

**ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.**

**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ**

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ**

**ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ Τ.Ε.**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Ανάπτυξη εφαρμογής Android με χρήση NFC**

**Κουσουρής Σπυρίδων Νικόλαος**

**Εισηγητής: Ιωάννης Έλληνας**

**ΑΘΗΝΑ**

**ΙΑΝΟΥΑΡΙΟΣ 2016**



Ανάπτυξη εφαρμογής Android με χρήση NFC

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Ανάπτυξη εφαρμογής Android με χρήση NFC**

**Κουσουρής Σπυρίδων Νικόλαος**

**A.M. 38266**

**Εισηγητής:**

**Ιωάννης Έλληνας, Καθηγητής**

**Εξεταστική Επιτροπή:**

**Ημερομηνία εξέτασης:**



## **ΕΥΧΑΡΙΣΤΙΕΣ**

Για την ολοκλήρωση και παρουσίαση της συγκεκριμένης εργασίας θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, κ. Ιωάννη Έλληνα, για την στήριξη του και την ευκαιρία που μου έδωσε να ασχοληθώ με ένα πολύ ενδιαφέρον αντικείμενο, την ανάπτυξη εφαρμογών Android και την τεχνολογία NFC.

Επίσης θα ήθελα να ευχαριστήσω την οικογένεια μου για την στήριξη τους καθ'ολη την διάρκεια των σπουδών μου και τον συνάδελφο Κωνσταντίνο Γιαννούλη για την βοήθεια του στην ανάπτυξη της συγκεκριμένης εφαρμογής.

## **ΠΕΡΙΛΗΨΗ**

Η παρούσα πτυχιακή εργασία ασχολείται με την τεχνολογία Επικοινωνίας Κοντινού Πεδίου (NFC) και περιλαμβάνει την ανάπτυξη μίας εφαρμογής σε για Android smartphones σχετικά με βαθμολογίες. Η τεχνολογία NFC είναι μια καινούρια τεχνολογία μεταφοράς δεδομένων η οποία αρχίζει να κατακλύζει την αγορά. Χρησιμοποιείται σε πολύ μεγάλο βαθμό από κινητά τηλέφωνα νεας γενιάς και ειδικά σε θέματα πληρωμών. Για την εφαρμογή που αναπτύχθηκε έγινε χρήση NFC Tags με σκοπό την προβολή συγκεκριμένων μαθημάτων όταν ένα tag σαρωθεί από το κινητό και την εύρεση της βαθμολογίας ενός φοιτητή.

## **ABSTRACT**

The present thesis concerns the development of an Android application using Near Field Communication (NFC) technology. NFC is a new technology used in order to send and receive data between two devices and has gained a lot of recognition in today's market. It is widely used by smartphones and especially in payments. For the application developed, NFC tags are used in order to display certain lessons once a tag is scanned by a smartphone and then allow the student to find his score.

Επιστημονική Περιοχή: Ανάπτυξη Εφαρμογών Κινητών Τηλεφώνων

Λέξεις Κλειδιά: NFC, Android, Βαθμολογίες, Application, java

## **ΠΕΡΙΕΧΟΜΕΝΑ**

<b>ΚΕΦΑΛΑΙΟ 1</b> .....	<b>11</b>
1.1 Περιγραφή του αντικειμένου της πτυχιακής εργασίας .....	11
1.2 Ιστορική Αναδρομή .....	11
<b>ΚΕΦΑΛΑΙΟ 2</b> .....	<b>13</b>
2.1 Ιστορική Αναδρομή .....	13
2.1.2 Η νέα εποχή .....	14
2.1.3 Ο ανταγωνισμός και οι διαμάχες .....	15
2.2. Τρέχουσα κατάσταση .....	17
2.2.1 Η κυριαρχία .....	17
2.2.1.1 Παράγοντες επιτυχίας .....	17
2.2.1.2 Android Παντού .....	18
2.2.2 Τα μειονεκτήματα .....	18
2.2.2.1 Πολυμορφικότητα .....	18
2.2.2.2 Αγορές Android και διανομή εφαρμογών .....	19
2.2.2.3 Πειρατεία λογισμικού .....	21
2.2.2.4 Ασφάλεια και απόρρητο ιδιωτικότητας .....	21
2.3 Διεπαφή Χρήστη .....	22
2.4 Εκδόσεις Android και κατά κερματισμός αγοράς .....	23
2.5 Πλατφόρμα Ανάπτυξης Εφαρμογών .....	24
2.6 Σύγκριση εγγενών και διαπλατφορμικών λύσεων .....	26
2.7 Βασικές αρχές εφαρμογών Android .....	29
2.7.1 Δραστηριότητες (Activities) .....	29
2.7.2 Υπηρεσίες (Services) .....	31
2.7.3 Πάροχοι περιεχομένου .....	32
2.7.4 Προθέσεις και Φίλτρα Προθέσεων (Intents and Intent Filters) .....	32
2.7.5 Διεργασίες και νήματα (Processes and Threads) .....	33
2.7.6 Δικαιώματα (Permissions) .....	33
2.7.7 Αρχείο βασικών ορισμών (Android Manifest) .....	34
<b>ΚΕΦΑΛΑΙΟ 3</b> .....	<b>37</b>
3.1 Ανάλυση της τεχνολογίας NFC .....	37
3.2 Συσκευές NFC .....	37
3.2.1 Παθητικές Συσκευές .....	38
3.2.2 Ενεργές Συσκευές .....	40
3.3 Είδη λειτουργίας NFC .....	41
3.4 Εφαρμογές της τεχνολογίας NFC .....	42
3.5 Η δομή NDEF .....	43
<b>ΚΕΦΑΛΑΙΟ 4</b> .....	<b>47</b>
4.1 Περιγραφή της εφαρμογής .....	47
4.2 Η υλοποίηση της εφαρμογής .....	48
<b>ΚΕΦΑΛΑΙΟ 5</b> .....	<b>74</b>
5.1 Συμπεράσματα .....	74
5.2 Μελλοντικές εργασίες .....	74
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ</b> .....	<b>77</b>
<b>ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ</b>	
Σχήμα 2.1: Android logo .....	13
Σχήμα 2.2 : Εκδόσεις Android και κατακερματισμός αγοράς .....	24
Σχήμα 2.3: Ροή δραστηριοτήτων .....	31
Σχήμα 3.1: Ανάλυση NFC Tag .....	38



Σχήμα 3.2: NDEF Overview .....	43
Σχήμα 4.1: Αρχική οθόνη εφαρμογής .....	49
Σχήμα 4.3: Layout κ. Έλληνα .....	50
Σχήμα 4.4: Layout κ. Λεβέντη.....	50
Σχήμα 4.5: Layout κ. Ζάχαρη .....	51
Σχήμα 4.6: Layout κ. Βελώνη .....	51
Σχήμα 4.7: Layout κ. Τσελίκα .....	52
Σχήμα 4.8: Layout κ. Πρεζεράκου.....	52
Σχήμα 4.9 Layout κ. Παλιατσού .....	53
Σχήμα 4.10: Layout κ. Νικολόπουλου.....	53
Σχήμα 4.11: Layout κ. Ματιάτου .....	54
Σχήμα 4.12: Layout κ. Κούρου.....	54
Σχήμα 4.13: Layout κ. Κουκουλέτσου .....	55
Σχήμα 4.14: Layout κ. Γιαννακόπουλου .....	55
Σχήμα 4.15: Layout κ. Φατούρου.....	56
Σχήμα 4.16: Layout κ. Διλιντά .....	56
Σχήμα 4.17: Layout κ. Αλατσαθιανού.....	57
Σχήμα 4.18: Δήλωση ImageView .....	58
Σχήμα 4.20: Αντιστοίχιση στοιχείων με το xml κείμενο .....	60
Σχήμα 4.21: Δήλωση κουμπιών .....	61
Σχήμα 4.22: Δήλωση Linear Layouts και μεταβλητών.....	62
Σχήμα 4.24: Αντιστοίχιση των Linear Layouts και των κουμπιών με τα xml.....	64
Σχήμα 4.27: URL Connection.....	66
Σχήμα 4.28: Ανάγνωση Tag και εμφάνιση του κατάλληλου layout.....	66
Σχήμα 4.29: Το layout της αναζήτησης .....	67
Σχήμα 4.30: Ειδοποίηση ότι δεν βρεθηκε ο αριθμός μητρώου .....	68
Σχήμα 4.31: Εμφάνιση βαθμολογίας φοιτητή.....	68
Σχήμα 4.32: Το layout search1.xml.....	69
Σχήμα 4.34: Δήλωση πινάκων και μεταβλητών .....	71

## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 3.1: NFC Forum Tag Types.....	40
---------------------------------------	----

## **ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ**

NFC Near Field Communication

RFID Radio Frequency Identification

ISO International Organization for Standardization

IEC International Electrotechnical Commission

NDEF NFC Data Exchange Format

## **ΚΕΦΑΛΑΙΟ 1**

### **ΕΙΣΑΓΩΓΗ**

Σε αυτό το κεφάλαιο αναλύεται το αντικείμενο της πτυχιακής εργασίας και γίνεται μια ιστορική αναδρομή σχετικά με την τεχνολογία NFC.

#### **1.1 Περιγραφή του αντικειμένου της πτυχιακής εργασίας**

Η τεχνολογία NFC (Near Field Communication) είναι μια μορφή τεχνολογίας που επιτρέπει την επικοινωνία μεταξύ κινητών και άλλων συσκευών με την επαφή ή φέρνοντας δύο συσκευές σε κοντινή απόσταση. Πρόκειται για μία καινούρια τεχνολογία η οποία ξεκίνησε το 2004 αλλά ουσιαστικά τα πρώτα NFC Tags άρχισαν να χρησιμοποιούνται το 2006.

Το δεύτερο σκέλος της πτυχιακής εργασίας περιλαμβάνει την ανάπτυξη μίας εφαρμογής για Smartphones με λογισμικό Android, όπου γίνεται χρήση των NFC Tags σε συνδυασμό με το τηλέφωνο για την πρόσβαση των φοιτητών του τμήματος στις βαθμολογίες τους.

#### **1.2 Ιστορική Αναδρομή**

Η επικοινωνία NFC βασίζεται σε έναν μεγάλο βαθμό στο Radio Frequency Identification γνωστό και ως RFID. Ουσιαστικά πρόκειται για ένα υποσύνολο του RFID το οποίο έχει περιορισμένη εμβέλεια για λόγους ασφαλείας.

Το 2004 οι εταιρίες Nokia, Sony και Philips ίδρυσαν τον λεγόμενο NFC Forum μέσω του οποίου προωθούν την ασφάλεια, την ευκολία και τις καινοτομίες στον χώρο του NFC.

Η αρχή για τα NFC Tags έγινε το 2006 και άρχισε με τις λεγόμενες «έξυπνες» αφίσες. Οι αφίσες αυτές περιλαμβάνουν πληροφορίες οι οποίες μπορούν να διαβαστούν όταν ένα κινητό που υποστηρίζει την τεχνολογία NFC έρθει εντός εμβέλειας (4 - 5cm). Παραδειγματικά μπορούν να δώσουν παραπάνω πληροφορίες για ένα έκθεμα σε ένα μουσείο ή ακόμα και την βιογραφία κάποιου διάσημου σε μία αφίσα με την φωτογραφία του.

Έκεινη την περίοδο, και συγκεκριμένα τον Φεβρουάριο του 2006, έκανε την εμφάνισή του και το πρώτο κινητό συμβατό με NFC, το Nokia 6131. Από τότε οι δυνατότητες του NFC έχουν πολλαπλασιαστεί και περιλαμβάνουν πληρωμές μέσω κινητών μέχρι και κοινοποιήσεις βίντεο, συνδέσμων καθώς και προσκλήσεις σε παιχνίδια μεταξύ 2 χρηστών. Το πρώτο Android τηλέφωνο συμβατό με NFC ήταν το Samsung Nexus S, το οποίο κυκλοφόρησε το 2010. Τέλος η Apple περιέλαβε το NFC για πρώτη φορά το 2015 στα iPhone 6 και 6+.

Σήμερα το NFC αναπτύσσεται ραγδαία στις αγορές τις Ασίας, της Ευρώπης αλλά και στην Αμερική. Αναμένεται ότι σύντομα θα εξελιχθεί σε πολύ δημοφιλή τρόπο πληρωμής και ανταλλαγής δεδομένων σε όλες τις ανεπτυγμένες αγορές.

## ΚΕΦΑΛΑΙΟ 2

### ΠΛΑΤΦΟΡΜΑ ANDROID

#### 2.1 Ιστορική Αναδρομή

Το Android είναι ένα λειτουργικό σύστημα ανοικτού κώδικα (Open Source) για έξυπνες κινητές συσκευές με οθόνη αφής (touch screen), βασισμένο στον πυρήνα του λειτουργικού συστήματος Linux. Απαριθμεί περισσότερες από 750 εκατομμύρια συσκευές και περισσότεροι από 60 κατασκευαστές τηλεφώνων βασίζονται σε αυτό για τα προϊόντα τους. Στα πιο γνωστά προϊόντα συγκαταλέγεται η ταμπλέτα Kindle Fire της εταιρείας Amazon, η σειρά έξυπνων κινητών τηλεφώνων Galaxy S της εταιρείας Samsung, καθώς και ταμπλετών Nexus της Google.

Το σήμα κατατεθέν του Android είναι το ευρέως αναγνωρίσιμο πράσινο ρομπότ, το οποίο φαίνεται παρακάτω



Σχήμα 2.1: Android logo

Η ανάπτυξη του Android ξεκίνησε το 2003 στο Palo Alto, California(μέρος της Silicon Valley – «Μέκκα» των μεγαλύτερων εταιρειών υψηλής τεχνολογίας στον κόσμο). Από μια μικρή ομάδα νέων και φιλόδοξων ανθρώπων υπό τη μορφή νεόφυτης εταιρείας (startup company) με το όνομα Android Inc. Λίγα είναι γνωστά για την αρχική έκδοση του λογισμικού (καθότι δούλευαν κάτω από άκρα μυστικότητα), παρά μόνο το γεγονός ότι δεν έμοιαζε πολύ, ούτε εικαστικά αλλά ούτε και τεχνικά με το αυτό που παρουσιάστηκε το 2008 ως το πρώτο επίσημο έξυπνο κινητό τηλέφωνο με οθόνη αφής Android (το HTC dream).

Μόλις δύο χρόνια αργότερα, το 2005 η Google εξαγοράζει την εταιρεία Android Inc. (παράλληλα απασχολώντας τους ιδρυτές) σε μία διαφανόμενη προσπάθεια να εισέλθει στον χώρο του πολλά υποσχόμενου και κερδοφόρου τομέα των έξυπνων κινητών συσκευών, κάτι σημαντικό για το μέλλον της εταιρείας. Η εξαγορά είχε σαν αποτέλεσμα να δώσει οικονομική ενίσχυση, μεγαλύτερη προοπτική και ορατότητα στο Android. Παράλληλα έκανε εκατομμυριούχους τους εμπνευστές του. Ανάμεσα στην ιδρυτική ομάδα του Android ήταν και ο Andy Rubin, ο οποίος συνέχισε μέχρι το Μάρτιο του 2013 να έχει ενεργή και σημαντική συμμετοχή στην εξέλιξη του Android. Θεωρείται από πολλούς, από τους βασικότερους εμπνευστές – οραματιστές και εκτελεστές του Android.

### **2.1.2 Η νέα εποχή**

Το 2007, το Android περνάει στην δικαιοδοσία και επίβλεψη μιας κοινοπραξίας εταιρειών τεχνολογίας και τηλεπικοινωνιών, της «Open Handset Alliance» (OHA). Σκοπός του νεοϊδρυθέντος οργανισμού, ήταν και είναι η ανάπτυξη και εξέλιξη ανοικτών προτύπων στις συσκευές κινητής τηλεφωνίας. Βασικός περιορισμός παραμένει η απαγόρευση παραγωγής και διάθεσης έξυπνων κινητών τηλεφώνων μη συμβατών με την επίσημη έκδοση του Android. Παράλληλα η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache License (έκδοση 2) μιας άδειας ελευθέρου λογισμικού, κίνηση που όπως θα δούμε αργότερα έδωσε τεράστια ώθηση στο λειτουργικό της σύστημα, αλλά

και παράλληλα οδήγησε σε ένα μεγάλο κατακερματισμό (fragmentation) της αγοράς του Android.

