen mode
        // *****
        requestWindowFeature(Window.FEATURE_NO_TITLE);

        getWindow().addFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN);
        // WindowManager.LayoutParams.NO_STATUS_BAR_FLAG);
        // *****

        setContentView(R.layout.splash);
        buttonNext = (Button) findViewById(R.id.Button01);
        buttonNext.setOnClickListener(this);
    }
}
```

```

@Override
public void onClick(View src) {
    // TODO Auto-generated method stub

    switch (src.getId()) {

        case R.id.Button01:
            Intent intent1 = new
Intent (src.getContext(), introScreenTest.class);
            //intent1.setClassName("com.IntroScrn",
            //    "com.IntroScrn.introScreenTest");
            startActivity(intent1);
            this.finish();
            break;

    }
}
}

```

---

### introScreenTest Class

```

/**
 * Η κύρια κλάση της εφαρμογής
 *
 *
 * Λαδής Γιώργος
 */
package com.IntroScrn;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Toast;

public class introScreenTest extends Activity implements OnClickListener
{

    /** Called when the activity is first created. */

    String arxikiTimi = "start"; // Για να περάσουμε στο Intent την
τιμή που έχουμε
    String neaTimi; //επιλέξει από το Alert Dialog
    Boolean item_has_been_selected = false;
    // final CharSequence[] items =
    // {"Fire","Reinforcement","Intruder","False Alarm"};
    final CharSequence[] items = { "Φωτιά", "Ενισχύσεις", "Εισβολέας",
        "Λάθος Συναγερμός" }; // πίνακας με τις τιμές των Radio
    Button
    String emailList[] = { "aaaaaaaa" };//το e-mail του παραλήπτη
    String emailCCList[] = { "bbbbbbbb" };//κοινοποίηση

```

```

Button mailBtn;
Button alertBtn;
Button cameraBtn;
Button formBtn;

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main_screen);

    cameraBtn = (Button) findViewById(R.id.Button01);
    formBtn = (Button) findViewById(R.id.Button02);
    mailBtn = (Button) findViewById(R.id.Button03);
    alertBtn = (Button) findViewById(R.id.Button04);

    formBtn.setOnClickListener(this);
    alertBtn.setOnClickListener(this);

    cameraBtn.setOnClickListener(this);
    mailBtn.setOnClickListener(this);
}

public void alert_Dialog() { //δημιουργία alert Dialog
    //που θα περιλαμβάνει τίτλο, εικονίδιο δυο κουμπια ένα "OK"
    και ένα "CANCEL" και 4 radio buttons
    AlertDialog.Builder builder = new AlertDialog.Builder(this);

    builder.setTitle(R.string.Choose_Alert_String).setCancelable(false)

    .setIcon(R.drawable.hai_global).setPositiveButton("OK",
        new DialogInterface.OnClickListener()
    {
        @Override
        public void
        onClick(DialogInterface dialog,
            int which) {
            // TODO Auto-generated
            if
            arxikiTimi =
            Intent smsIntent1 =

            Intent.ACTION_SENDTO, Uri
            .parse("sms:123456"));
            smsIntent1.putExtra("sms_body", arxikiTimi
            .toString());
            startActivity(smsIntent1);
        }
    }
}

```

```

        } else {

            Toast.makeText(getApplicationContext(),
                "No
                Item Was Chosen.",
                Toast.LENGTH_LONG).show();

            item_has_been_selected = false;
        }
    })

    .setNeutralButton("Cancel",
        new DialogInterface.OnClickListener()
    {
        @Override
        public void
        onClick(DialogInterface dialog,
            int which) {
            // TODO Auto-generated
            item_has_been_selected =
            false;
        }
    })

    .setSingleChoiceItems(items, -1,
        new DialogInterface.OnClickListener()
    {
        @Override
        public void
        onClick(DialogInterface dialog,
            int which) {
            // TODO Auto-generated

            neaTimi = (String)
            items[which];

            item_has_been_selected =
            true;
        }
    });

    AlertDialog alert = builder.create();
    alert.show();

}

@Override
public void onClick(View v) { //σύνδεση των τεσσάρων κουμπιών με
τις αντίστοιχες κλάσεις τους μέσω intent

    switch (v.getId()) {

```

```
        case R.id.Button01:
            Intent mapIntent = new Intent(v.getContext(),
myMapV2.class);
            startActivity(mapIntent);
            break;
        case R.id.Button02:
            Intent formIntent = new Intent(v.getContext(),
Form.class);
            startActivity(formIntent);

            break;
        case R.id.Button03:

