



**ΑΤΕΙ ΛΑΡΙΣΑΣ**  
**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ**

**Τμήμα Τεχνολογίας Πληροφορικής  
και Τηλεπικοινωνιών**

**Η ΤΕΧΝΟΛΟΓΙΑ NFC ΚΑΙ Η ΑΝΑΠΤΥΞΗ  
ΕΦΑΡΜΟΓΗΣ ΔΙΑΧΕΙΡΙΣΗΣ ΣΠΟΥΔΑΣΤΩΝ  
Γ-ΒΑΘΜΙΑΣ ΕΚΠΑΙΔΕΥΣΗΣ ΜΕ ΧΡΗΣΗ ΤΟΥ  
ANDROID SDK**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**ΑΡΓΥΡΑΚΗΣ ΑΘΑΝΑΣΙΟΣ**

**Επιβλέπων: Δρ. Κακαρόντζας Γεώργιος, Καθηγητής Εφαρμογών**

**ΛΑΡΙΣΑ 2013**



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

Αθανάσιος Αργυράκης

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Τόπος: .....

Ημερομηνία: .....

### ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. ....
2. ....
3. ....

# Περίληψη

Η παρούσα πτυχιακή εργασία έχει ως θέμα **την τεχνολογία επικοινωνίας κοντινού πεδίου (Near Field Communication - NFC) και την ανάπτυξη μιας ολοκληρωμένης πλατφόρμας διαχείρισης σπουδαστών τριτοβάθμιας εκπαίδευσης.**

Η πλατφόρμα αποτελείται από μια Desktop εφαρμογή για υπολογιστές με λειτουργικό σύστημα Microsoft Windows και από μια εφαρμογή για «έξυπνα» κινητά τηλέφωνα (smartphones) με λειτουργικό σύστημα Android. Η εφαρμογή Android επιτρέπει στη Γραμματεία ενός Ιδρύματος να προγραμματίζει τις «έξυπνες» κάρτες των φοιτητών, όπως επίσης και στον εκπαιδευτικό να παίρνει τις παρουσίες των φοιτητών με ένα απλό άγγιγμα της φοιτητικής κάρτας του σπουδαστή στο NFC κινητό του. Μέσω της Desktop εφαρμογής η Γραμματεία του Εκπαιδευτικού Ιδρύματος θα καταχωρεί στη βάση δεδομένων τα εργαστηριακά τμήματα τα οποία τροφοδοτούν την εφαρμογή Android με σημεία ελέγχου. Τέλος, η πλατφόρμα μέσω του NFC κινητού τηλεφώνου παρέχει επιπρόσθετα τη δυνατότητα ελέγχου πρόσβασης στη φοιτητική λέσχη σίτισης επιτρέποντας την παροχή δωρεάν γεύματος μόνο στους φοιτητές που δικαιούνται δωρεάν σίτιση.

Στην πτυχιακή εργασία περιγράφεται πλήρως η τεχνολογία NFC καθώς και οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της πλατφόρμας. Επιπλέον, σκοπός της είναι η προβολή της χρησιμότητας του NFC στον τομέα της εκπαίδευσης καθώς και η εφαρμογή της σε διάφορους τομείς.



# Ευχαριστίες

Θέλω να ευχαριστήσω την οικογένεια μου για την στήριξη που μου έδειξε σε όλα τα χρόνια των σπουδών μου καθώς και τον καθηγητή μου Γεώργιο Κακαρόντζα για την υπομονή που έδειξε κατά την δημιουργία της παρούσας πτυχιακής εργασίας.