### 2.1.3 Ο ανταγωνισμός και οι διαμάχες

Στο μεταξύ, το καλοκαίρι του 2007, η εταιρεία Apple αποκάλυψε την πρωτοποριακή και επαναστατική για τον κλάδο, έξυπνη κινητή συσκευή με οθόνη αφής. Αυτή δεν ήταν άλλη από το πασίγνωστο πετυχημένο και ακριβό iPhone με το ιδιόκτητο (proprietary) λειτουργικό σύστημα iPhone OS (μετέπειτα γνωστό ως iOS). Η κίνηση αυτή της Apple αιφνιδίασε όλη την αγορά κινητών των τηλεφώνων, η οποία βρέθηκε τουλάχιστον μία γενιά πίσω όσον αφορά στην τεχνολογική καινοτομία καθώς καμία άλλη εταιρεία δεν χρησιμοποιούσε με επιτυχία οθόνη αφής και εξελιγμένο λειτουργικό σύστημα στη σειρά των προϊόντων της για κινητή τηλεφωνία, και μέσα σε ένα πολύ μικρό χρονικό διάστημα κατάφερε να αποκτήσει πολύ σημαντικό μερίδιο αγοράς με τεράστια μάλιστα κέρδη. Το γεγονός αυτό σήμανε συναγερμό για τις εταιρείες πληροφορικής/τηλεπικοινωνιών, όπως η Google, η οποία αρχικά έσπευσε, σε διάστημα λίγων ετών, να εξελίξει το Android στο ποιοτικό επίπεδο του iOS και αργότερα ίσως, για πρώτη φορά να το ξεπεράσει. Η Apple, βλέποντας να ανεβαίνει σταδιακά αλλά σταθερά το μερίδιο αγοράς του Android, κατέφυγε σε μηνύσεις. Η όλη αυτή διαδικασία τελικά αποδείχτηκε γενικά πολύ επίπονη και δαπανηρή για τις εμπλεκόμενες εταιρείες (όπως και για τις μεγαλύτερες εταιρείες κατασκευής συσκευών που λειτουργούν με το λειτουργικό σύστημα Android (όπως η Samsung) καθώς η Apple προσέφυγε δικαστικά ενάντια σε οτιδήποτε έμοιαζε να χρησιμοποιεί εμπορικά ή να σχετίζεται δικαιωματικά με το Android, σε αυτό που ανεπίσημα πλέον αποκαλείται Apple – Google Patent War.

Αργότερα ενεπλάκη στη διαμάχη και η εταιρεία Microsoft, η οποία εικάζεται ότι βλέποντας να έχει μείνει πίσω στις εξελίξεις, δημιούργησε κρυφά συμμαχία με την



Apple σχετικά με πνευματικά δικαιώματα και πατέντες ευρεσιτεχνίας και στράφηκε επίσης ενάντια στο Android.

Ακολούθησαν, η εταιρεία Oracle, η οποία ισχυρίστηκε καταπάτηση πνευματικών δικαιωμάτων όσον αφορά στη γλώσσα Java και η εταιρεία Nokia, η οποία επίσης ισχυρίστηκε παρόμοια πράγματα για διάφορες τεχνολογίες, βλέποντας ότι χάνει μεγάλο μερίδιο αγοράς (με το πλέον απαρχαιωμένο ιδιόκτητο λειτουργικό σύστημα Symbian OS) προς τους κατασκευαστές που χρησιμοποιούσαν Android. Όλα αυτά οδήγησαν την Google να αναγκαστεί να ξοδέψει τεράστια ποσά (δισεκατομμύρια δολάρια) για να αμυνθεί (κυρίως αγοράζοντας διπλώματα ευρεσιτεχνίας) και να προστατέψει το οικοσύστημα Android.

Στο μεταξύ, την περίοδο 2008-2011 οι κυρίαρχες εταιρείες του χώρου, δηλαδή η RIM ( με το λειτουργικό σύστημα Blackberry OS) και η Nokia (με το λειτουργικό σύστημα Symbian OS ή το πλέον παρωπλισμένο λειτουργικό σύστημα MeeGo – συνεργασία της Nokia με την εταιρεία Intel), βλέπουν να χάνουν τεράστια ποσοστά της αγοράς και ξαφνικά να βρίσκονται στην τελευταία θέση της κατάταξης, από κυρίαρχοι της αγοράς (market leaders). Η Microsoft στο διάστημα αυτό, προκειμένου να γίνει ανταγωνιστική, προσπαθεί και φτιάχνει ένα καινούριο ιδιόκτητο λειτουργικό σύστημα για έξυπνα κινητά τηλέφωνα, το Windows Phone (έχοντας εγκαταλείψει πλέον το παλιό λειτουργικό σύστημα Windows Mobile). Το νέο λειτουργικό σύστημα φαίνεται να μην έχει αποκτήσει ακόμα σημαντική διείσδυση στην αγορά εν έτει 2015. Η Nokia, συνεχίζοντας να χάνει μερίδιο αγοράς προς το Android, τελικά υιοθέτησε το 2011 το λειτουργικό Windows Phone (αντί να εξελίξει το δικό της) σε μια στρατηγική συνεργασία με τη Microsoft. Η κίνηση αυτή δεν φαίνεται να έχει αποδώσει τα αναμενόμενα οφέλη μέχρι και το τέλος του 2012 – αρχές 2013. Η RIM στις αρχές του 2013 μετονομάστηκε σε Blackberry και κυκλοφορεί το ολοκαίνουριο και φέρελπις λειτουργικό σύστημα Blackberry 10, μετά από τα μεγάλα οικονομικά και οργανωτικά προβλήματα που αντιμετώπισε λόγω της απώλειας τεράστιου μεριδίου αγοράς στον ανταγωνισμό (κυρίως προς το Android αλλά και το iOS).

## **2.2. Τρέχουσα κατάσταση**

### **2.2.1 Η κυριαρχία**

Το Android συνεχίζει την ανοδική του πορεία στη διάρκεια του 2015 όντας πλέον κυρίαρχος της αγοράς των έξυπνων κινητών συσκευών, παρά τα προβλήματα που προσπάθησαν να δημιουργήσουν ή δημιούργησαν οι ανταγωνιστικές εταιρείες μέσω του συστήματος κατοχύρωσης ευρεσιτεχνιών των Ηνωμένων Πολιτειών της Αμερικής. Αναλυτές της αγοράς εικάζουν ότι η επικράτηση του Android και το 2016 είναι σχεδόν σίγουρη υπόθεση

#### **2.2.1.1 Παράγοντες επιτυχίας**

Σε αυτή την πορεία συνέβαλε η Google η οποία επένδυσε σοβαρούς πόρους στο έργο (από το να το βελτιώνει συνεχώς, μέχρι να το προστατέψει στα δικαστήρια ή ξοδεύοντας δισεκατομμύρια δολάρια διπλώματα ευρεσιτεχνίας προκειμένου να θωρακιστεί νομικά η πλατφόρμα) καθώς και η ανοικτή φύση της πλατφόρμας που έδωσε τη δυνατότητα σε πληθώρα κατασκευαστών να προσφέρουν σε αφθονία άκρως ανταγωνιστικά προϊόντα και να λειτουργήσει ο ανταγωνισμός και η ελεύθερη αγορά καλύτερα. Σημειώνεται ότι το Android καλύπτει μια ευρεία γκάμα συσκευών από ταμπλέτες και κινητά έως και netbooks, με τις συσκευές να διατίθενται σε ένα εύρος τιμών που ξεκινά από τα 50 -70 και φθάνει ως τα επίπεδα των συσκευών iOS δηλαδή στην περιοχή των 800 – 900 ευρώ.

Επιπρόσθετα, το γεγονός ότι οι περισσότερες εφαρμογές για Android προσφέρονται δωρεάν (και μάλιστα δεν χρειάζονται και ιδιαίτερη έγκριση όπως οι εφαρμογές της Apple στο App store, μειώνοντας έτσι το χρόνο διάθεσης

Ανάπτυξη εφαρμογής Android με χρήση NFC

(deployment) στην αγορά Play Store σε τάξη λεπτών αντί ημερών – εβδομάδων) είναι ένας ακόμα βασικός λόγος επικράτησης της πλατφόρμας.

Τέλος, σημαντικό ρόλο φυσικά έπαιξαν και τα σφάλματα των ανταγωνιστών, όπως η Nokia ή οποία εγκατέλειψε το δικό της λειτουργικό σύστημα Symbian) και RIM (πλέον BlackBerry, η οποία άργησε σημαντικά να εκσυγχρονίσει το ιδιωτικό λειτουργικό της σύστημα), της Microsoft (η οποία καθυστέρησε να κυκλοφορήσει το δικό της λειτουργικό σύστημα για κινητές συσκευές) και φυσικά της Apple με κύριο ζήτημα το πρόβλημα κεραίας στο iPhone, τα προβλήματα του προγράμματος Maps ( μετά την απόφαση της εταιρείας να εγκαταλείψει το πρόγραμμα Google maps).

#### **2.2.1.2. Android Παντού**

Πέρα από τα κινητά τηλέφωνα, τις ταμπλέτες και τα netbooks, το Android έχει επεκταθεί και σε μια πληθώρα συσκευών όπως τα παρακάτω:

- Φωτογραφικές μηχανές
- Τηλεοράσεις
- Συσκευές αναπαραγωγής πολυμέσων (media players)
- Παιχνιδομηχανές
- Συστήματα αυτοματισμού κατοικιών και κτηρίων
- Έξυπνα ρολόγια
- Πλυντήρια κ.α.

#### **2.2.2 Τα μειονεκτήματα**

##### **2.2.2.1 Πολυμορφικότητα**

Η τεράστια επιτυχία του Android έχει και το τίμημα της, κάτι που πληρώνουν κυρίως οι καταναλωτές που δέχονται βομβαρδισμό από νέες συσκευές (αφού υπάρχει υπερπροσφορά και συνεχώς κατακλύζουν την αγορά πάρα πολλά νέα μοντέλα κάτι που οδηγεί σε μεγάλο βαθμό την απαξίωση της αξίας των συσκευών καθώς αποκτούν ευκολότερα την ιδιότητα των αναλώσιμων συσκευών) καθώς και οι προγραμματιστές Android καθότι η κάθε συσκευή μπορεί να παρουσιάσει ιδιαιτερότητες (εφόσον είναι ανοικτός ο κώδικας και κάθε κατασκευαστής κάνει αλλαγές βάσει των αναγκών του) επομένως είναι αναγκαία η δοκιμή της καλής λειτουργίας των εφαρμογών προς έκδοση σε μια πολύ μεγάλη ποικιλία συσκευών και διαφορετικών εκδόσεων (πολλές φορές με ασυμβατότητα μεταξύ κυρίων εκδόσεων).

Επιπρόσθετα παρατηρείται το φαινόμενο να μη λαμβάνεται από τις εταιρείες μέριμνα για την αναβάθμιση του λειτουργικού συστήματος σε συσκευές μικρής ή μεσαίας δυναμικότητας, λόγω του μικρού σχετικά κόστους των συσκευών και άρα μικρότερου περιθωρίου κέρδους. Συνεπώς οι συσκευές δεν συντηρούνται κατάλληλα από τις εταιρείες που τις παράγουν, με αποτέλεσμα να καθίστανται παρωχημένες πολύ εύκολα. Αυτό αποτελεί και ένα μεγάλο παράπονο της κοινότητας Android, ενώ δε συμβαίνει το ίδιο με την Apple όπου συσκευές που βρίσκονται τρεις γενιές πίσω από άποψη υλικού μπορούν να τρέξουν σχεδόν απροβλημάτιστα την τελευταία έκδοση του λειτουργικού συστήματος (π.χ. iPhone 4 μπορεί να τρέξει την τελευταία έκδοση, iOS9.X).

### **2.2.2.2 Αγορές Android και διανομή εφαρμογών**

Η μεταφορτώση και διαχείριση των εφαρμογών Android γίνεται κυρίως από το Play Store (πρώην Android Market), την επίσημη αγορά λογισμικού (marketplace) Android από την Google.

Ωστόσο, υπάρχουν πλέον πολλές αγορές λογισμικού (Application marketplaces) εφαρμογών Android, η καθεμιά με τους δικούς της περιορισμούς.

Μερικές τέτοιες αγορές είναι:

- Play Store (το διαχειρίζεται η Google)
- Amazon App Store (το διαχειρίζεται η Amazon)
- Opera Mobile App Store (το διαχειρίζεται η Opera, δημιουργός της εφαρμογής Opera browser).
- Samsung Apps Store (διαχειρίζεται από τη Samsung και περιορίζεται μόνο για τα προϊόντα της)
- GetJar (δεν σχετίζεται με κάποια μεγάλη εταιρεία πληροφορικής)
- SlideMe (δεν σχετίζεται με κάποια μεγάλη εταιρεία πληροφορικής).

Γενικότερα υπάρχουν δύο τρόποι «φορτώματος» εφαρμογών σε Android συσκευές:

1. Ο πρωταρχικός τρόπος απόκτησης εφαρμογών είναι από κάποια από τις διαθέσιμες αγορές εφαρμογών Android. Η κάθε αγορά λογισμικού έχει και τη δική της εφαρμογή διαχείρισης, που παρέχει στους χρήστες πρόσβαση στις εφαρμογές που προσφέρει.
2. Ο δεύτερος τρόπος είναι πιο χειροκίνητος και αφορά τη μεταφόρτωση εκτελέσιμου αρχείου (τύπου APK) της εφαρμογής στη συσκευή. Το Android επιτρέπει την εγκατάσταση εφαρμογών (ως πακέτα APK) της εφαρμογής στη συσκευή, αρκεί ο χρήστης να έχει αποδεχθεί στις ρυθμίσεις της συσκευής του να γίνεται εγκατάσταση πακέτων από μη έμπιστες πηγές. Ο τρόπος αυτός είναι και ο συνηθέστερος εγκατάστασης πειρατικών εφαρμογών ή εφαρμογών που δε δημιουργούνται στο Play Store για ποικίλους λόγους (π.χ. επειδή είναι ιδιωτικά είτε επειδή δεν συμφωνούν με τους όρους της εκάστοτε αγοράς). Η διαδικασία αυτή λέγεται «κατά περίπτωση διανομή εφαρμογής» ή αλλιώς ad-hoc app distribution.

### **2.2.2.3 Πειρατεία λογισμικού**

Η Google έχει δεχθεί σφοδρή κριτική όσον αφορά στην ευκολία απόκτησης και χρήσης εμπορικών εφαρμογών Android με παράνομο τρόπο (χωρίς να πληρωθεί το σχετικό αντίτιμο από το χρήστη). Κάποιες προσπάθειες έχουν γίνει προς αυτή την κατεύθυνση χωρίς όμως το πρόβλημα να λυθεί. Το ζήτημα της πειρατείας των εφαρμογών Android παραμένει και σήμερα ένα σοβαρό θέμα για τους εκδότες και προγραμματιστές εφαρμογών Android.

### **2.2.2.4 Ασφάλεια και απόρρητο ιδιωτικότητας**

Δυστυχώς όπως και με κάθε άλλο παραγωγικό σύστημα, η ασφάλεια του Android είναι ένα μεγάλο ζήτημα που παραμένει ανοικτό και η θωράκιση του από εξωγενείς κακοήθεις παράγοντες δεν έχει καταστεί απόλυτα επιτυχής. Αποτέλεσμα αυτού είναι να έχουν υπάρξει διάφορα και σημαντικά κρούσματα υποκλοπής στοιχείων και ανεξέλεγκτης των τελικών χρηστών τα τελευταία χρόνια.