            Intent emailIntent = new
Intent(android.content.Intent.ACTION_SEND);

            emailIntent.putExtra(android.content.Intent.EXTRA_EMAIL,
emailList);
            emailIntent.putExtra(android.content.Intent.EXTRA_CC,
emailCCList);
            emailIntent.setType("plain/text");
            startActivity(Intent.createChooser(emailIntent,
"Send your email in:"));

            break;
        case R.id.Button04:
            alert_Dialog();
            break;
    }
}
}
```

---

### Form Class

```
package com.IntroScrn;

/**
 * η κλάση περιέχει πεδία για συμπλήρωση και drop-down list
 *
 * Λαδής Γεώργιος
 */

import java.text.SimpleDateFormat;
import java.util.Date;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
```



```

import android.widget.Spinner;
import android.widget.Toast;

public class Form extends Activity{
    //String emailList[] = { "geladis@it.teithe.gr"};
    //String emailCCList[] = { "giorgosladis@gmail.com"};
    private Spinner alertSp,armedSp,eventSp,placeSp;
    private EditText officerId,numIntruders;
    private String alertChoice,armedChoice,eventChoice,placeChoice;
    private int offId,numIntr;
    private Button sendBtn;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.form);

        alertSp=(Spinner) findViewById(R.id.SpinnerAlert);
        armedSp=(Spinner) findViewById(R.id.SpinnerArmed);
        eventSp=(Spinner) findViewById(R.id.SpinnerEvent);
        placeSp=(Spinner) findViewById(R.id.SpinnerPlace);
        officerId=(EditText) findViewById(R.id.EditText01);
        numIntruders=(EditText) findViewById(R.id.EditText02);
        sendBtn=(Button) findViewById(R.id.Button06);

        sendBtn.setOnClickListener(new Button.OnClickListener(){
            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub

                alertChoice=alertSp.getSelectedItem().toString();
                armedChoice=armedSp.getSelectedItem().toString();
                eventChoice=eventSp.getSelectedItem().toString();
                placeChoice=placeSp.getSelectedItem().toString();
                offId=new
Integer(officerId.getText().toString());
                numIntr=new
Integer(numIntruders.getText().toString());

                SimpleDateFormat dateFormat=new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
                Date date=new Date();
                DataLayer dl=new
DataLayer(getBaseContext());//προσθέτουμε στην μέθοδο ια στοιχεία που
έχουμε επιλέξει
                dl.addForm(offId,numIntr,armedChoice,alertChoice
, eventChoice,placeChoice,dateFormat.format(date));
                Toast.makeText(getApplicationContext(),"Table
updated", Toast.LENGTH_SHORT).show();
            }
        });
    }
}

```

```
}  
}
```

---

### DataLayer Class

```
package com.IntroScrn;  
  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.sqlite.SQLiteDatabase;  
  
/**  
 * Εισαγωγή τιμών στον πίνακα  
 *  
 * armed=true/false  
 * alertLevel=red/yellow/green  
 * eventChar=observation/sabotage/theft  
 * placeEvent=in HAI/perimeter/outside;  
 *  
 * Giorgos Ladis  
 */  
  
public class DataLayer {  
  
    private DBHelper _dbHelper;  
  
    public DataLayer(Context context) {  
        _dbHelper = new DBHelper(context);  
    }  
  
    public void addForm(int officer_Id, int no_Intruders, String  
_armed,  
                        String alert_Level, String event_Char, String  
place_Event,  
                        String date_Time) {  
        SQLiteDatabase db = _dbHelper.getWritableDatabase();  
        try {  
            ContentValues values = new ContentValues();  
  
            values.put("officerId", officer_Id);  
            values.put("noIntruders", no_Intruders);  
            values.put("armed", _armed);  
            values.put("alertLevel", alert_Level);  
            values.put("eventChar", event_Char);  
            values.put("placeEvent", place_Event);  
            values.put("timeStamp", date_Time);  
  
            db.insert("formData", null, values);  
        }  
    }  
}
```

```

        } finally {
            if (db != null)
                db.close();
        }
    }
}

```

---

### DBHelper Class

```

package com.IntroScrn;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.database.sqlite.SQLiteDatabase.CursorFactory;