Αργυράκης Αθανάσιος

19/9/2013





# Περιεχόμενα

ΠΕΡΙΛΗΨΗ .....	I
ΕΥΧΑΡΙΣΤΙΕΣ.....	III
ΠΕΡΙΕΧΟΜΕΝΑ.....	V
1 ΕΙΣΑΓΩΓΗ .....	1
2 ΤΕΧΝΟΛΟΓΙΑ NFC .....	3
2.1 ΤΙ ΕΙΝΑΙ ΤΟ RFID .....	3
2.2 ΤΙ ΕΙΝΑΙ ΤΟ NFC.....	5
2.3 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ .....	6
2.4 ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ .....	7
2.4.1 Το Πρωτόκολλο .....	7
2.4.2 Τρόπος Επικοινωνίας.....	7
2.4.3 NDEF.....	9
2.4.4 NFC Tags.....	11
2.4.5 Contactless Smart Cards.....	12
2.4.6 Σύγκριση RFID Tag με Contactless Smart Card.....	15
2.4.7 NFC Reader.....	15
2.5 ΕΦΑΡΜΟΓΕΣ NFC .....	17
2.5.1 Πληρωμές.....	17
2.5.2 Μεταφορές.....	17
2.5.3 Υγεία.....	17
2.5.4 Επιχειρήσεις.....	17
2.5.5 Εκπαίδευση.....	18
2.6 ΑΣΦΑΛΕΙΑ ΣΤΟ NFC.....	18
2.6.1 Network Sniffing .....	18
2.6.2 Αλλοίωση Δεδομένων.....	19
2.7 ΠΛΕΟΝΕΚΤΗΜΑΤΑ NFC .....	20

<b>3</b>	<b>ANDROID .....</b>	<b>23</b>
3.1	ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ .....	23
3.2	ΕΚΔΟΣΕΙΣ ANDROID.....	24
3.3	ΑΡΧΙΤΕΚΤΟΝΙΚΗ .....	27
3.3.1	<i>Πυρήνας.....</i>	<i>27</i>
3.3.2	<i>Βιβλιοθήκες.....</i>	<i>28</i>
3.3.3	<i>Χρόνος Εκτέλεσης .....</i>	<i>28</i>
3.3.4	<i>Πλαίσιο Εφαρμογής.....</i>	<i>28</i>
3.3.5	<i>Εφαρμογές.....</i>	<i>29</i>
3.4	ΕΡΓΑΛΕΙΑ ΓΙΑ ΤΗΝ ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ANDROID.....	30
3.4.1	<i>Γλώσσα Προγραμματισμού JAVA.....</i>	<i>30</i>
3.4.2	<i>Eclipse IDE.....</i>	<i>30</i>
3.4.3	<i>Android SDK.....</i>	<i>31</i>
3.4.4	<i>Δομή μιας Android Εφαρμογής.....</i>	<i>32</i>
<b>4</b>	<b>ΕΦΑΡΜΟΓΗ.....</b>	<b>35</b>
4.1	ΠΕΡΙΓΡΑΦΗ ΠΛΑΤΦΟΡΜΑΣ .....	35
4.2	SQL SERVER DATABASE .....	36
4.3	WEB SERVICES .....	38
4.4	ΕΦΑΡΜΟΓΗ ANDROID .....	45
4.4.1	<i>TEI LARISSA NFC APPLICATION.....</i>	<i>45</i>
4.4.2	<i>NFC Attendance .....</i>	<i>61</i>
4.5	DESKTOP ΕΦΑΡΜΟΓΗ.....	64
4.6	MERODROID .....	70
<b>5</b>	<b>ΣΥΜΠΕΡΑΣΜΑΤΑ .....</b>	<b>75</b>
	<b>ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>	<b>77</b>
	<b>ΠΑΡΑΡΤΗΜΑ Α .....</b>	<b>ΣΦΑΛΜΑ! ΔΕΝ ΕΧΕΙ ΟΡΙΣΤΕΙ ΣΕΛΙΔΟΔΕΙΚΤΗΣ.</b>

# 1 Εισαγωγή

Είναι γεγονός πως διανύουμε μια περίοδο, κατά την οποία η ανάπτυξη της κινητής τηλεφωνίας βρίσκεται στο πιο κορυφαίο σημείο συγκριτικά με κάθε άλλη στιγμή στην ιστορία της. Τα κινητά τηλέφωνα, ολοένα και περισσότερο, μετατρέπονται σε μικρούς υπολογιστές έτοιμους να εξυπηρετήσουν, ανά πάσα ώρα και στιγμή, κάθε λογής ανάγκη. Εύλογα χαρακτηρίζονται ως «έξυπνα κινητά» (smartphones) διότι έχοντας τη δυνατότητα συνδεσιμότητας μέσω WiFi ή 3G και διαθέτοντας ενσωματωμένη κάμερα υψηλής ανάλυσης και δέκτη GPS, παρέχουν στον χρήστη υπηρεσίες όπως είναι η αποστολή/ λήψη μηνυμάτων ηλεκτρονικού ταχυδρομείου, η πλοήγηση στο διαδίκτυο, η λήψη φωτογραφιών, η διασκέδαση μέσω παιχνιδιών και μουσικής, η πλοήγηση μέσω GPS, κλπ.