Οι πηγές του κακού εντοπίζονται στα παρακάτω τέσσερα σημεία:

1. Η έλλειψη ελέγχου κατά την υποβολή ενός προγράμματος στο Play Store. Ενώ η πολιτική αυτή βοηθάει στο να μειωθούν οι απαιτήσεις δημοσίευσης εφαρμογών από τους νομότυπους προγραμματιστές/εκδότες, ταυτόχρονα μπορεί να αποτελέσει σημείο εκμετάλλευσης για όσους θέλουν να διαθέσουν κακόβουλες και ζημιογόνες εφαρμογές. Ίσως μια πολύ γρήγορη και τυπική διαδικασία ανθρωπίνου ελέγχου από την Google να βοηθούσε στο να μειωθεί η έκταση αυτού του φαινομένου. Σημειώνεται ότι το Play Store έχει ήδη ένα αυτοματοποιημένο σύστημα ελέγχου εφαρμογών,

γνωστό ως Bouncer, εντούτοις η δυνατότητα του στην πρόληψη αμφισβητείται.

2. Ο μηχανισμός ασφαλείας του Android μερικές φορές δεν γίνεται απόλυτα κατανοητός από τους προγραμματιστές ή γίνεται κατάχρηση του μηχανισμού των δικαιωμάτων, με αποτέλεσμα ο χρήστης να συμφωνεί και να παραχωρεί πρόσβαση σε περισσότερα πράγματα από ότι θα επιθυμούσε. Στην ενότητα «2.7.6 Δικαιώματα (Permissions)» γίνεται εκτενέστερη αναφορά στο μηχανισμό δικαιωμάτων.
3. «Κερκόπορτες» (Back Doors) από κατασκευαστές κινητών Android οι οποίες ανακαλύπτονται και χρησιμοποιούνται από διάφορους χάκερ (hackers).
4. Κενά ασφαλείας του Android που επιτρέπουν στους χρήστες να γίνουν υπερχρήστες (root) και να αποκτήσουν ελεύθερη πρόσβαση σε όλο το λειτουργικό σύστημα.

### 2.3 Διεπαφή Χρήστη

Το Android είναι κατά βάση λειτουργικό σύστημα κινητών συσκευών με οθόνες αφής και στηρίζεται στις χειρονομίες αφής του χρήστη για τη βασική λειτουργικότητα του. Αξιοποιεί μηχανισμούς παροχής αντίδρασης πληροφορίας (information feedback) μέσω

διάφορες εφαρμογές αξιοποιώντας, μόνα τους ή σε συνδυασμό, υλικά εξαρτήματα όπως:

- Πυξίδα (compass)
- Επιταχυνσιόμετρο (accelerator)
- Βαρόμετρο (barometer)
- Γυροσκόπιο (gyroscope)
- Αισθητήρες εγγύτητας (proximity sensors)
- Δέκτη GPS (GPS receiver)
- κ.α.

Παρέχει βολικό τρόπο προβολής και επιλογής των διαθέσιμων εφαρμογών υπό τη μορφή εικονιδίων σε ένα εικονικό πλέγμα (grid). Υποστηρίζει πληθώρα βασικών εργοστασιακών εφαρμογών (από αναγνώστη ηλεκτρονικού ταχυδρομείου, e-mail client, μέχρι και εφαρμογή διαχείρισης επαφών) καθώς και χιλιάδες άλλες εφαρμογές και παιχνίδια.

Το κάθε εικονίδιο στην οθόνη αντιπροσωπεύει μια εφαρμογή και ανοίγει όταν επιλεγεί μέσω ενός αγγίγματος (tap). Πέρα από το απλό άγγιγμα, υποστηρίζονται διάφορες ενέργειες που μπορούν να γίνουν με το χέρι σε οθόνη αφής (κίνηση σάρωσης – swipe, τσίμπημα – pinch, διπλό άγγιγμα – double tap) και που έχουν αντιστοιχιστεί σε πολυπλοκότερες ενέργειες και λειτουργίες (π.χ. η κίνηση σάρωσης χρησιμοποιείται συχνά στο να αλλάξει η ορατή οθόνη με μια άλλη που την ακολουθεί στη σειρά). Περισσότερα στοιχεία για τα παραπάνω μπορεί να αναζητήσει κανείς στον επίσημο οδηγό διεπαφής της τεκμηρίωσης του Android.

### **2.4 Εκδόσεις Android και κατά κερματισμός αγοράς**

Το λογισμικό Android διαρκώς ανανεώνεται με διαφορετικές εκδόσεις σχεδόν ανά εξάμηνο και με συνεχείς αναβαθμίσεις για διόρθωση σφαλμάτων. Παρακάτω βλέπουμε τις τελευταίες εκδόσεις Android και τον κατακερματισμό της αγοράς.



Version	Codename	API	Distribution
2.2	Froyo	8	0.2%
2.3.3 - 2.3.7	Gingerbread	10	3.0%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2.7%
4.1.x	Jelly Bean	16	9.0%
4.2.x		17	12.2%
4.3		18	3.5%
4.4	KitKat	19	36.1%
5.0	Lollipop	21	16.9%
5.1		22	15.7%

Σχήμα 2.2 : Εκδόσεις Android και κατακερματισμός αγοράς

## 2.5 Πλατφόρμα Ανάπτυξης Εφαρμογών

Η ανάπτυξη εφαρμογών για την πλατφόρμα Android γίνεται (ως προεπιλογή) με τη χρήση της γλώσσας προγραμματισμού Java σε συνδυασμό με το Πακέτο Ανάπτυξης Λογισμικού για Android (Android Software Development Kit ή Android SDK).

Το Πακέτο ανάπτυξης περιλαμβάνει μια εκτενή γκάμα εργαλείων όπως:

- εργαλείο διόρθωσης σφαλμάτων (Debugger)
- βιβλιοθήκες ανάπτυξης (π.χ. χάρτες Google)
- προσομοιωτή/εξομοιωτή φυσικής συσκευής (εικονικές συσκευές δηλαδή)
- οδηγίες χρήσης
- τεκμηρίωση κώδικα καθώς και

- παραδείγματα λειτουργικού κώδικα μεταξύ άλλων

Το επίσημο ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment, IDE) του Android είναι μια τροποποιημένη έκδοση του γνωστού περιβάλλοντος Eclipse, το οποίο περιλαμβάνει την τελευταία έκδοση του Android SDK και στο σύνολο ονομάζεται «Εργαλεία Ανάπτυξης Λογισμικού για Android», Android Development Tools (ATD).

Αξίζει να αναφερθεί ότι υπάρχουν εναλλακτικές προσεγγίσεις για την ανάπτυξη εφαρμογών Android με τη χρήση διαπλατφορμικού πακέτου λογισμικού δημιουργίας κινητών εφαρμογών ιστού (cross platform mobile web applications framework). Στην ενότητα 2.6 «Σύγκριση εγγενών και διαπλατφορμικών λύσεων» αναλύονται τα πλεονεκτήματα καθώς και τα μειονεκτήματα της κάθε προσέγγισης και η λογική πίσω από την επιλογή του επίσημου τρόπου ανάπτυξης για την εφαρμογή.

Αξίζει επίσης να σημειωθεί σε αυτό το σημείο ότι λόγω του κατακερματισμού της αγοράς του Android η διαδικασία δοκιμής καλής λειτουργίας (testing) μιας εφαρμογής έχει γίνει αρκετά δύσκολη και επίπονη καθώς χρειάζονται στην κυριολεξία πολλές δεκάδες διαφορετικές Android συσκευές προκειμένου να καλύψει κανείς μεγάλο μέρος της αγοράς. Η ύπαρξη δυνατότητας προσομοίωσης διευκολύνει σημαντικά τη διαδικασία δοκιμών, παρόλα αυτά στην πράξη η προσομοίωση δεν προσφέρει στην πιστότητα των πραγματικών συσκευών και είναι αργή.

Ένα ενδεικτικό παράδειγμα είναι το μέγεθος του εργαστηρίου δοκιμών (testing lab) της εταιρείας animoca η οποία ειδικεύεται στην ανάπτυξη εφαρμογών ψυχαγωγίας για πλατφόρμες κινητών συσκευών (mobile platforms).

## 2.6 Σύγκριση εγγενών και διαπλατφορμικών λύσεων

Για κάθε λειτουργικό κινητών συσκευών υπάρχει και η αντίστοιχη επίσημη υποδομή (IDE και SDK) δημιουργίας εγγενών εφαρμογών για αυτό.

Για το Android είδαμε ότι είναι το πακέτο λογισμικού Android Development Tools το οποίο περιλαμβάνει το Android SDK και κάνει χρήση της γλώσσας προγραμματισμού Java καθώς και μιας τροποποιημένης έκδοσης του προγράμματος Eclipse ως IDE.

Για το Windows Phone υπάρχει το πακέτο λογισμικού Windows Phone SDK το οποίο περιλαμβάνει το ομώνυμο SDK, το πρόγραμμα Visual Studio Express ως IDE και κάνει χρήση οποιασδήποτε γλώσσας προγραμματισμού υποστηρίζεται από το πλαίσιο λογισμικού .NET (.NET framework).

Για το iOS υπάρχει το πακέτο λογισμικού «Εργαλεία Προγραμματιστή iOS» (iOS Developer Toolset) το οποίο περιλαμβάνει το iOS SDK το πρόγραμμα XCode ως IDE και κάνει χρήση της γλώσσας προγραμματισμού Objective-C (υπερσύνολο της C).

Κανένας του δε μοιράζεται την ίδια γλώσσα προγραμματισμού γεγονός που έχει επιπτώσεις στους προγραμματιστές.

*Σημείωση: Τα προαναφερθέντα λειτουργικά δεν είναι τα μόνα λειτουργικά κινητών συσκευών, ωστόσο είναι τα επικρατέστερα και τα πιο γνωστά. Τα μικρότερα λειτουργικά δεν αναφέρονται και δεν έχουν ληφθεί υπόψη στη μελέτη λόγω του μικρού μεριδίου αγοράς που κατέχουν.*

Όλα τα παραπάνω παρέχονται δωρεάν στον προγραμματιστή, ωστόσο υπάρχουν κάποιοι βασικοί περιορισμοί στη χρήση των δύο τελευταίων:

## Ανάπτυξη εφαρμογής Android με χρήση NFC

- Το Windows Phone SDK λειτουργεί μόνο στα πλαίσια του λειτουργικού Microsoft Windows.
- Το iOS Developer Toolset λειτουργεί μόνο στα πλαίσια του λειτουργικού Mac OSX.

Αντίθετα, το Android Development Tool μπορεί να λειτουργήσει σε όλα τα γνωστά λειτουργικά συστήματα (Windows, Mac OS X, Linux) χωρίς πρόβλημα ή περιορισμό.

Επομένως η ανάπτυξη μιας εγγενούς iOS εφαρμογής προϋποθέτει την ύπαρξη ενός Apple υπολογιστή. Αυτό διότι το λειτουργικό Mac OS X δεν μπορεί να λειτουργήσει σε έναν οποιονδήποτε υπολογιστή λόγω τεχνικών και νομικών περιορισμών.

Παρόμοια, η ανάπτυξη μιας εγγενούς Windows Phone εφαρμογής προϋποθέτει την κατοχή άδειας χρήσης του λειτουργικού συστήματος Microsoft Windows. Και τα δύο, ανεξαρτήτως επιλογής έχουν κόστος.

Αντίθετα, η ανάπτυξη μιας εγγενούς εφαρμογής Android μπορεί να γίνει με μηδενικό κόστος στην περίπτωση που χρησιμοποιηθεί οποιαδήποτε δωρεάν έκδοση του λειτουργικού Linux.

Οι εν λόγω περιορισμοί καθώς και η ανάγκη ενός ενιαίου πηγαίου κώδικα (single codebase), οδήγησαν στη δημιουργία εργαλείων ανάπτυξης διαπλατφορμικών εφαρμογών, χρησιμοποιώντας μόνο μία τεχνολογία προγραμματισμού. Θεωρείται λογικό να υποτεθεί ότι η ανάπτυξη τριών διαφορετικών εγγενών εκδόσεων, μια για κάθε λειτουργικό κινητών συσκευών σχεδόν τριπλασιάζει την συνολική απαιτούμενη προσπάθεια.

Η χρήση εγγενών εργαλείων ανάπτυξης για κάθε λειτουργικό παρουσιάζει όμως και σημαντικά πλεονεκτήματα όπως η εκμετάλλευση εξεζητημένων δυνατοτήτων των συσκευών (π.χ. augmented reality σε συσκευές iPhone) και ταχύτητα εκτέλεσης του τελικού προϊόντος. Για αυτό το λόγο είναι σύνηθες

μεγάλες εμπορικές εφαρμογές να χρησιμοποιούν συνδυαστικά εγγενείς πλατφόρμες ανάπτυξης με διαπλατφορμικά εργαλεία.

Σημαντικό είναι επίσης και το κόστος δημοσίευσης εφαρμογών (δωρεάν ή επί πληρωμή) στις αντίστοιχες αγορές του iOS και του Windows Phone. Το ποσό αυτό ανέρχεται σε περίπου \$100 ανά έτος, ενώ για το Android το κόστος δημοσίευσης στο Play Store είναι \$25 εφάπαξ. Σε κάθε περίπτωση, το κόστος ισχύει ακόμα κι αν χρησιμοποιηθεί διαπλατφορμική λύση για τη δημιουργία μιας εφαρμογής (εφόσον αυτή δημοσιεύεται στις εν λόγω αγορές).

Γενικότερα οι εφαρμογές χωρίζονται σε τρεις κατηγορίες:

1. Εγγενείς εφαρμογές παράγωγα των επίσημων SDK
2. Εφαρμογές παράγωγα όλων των ανεπίσημων SDK που μετατρέπονται τελικά από το SDK σε εγγενείς εφαρμογές. Ένα τέτοιο παράδειγμα είναι το Titanium SDK. Οι εφαρμογές που έχουν αναπτυχθεί με αυτό, θεωρητικά έχουν την ταχύτητα των εγγενών εφαρμογών. Στην πράξη αποδεικνύεται ότι έχουν προβλήματα καθυστέρησης κυρίως στα εφέ των διαφόρων παραθύρων της εφαρμογής καθώς και στην επιστροφή από προηγούμενη κατάσταση (π.χ. από παύση της εφαρμογής).
3. Τοπικές εφαρμογές ιστού (local web applications) γραμμένες στις κλασικές τεχνολογίες συγγραφής ιστοσελίδων και «περιτυλιγμένες» (native wrapper) ως εγγενείς. Η απόκριση τους και η ταχύτητα τους εξαρτάται από την ταχύτητα της υποδομής της επίσημης εφαρμογής πλοήγησης του λειτουργικού συστήματος της συσκευής. Γενικά υπολείπονται πολύ της ταχύτητας των εγγενών εφαρμογών, κάτι που είναι προφανές σε εφαρμογές σημαντικού όγκου. Τέλος δεν υποστηρίζουν πλήρως τοπικές βάσεις δεδομένων (π.χ. SQLite).

Η χρήση της τεχνολογίας NFC, και οι απαιτήσεις της εφαρμογής «Βαθμολογίες NFC» να είναι γρήγορη και αποκρίσιμη, και με το χαμηλότερο δυνατό κόστος, οδήγησαν στην απόφαση της ανάπτυξης στην επίσημη πλατφόρμα Android. Σημαντικό ρόλο επίσης έπαιξε και η γλώσσα προγραμματισμού καθώς η Java

είναι ευρέως διαδεδομένη και πολύ γνωστή και κατ' επέκταση γίνεται ευκολότερη η συνεισφορά και συμμετοχή άλλων προγραμματιστών.

Στα μειονεκτήματα αυτής της επιλογής συγκαταλέγεται το γεγονός ότι η εφαρμογή δεν θα είναι διαθέσιμη για τα άλλα δύο μεγάλα λειτουργικά συστήματα κινητών συσκευών.