/**
 * Η κλάση δημιουργεί έναν πίνακα ΒΔ που αποθηκεύεται στην μνήμη του
 * τηλεφώνου
 *
 * Giorgos Ladis
 */

public class DBHelper extends SQLiteOpenHelper {

    public DBHelper(Context context) {
        super(context, "formDt.db", null, 1);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String createSql="CREATE TABLE formData "(eventId INTEGER
PRIMARY KEY AUTOINCREMENT NOT NULL , "+
        "officerId INTEGER, "+ "noIntruders INTEGER, "+ "armed TEXT,
"+ "alertLevel TEXT, "+ "eventChar TEXT, "+ "placeEvent TEXT, "+
        "timeStamp DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
"+ "latitude REAL , "+ "longitude REAL);";
        db.execSQL(createSql);
    }

    @Override
    public void onUpgrade(SQLiteDatabase arg0, int arg1, int arg2) {
        // TODO Auto-generated method stub
    }
}

```

---

**myMapV2 Class**

```
package com.IntroScrn;

import java.util.List;

import android.graphics.drawable.Drawable;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.widget.Toast;

import com.google.android.maps.GeoPoint;
import com.google.android.maps.MapActivity;
import com.google.android.maps.MapController;
import com.google.android.maps.MapView;
import com.google.android.maps.Overlay;
import com.google.android.maps.OverlayItem;

/**
 * Η κλάση τοποθετεί στο χάρτη σε συγκεκριμένα σημεία της κάμερας και τον
 φρουρό
 *
 * Λαδής Γεώργιος
 */
public class myMapV2 extends MapActivity {

    private MapView mView;
    private MapController mc;
    private GeoPoint p1, p2, p3;
    private LocationManager locationManager;
    private LocationListener locationListener;
    private Boolean getGPSCoordinateUpdates;
    private ItemOverlay cameras, guards;
    private OverlayItem guard;

    private double latGPS, longGPS, accuracy;
```

```

private int latGP, longGP;// latitude,longitude Geopoint for map

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.map_view);

    p1 = new GeoPoint(38326062, 23588784);
    OverlayItem overlayitem = new OverlayItem(p1, "CAMERAS",
"Camera1");

    p2 = new GeoPoint(38326495, 23588478);
    OverlayItem overlayitem2 = new OverlayItem(p2, "CAMERAS",
"Camera2");

    p3 = new GeoPoint(38326619, 23588977);
    OverlayItem overlayitem3 = new OverlayItem(p3, "CAMERAS",
"Camera3");

    guard = new OverlayItem(new GeoPoint(38327686, 23586059),
"GUARDS",
        "Guard1");// initial location sti mesi

    // g2 = new GeoPoint(38326525, 23586992);
    // OverlayItem overlayitem4 = new OverlayItem(g2, "HAI",
"Guard2");

    // g3 = new GeoPoint(38325245, 23589664);
    // OverlayItem overlayitem5 = new OverlayItem(g3, "HAI",
"Guard3");

    mview = (MapView) findViewById(R.id.mapView);
    mview.setBuiltInZoomControls(true);//zoom controls
    mview.setSatellite(true); //satellite view
    mc = mview.getController();
    mc.setZoom(18); // zoom level
    mc.animateTo(p1);//ο χάρτης θα ξεκινήσει απο το πρώτο σημείο(
κάμερα p1)
    List<Overlay> mapOverlays = mview.getOverlays();

    Drawable drawable =
this.getResources().getDrawable(R.drawable.camera);
    // Drawable drawable3 =
    // this.getResources().getDrawable(R.drawable.axis);
    Drawable drawable2 = this.getResources().getDrawable(
        R.drawable.security);

    cameras = new ItemOverlay(drawable, this);
    // axis= new ItemOverlay(drawable3, this);
    guards = new ItemOverlay(drawable2, this);

    cameras.addOverlay(overlayitem);
    cameras.addOverlay(overlayitem2);
    cameras.addOverlay(overlayitem3);
    // axis.addOverlay(overlayitem3);

    guards.addOverlay(guard);
    // guards.addOverlay(overlayitem4);

```

```

// guards.addOverlay(overlayitem5);

mapOverlays.add(cameras);
// mapOverlays.add(axis);
mapOverlays.add(guards);
locationManager = (LocationManager) this
    .getSystemService(LOCATION_SERVICE);
locationListener = new LocationListener() {