Οι παραπάνω υπηρεσίες υποστηρίζονται χάρη στο λειτουργικό σύστημα που βασίζονται τα smartphones. Τα πιο γνωστά λειτουργικά συστήματα κινητών τηλεφώνων είναι το Android της Google, το iOS της Apple και το Windows Phone της Microsoft. Επικρατέστερο στις μέρες μας θεωρείται το Android, το οποίο έχει κυριαρχήσει στις πωλήσεις και συναντάται στις περισσότερες συσκευές κινητών τηλεφώνων.

Το 2010, πρώτο το Android υποστήριξε την τεχνολογία Near Field Communication (NFC), η οποία είχε αναπτυχθεί από το NFC Forum ήδη από το 2004. Πρόκειται για μια τεχνολογία που επιτρέπει τη διασύνδεση δύο συσκευών φέρνοντάς τα σε μικρή απόσταση. Η Samsung το 2010 ενσωμάτωσε την τεχνολογία NFC στο μοντέλο Nexus S. Συνεπώς, ξεκίνησε η ανάπτυξη εφαρμογών βασισμένων στο NFC με τη βοήθεια του Android SDK από την έκδοση 2.3 και μετά. Στα κεφάλαια που ακολουθούν θα περιγραφεί πλήρως η τεχνολογία NFC καθώς επίσης και οι πολλαπλές χρήσεις της σε διαφορετικούς τομείς.



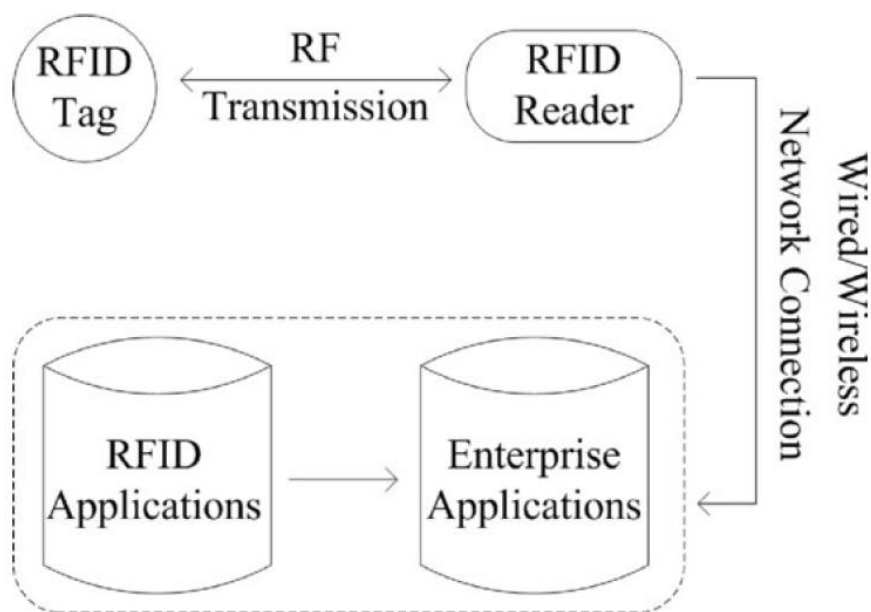
## 2 Τεχνολογία NFC

Η τεχνολογία NFC αποτελεί μια πρότυπη τεχνολογία ασύρματης συνδεσιμότητας που επιτρέπει τη διασύνδεση δύο συσκευών ή μιας συσκευής NFC με μία NFC ετικέτα (NFC tag) **με σκοπό την ανταλλαγή δεδομένων με ένα απλό άγγιγμα**. Στο κεφάλαιο που ακολουθεί θα αναλυθεί πλήρως η λειτουργία της καθώς και η χρησιμότητά της στις ασύρματες επικοινωνίες.

### 2.1 Τι είναι το RFID

Η τεχνολογία Radio-frequency identification (RFID) είναι μια ασύρματη τεχνολογία που επιτρέπει την ανταλλαγή δεδομένων μεταξύ ενός RFID reader (αναγνώστης) και μιας ηλεκτρονικής RFID ετικέτας (RFID Tag) μέσω ραδιοκυμάτων. Τα ηλεκτρομαγνητικά κύματα βοηθούν στη μεταφορά των δεδομένων έτσι ώστε ο reader να μπορεί να διαβάσει ή να γράψει δεδομένα από και προς το RFID tag.