## 2.7 Βασικές αρχές εφαρμογών Android

Οι βασικές δομικές οντότητες που συνθέτουν μια εφαρμογή Android, και που έχουν αξιοποιηθεί στην πράξη στην εφαρμογή είναι οι ακόλουθες:

### 2.7.1 Δραστηριότητες (Activities)

Ως δραστηριότητα ορίζεται μια συγκεκριμένη ενέργεια που μπορεί να κάνει ένας χρήστης σε μία Android εφαρμογή. Κάθε δραστηριότητα εφόσον ορίζει μια διεπαφή με το χρήστη, είναι υπεύθυνη για τη δημιουργία του παραθύρου που θα περιέχει τη διεπαφή αυτή.

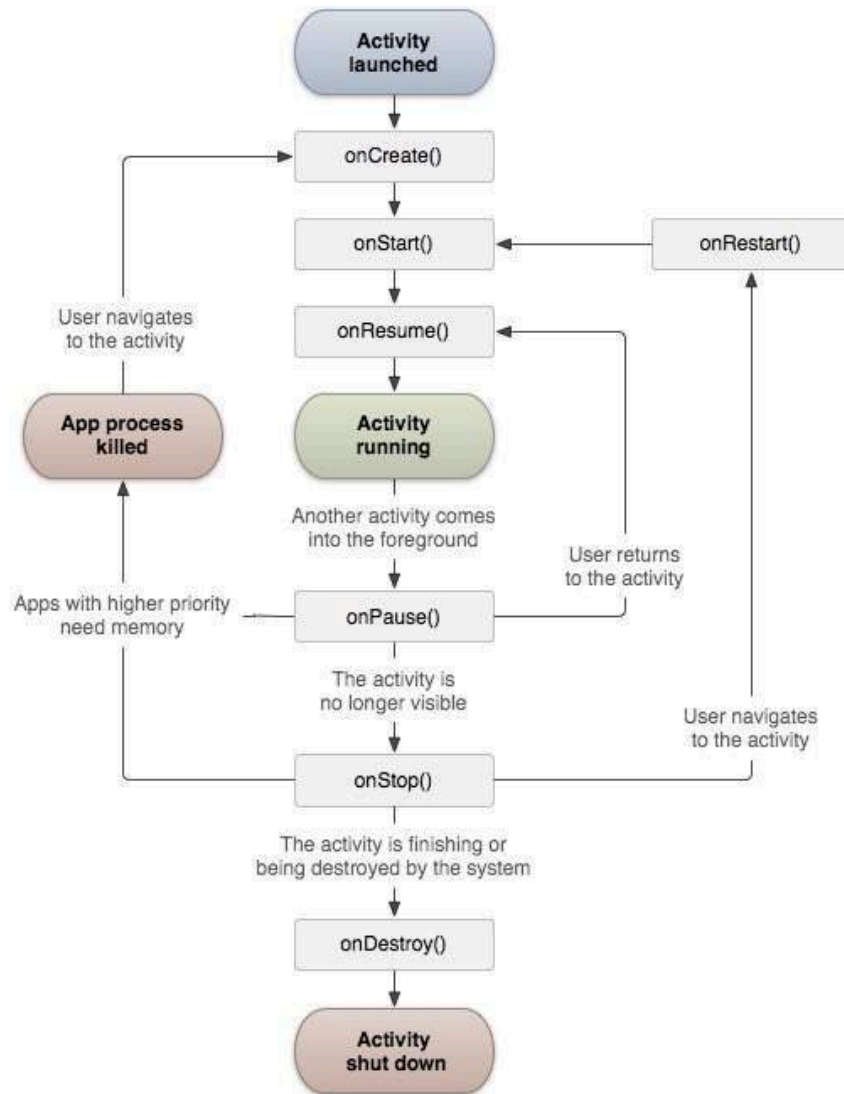
Μια εφαρμογή αποτελείται συνήθως από πολλαπλές δραστηριότητες, μία εκ των οποίων είναι η λεγόμενη «κύρια δραστηριότητα» (main activity) και η οποία αποτελεί την οθόνη που εμφανίζεται στο χρήστη όταν εκκινεί την εφαρμογή. Οι δραστηριότητες εναλλάσσονται ξεκινώντας άλλες δραστηριότητες. Για να γίνει αυτό, η εν ενεργεία δραστηριότητα σταματά την εκτέλεση της και τοποθετείται από το λειτουργικό σύστημα σε μία στοίβα (back stack). Η νέα δραστηριότητα που την αντικαθιστά τοποθετείται στην κορυφή της στοίβας με τη δραστηριότητα που έτρεχε πριν από αυτή να βρίσκεται ακριβώς από κάτω της.

Αξίζει να σημειωθεί ότι η βασική αυτή αρχή της στοίβας, καθιστά δυνατή την ύπαρξη ενός καθολικού «πίσω» κουμπιού (back button) σε όλες τις κινητές συσκευές Android και έτσι καθίσταται εφικτή η μεταφορά στην αμέσως προηγούμενη οθόνη.

Μια δραστηριότητα έχει τέσσερις κατά στάσεις (states):

- Αν είναι στο προσκήνιο (στην κορυφή της στοίβας) θεωρείται ότι είναι ενεργή (active).
- Αν έχει χάσει την εστίαση (focus) αλλά είναι ακόμα ορατή, θεωρείται ότι είναι σε παύση (paused).
- Αν έχει αντικατασταθεί από μια άλλη δραστηριότητα, θεωρείται ότι είναι σταματημένη (stopped).
- Αν είναι σε παύση ή σταματημένη, το λειτουργικό σύστημα μπορεί να αποδεσμεύσει χώρο που καταλαμβάνει στη μνήμη ζητώντας να σταματήσει τη λειτουργία της ή τερματίζοντας τη διεργασία που την εκτελεί.

Οι καταστάσεις αυτές και οι μεταξύ τους μεταβιβάσεις παρουσιάζονται λεπτομερώς στην εικόνα που ακολουθεί:



Σχήμα 2.3: Ροή δραστηριοτήτων.

## 2.7.2 Υπηρεσίες (Services)

Μια υπηρεσία είναι ένα κομμάτι μιας εφαρμογής Android το οποίο χρησιμοποιείται όταν θέλουμε να εκτελέσουμε διαδικασίες μακράς διάρκειας (long running) που δεν απαιτούν κάποια ενέργεια από το χρήστη. Για παράδειγμα, μια υπηρεσία μπορεί να αναλάβει την επικοινωνία με το δίκτυο, την αναπαραγωγή μουσικής, την αναγνώριση/εγγραφή σε κάποιο αρχείο ή να αλληλοεπιδρά



(interact) με έναν πάροχο περιεχομένου (content provider). Όλα αυτά γίνονται στο παρασκήνιο.

Αξίζει να σημειωθεί ότι μια υπηρεσία δεν είναι μια ξεχωριστή διεργασία. Εκτελείται στην ίδια διεργασία με την εφαρμογή της οποίας είναι μέρος. Επίσης δεν είναι νήμα (thread) και δε μας δίνει τη δυνατότητα να εκτελέσουμε κάποια ενέργεια εκτός του κυρίως νήματος της εφαρμογής του λεγόμενου Event Dispatching Thread (EDT) στον κόσμο της Java.

### 2.7.3 Πάροχοι περιεχομένου

Ένας πάροχος περιεχομένου χειρίζεται την πρόσβαση στα δεδομένα που υπάρχουν σε μια κεντρική πηγή πληροφοριών. Είναι κομμάτι μιας εφαρμογής Android και παρέχει μια διεπαφή μέσω της οποίας χειρίζεται τα εν λόγω δεδομένα. Με τον τρόπο αυτό δίνει τη δυνατότητα σε όποιον αναπτύσσει μια τέτοια εφαρμογή να διαχωρίσει τα δεδομένα από την υλοποίηση της εφαρμογής του (encapsulation) ενώ ταυτόχρονα του παρέχει μηχανισμούς που διασφαλίζουν την ασφάλεια τους.

### 2.7.4 Προθέσεις και Φίλτρα Προθέσεων (Intents and Intent Filters)

Τρία βασικά κομμάτια μιας εφαρμογής Android – οι δραστηριότητες, υπηρεσίες και δέκτες καθολικών εκπομπών (broadcast receivers) – ενεργοποιούνται μέσω μηνυμάτων, που στην πλατφόρμα Android ονομάζονται προθέσεις (intents). Ένα φίλτρο προθέσεων (intent filter) είναι στην ουσία μια δέσμη δεδομένων η οποία περιέχει πληροφορίες που αφορούν την οντότητα που θα λάβει το μήνυμα (όπως την ενέργεια που θα πρέπει να εκτελεστεί και τα δεδομένα που θα επηρεαστούν από την ενέργεια αυτή), καθώς και πληροφορίες που αφορούν το λειτουργικό (όπως το είδος της οντότητας που θα χειριστεί την πρόθεση).

Σε κάθε περίπτωση, το σύστημα εντοπίζει τη δραστηριότητα ή υπηρεσία για την οποία προορίζεται η πρόθεση και την αρχικοποιεί, εφόσον είναι απαραίτητο.

Μια πρόθεση συνήθως αποτελείται από τα εξής δεδομένα:

- Το όνομα της οντότητας που θα την χειριστεί
- Το είδος της οντότητας για την οποία προορίζεται η πρόθεση
- Την ενέργεια που θα εκτελεστεί
- Τα δεδομένα που θα επηρεαστούν από την ενέργεια αυτή

### 2.7.5 Διεργασίες και νήματα (Processes and Threads)

Κάθε φορά που μια οντότητα μιας εφαρμογής Android ξεκινά και η εφαρμογή δεν έχει άλλες οντότητες που εκτελούνται εκείνη τη στιγμή, το σύστημα ξεκινά μια νέα διεργασία Linux για την εφαρμογή καθώς και ένα εκτελέσιμο νήμα. Εξ ορισμού, όλες οι οντότητες μιας εφαρμογής εκτελούνται στην ίδια διεργασία και νήμα. Σε περίπτωση που μια οντότητα ξεκινά και υπάρχει ήδη μια διεργασία για την εφαρμογή, τότε αυτή ξεκινά στην ίδια διεργασία, κάνοντας χρήση του υπάρχοντος εκτελέσιμου νήματος. Παρόλα αυτά υπάρχει η δυνατότητα (κατ' απαίτηση του προγραμματιστή) διαφορετικές οντότητες σε μια εφαρμογή να εκτελεστούν σε διαφορετικές διεργασίες και να δημιουργηθούν επιπλέον νήματα για κάθε διεργασία.

### 2.7.6 Δικαιώματα (Permissions)

Το Android είναι ένα λειτουργικό σύστημα βασισμένο στην απομόνωση προνομίων (privilege separated), στο οποίο κάθε εφαρμογή που εκτελείται έχει και μια μοναδική ταυτότητα. Μέρη του συστήματος έχουν επίσης μοναδικές ταυτότητες. Έτσι διασφαλίζεται ότι κάθε εφαρμογή που εκτελείται είναι απομονωμένη από τις υπόλοιπες εφαρμογές αλλά και από το σύστημα.

Μια σημαντική σχεδιαστική αρχή στην πολιτική ασφάλειας του λειτουργικού συστήματος Android είναι ότι καμία εφαρμογή δεν έχει εξ ορισμού δικαίωμα να εκτελέσει κανενός είδους ενέργεια που μπορεί να επηρεάσει άλλες εφαρμογές, το σύστημα ή το χρήστη. Αυτό περιλαμβάνει την ανάγνωση ή την εγγραφή στα προσωπικά δεδομένα του χρήστη (π.χ. στις επαφές ή τα e-mail του) την ανάγνωση ή εγγραφή σε αρχεία που ανήκουν σε άλλες εφαρμογές, την πρόσβαση στο δίκτυο κ.λπ.

Το γεγονός ότι κάθε εφαρμογή είναι απομονωμένη (Sandboxed) από τις υπόλοιπες, σημαίνει ότι πρέπει να δηλώσει ρητά τις επιπλέον λειτουργίες που χρειάζεται να εκτελέσει και οι οποίες δεν περιλαμβάνονται στο βασικό χώρο που δίνεται από το λειτουργικό για την εκτέλεση της (sandbox). Η εφαρμογή αιτείται αυτές τις λειτουργίες στατικά, μέσω ενός μηχανισμού δικαιωμάτων, ο οποίος ζητά τη συγκατάθεση του χρήστη τη στιγμή που εγκαθιστά την εφαρμογή.

### 2.7.7 Αρχείο βασικών ορισμών (Android Manifest)

Κάθε εφαρμογή Android πρέπει να έχει ένα αρχείο “AndroidManifest.xml” (με αυτό ακριβώς το όνομα) στον κεντρικό κατάλογο (root directory) του έργου (project). Το αρχείο manifest περιέχει τις απαραίτητες πληροφορίες που πρέπει να γνωρίζει το λειτουργικό σύστημα για την εφαρμογή που το αρχείο συνοδεύει. Μεταξύ άλλων ορίζει τα εξής:

- Ορίζει το όνομα του πακέτου Java που περιέχει την εφαρμογή
- Περιγράφει τις δομικές οντότητες και στοιχεία εφαρμογής – δραστηριότητες, υπηρεσίες και παρόχους περιεχομένου που την αποτελούν
  - Ορίζει τα δικαιώματα που χρειάζεται η εφαρμογή από το λειτουργικό για να εκτελεστεί.
  - Ορίζει την ελάχιστη έκδοση του λειτουργικού με την οποία η εφαρμογή είναι συμβατή

- Ορίζει τις βιβλιοθήκες με τις οποίες πρέπει να συνδεθεί (ή από τις οποίες εξαρτάται) η εφαρμογή.

Σημειώνεται ότι οι αντίστοιχες βασικές οντότητες που αποτελούν το περιβάλλον χρήστη δεν αναλύονται καθώς συναντώνται με παρόμοια μορφή σε όλες σχεδόν τις διεπαφές κινητών εφαρμογών. Για περισσότερες λεπτομέρειες ο ενδιαφερόμενος αναγνώστης μπορεί να ανατρέξει στον σύντομο οδηγό Android για προγραμματιστές.



## ΚΕΦΑΛΑΙΟ 3

### Η ΤΕΧΝΟΛΟΓΙΑ NFC

#### 3.1 Ανάλυση της τεχνολογίας NFC

Όπως αναφέρθηκε παραπάνω το NFC βασίζεται στην τεχνολογία RFID. Πιό συγκεκριμένα χρησιμοποιούν το ίδιο πρωτόκολλο ISO/IEC στο οποίο το NFC εισήλθε το 2003. Φέρνοντας 2 συσκευές με NFC εντός εμβέλειας δημιουργείται μια μαγνητική επαγωγή παρόμοια με αυτή που γίνεται, εδώ και χρόνια, για την χρήση καρτών με σκοπό το ξεκλείδωμα μιας πόρτας. Αφού οι 2 συσκευές συνδεθούν χρησιμοποιούν μια συχνότητα 13.56 MHz , ίδια με τις συνδέσεις RFID, και ξεκινά η μετάδοση δεδομένων με τον σχετικά αργό ρυθμό των 424 Kbps.

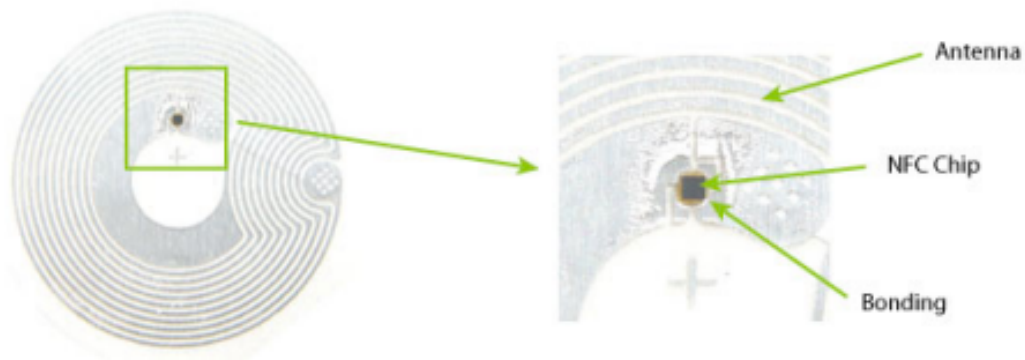
Η τεχνολογία NFC υποστηρίζεται απο κρυπτογράφηση και το γεγονός ότι έχει πολύ μικρή εμβέλεια λειτουργίας προσθέτει επιπλέον ασφάλεια καθώς οι πιθανότητες να υποκλαπεί το σήμα σε τόση μικρή εμβέλεια είναι πολύ μικρές έως μηδαμινές.