    @Override
    public void onLocationChanged(Location location) {
        // TODO Auto-generated method stub
        latGPS = location.getLatitude(); // se double
        morfh px 38.326062,
        // 23.588784

        longGPS = location.getLongitude();
        accuracy = location.getAccuracy();
        if (accuracy < 100.0) {
            latGPS = (location.getLatitude()) * 1E06; //
        }

        // integer gia
        // na mpei sto
        longGPS = location.getLongitude() * 1E06; //
        sto geopoint

        latGP = (int) latGPS;
        longGP = (int) longGPS;
        Toast.makeText(getApplicationContext(),
            "find gps",
            Toast.LENGTH_SHORT).show();

        //
        locationManager.removeUpdates(locationListener);
        guard = guards.moveOverlayItem(guard, new
        GeoPoint(latGP, //καλούμε τη μέθοδο που θα επιτρέψει στο εικονίδιο του
        φρουρού
        //να αλλάξει θέση καθώς
        μετακινώμαστε με την συσκευή
        longGP));
        // guard = new OverlayItem(new
        GeoPoint(latGP, longGP),
        // "HAI", "Guard");
        // guards.addOverlay(guard);
        // mView.postInvalidate();
        mView.invalidate();
    }

}

@Override
public void onProviderDisabled(String provider) {
    // TODO Auto-generated method stub

}

@Override
public void onProviderEnabled(String provider) {

```

```

        // TODO Auto-generated method stub
    }

    @Override
    public void onStatusChanged(String provider, int
status,
        Bundle extras) {
        // TODO Auto-generated method stub
    }
};

locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER
, 0,
        0, locationManager);
}

@Override
protected boolean isRouteDisplayed() {
    return false;
}
}

```

#### ItemOverlay Class

```

package com.IntroScrn;

import java.util.ArrayList;
import android.app.AlertDialog;
import android.content.Context;
import android.content.Intent;
import android.graphics.drawable.Drawable;
import android.util.Log; //import android.widget.MediaController;
//import android.widget.VideoView;

import com.google.android.maps.GeoPoint;
import com.google.android.maps.ItemizedOverlay;
import com.google.android.maps.OverlayItem;

/**
 * Η κλάση προσθέτει σε μία λίστα τα αντικείμενα του χάρτη (κάμερες,
φρουρός)
 * περιέχει την μέθοδο για την μετακίνηση του φρουρού
 * μια μέθοδο για την ενεργεια που θα γίνει κατά την επιλογή της κάμερας
onTap()
 *
 * Λαδής Γεώργιος
 *
 */

public class ItemOverlay extends ItemizedOverlay<OverlayItem> {

    private ArrayList<OverlayItem> mOverlays = new
ArrayList<OverlayItem>();
    private Context ctx;

```

```
public ItemOverlay(Drawable defaultMarker, Context context) {
    super(boundCenterBottom(defaultMarker));
    this.ctx = context;
}

public void addOverlay(OverlayItem overlay) {
    mOverlays.add(overlay);
    populate();
}

@Override
protected OverlayItem createItem(int i) {
    return mOverlays.get(i);
}

@Override
public int size() {
    return mOverlays.size();
}

//μέθοδος για την μετακίνηση του φρουρου
public OverlayItem moveOverlayItem(OverlayItem overlay, GeoPoint
newPos) {
    OverlayItem newOverLay = new OverlayItem(newPos,
overlay.getTitle(),
                overlay.getSnippet());
    mOverlays.remove(overlay);
    this.addOverlay(newOverLay);
    return newOverLay;
}

//μέθοδος για την επιλογή των εικονιδίων πάνω στο χάρτη
protected boolean onTap(int index) {
    Intent camInt;
    OverlayItem item=mOverlays.get(index);
    String title=item.getSnippet();

    switch (index) {
    case 0:
        // Log.d("camTap1", "index=" + index);
        if(title=="Camera1"){
            camInt = new Intent(ctx, CameraStream.class);
            camInt.putExtra("CAM_IP", "192.168.2.85");
            camInt.putExtra("CAM_VENDOR", "VIVOTEK1");
            camInt.putExtra("CAM_MOVE", "camctrl.cgi");
            ctx.startActivity(camInt);
        }
        else if(title=="Guard1"){

            AlertDialog.Builder dialog = new
AlertDialog.Builder(ctx);
            dialog.setTitle(item.getTitle());
            dialog.setMessage(item.getSnippet());
            dialog.show();
        }
    }
}
```