Η αρχιτεκτονική ενός RFID συστήματος είναι η εξής (Εικόνα 1):

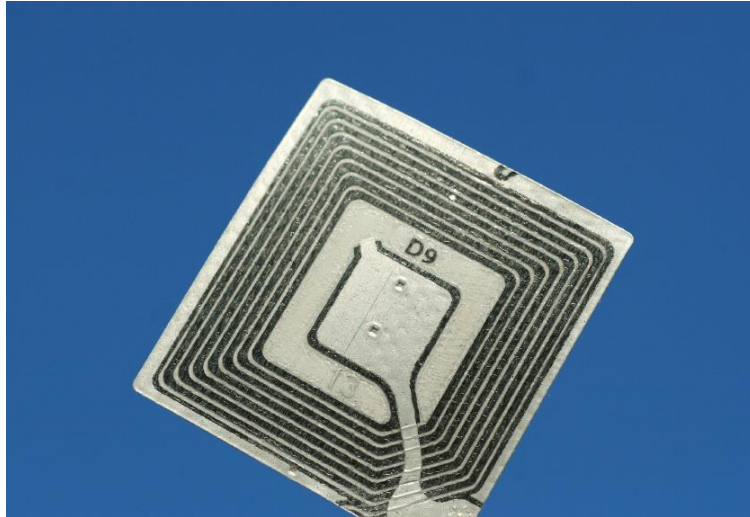


Εικόνα 1: Αρχιτεκτονική RFID συστήματος [1]

Το σύστημα τίθεται σε λειτουργία όταν ένα RFID Tag βρεθεί εντός της εμβέλειας του reader. Στο σημείο αυτό ενεργοποιείται το εσωτερικό του κύκλωμα και ξεκινάει η μεταφορά των δεδομένων. Ένα RFID tag αποτελείται από ένα ολοκληρωμένο κύκλωμα (IC-Integrated Circuit) και μια κεραία (antenna) . Το κύκλωμα είναι υπεύθυνο για την αποθήκευση και την επεξεργασία των δεδομένων όπως επίσης και για τη διαμόρφωση / αποδιαμόρφωση του ραδιοκύματος (RF Signal). Η κεραία με τη σειρά της είναι υπεύθυνη για την αποστολή και τη λήψη του σήματος.

Τα RFID Tags (Εικόνα 2) χωρίζονται σε δύο (2) κατηγορίες, στα ενεργά και στα παθητικά.

- **Ενεργό RFID tag:** Τα ενεργά tags έχουν ενσωματωμένο ένα ολοκληρωμένο κύκλωμα και μια κεραία, και διαθέτουν μια πηγή ενέργειας. Η πηγή ενεργοποιεί το κύκλωμα ώστε να παραχθεί το σήμα που θα αποσταλεί. Τα ενεργά RFID tags έχουν μεγαλύτερη εμβέλεια και μεγαλύτερη μνήμη σε σχέση με τα παθητικά και το μέγεθός τους είναι σαφώς μεγαλύτερο από αυτό των παθητικών λόγω της ύπαρξης πηγής ενέργειας [3].
- **Παθητικό RFID tag:** Τα παθητικά RFID tags έχουν, παρομοίως με τα ενεργά, ενσωματωμένο ένα ολοκληρωμένο κύκλωμα και μία κεραία . Η έλλειψη πηγής ενέργειας είναι ο λόγος που λειτουργεί παθητικά. Το παθητικό tag ενεργοποιείται όταν δεχτεί το ραδιοκύμα ώστε να στείλει και αυτό με τη σειρά του την απάντηση μέσω της κεραίας. Λόγω της έλλειψης πηγής ενέργειας είναι μικρά σε μέγεθος και η τιμή τους στην αγορά είναι πολύ χαμηλή. Αξίζει να αναφερθεί ότι η διάρκεια ζωής τους είναι μεγάλη [3].