#### 3.2 Συσκευές NFC

Οι συσκευές που χρησιμοποιούν NFC χωρίζονται σε δύο κύριες κατηγορίες, τις ενεργές και τις παθητικές. Οι ενεργές συσκευές είναι εκείνες οι οποίες μπορούν να παράγουν ηλεκτρομαγνητικό πεδίο. Αντίθετα οι παθητικές συσκευές δεν έχουν αυτήν την δυνατότητα και βασίζονται στην συσκευή με την οποία επικοινωνούν για να λειτουργήσουν.

### 3.2.1 Παθητικές Συσκευές

Το κυριότερο παράδειγμα παθητικής συσκευής είναι τα NFC Tags. Τα NFC Tags είναι συσκευές που επιτρέπουν την αποθήκευση δεδομένων. Ανάλογα με τον τύπο τους ενδέχεται να υποστηρίζουν και κάποιες επιπλέον λειτουργίες όπως η κρυπτογράφηση. Είναι συνήθως μικρά και λεπτά σε μέγεθος και υπάρχουν στην αγορά σε διάφορες μορφές, όπως τα αυτοκόλλητα ή μπρελόκ.



Σχήμα 3.1: Ανάλυση NFC Tag

Όπως βλέπουμε από το σχήμα το μεγαλύτερο μέρος της επιφάνειας ενός NFC Tag καλύπτεται από την κεραία του. Η κεραία αυτή είναι ένα πηνίο κατασκευασμένο με τέτοιο τρόπο ώστε να μπορεί να μετατρέπει το μαγνητικό πεδίο σε ενέργεια. Τα υλικά κατασκευής είναι το αλουμίνιο ή ο χαλκός, αν και το αλουμίνιο είναι πιο διαδεδομένο στην αγορά λόγω κόστους. Το πάχος της κεραίας δεν είναι ανάλογο της αποδοτικότητας της, για τον λόγο αυτό βλέπουμε ότι οι κεραίες είναι τόσο λεπτές που χωράνε σε ένα αυτοκόλλητο. Σε αντίθεση με το πάχος, η διάμετρος της κεραίας έχει σημασία για την αποδοτικότητα. Η καλύτερη απόδοση σε ένα NFC Tag του δίνει την δυνατότητα να λειτουργεί σε

μεγαλύτερη απόσταση, χωρίς αυτή να περνά ποτέ τα 10 εκατοστά. Για παράδειγμα ένα Tag με κεραία διαμέτρου 25mm μπορεί να λειτουργήσει σε απόσταση μέχρι και 7 εκατοστών, ενώ ένα με διάμετρο κεραίας 38mm μπορεί να λειτουργήσει σε απόσταση μέχρι και 9 εκατοστών. Ωστόσο, όσο μεγαλύτερη είναι η κεραία τόσο μεγαλύτερο είναι και το μαγνητικό πεδίο το απαιτείται προκειμένου να λειτουργήσει το Tag. Για τον λόγο αυτό δεν συναντάμε ιδιαίτερα μεγάλες κεραίες αφού τα κινητά, τα οποία χρησιμοποιούνται κυρίως σε αυτές τις αλληλεπιδράσεις, δεν παρέχουν τόσο δυνατό μαγνητικό πεδίο.

Εκτός της κεραίας ένα ακόμα βασικό μέρος του NFC Tag είναι το NFC Chip. Το NFC Chip είναι ένα πολύ μικρό σε μέγεθος ολοκληρωμένο κύκλωμα που είναι υπεύθυνο για όλες τις λειτουργίες του NFC Tag. Τα NFC chips διαφέρουν σε τομείς όπως ο διαθέσιμος αποθηκευτικός χώρος, η δυνατότητα προγραμματισμού ώστε να είναι μόνο για ανάγνωση, η υποστήριξη κρυπτογραφημένης επικοινωνίας, τα υποστηριζόμενα ISO, και άλλα.

Το NFC Forum έχει κατηγοριοποιήσει τα NFC Tags σε τέσσερις τύπους. Στον πίνακα 2.1 , μπορούμε να δούμε τα βασικά χαρακτηριστικά του κάθε τύπου και μερικά προϊόντα που ανήκουν στην κάθε κατηγορία.

Ο προγραμματισμός των Tags ξεφεύγει από την απλή ανάγνωση κειμένου και δίνει πραγματικά αμέτρητες δυνατότητες. Μπορούμε για παράδειγμα να προγραμματίσουμε ένα Tag να οδηγεί σε μία ιστοσελίδα η οποία θα ανοίγει αυτόματα μόλις πλησιάσουμε το κινητό. Ένα Tag μπορεί επίσης να περιέχει πληροφορίες για ένα άτομο σε μορφή επαφής δημιουργώντας έτσι μια διαφορετική κάρτα εργασίας. Επίσης μπορούμε να προγραμματίσουμε ένα Tag να επηρεάζει διάφορες λειτουργίες του κινητού. Για παράδειγμα με ένα πέρασμα πάνω από ένα Tag μπορεί το κινητό να κλείνει το WiFi να μπαίνει στο αθόρυβο και να ενεργοποιείται ένα ξυπνητήρι πριν πέσουμε για ύπνο. Μια ακόμα λειτουργία θα μπορούσε να είναι ένα NFC Tag στο αυτοκίνητο το οποίο θα είναι προγραμματισμένο να συνδέεται με μία Bluetooth συσκευή για τις κλήσεις και ταυτόχρονα να ανοίγει και την εφαρμογή του GPS.



Η διαδικασία προγραμματισμού των Tags είναι πολύ απλή καθώς οι εφαρμογές παρέχουν όλα τα εργαλεία για να προγραμματιστεί οποιαδήποτε λειτουργία επιθυμούμε με ευκολία.

<b>Τύπος NFC Tag</b>	<b>Σύντομη περιγραφή</b>	<b>Παραδείγματα προϊόντων</b>
NFC Forum Type 1	Βασίζονται στο ISO/IEC 14443A. Μπορούν να διαβαστούν και να γραφούν/ επανεγγραφούν. Μπορούν να προγραμματιστούν και ώστε να είναι μόνο για ανάγνωση. Η διαθέσιμη χωρητικότητα είναι από 96 bytes μέχρι και 2 KB.	Broadcom Topaz
NFC Forum Type 2	Βασίζονται στο ISO/IEC 14443A. Μπορούν να διαβαστούν και να γραφούν/ επανεγγραφούν. Μπορούν να προγραμματιστούν και ώστε να είναι μόνο για ανάγνωση. Η διαθέσιμη χωρητικότητα είναι από 48 bytes μέχρι και 2 KB.	NXP NTAG203, NXP Mifare Ultralight
NFC Forum Type 3	Βασίζονται στο Japanese Industrial Standard (JIS) X 6319-4 που είναι γνωστό και ως FeliCa. Είναι ρυθμισμένα από τον κατασκευαστή να είναι είτε με δυνατότητα επανεγγραφής είτε μόνο για ανάγνωση. Η διαθέσιμη μνήμη διαφέρει με το θεωρητικό όριο να αγγίζει το 1 MB ανά υπηρεσία.	Sony FeliCa
NFC Forum Type 4	Είναι πλήρως συμβατά με το ISO/IEC 14443, το περιβάλλον επικοινωνίας μπορεί να είναι τύπου A ή του τύπου B. Είναι ρυθμισμένα από τον κατασκευαστή να είναι είτε με δυνατότητα επανεγγραφής είτε μόνο για ανάγνωση. Η διαθέσιμη μνήμη διαφέρει και φτάνει τα 32 KB ανά υπηρεσία.	NXP DesFire, NXP SmartFX with JCOP

Πίνακας 3.1: NFC Forum Tag Types

### 3.2.2 Ενεργές Συσσκευές

Οι ενεργές συσκευές μπορούν να παράγουν το ηλεκτρομαγνητικό πεδίο για να υπάρξει επικοινωνία όπως αναφέρθηκε παραπάνω. Υπάρχουν στο εμπόριο συσκευές που συνδέονται και ελέγχονται με τον υπολογιστή, τα NFC Readers/Writers τα οποία μπορούν να διαβάσουν/γράψουν συγκεκριμένα είδη NFC Tags, ανάλογα με το κάθε μοντέλο. Η διαφορά αυτών των συσκευών με

τα κινητά είναι ότι δεν είναι φορητές. Ένα παράδειγμα τέτοιας συσκευής είναι το μηχάνημα στο οποίο πραγματοποιούνται οι ανέπαφες συναλλαγές σε ένα κατάστημα με μία χρεωστική κάρτα.

### 3.3 Είδη λειτουργίας NFC

Το NFC Forum προσδιορίζει τρία είδη λειτουργίας NFC συσκευών οι οποίες ορίζονται ως:

- Reader/Writer Mode
- Peer to Peer Mode
- Card Emulation Mode

**Reader/Writer Mode:** Σε αυτήν την λειτουργία έχουμε επικοινωνία ανάμεσα σε μία ενεργή και μία παθητική συσκευή. Χαρακτηριστικό παράδειγμα είναι ένα κινητό το οποίο διαβάζει ή γράφει ένα NFC Tag. Όταν έρθουν σε κοντινή απόσταση το κινητό τηλέφωνο δημιουργεί το ηλεκτρομαγνητικό πεδίο που βάζει σε λειτουργία το NFC Tag ώστε να ξεκινήσει η ανταλλαγή δεδομένων. Το δεδομένα που ανταλλάσσονται έχουν συνήθως μια συγκεκριμένη δομή, ορισμένη απο το NFC Forum, την NFC Data Exchange Format (NDEF).

**Peer to Peer Mode:** Σε αυτήν την λειτουργία έχουμε επικοινωνία ανάμεσα σε δύο ενεργές συσκευές. Για παράδειγμα δύο κινητά τηλέφωνα τα οποία ανταλλάσσουν πληροφορίες μεταξύ τους. Δεν υποστηρίζεται η αμφίδρομη επικοινωνία, όταν μία συσκευή στέλνει δεδομένα, η άλλη μπορεί μόνο να τα διαβάζει. Το πρωτόκολλο που χρησιμοποιείται λέγεται Logical Link Control Protocol (LLCP) και έχει δημιουργηθεί από το NFC Forum. Οι συσκευές ανταλλάσσουν δεδομένα οποιουδήποτε τύπου, αρκεί να έχουν την δομή NDEF.

**Card Emulation Mode:** Σε αυτήν την λειτουργία έχουμε επίσης δύο ενεργές συσκευές που αλληλεπιδρούν. Η μία από τις συσκευές προσομοιώνει ένα NFC

Tag ενώ η άλλη διαβάζει ή γράφει σε αυτό το εικονικό NFC Tag. Αυτού του είδους η λειτουργία χρησιμοποιείται κυρίως για ηλεκτρονικές πληρωμές, ηλεκτρονικά εισιτήρια ή έλεγχο πρόσβασης. Όταν μια συσκευή προσομοιώνει ένα NFC Tag τα κυριότερα ζητήματα που προκύπτουν αφορούν την ασφάλεια των δεδομένων που υπάρχουν σε αυτό το tag. Για να προστεθεί ένα επιπλέον επίπεδο ασφαλείας στα δεδομένα που αποθηκεύονται στο NFC Tag, οι κατασκευαστές επέλεξαν να ασφαλίσουν τα δεδομένα με την χρήση εξειδικευμένων συσκευών. Οι συσκευές αυτές ονομάζονται Secure Elements

### 3.4 Εφαρμογές της τεχνολογίας NFC

Πλέον η τεχνολογία NFC έχει εισέλθει δυναμικά στον χώρο της τεχνολογίας. Όλα τα τηλέφωνα νέας γενιάς υποστηρίζουν NFC και σταδιακά καλύπτεται και ο τομέας των χρεωστικών και πιστωτικών καρτών. Πρόσφατα παρουσιάστηκαν στην Ελλάδα οι λεγόμενες «ανέπαφες συναλλαγές». Οι συναλλαγές αυτές πραγματοποιούνται φέρνοντας μία κάρτα σε κοντινή απόσταση με μία συσκευή πληρωμής η οποία αυτόματα αφαιρεί το ποσό της συναλλαγής από την κάρτα, προσφέροντας ευκολία και ταχύτητα σε μικρές καθημερινές συναλλαγές καθώς το όριο συναλλαγής, λόγω ασφάλειας, είναι 25 ευρώ.

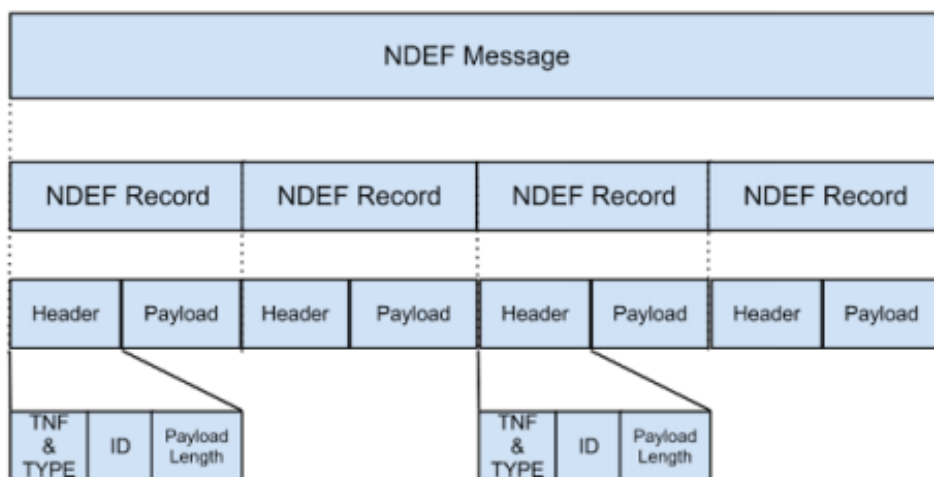
Σε διεθνές επίπεδο τα πράγματα είναι πολύ πιο εξελιγμένα καθώς έχουμε πολλές εφαρμογές που υποστηρίζουν την τεχνολογία NFC και την χρήση ενός λεγόμενου «ψηφιακού πορτοφολιού» το οποίο ουσιαστικά βρίσκεται στο κινητό μας. Πρώτη η Google το 2011 παρουσίασε το Google Wallet το οποίο χρησιμοποιεί NFC για την πραγματοποίηση πληρωμών. Έκτοτε μεγάλες εταιρίες επικοινωνιών στην Αμερική άρχισαν να χρησιμοποιούν παρόμοιες εφαρμογές για πληρωμές.

Σε γενικές γραμμές η χρήση του NFC στην αγορά περιλαμβάνει την πραγματοποίηση πληρωμών. Πέραν των πληρωμών οι χρήστες των συσκευών μπορούν να κάνουν χρήση του NFC με άλλους τρόπους, για ευκολία ή και

διασκέδαση. Όπως προαναφέρθηκε το NFC υποστηρίζει μεταφορά δεδομένων. Στην πράξη η μεταφορά δεδομένων με χρήση NFC δεν είναι συνήθης καθώς ο ρυθμός μετάδοσης υστερεί σε σχέση με τεχνολογίες όπως το Bluetooth. Ακόμα 'ένα εργαλείο το οποίο μας δίνει πολλές δυνατότητες για χρήση του NFC είναι τα NFC Tags. Τα NFC Tags πραγματικά μπορούν να μας προσφέρουν αμέτρητες προοπτικές από το απλό διάβασμα ενός κειμένου, μέχρι την αλλαγή λειτουργιών σε ένα κινητό, τις οποίες θα έχουμε προκαθορίσει εμείς οι ίδιοι

### 3.5 Η δομή NDEF

Το NDEF αποτελεί μια δυαδική δομή δεδομένων που μπορεί να χρησιμοποιηθεί για την ανταλλαγή πληροφοριών μεταξύ NFC συσκευών που υποστηρίζουν τα πρότυπα του NFC Forum. Οι πληροφορίες ενθυλακώνονται και μεταφέρονται στα επονομαζόμενα NDEF μηνύματα. Κάθε NDEF μήνυμα αποτελείται από μία ή περισσότερες εγγραφές που ονομάζονται NDEF records. Στο σχήμα 2.2 φαίνεται η δομή του NDEF μηνύματος καθώς και της NDEF εγγραφής.