```
        break;
    case 1:
        // Log.d("camTap2", "index=" + index);
        camInt = new Intent(ctx, CameraStream.class);
        camInt.putExtra("CAM_IP", "192.168.2.87");
        camInt.putExtra("CAM_VENDOR", "VIVOTEK2");
        camInt.putExtra("CAM_MOVE", "viewer/camctrl.cgi");
        ctx.startActivity(camInt);
        break;
    case 2:
        // Log.d("camTap3", "index=" + index);
        camInt = new Intent(ctx, AxisView.class);
        camInt.putExtra("CAM_IP", "192.168.2.86");
        camInt.putExtra("CAM_VENDOR", "AXIS");
        // camInt.putExtra("CAM_MOVE", "viewer/camctrl.cgi");
        ctx.startActivity(camInt);
        break;
    }

    return true;
}
}
```

---

### AxisView Class

```
package com.IntroScrn;

import android.app.Activity;
import android.content.Context;
import android.content.pm.ActivityInfo;
import android.os.Bundle;
import android.os.PowerManager;
import android.os.PowerManager.WakeLock;
import android.view.Window;
import android.webkit.WebChromeClient;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebSettings.LayoutAlgorithm;

/**
 * Η κλάση για εμφάνισει της κάμερας τύπου Axis με την βοήθεια ενός
 * browser
 * Λαδής Γεώργιος
 *
 */

public class AxisView extends Activity {
    /** Called when the activity is first created. */
```

```
WebSettings myWebSettings;
WebView myWebView;
private String camIp;
private String camVendor;
PowerManager powerManager;
WakeLock wakeLock;

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    this.getWindow().requestFeature(Window.FEATURE_PROGRESS); //
add progress

        // bar

        // support
    setContentView(R.layout.axis_webview);

    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);
    powerManager = (PowerManager)
    getSystemService(Context.POWER_SERVICE);
    wakeLock = powerManager.newWakeLock(
        PowerManager.SCREEN_BRIGHT_WAKE_LOCK,
        "AXIS Camera");
    wakeLock.acquire();

    myWebView = (WebView) findViewById(R.id.WebView01);
    myWebSettings = myWebView.getSettings();

    // Makes Progress bar Visible
    getWindow().setFeatureInt(Window.FEATURE_PROGRESS,
        Window.PROGRESS_VISIBILITY_ON);

    Bundle extras = getIntent().getExtras();
    if (extras != null) {
        camIp = extras.getString("CAM_IP");
        camVendor = extras.getString("CAM_VENDOR");
    }

    myWebSettings.setJavaScriptEnabled(true);
    myWebSettings.setAllowFileAccess(true);
    myWebSettings.setPluginsEnabled(true); // flash
    myWebSettings.setSupportZoom(true);

    myWebSettings.setUseWideViewPort(true);

    myWebSettings.setLayoutAlgorithm(LayoutAlgorithm.NARROW_COLUMNS);
    myWebView.setInitialScale(60);

    myWebSettings.setCacheMode(WebSettings.LOAD_NO_CACHE);
    myWebView.getSettings().setBuiltInZoomControls(true);

    myWebView.loadUrl("http://" + camIp + "/axis-
cgi/mjpg/video.swf");
```

```
        // Sets the Chrome Client, and defines the onProgressChanged
        // This makes the Progress bar be updated.
        final Activity MyActivity = this;
        myWebView.setWebChromeClient(new WebChromeClient() {
            public void onProgressChanged(WebView view, int
progress) {
                // Make the bar disappear after URL is loaded,
and changes
                // string to Loading...
                MyActivity.setTitle("Loading...");
                MyActivity.setProgress(progress * 100); // Make
the bar

                // disappear after URL
                // is loaded

                // Return the app name after finish loading
                if (progress == 100)
                    MyActivity.setTitle(R.string.app_name);
            }
        });
    }
}
```

---

### CameraStream Class

```
package com.IntroScrn;

import java.io.BufferedInputStream;
import java.io.IOException;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLConnection;
import android.app.Activity;
import android.content.pm.ActivityInfo;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.view.Window;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.MediaController;
import android.widget.VideoView;

/**
 * Η κλάση για των τηλεχειρισμό των καμερων vivotek
```

```
*
*
* Λαδής Γεώργιος
*
*/
public class CameraStream extends Activity implements OnClickListener{
    /** Called when the activity is first created. */
    private Button upBtn, downBtn, leftBtn, rightBtn, homeBtn, startStopBtn;
    private VideoView myVideoView;
    private MediaController myMC;

    private String camIp;
    private String camVendor;
    private String camMove;
    private boolean videoPlaying;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.camera_view);