Εικόνα 2: RFID tag [2]

## 2.2 Τι είναι το NFC

Το Near Field Communication (NFC) είναι μια καινούργια τεχνολογία βασισμένη στο RFID, η οποία υποστηρίζει την ασύρματη διασύνδεση μεταξύ συσκευών σε μικρή απόσταση. Η συγκεκριμένη τεχνολογία αναπτύχθηκε με σκοπό να λύσει υπάρχοντα και μελλοντικά προβλήματα στο χώρο των επικοινωνιών και στη μεταφορά δεδομένων. Το NFC εφαρμόζεται κυρίως στις συσκευές κινητής τηλεφωνίας (smartphones) και έχει τρεις βασικές λειτουργίες:

- Λειτουργία αναγνώστη (reader mode): η NFC συσκευή είναι ενεργή ώστε να διαβάσει ή να γράψει ένα NFC Tag
- Peer-to-Peer λειτουργία: δυο NFC συσκευές βρίσκονται σε ενεργή σύνδεση ώστε να ανταλλάξουν δεδομένα
- Εξομοιωτής κάρτας (card emulation): η NFC συσκευή παίζει το ρόλο ασύρματης κάρτας.

## 2.3 Ιστορική Αναδρομή

Το NFC (Near Field Communication) αναπτύχθηκε το 2004 από το NFC Forum, το οποίο είναι ένας μη κερδοσκοπικός οργανισμός, και συγκεκριμένα από τις εταιρίες Nokia και Sony. Το NFC Forum καθόρισε τις τεχνικές προδιαγραφές και την αρχιτεκτονική που πρέπει να εφαρμόζονται σε κάθε συσκευή που είναι συμβατή με το NFC. Σκοπός του NFC Forum είναι η δημιουργία εφαρμογών και λύσεων βασισμένων στην τεχνολογία NFC. [1]

Το 2004, στο ξεκίνημα του, το NFC Forum είχε 140 εγγεγραμμένα μέλη ενώ υπολογίζεται πως μέχρι σήμερα τα εγγεγραμμένα μέλη υπερβαίνουν τα 170. Το 2006 ανακοίνωσε επίσημα την αρχιτεκτονική της τεχνολογίας NFC για τις συσκευές και τις «έξυπνες» ετικέτες (NFC Tag), οι οποίες χρησιμοποιούν το NFC προσελκύοντας έτσι οργανισμούς όπως ο GSMA (GSM Association), EMVCo και ο Smart Card Alliance. Το Forum υπογράφοντας συμφωνίες με τους παραπάνω οργανισμούς κατάφερε να κατοχυρώσει τις απαιτούμενες τεχνικές προδιαγραφές για τη σωστή και ασφαλή χρήση του NFC. Την ίδια χρονιά, η Nokia κυκλοφόρησε το πρώτο κινητό τηλέφωνο που υποστήριζε NFC, το μοντέλο Nokia 6131.

Το 2009 το NFC Forum ανακοίνωσε τα πρότυπα για την Peer-to-Peer επικοινωνία, την μεταφορά επαφών, url και την εκκίνηση Bluetooth επικοινωνίας.

Μετά από ένα χρόνο, το 2010 η Samsung κυκλοφόρησε το πρώτο κινητό τηλέφωνο NFC με λειτουργικό Android, το μοντέλο Nexus S.



Εικόνα 3: Λογότυπο NFC [3]



## **2.4 Τεχνικά Χαρακτηριστικά**

### **2.4.1 Το Πρωτόκολλο**

Το πρωτόκολλο επικοινωνίας του NFC βασίζεται στην ασύρματη διασύνδεση συνδυάζοντας δύο παλαιότερες μορφές επικοινωνίας, το Bluetooth και το RFID. Στην επικοινωνία συμμετέχουν πάντοτε δύο συσκευές. Το πρωτόκολλο δημιουργεί ένα ασύρματο δίκτυο μεταξύ NFC συσκευών όπως κινητά τηλέφωνα, NFC tags και περιφερειακές συσκευές του ηλεκτρονικού υπολογιστή. Το πρωτόκολλο λειτουργεί στη συχνότητα 13,56 MHz και προσφέρει ταχύτητα για μεταφορά δεδομένων 106 kbit/sec, 212 kbit/sec και φτάνει μέχρι και 424 kbit/sec σε απόσταση έως και 5 εκατοστά.