Σχήμα 3.2: NDEF Overview

Κάθε NDEF εγγραφή μπορεί να έχει θεωρητικά μέγεθος μέχρι και 4GB, στην πραγματικότητα όμως δεν υπάρχει κανένα NFC Tag που να υποστηρίζει τέτοια χωρητικότητα. Όπως φαίνεται στο σχήμα 2.2 η κεφαλίδα του NDEF record αποτελείται από τρία μέρη, το ID που είναι ένα μοναδικό αναγνωριστικό, το Payload Length που είναι το μέγεθος του φορτίου, και το TNF & Type που υποδεικνύει τον τύπο δεδομένων όπως είναι οι ηλεκτρονικές διευθύνσεις για παράδειγμα, πέραν αυτών όμως μια NDEF εγγραφή μπορεί να περιέχει δεδομένα που δεν ανήκουν στους υποστηριζόμενους τύπους





## ΚΕΦΑΛΑΙΟ 4

### Η ΕΦΑΡΜΟΓΗ

#### 4.1 Περιγραφή της εφαρμογής

Η εφαρμογή που αναπτύχθηκε για την παρούσα πτυχιακή εργασία αφορά την χρήση NFC Tags προκειμένου οι φοιτητές να ενημερώνονται για την βαθμολογία τους. Για την ανάπτυξη της εφαρμογής έγινε χρήση του Android Studio της Google σε γλώσσα προγραμματισμού Java. Η λογική είναι η εξής: Κάθε καθηγητής θα έχει έξω από το γραφείο του ένα NFC Tag με έναν μοναδικό κωδικό. Αφού ανοιχτεί η εφαρμογή και εμφανιστεί η αρχική οθόνη, ο φοιτητής φέρνει το κινητό του σε κοντινή απόσταση με το Tag και αυτόματα εμφανίζονται στην οθόνη κουμπιά με τα μαθήματα του συγκεκριμένου καθηγητή. Στην συνέχεια, αφού ο φοιτητής επιλέξει το μάθημα που τον αφορά, γράφει τον αριθμό μητρώου του και επιλέγοντας την αναζήτηση θα ανοίξει ένα παράθυρο που θα τον ενημερώνει για την βαθμολογία του στο συγκεκριμένο μάθημα. Στην περίπτωση που δεν υπάρχει βαθμολογία για τον συγκεκριμένο αριθμό μητρώου εμφανίζεται το κατάλληλο μήνυμα.

Για τον προγραμματισμό των Tags απαιτείται η ύπαρξη μίας εφαρμογής οποία δίνει αυτή τη δυνατότητα στον χρήστη. Υπάρχουν πολλές δωρεάν εφαρμογές στο Play Store της Google για να επιλεξει κανείς ανάλογα με τις απαιτήσεις του. Στην εφαρμογή που υλοποιήθηκε για την συγκεκριμένη πτυχιακή εργασία έγινε χρήση της εφαρμογής NFC Tools και τα Tags προγραμματίστηκαν ώστε να περιέχουν έναν κωδικό ο οποίος μόλις διαβαστεί από την εφαρμογή θα παραπέμπει τον χρήστη στα μαθήματα του εκάστοτε καθηγητή. Στην συνέχεια τα Tags κλειδώθηκαν έτσι ώστε να μην μπορεί κάποιος να γράψει εκ νέου πάνω σε αυτά για να επηρεάσει την εφαρμογή.



Οι βαθμολογίες των φοιτητών βρίσκονται στο Google Drive και τα αρχεία διαχειρίζονται από έναν λογαριασμό της σχολής. Κάθε μάθημα έχει το δικό του αρχείο το οποίο περιέχει δύο στήλες, μία με τον αριθμό μητρώου και μία με τον βαθμό. Συνεπώς όταν επιλεγεί ένα συγκεκριμένο μάθημα η εφαρμογή κατεβάζει το αντίστοιχο αρχείο στο background σε μορφή CSV. Στην συνέχεια τα δεδομένα του αρχείου τοποθετούνται σε έναν δισδιάστατο πίνακα όπου και γίνεται η αναζήτηση του φοιτητή. Με τον τρόπο αυτό αποφεύγουμε να γίνεται update στην εφαρμογή κάθε εξάμηνο με την αλλαγή των βαθμολογιών και με την αλλαγή μόνο των στοιχείων στο αρχείο του Google Drive πετυχαίνουμε το επιθυμητό αποτέλεσμα. Τέλος με την επιστροφή στην αρχική οθόνη της εφαρμογής διαγράφεται το κατεβασμένο αρχείο για να μην επιβαρύνεται η μνήμη του τηλεφώνου.

### 4.2 Η υλοποίηση της εφαρμογής

Ξεκινώντας την εφαρμογή, όπως βλέπουμε στο σχήμα 4.1, παρουσιάζονται τα στοιχεία της εφαρμογής. Στην συνέχεια πρέπει ο χρήστης να φέρει το κινητό του κοντά σε ένα NFC Tag ενός καθηγητή. Ουσιαστικά το κάθε Tag παραπέμπει σε ένα διαφορετικό Layout και κάθε καθηγητής έχει το δικό του Layout που εμφανίζεται με την ανάγνωση του συγκεκριμένου NFC Tag. Ταυτόχρονα με την εμφάνιση του Layout που επιθυμούμε, εξαφανίζονται και όλα τα υπόλοιπα Layouts και το ImageView της αρχικής σελίδας. Θα δούμε την χρησιμότητα αυτής της υλοποίησης παρακάτω.

Επίσης κατά την εκκίνηση της εφαρμογής θα ζητηθεί η ενεργοποίηση του NFC, αν αυτό δεν είναι ενεργοποιημένο, όπως φαίνεται στο σχήμα 4.2. Επιλέγοντας τις ρυθμίσεις, ενεργοποιούμε το NFC και επιστρέφουμε στην εφαρμογή όπου είμαστε έτοιμοι να διαβάσουμε τα διαθέσιμα NFC Tags για κάθε καθηγητή. Τα Layout των καθηγητών τα βλέπουμε παρακάτω, στα σχήματα 4.3 – 4.17.



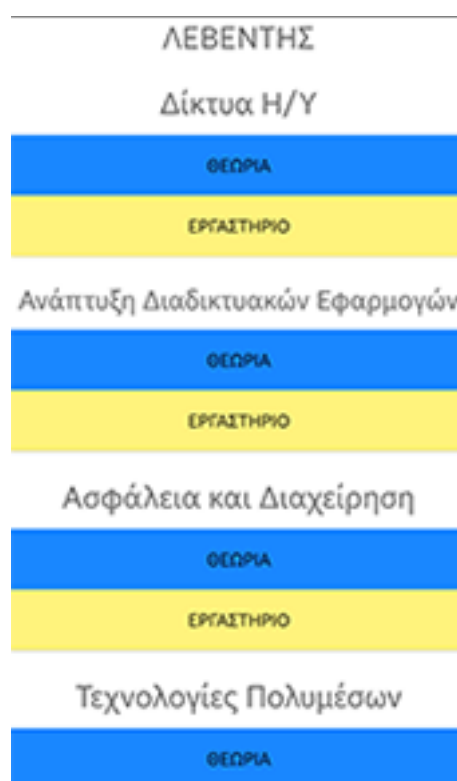
Σχήμα 4.1: Αρχική οθόνη εφαρμογής



Σχήμα 4.2: Ειδοποίηση ενεργοποίησης NFC



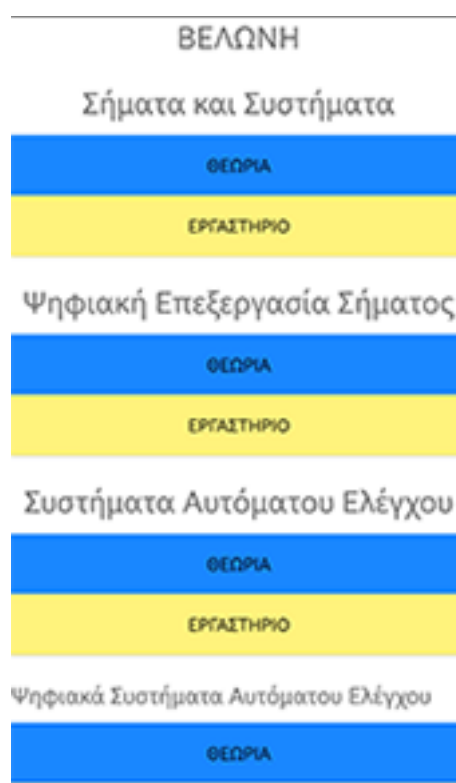
Σχήμα 4.3: Layout κ. Έλληνα



Σχήμα 4.4: Layout κ. Λεβέντη



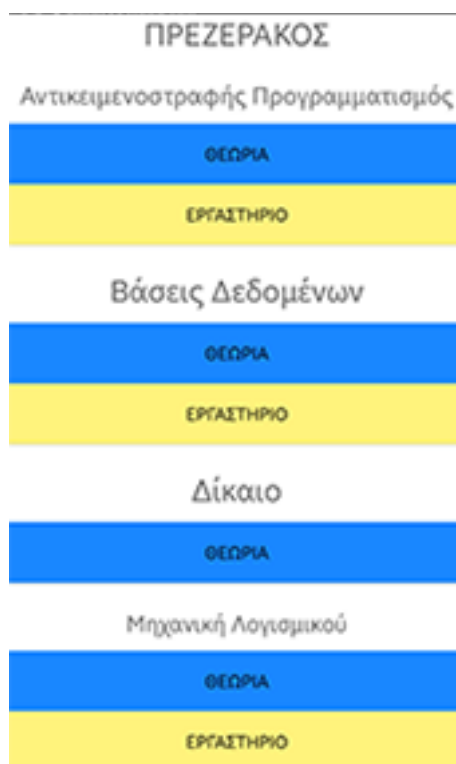
Σχήμα 4.5: Layout κ. Ζάχαρη



Σχήμα 4.6: Layout κ. Βελώνη



Σχήμα 4.7: Layout κ. Τσελίκας



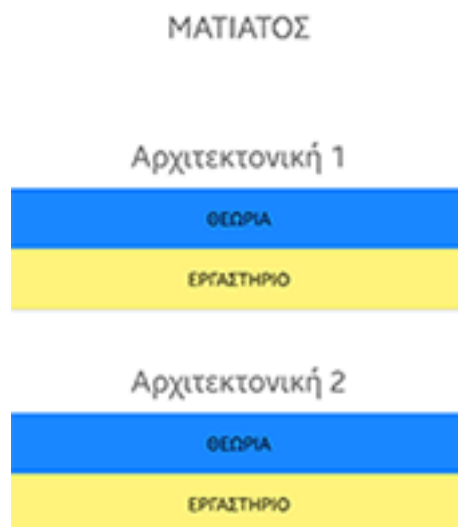
Σχήμα 4.8: Layout κ. Πρεζεράκου



Σχήμα 4.9 Layout κ. Παλιατσού



Σχήμα 4.10: Layout κ. Νικολόπουλου



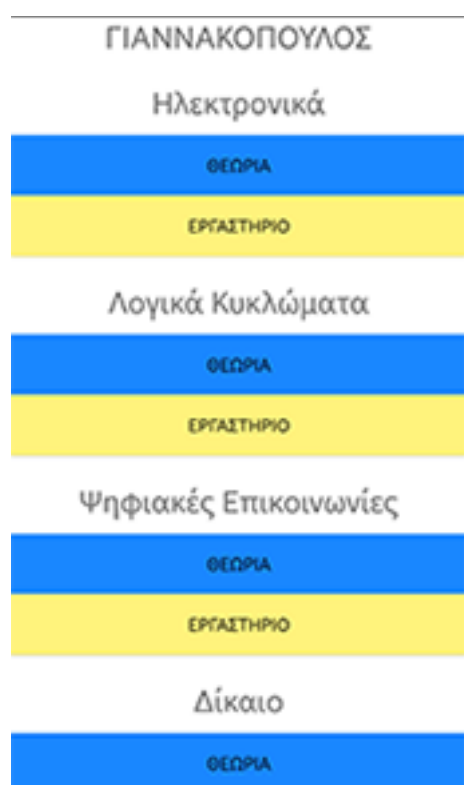
Σχήμα 4.11: Layout κ. Ματιάτου



Σχήμα 4.12: Layout κ. Κούρου



Σχήμα 4.13: Layout κ. Κουκουλέτσου



Σχήμα 4.14: Layout κ. Γιαννακόπουλου

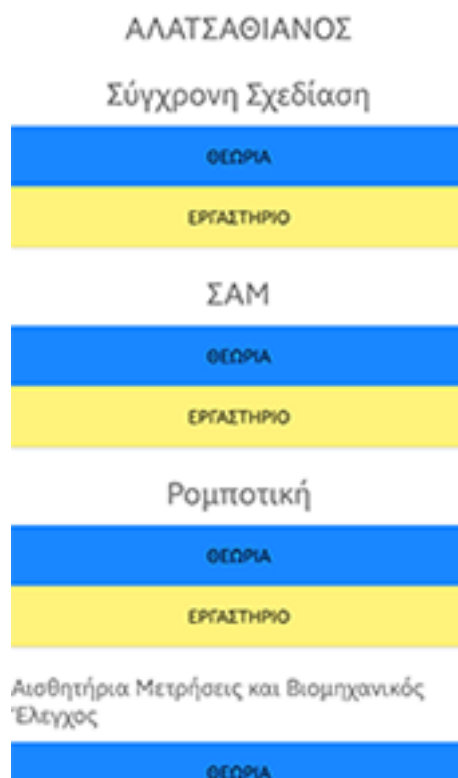




Σχήμα 4.15: Layout κ. Φατούρου.



Σχήμα 4.16: Layout κ. Διλιντά



Σχήμα 4.17: Layout κ. Αλατσαθιανού

Όλα τα παραπάνω υλοποιούνται μέσω ενός κειμένου xml το οποίο ονομάζεται `activity_main.xml`. Στο xml κείμενο αυτό έχουμε ορίσει οτιδήποτε έχει να κάνει με την εμφάνιση της εφαρμογής μας, έως τώρα ,καθώς υπάρχει και άλλο ένα xml κείμενο για την αναζήτηση. Επομένως η αρχική οθόνη και η εμφάνιση όλων των Layouts καθηγητών μαζί με όλα τα στοιχεία εντός του Layout (κείμενα,κουμπια,ορατότητα) δημιουργούνται στο xml κείμενο `activity_main.xml`.