        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);

        myVideoView=(VideoView) findViewById(R.id.VideoView01);
        myMC= new MediaController(this);
        myVideoView.setMediaController(myMC);
        upBtn=(Button) findViewById(R.id.BtnUp);
        downBtn=(Button) findViewById(R.id.BtnDown);
        leftBtn=(Button) findViewById(R.id.BtnLeft);
        rightBtn=(Button) findViewById(R.id.BtnRight);
        homeBtn=(Button) findViewById(R.id.BtnHome);
        startStopBtn=(Button) findViewById(R.id.BtnStartStop);

        videoPlaying=false;
        upBtn.setOnClickListener(this);
        downBtn.setOnClickListener(this);
        leftBtn.setOnClickListener(this);
        rightBtn.setOnClickListener(this);
        homeBtn.setOnClickListener(this);
        startStopBtn.setOnClickListener(this);

        Bundle extras=getIntent().getExtras();
        if (extras!=null) {
            camIp=extras.getString("CAM_IP");
            camVendor=extras.getString("CAM_VENDOR");
            camMove=extras.getString("CAM_MOVE");
        }
    }

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
    }
}
```

```
URL
urlHome=null,urlUp=null,urlDown=null,urlLeft=null,urlRight=null;
    try {
        urlHome=new URL("http://"+camIp+"/cgi-
bin/"+camMove+"?move=home");
    } catch (MalformedURLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    try {
        urlUp=new URL("http://"+camIp+"/cgi-
bin/"+camMove+"?move=up");
    } catch (MalformedURLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    try {
        urlDown=new URL("http://"+camIp+"/cgi-
bin/"+camMove+"?move=down");
    } catch (MalformedURLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    try {
        urlLeft=new URL("http://"+camIp+"/cgi-
bin/"+camMove+"?move=left");
    } catch (MalformedURLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    try {
        urlRight=new URL("http://"+camIp+"/cgi-
bin/"+camMove+"?move=right");
    } catch (MalformedURLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    switch(v.getId()){
    case R.id.BtnHome:
        try {
            URLConnection urlc = urlHome.openConnection();
            BufferedInputStream buffer = new
BufferedInputStream(urlc.getInputStream());

            StringBuilder builder = new StringBuilder();
            int byteRead;
            while ((byteRead = buffer.read()) != -1)
                builder.append((char) byteRead);

            buffer.close();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
```

```
        break;
    case R.id.BtnLeft:
        try {
            URLConnection urlc = urlLeft.openConnection();
            BufferedInputStream buffer = new
BufferedInputStream(urlc.getInputStream());

            StringBuilder builder = new StringBuilder();
            int bytesRead;
            while ((byteRead = buffer.read()) != -1)
                builder.append((char) byteRead);

            buffer.close();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        break;
    case R.id.BtnRight:
        try {
            URLConnection urlc = urlRight.openConnection();
            BufferedInputStream buffer = new
BufferedInputStream(urlc.getInputStream());

            StringBuilder builder = new StringBuilder();
            int bytesRead;
            while ((byteRead = buffer.read()) != -1)
                builder.append((char) byteRead);

            buffer.close();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        break;
    case R.id.BtnUp:
        try {
            URLConnection urlc = urlUp.openConnection();
            BufferedInputStream buffer = new
BufferedInputStream(urlc.getInputStream());

            StringBuilder builder = new StringBuilder();
            int bytesRead;
            while ((byteRead = buffer.read()) != -1)
                builder.append((char) byteRead);

            buffer.close();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        break;
    case R.id.BtnDown:
        try {
            URLConnection urlc = urlDown.openConnection();
            BufferedInputStream buffer = new
BufferedInputStream(urlc.getInputStream());
```

```
        StringBuilder builder = new StringBuilder();
        int bytesRead;
        while ((byteRead = buffer.read()) != -1)
            builder.append((char) byteRead);

        buffer.close();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    break;
case R.id.BtnStartStop:
    if (videoPlaying==false){
        Uri
videoUri=Uri.parse("rtsp://" + camIp + "/live.sdp");
        myVideoView.setVideoURI(videoUri);
        myVideoView.start();
        videoPlaying=true;
        startStopBtn.setText(R.string.stopCam);
    }
    else{
        myVideoView.stopPlayback();
        videoPlaying=false;
        startStopBtn.setText(R.string.startCam);
    }
    break;
}
}
}
```

---