Το NFC πρωτόκολλο υποστηρίζει δύο μορφές επικοινωνίας: την ενεργή και την παθητική. Κατά την ενεργή μορφή επικοινωνίας και οι δύο συσκευές δημιουργούν το δικό τους ηλεκτρομαγνητικό πεδίο ώστε να μεταφέρουν τα δεδομένα. Αντιθέτως στην παθητική μορφή, η μία από τις δύο συσκευές δημιουργεί το ηλεκτρομαγνητικό πεδίο και όπως ορίζει το πρωτόκολλο, ο πομπός (Initiator) είναι ο υπεύθυνος για τη δημιουργία του πεδίου. Το NFC, λόγω της παθητικής λειτουργίας που προσφέρει, συμβάλλει σε έναν από τους βασικούς στόχους των συσκευών που λειτουργούν με μπαταρία, στην εξοικονόμηση ενέργειας. Το πρωτόκολλο θέτει σε κατάσταση εξοικονόμησης ενέργειας τη μία συσκευή, περιμένοντας η δεύτερη συσκευή, με ένα απλό άγγιγμα, να ενεργοποιήσει το μαγνητικό πεδίο και να εξασφαλίσει τη διασύνδεση τους.

### **2.4.2 Τρόπος Επικοινωνίας**

Η επικοινωνία μεταξύ δύο συσκευών NFC είναι ημι-αμφίδρομη (half-duplex), δηλαδή κάθε συσκευή πρέπει να λάβει πρώτα το αίτημα και έπειτα να στείλει την ανάλογη απάντηση. Για την ορθή επικοινωνία απαιτείται να υπάρχει ο πομπός (Initiator) και ο δέκτης (Target). Κάθε μία από τις συσκευές που συμμετέχουν στην διαδικασία αυτή είναι εφικτό να δρουν και ως πομπός και ως δέκτης. Ο πομπός είναι εκείνος που διαχειρίζεται τη μεταφορά των δεδομένων στέλνοντας το κατάλλη-

λο αίτημα στο δέκτη, ώστε ο τελευταίος να γνωρίζει τις ενέργειες που πρέπει να εκτελέσει.

Η ευκολία που προσφέρει στην επικοινωνία, σε σχέση με τα άλλα πρωτόκολλα, είναι μεγάλη. Λόγω της μικρής απόστασης που χρειάζονται οι συσκευές για να πραγματοποιηθεί η σύνδεση, το ίδιο το πρωτόκολλο είναι υπεύθυνο για την ασφάλεια των δεδομένων καθώς και για τη μεταφορά τους.

#### **2.4.2.1 Λειτουργία Ανάγνωσης/Εγγραφής**

Το NFC επιτρέπει την εγγραφή και ανάγνωση δεδομένων από ένα NFC κινητό τηλέφωνο σε ένα NFC tag. Σε αυτή την περίπτωση έχουμε ως πομπό (initiator) το κινητό τηλέφωνο και ως δέκτη (target) το NFC tag. Η συσκευή όταν έρθει στην απόσταση που ορίζει το NFC πρωτόκολλο, δηλαδή μικρότερη των 5 εκατοστών, μπορεί μέσω μίας εφαρμογής να διαβάσει ή να γράψει δεδομένα από και προς το tag αντίστοιχα. Η χρησιμότητα είναι σημαντική διότι αυτοματοποιούνται διαδικασίες, όπως για παράδειγμα η διαδικασία ανοίγματος του περιηγητή και η είσοδος σε μια ηλεκτρονική διεύθυνση, οι οποίες πραγματοποιούνται με ένα απλό άγγιγμα (touch) του tag. Οι περιπτώσεις χρήσης είναι πάρα πολλές και αυξάνονται συνεχώς καλύπτοντας ανάγκες επικοινωνίας και αυτοματοποίησης καθημερινών ενεργειών-συνηθειών μέσω έξυπνων κινητών τηλεφώνων (smartphones).

#### 2.4.2.2 Λειτουργία Peer-to-peer

Η Peer-to-Peer (σημείο-προς-σημείο) επικοινωνία επιτρέπει σε δύο συσκευές να συνδεθούν μεταξύ τους και να ανταλλάξουν δεδομένα αυτόματα όταν βρεθούν σε απόσταση μικρότερη των 5 εκατοστών. Η επικοινωνία αυτή βασίζεται στο πρότυπο ISO/IEC 18092 το οποίο λέγεται NFCIP-1 (Near Field Communication Interface Protocol). Η Google ήδη από την έκδοση 4 του Android ενσωμάτωσε την εφαρμογή Android Beam (Εικόνα 4) η οποία κάνει χρήση της peer-to-peer σύνδεσης για Android NFC κινητά για την ανταλλαγή φωτογραφιών, κειμένων, επαφών και πολλών άλλων αρχείων.