Στο σχήμα 4.18 βλέπουμε την δήλωση του αρχικού `ImageView` εντός του xml κειμένου και στο σχήμα 4.19 την δήλωση του `Linear Layout` για τον κ. Τσελίκα και θα αναλύσουμε τις επιμέρους εντολές.

```
<ImageView
    android:id="@+id/main"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:weightSum="1"

    android:background="@drawable/splash"

>
</ImageView>
```

Σχήμα 4.18: Δήλωση ImageView

Όπως βλέπουμε είναι απαραίτητο να ορίσουμε ένα id το οποίο θα ξεχωρίζει το στοιχείο μας από τα υπόλοιπα στοιχεία. Στην συνέχεια ορίζουμε τις διαστάσεις του View και τέλος ορίζουμε μία εικόνα ως εικόνα background. Μία εντολή την οποία δεν βλέπουμε εδώ αλλά υπάρχει σε όλα τα Linear Layouts είναι η εντολή `android:visibility="gone"`. Αυτό γίνεται γιατί κατά την εκκίνηση της εφαρμογής θέλουμε να βλέπουμε την αρχική εικόνα και όχι τα Linear Layouts. Μετά όταν θελήσουμε να δούμε το Layout κάθε καθηγητή το `visibility` της αρχικής οθόνης θα αλλάξει, κάνοντας την εικόνα να «εξαφανιστεί».

```
<LinearLayout
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:weightSum="1"
    android:id="@+id/TS"
    android:visibility="gone">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="ΤΣΕΛΙΚΑΣ"
        android:id="@+id/LabelTS"
        android:layout_gravity="center_horizontal"
        android:textSize="24dp"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Τεχνολογίες Ευρυζωνικών Δικτύων"
        android:id="@+id/ted"
        android:textSize="20dp"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="200dp"
    />

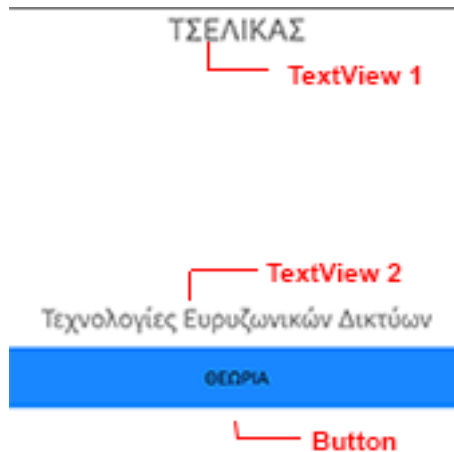
    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_weight="0.05"
        android:text="ΘΕΩΡΙΑ"
        android:id="@+id/btnted"

        android:layout_marginTop="10dp"
        android:background="#ff1987ff"
    />
</LinearLayout>
```

Σχήμα 4.19: Linear Layout κ. Τσελικά σε μορφή xml

Αναλύοντας το xml κείμενο για το Layout του κ. Τσελικά βλέπουμε αρχικά την δήλωση ενός νέου Linear Layout και στην συνέχεια τα στοιχεία που το χαρακτηρίζουν. Τα στοιχεία orientation, layout\_width, layout\_height και weightSum αφορούν τις διαστάσεις του Layout. Παρακάτω ορίζουμε το id του Layout (ονομάζουμε το συγκεκριμένο TS) προκειμένου μετά να μπορούμε να χειριστούμε το πότε θέλουμε να το βλέπουμε και πότε όχι. Στην συνέχεια ορίζουμε το visibility ως gone για να μην εμφανίζεται το συγκεκριμένο Layout αρχικά. Επίσης βλέπουμε και την δήλωση του TextView με το όνομα του καθηγητή, στην συνέχεια την δήλωση ενός TextView με το όνομα του μαθήματος και τέλος την

δήλωση του κουμπιού που χρησιμοποιείται μαζί με τις διαστάσεις του καθώς και ένα μοναδικό id για κάθε κουμπί. Παρακάτω στο σχήμα 4.20 βλέπουμε την αντιστοίχιση των στοιχείων του layout με το xml κείμενο του σχήματος 4.19



Σχήμα 4.20: Αντιστοίχιση στοιχείων με το xml κείμενο

Είδαμε λοιπόν πως ορίζουμε ένα Linear Layout. Η διαδικασία επαναλαμβάνεται 14 ακόμα φορές για τα Layouts των υπόλοιπων καθηγητών. Η συνέχεια της διαδικασίας αποτελεί και το δύσκολο κομμάτι της εφαρμογής. Αυτό που θέλουμε είναι να κάνουμε την σύνδεση του xml κειμένου activity\_main με την κύρια κλάση της εφαρμογής MainActivity.java. Αρχικά στην MainActivity πρέπει να δηλώσουμε όλα τα κουμπιά και να τα συσχετίσουμε με τα κουμπιά στο xml κείμενο και να δηλώσουμε αντίστοιχα και όλα τα Layouts και να τα συσχετίσουμε με τα Layouts του xml κειμένου. Στα σχήματα 4.21 και 4.22 βλέπουμε την αρχική δήλωση των κουμπιών και των Linear Layouts και στην συνέχεια θα δουμε και την συσχέτιση τους.

```
Button btnarch1th;  
Button btnarch2th;  
Button btneeth;  
Button btnarch1erg;  
Button btnarch2erg;  
Button btneeerg;  
Button btnssth;  
Button btnsserg;  
Button btnsamth;  
Button btnsamerg;  
Button btnrobth;  
Button btnroberg;  
Button btnamvth;  
Button btnamverg;  
Button btnmixth;  
Button btnmixerg;  
Button btnmath;  
Button btnefmth;  
Button btnfysth;  
Button btnfyserg;  
Button btnpseth;  
Button btnpseerg;  
Button btnvbth;  
Button btnvberg;  
Button btndomth;  
Button btndomerg;
```

Σχήμα 4.21: Δήλωση κουμπιών

```
String selectedLesson;  
NfcAdapter nfcAdpt;  
PendingIntent pendingIntent;  
LinearLayout EL;  
LinearLayout FT;  
LinearLayout AL;  
LinearLayout NK;  
LinearLayout KK;  
LinearLayout TS;  
LinearLayout GN;  
LinearLayout VL;  
LinearLayout DR;  
LinearLayout KR;  
LinearLayout PZ;  
LinearLayout MT;  
LinearLayout DL;  
LinearLayout LV;  
LinearLayout PL;  
LinearLayout ZX;  
ImageView main;
```

Σχήμα 4.22: Δήλωση Linear Layouts και μεταβλητών

Στην συνέχεια κάνουμε την σύνδεση της κλάσης MainActivity με το xml κείμενο activity\_main καθώς και τον έλεγχο για την χρήση του NFC. Ελέγχεται αν η συσκευή υποστηρίζει NFC και αν είναι ενεργοποιημένο. Σε περίπτωση που δεν είναι ενεργοποιημένο εμφανίζεται ένα κείμενο το οποίο παραπέμπει τον χρήστη στις ρυθμίσεις όπως είδαμε στο σχήμα 4.2. Αυτό υλοποιείται μέσω της μεθόδου onCreate μαζί με μερικές ακόμα λειτουργίες όπως βλέπουμε στο σχήμα 4.23.

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    nfcAdpt = NfcAdapter.getDefaultAdapter(this);
    if (nfcAdpt != null) {
        if (nfcAdpt.isEnabled()) {
        } else {
            AlertDialog.Builder builder = new AlertDialog.Builder(
                MainActivity.this);
            builder.setMessage(
                "Για να χρησιμοποιήσεις την εφαρμογή πρέπει το NFC να είναι ενεργοποιημένο.")
                .setCancelable(false)
                .setPositiveButton("Ρυθμίσεις",
                    (dialog, id) -> {
                        if (android.os.Build.VERSION.SDK_INT >= 16) {
                            startActivity(new Intent(
                                android.provider.Settings.ACTION_NFC_SETTINGS));
                        } else {
                            startActivity(new Intent(
                                android.provider.Settings.ACTION_WIRELESS_SETTINGS));
                        }
                    });
            builder.setNegativeButton("Ακύρωση",
                (dialog, id) -> { dialog.cancel(); });
            AlertDialog alert = builder.create();
            alert.show();
        }
    } else {
        Toast.makeText(
            MainActivity.this,
            "Για να χρησιμοποιήσεις την εφαρμογή η συσκευή σου θα πρέπει να υποστηρίζει NFC τεχνολογία",
            Toast.LENGTH_LONG).show();
    }
}

```

Σχήμα 4.23: Η μέθοδος onCreate (MainActivity.java)

Η μέθοδος onCreate είναι μια μέθοδος που παράγεται αυτόματα στην δημιουργία της κλάσης. Σημαντικό κομμάτι κώδικα είναι η εντολή setContentView(R.layout.activity\_main) η οποία συνδεεί την κλάση μας με το xml κείμενο, δηλαδή η MainActivity «βλέπει» πλέον στην activity\_main.xml. Η μεταβλητή nfcAdpt ελέγχει για την χρήση του NFC και εισάγεται ως παράμετρος σε ένα if για να παρουσιαστούν τα κατάλληλα μηνύματα σε περίπτωση που το NFC δεν είναι ανοιχτό ή δεν υποστηρίζεται. Σε περίπτωση που το NFC λειτουργεί κανονικά το πρόγραμμα συνεχίζει χωρίς κάποια αλλαγή.

Σημαντικό κομμάτι σε αυτό το σημείο παίζει ο ορισμός των μαθημάτων και η σύνδεση του καθε μαθήματος με το αντίστοιχο link στο Google Drive, όπως και η σύνδεση των κουμπιών και των Linear Layouts με τα xml.



```
//Tselikas
TS = (LinearLayout) findViewById(R.id.TS);
btnted = (Button) findViewById(R.id.btnted);
btnted.setOnClickListener((v28) -> {
    selectedLesson = "ted";
    new Download().execute();
});
```

Σχήμα 4.24: Αντιστοίχιση των Linear Layouts και των κουμπιών με τα xml

Συνεχίζουμε τα παραδείγματα με το Layout του κ. Τσελίκας λόγω του ενός μαθήματος που έχει. Στο σχήμα 4.24 βλέπουμε το Linear Layout TS να συνδέεται με το αντίστοιχο TS του xml μέσω της εντολής `findViewById(R.id.TS)`. Το ίδιο ισχύει και για το κουμπί `btnted` το οποίο συνδέεται με το αντίστοιχο κουμπί με `id btnted` μέσω της εντολής `findViewById(R.id.btnted)`. Στην συνέχεια ορίζουμε στο κουμπί `btnted` ένα `OnClickListener` δηλαδή μία ενέργεια μόλις πατηθεί. Οι ενέργειες είναι δύο. Η μεταβλητή `selectedLesson` παίρνει την τιμή "ted" και καλείται η κλάση `Download` η οποία ξεκινά να κατεβάζει το αρχείο μας λαμβάνοντας ως παράμετρο το `selectedLesson` (στην περίπτωση μας το μάθημα με κωδικό `ted`: Τεχνολογίες Ευρυζωνικών Δικτύων). Η ίδια διαδικασία επαναλαμβάνεται για κάθε `Linear Layout` (καθηγητές) και για κάθε κουμπί (μαθήματα) που υπάρχουν στην εφαρμογή.

```
public class Download extends AsyncTask<Void, Void, Void> {
    @Override
    protected Void doInBackground(Void... params) {
        try {
            downloadFile(selectedLesson);
        } catch (IOException e) {
            e.printStackTrace();
        }
        return null;
    }
}
```

Σχήμα 4.25: Η κλάση Download

Η κλάση Download λαμβάνει ως παράμετρο το `selectedLesson` και καλεί την μέθοδο `downloadFile` (σχήμα 4.25) η οποία πέρνει ως παράμετρο το `selectedLesson` με την ονομασία `lesson` και με ένα `if statement` βρίσκει το σωστό μάθημα μαζί με το `link` του αρχείου του (σχήμα 4.26) για να ξεκινήσει το `download` με τον κώδικα το `URLConnection` του σχήματος 4.27

```
private void downloadFile(String lesson) throws IOException {
    URL url;
    if (lesson.equalsIgnoreCase("arx1")) {
        url = new URL("https://docs.google.com/spreadsheets/d/1ZJya16QkCpwTURuH3G9qcLWildYcT9_w-tAa3308Tz4/export?format=csv");
    } else if (lesson.equalsIgnoreCase("arx2")) {
        url = new URL("https://docs.google.com/spreadsheets/d/1ytrGtwv36THldEshltMPEmeLg2dVukKHUdxJqU85Ti8/export?format=csv");
    } else if (lesson.equalsIgnoreCase("epe3")) {
        url = new URL("https://docs.google.com/spreadsheets/d/13ViM560aoqeg-YBGNaIpnK7N1o1qsJUnHazsa00k9fA/export?format=csv");
    } else if (lesson.equalsIgnoreCase("ss")) {
        url = new URL("https://docs.google.com/spreadsheets/d/1MN75LuxMsDns1Vb2pbwrK_QnNFyG5Vs--spPBkib_cvA/export?format=csv");
    }
}
```

Σχήμα 4.26: Η μέθοδος `downloadFile`

```
try {
    URLConnection ucon = url.openConnection();
    ucon.setReadTimeout(5000);
    ucon.setConnectTimeout(10000);

    InputStream is = ucon.getInputStream();
    BufferedInputStream inStream = new BufferedInputStream(is, 1024 * 5);

    File file = new File(getCacheDir(), lesson);

    if (file.exists()) {
        file.delete();
    }
    file.createNewFile();

    FileOutputStream outputStream = new FileOutputStream(file);
    byte[] buff = new byte[5 * 1024];

    int len;
    while ((len = inStream.read(buff)) != -1) {
        outputStream.write(buff, 0, len);
    }

    outputStream.flush();
    outputStream.close();
    inStream.close();
} catch (Exception e) {
    e.printStackTrace();
}
```

Σχήμα 4.27: URL Connection

Προκειμένου να γίνει εμφάνιση όλων αυτών βέβαια έχουμε την αναγνώριση του NFC Tag. Η αναγνώριση αυτή γίνεται με ένα if statement το οποίο ελέγχει τι περιέχει το NFC Tag και αναλόγως λαμβάνει τις απαραίτητες ενέργειες. Για παράδειγμα όταν αναγνωριστεί το Tag του κ. Τσελίκας θέλουμε να εξαφανίσουμε οτιδήποτε άλλο και να εμφανίσουμε το δικό του Layout μόνο. Αυτό φαίνεται στο σχήμα 4.28 και αντίστοιχα γίνεται το ίδιο με τα Layout άλλων καθηγητών. Ορίζουμε δηλαδή το visibility του layout που θέλουμε ως VISIBLE αλλάζοντας την κατάσταση του από gone που ήταν η default δήλωση στο xml κείμενο όπως είδαμε παραπάνω.