Εικόνα 4: Λειτουργία Android Beam για peer-to-peer επικοινωνία [4]

#### 2.4.2.3 Εξομοίωση «Έξυπνης» Κάρτας

Ένα κινητό τηλέφωνο NFC μπορεί επίσης να παίξει το ρόλο της έξυπνης κάρτας που λειτουργεί χωρίς επαφή (contactless smart card). Οι συσκευές που λειτουργούν ως κάρτες μπορούν να εξομοιώσουν όλους τους τύπους καρτών, οι οποίες είναι συμβατές με το πρότυπο του NFC (ISO/IEC 14443), εκτελώντας όλες τις λειτουργίες και συναλλαγές που είναι εφικτό να πραγματοποιηθούν με τις έξυπνες κάρτες.

#### 2.4.3 NDEF

Το NFC Forum δημιούργησε ένα πρότυπο ή καλύτερα ένα τύπο μηνύματος για την ανταλλαγή δεδομένων μεταξύ NFC συσκευών που κάνουν χρήση της τεχνολογίας NFC.

Το πρότυπο αυτό ονομάστηκε NDEF (NFC Data Exchange Format). Το NDEF επιτρέπει την μεταφορά οποιουδήποτε τύπου δεδομένων μεταξύ συσκευών ή μεταξύ συσκευής και κάποιου παθητικού tag στο οποίο είναι αποθηκευμένο το NDEF μήνυμα (NDEF message). Έτσι, λοιπόν, κάθε εφαρμογή αρκεί να ενθυλακώσει τα δεδομένα που έχει προς μεταφορά σε ένα NDEF message.

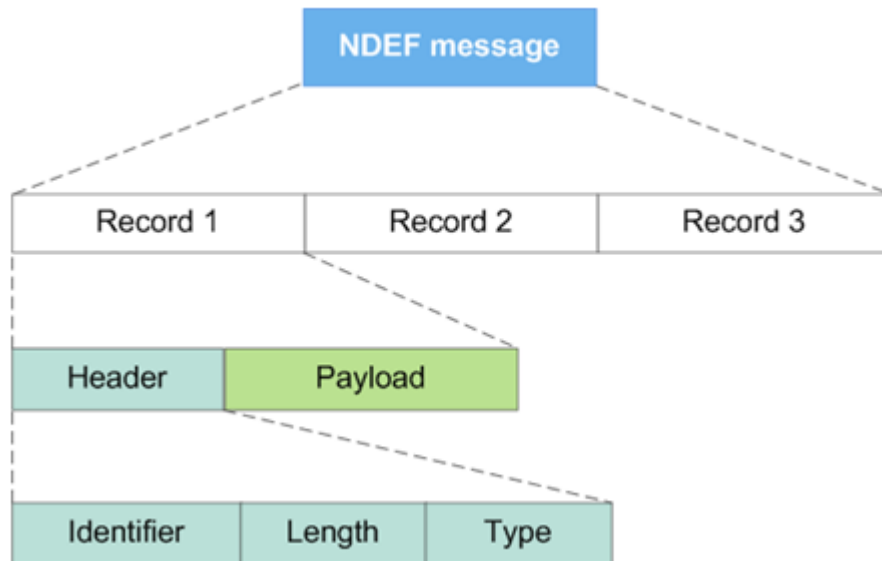
Το NDEF message, η δομή του οποίου παρουσιάζεται στην Εικόνα 5, είναι ένα μήνυμα σε δυαδική μορφή (binary), το οποίο περιέχει από μία έως άπειρες εγγραφές (NDEF records). Το NDEF record ενδέχεται να είναι μήνυμα κειμένου (text message) προς μεταφορά ή οποιοσδήποτε άλλος τύπος δεδομένων όπως επαφές, ηλεκτρονικές διευθύνσεις, εικόνες, κλπ.