```
}else if (body.equalsIgnoreCase("tselikas")){  
    main.setVisibility(View.GONE);  
    KR.setVisibility(View.GONE);  
    DR.setVisibility(View.GONE);  
    VL.setVisibility(View.GONE);  
    GN.setVisibility(View.GONE);  
    AL.setVisibility(View.GONE);  
    FT.setVisibility(View.GONE);  
    NK.setVisibility(View.GONE);  
    KK.setVisibility(View.GONE);  
    EL.setVisibility(View.GONE);  
    PZ.setVisibility(View.GONE);  
    MT.setVisibility(View.GONE);  
    DL.setVisibility(View.GONE);  
    LV.setVisibility(View.GONE);  
    PL.setVisibility(View.GONE);  
    ZX.setVisibility(View.GONE);  
    TS.setVisibility(View.VISIBLE);  
}
```

Σχήμα 4.28: Ανάγνωση Tag και εμφάνιση του κατάλληλου layout

Έτσι μόλις έχουμε εμφάνιση του layout επιλέγουμε το μάθημά μας και ξεκινάει η διαδικασία που περιγράψαμε παραπάνω. Το τελικό στάδιο με την αναζήτηση της βαθμολογίας του φοιτητή υλοποιείται μέσω της κλάσης Search1.java και του xml κειμένου search1.xml. Το layout του search1.xml το βλέπουμε στο σχήμα 4.29. Στην συνέχεια ο φοιτητής γράφει τον αριθμό μητρώου του και εμφανίζεται μήνυμα ειδοποίησης σε περίπτωση που ο βαθμός δεν βρεθεί(Σχήμα 4.30) ή

Ανάπτυξη εφαρμογής Android με χρήση NFC

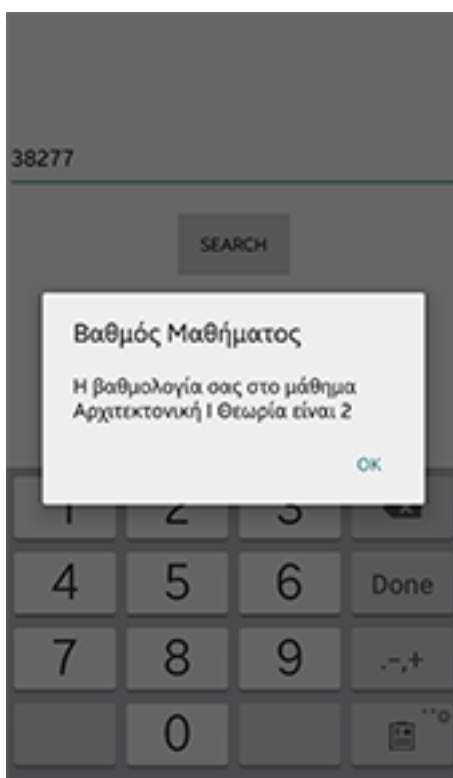
εμφανίζεται ένα παράθυρο το οποίο δείχνει το βαθμό του φοιτητή στο προεπιλεγμένο μάθημα(Σχημα 4.31).



Σχήμα 4.29: Το layout της αναζήτησης



Σχήμα 4.30: Ειδοποίηση ότι δεν βρεθηκε ο αριθμός μητρώου



Σχήμα 4.31: Εμφάνιση βαθμολογίας φοιτητή

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:weightSum="1"
    android:id="@+id/lala"
    android:background="@android:color/background_light">

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="number"
        android:ems="10"
        android:id="@+id/txt1"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="100dp" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Search"
        android:id="@+id/btnSearch1"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="20dp"
        android:background="@color/button_material_light" />

</LinearLayout>
```

Σχήμα 4.32: Το layout search1.xml

Όπως βλέπουμε στο σχήμα 4.32, το xml κείμενο της Search1 είναι κατά πολύ μικρότερο από το προηγούμενο. Ουσιαστικά έχουμε μόνο την δήλωση EditText για να εισαχθεί ο αριθμός μητρώου, με περιορισμό στο android:inputType="number" για να δέχεται μόνο αριθμούς και την δήλωση του κουμπιού Search για να δωθεί η εντολή αναζήτησης.

Η διαδικασία σε αυτό το σημείο της εφαρμογής είναι αρκετά πιο απλή από το προηγούμενο στάδιο. Έχει γίνει λήψη του επιθυμητού CSV αρχείου για το μάθημα που επιθυμούμε και πραγματοποιείται η μεταφορά των δεδομένων του σε έναν δισδιάστατο πίνακα με όνομα lista και στην συνέχεια γίνεται ο διαχωρισμός σε δύο ακόμα πίνακες, έναν με όνομα AM και έναν με όνομα grades, όπου βρίσκουμε τον φοιτητή με την αντίστοιχη βαθμολογία του. Η διαδικασία αυτή πραγματοποιείται με μία μέθοδο η οποία ονομάζεται readCsv και την βλέπουμε στο σχήμα 4.33.

```
public final void readCsv(String file) throws IOException {
    String[] next = {};

    File f = new File(getCacheDir(), file);
    FileInputStream fileInputStream = new FileInputStream(f);
    CSVReader reader1 = new CSVReader(new InputStreamReader(fileInputStream));

    for (; ; ) {
        next = reader1.readNext();
        if (next != null) {
            lista.add(next);
        } else {
            break;
        }
    }
    reader1.close();
    for (int i = 0; i < lista.size(); i++) {
        if (lista.get(i).length > 1) {
            AM.add(lista.get(i)[0]);
            grades.add(lista.get(i)[1]);
        }
    }
}
```

Σχήμα 4.33: Η μέθοδος readCsv

Η μέθοδος readCsv λαμβάνει ως παράμετρο το αρχείο που λήφθηκε από την μέθοδο downloadFile, την οποία χρησιμοποιήσαμε στην προηγούμενη κλάση και κατέβασε το κατάλληλο αρχείο με το όνομα του επιθυμητού μαθήματος. Στην συνέχεια η μέθοδος readCsv τοποθετεί τα δεδομένα του αρχείου σε έναν πίνακα με το όνομα lista και διαχωρίζουμε τον πίνακα σε δύο επιμέρους πίνακες, έναν για τους αριθμούς μητρώου με όνομα AM και έναν για τους βαθμούς με όνομα grades. Οι πίνακες αυτοί δηλώνονται στην αρχή της κλάσης Search1 μαζί με δύο ακόμα μεταβλητές. Η μία ονομάζεται lesson η οποία αρχικά είναι κενή και συμπληρώνεται με το μάθημα που επιλέγεται και μια Boolean μεταβλητή η οποία είναι αρχικά false και παρακάτω πέρνει την τιμή true μόνο αν βρεθεί ο επιθυμητός αριθμός μητρώου, για να διακοπεί ο βρόγχος αναζήτησης, όπως βλέπουμε στο σχήμα 4.34

```
public class search1 extends Activity {  
    // String filename = "";  
    ArrayList<String[]> lista;  
    ArrayList<String> AM;  
    ArrayList<String> grades;  
  
    String lesson = "";  
    boolean found = false;
```

Σχήμα 4.34: Δήλωση πινάκων και μεταβλητών

Η αναζήτηση του αριθμού μητρώου γίνεται στην μέθοδο onCreate(σχήμα 4.35), όπως επίσης και η σύνδεση της κλάσης με το αντίστοιχο xml layout της αναζήτησης. Γίνεται κλήση της readCsv με το επιθυμητό μάθημα και στην συνέχεια ένας βρόγχος αναζητά στον πίνακα AM την τιμή που εισήγαγε ο φοιτητής. Αν δεν βρεθεί ο αριθμός μητρώου εμφανίζεται το κατάλληλο μήνυμα. Στην περίπτωση που βρεθεί παίρνουμε τον βαθμό του φοιτητή από την αντίστοιχη θέση του πίνακα grades και διακόπτεται η διαδικασία αναζήτησης αλλάζοντας την boolean μεταβλητή found σε true. Δηλαδή αν ο αριθμός μητρώου βρεθεί στην 5<sup>η</sup> θέση του πίνακα AM, «τραβάμε» το στοιχείο από την 5<sup>η</sup> θέση του πίνακα grades και καλείται η μέθοδος createpopup με παράμετρο εισόδου την βαθμολογία.



```

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.search1);
    lista = new ArrayList<String[]>();
    AM = new ArrayList<String>();
    grades = new ArrayList<String>();

    if (getIntent().hasExtra("file")) {
        lesson = getIntent().getStringExtra("file");
    }
    try {
        readCsv(lesson);
    } catch (IOException e) {
        e.printStackTrace();
    }
    Button btnSearch1 = (Button) findViewById(R.id.btnSearch1);
    btnSearch1.setOnClickListener((view1) -> {
        EditText st1 = (EditText) findViewById(R.id.txt1);
        String searchVariable = st1.getText().toString();
        for (int i = 0; i < AM.size(); i++) {
            if (searchVariable.equalsIgnoreCase(AM.get(i))) {
                createpopup(grades.get(i));
                found = true;
                break;
            }
        }

        if (!found) {
            Toast.makeText(search1.this, "Δεν βρέθηκε εγγραφή για τον συγκεκριμένο αριθμό μητρώου", Toast.LENGTH_LONG).show();
        }
        if (grades.isEmpty()){
            Toast.makeText(search1.this, "Δεν έχει περαστεί ο βαθμός σας", Toast.LENGTH_LONG).show();
        }
    });
}

```

Σχήμα 4.35: Η μέθοδος onCreate (Search1.java)

Στην μέθοδο createpopup πραγματοποιείται το τελικό στάδιο της εφαρμογής. Παίρνει ως είσοδο την βαθμολογία του φοιτητή που βρέθηκε στην onCreate και με ένα if statement ελέγχει ποιό είναι το επιλεγμένο μάθημα. Όταν αυτό βρεθεί δημιουργείται ένα popup παράθυρο το οποίο εμφανίζει ένα κείμενο που λείει το μάθημα και την βαθμολογία του φοιτητή(σχήμα 4.36).

```

private void createpopup(String vathmologia) {
    AlertDialog.Builder builder = new AlertDialog.Builder(
        search1.this);
    builder.setTitle("Βαθμός Μαθήματος");
    builder.setCancelable(false);
    if (lesson.equalsIgnoreCase("arx1")) {
        builder.setMessage("Η βαθμολογία σας στο μάθημα Αρχιτεκτονική I θεωρία είναι " + vathmologia);
    } else if (lesson.equalsIgnoreCase("arx2")) {
        builder.setMessage("Η βαθμολογία σας στο μάθημα Αρχιτεκτονική II θεωρία είναι " + vathmologia);
    } else if (lesson.equalsIgnoreCase("epe3")){
        builder.setMessage("Η βαθμολογία σας στο μάθημα Επεξεργασία Εικόνας θεωρία είναι " + vathmologia);
    }else if (lesson.equalsIgnoreCase("ss")){
        builder.setMessage("Η βαθμολογία σας στο μάθημα Σύγχρονη Σχεδίαση θεωρία είναι " + vathmologia);
    }else if (lesson.equalsIgnoreCase("sam")){
        builder.setMessage("Η βαθμολογία σας στο μάθημα ΣΑΜ θεωρία είναι " + vathmologia);
    }else if (lesson.equalsIgnoreCase("rob")){
        builder.setMessage("Η βαθμολογία σας στο μάθημα Ρομποτική θεωρία είναι " + vathmologia);
    }else if (lesson.equalsIgnoreCase("amv")){
        builder.setMessage("Η βαθμολογία σας στο μάθημα Αισθητήρια Μετρήσεις & Βιομηχανικός Έλεγχος θεωρία είναι " + vathmologia);
    }else if (lesson.equalsIgnoreCase("mix")){

```

Σχήμα 4.36: Η μέθοδος createpopup



## ΚΕΦΑΛΑΙΟ 5

### ΣΥΜΠΕΡΑΣΜΑΤΑ – ΜΕΛΛΟΝΤΙΚΕΣ ΕΡΓΑΣΙΕΣ

#### 5.1 Συμπεράσματα

Τα συμπεράσματα τα οποία έβγαλα δημιουργώντας την εφαρμογή και αναζητώντας πληροφορίες για την τεχνολογία NFC είναι πως πρόκειται για μια πολύ ενδιαφέρουσα μορφή τεχνολογίας με πάρα πολλές εφαρμογές. Η κύρια εξυπηρέτηση του NFC είναι η αυτοματοποίηση πολλών καθημερινών λειτουργιών με ένα απλό «σκανάρισμα» του κινητού μας. Ήδη η αγορά έχει κατακλυστεί από συσκευές που χρησιμοποιούν NFC και εκτιμάται ότι θα συνεχίσει να αναπτύσσεται ραγδαία η χρήση του. Αυτό οφείλεται στην ασφάλεια που παρέχει στον απλό χρήστη, μιας και το σήμα του είναι πάρα πολύ δύσκολο να υποκλαπεί, αλλά και στις διαφορετικές λειτουργίες που παρέχει σχετικά με παρόμοιες τεχνολογίες όπως το Bluetooth και το RFID. Για παράδειγμα θα ήταν αδύνατον να πραγματοποιούμε οικονομικές συναλλαγές με χρήση Bluetooth, μιας και η εμβέλεια του δημιουργεί τις προϋποθέσεις για υποκλοπές.

Τέλος, η προσωπική μου εκτίμηση, είναι ότι μόλις αρχίσει η ευρεία διάδοση του NFC, ειδικά στην Ελλάδα, θα αποτελέσει έναν τομέα πάνω στον οποίο θα στηριχθούν πολλά projects. Επίσης θα εξυπηρετήσει πολύ σε καθημερινές λειτουργίες ενός χρήστη με τις αμέτρητες δυνατότητες που παρέχει, όπως προαναφέρθηκε, με τα NFC Tags, είτε για πραγματική ευκολία, είτε για απλή διασκέδαση.

#### 5.2 Μελλοντικές εργασίες

Η τεχνολογία NFC μπορεί μέσα στα επόμενα χρόνια να βρεί πολλές εφαρμογές στις αγορές καθώς και σε καθημερινές δραστηριότητες. Ένα θέμα προς ανάπτυξη

είναι η αγορά ηλεκτρονικού εισιτηρίου Μέσων Μαζικής Μεταφοράς. Μέσω μιας εφαρμογής ο χρήστης θα μπορεί να αγοράζει μονάδες τις οποίες θα εξαργυρώνει κατά την είσοδό του σε κάποιο μέσο περνώντας το κινητό πάνω από μια NFC συσκευή η οποία θα αφαιρεί το αντίτιμο του εισιτηρίου και θα εμφανίζει στο κινητό την απόδειξη αγοράς εισιτηρίου. Στο σενάριο που υπάρχει είσοδος με μπάρκες θα μπορεί ακόμα με το πέρασμα του κινητού να ανοίγει την μπάρκα.

Άλλη εφαρμογή με χρήση NFC θα μπορούσε να είναι ένα παρουσιολόγιο τάξης όπου κάθε φοιτητής εισερχόμενος στην αίθουσα θα περνάει το κινητό του πάνω από μια συσκευή NFC δηλώνοντας έτσι την φυσική του παρουσία. Παρόμοιο παράδειγμα είναι μια NFC κλειδαριά, σε μία εταιρία, η οποία θα περιέχει τα προφίλ των εργαζομένων και προκειμένου να εισέλθουν θα πρέπει να κάνουν ένα πέρασμα με το κινητό τους. Κατ'επέκταση αυτό μπορεί να συνοδευτεί με ένα μήνυμα καλωσορίσματος και με τον έλεγχο της ώρας άφιξης των εργαζομένων. Ακόμα μία εφαρμογή θα μπορούσε να περιλαμβάνει διαδραστικά αντικείμενα, όπως για παράδειγμα, ένα μενού σε μία καφετέρια ή σε ένα εστιατόριο να εμφανίζεται όταν ακουμπάμε το κινητό μας πάνω στο τραπέζι.

Πρόσφατα έκανε την εμφάνιση της στην αγορά μία άκρως καινοτόμος ιδέα μιας ελληνικής εταιρίας η οποία παρήγαγε το ονομαζόμενο Boxaki. Το Boxaki αποτελεί ένα μικρό κιβώτιο το οποίο έχει εγκατασταθεί σε παραλίες και χρησιμοποιείται για να αποθηκεύουμε με ασφάλεια τα προσωπικά μας αντικείμενα. Μόλις βάλουμε τα αντικείμενα μέσα στο κιβώτιο αυτό κλειδώνει και με την χρήση ενός μοναδικού βραχιολιού, το οποίο περιέχει μέσα ένα NFC Tag, το κιβώτιο ανοίγει. Το project αυτό εφαρμόστηκε για πρώτη φορά το καλοκαίρι του 2015 σε παραλίες.

Είναι σαφές ότι η τεχνολογία NFC θα αποτελέσει στο άμεσο μέλλον την βάση για την ανάπτυξη πολλών ακόμα εφαρμογών. Η παρουσία της πλέον σε όλα τα καινούρια κινητά τηλέφωνα δείχνει την τάση των εταιριών να κάνουν χρήση του NFC όλο και περισσότερο δίνοντας νέες δυνατότητες στους χρήστες.



## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Tom Igoe, Don Coleman, Brian Jepson Beginning NFC Near Field Communication with Arduino, Android, and PhoneGap 2014
- [2] Vedat Coskun, Kerem Ok, Busra Ozdenizci Professional NFC Application Development for Android 2013
- [3] <http://nfc-forum.org/>
- [4] <https://developer.android.com/guide/topics/connectivity/nfc/index.html>
- [5] <http://www.nearfieldcommunication.org/>
- [6] Android Studio Application Development Belen Cruz Zapata (<http://www.it-ebooks.info>)
- [7] [https://en.wikipedia.org/wiki/Near\\_field\\_communication](https://en.wikipedia.org/wiki/Near_field_communication)
- [8] <http://stackoverflow.com/>