NDEF Message						
R <sub>1</sub> MB=1	...	R <sub>r</sub>	...	R <sub>s</sub>	...	R <sub>t</sub> ME=1

Εικόνα 5: NDEF Message

Πάντοτε η πρώτη εγγραφή (NDEF record) ονομάζεται MB (Message Begin) και η τελευταία ME (Message End). Κάθε NDEF μήνυμα μπορεί να έχει τουλάχιστον μία εγγραφή, όπου η εγγραφή MB και ME συμπίπτουν, ενώ δεν υπάρχει περιορισμός στο μέγιστο πλήθος των εγγραφών. Κάθε NDEF record έχει τρεις παραμέτρους που καθορίζουν το payload:

- Payload type: το οποίο καθορίζει τον τύπο δεδομένων ο οποίος περιέχεται στην εγγραφή.
- Payload length: το οποίο καθορίζει το μέγεθος του payload μιας εγγραφής σε οκτάδες (bit)
- Payload identifier: είναι ένα αναγνωριστικό πεδίο ώστε να είναι εφικτή η συσχέτιση μεταξύ εγγραφών



Εικόνα 6 [5]

Τύποι δεδομένων που υποστηρίζει το NDEF

- URI: URL: "http://www.google.gr"  
Αριθμός Τηλεφώνου: "+306971234567"  
E-mail: info@teilar.gr
- Text: "Hello World", "Hello Android"
- Smart Poster: Text + URI
- Business Card

#### 2.4.4 NFC Tags

Ένα NFC Tag είναι στην πραγματικότητα μια παθητική RFID ετικέτα. Σε ένα NFC Tag μέχρι στιγμής μπορεί να αποθηκευτεί μικρός όγκος δεδομένων. Λόγω του ότι είναι σε παθητική μορφή λειτουργίας, για να ενεργοποιηθεί το Tag αρκεί μόνο να έρθει σε επαφή με ένα κινητό τηλέφωνο NFC ή με έναν αναγνώστη (NFC Reader). Η συσκευή αυτή θα δημιουργήσει το ηλεκτρομαγνητικό κύμα στο οποίο θα μεταφερθεί το αίτημα και το Tag με τη σειρά του θα επιστρέψει τα δεδομένα. Σε αυτό το σημείο πρέπει να σημειωθεί ότι το πρωτόκολλο ορίζει ότι μόνο ένα Tag μπορεί να συμμετέχει στη διαδικασία και όχι περισσότερα. Στα Tags μπορούν να αποθηκευτούν όλοι οι τύποι δεδομένων με τη διασφάλιση πάντα του περιορισμένου χώρου αποθήκευσης. Το NFC Forum όρισε τέσσερις τύπους Tag: Type 1, Type 2, Type 3, Type 4. Κάθε τύπος κάρτας, όπως

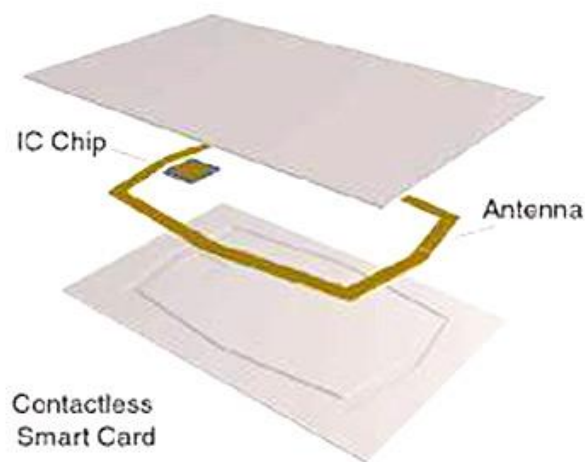
θα αναφερθεί αναλυτικότερα παρακάτω, έχει τη δική του διαμόρφωση και χωρητικότητα.

- Type 1: Το NFC Tag Type 1 είναι βασισμένο στο πρότυπο ISO/IEC 14443 Type A και επιτρέπει λειτουργίες ανάγνωσης και εγγραφής δεδομένων. Τα δεδομένα που έχουν αποθηκευτεί στο Tag μπορούν να τροποποιηθούν εκτός και αν έχει τεθεί σε λειτουργία μόνο ανάγνωσης (read only). Το μέγεθος της μνήμης του συγκεκριμένου τύπου Tag περιορίζεται σε χωρητικότητες από 96bytes μέχρι και 2KB. Η ταχύτητα μεταφοράς δεδομένων που προσφέρει είναι 106 kbps . [4]
- Type 2: Το NFC Tag Type 2 είναι βασισμένο στο πρότυπο ISO/IEC 14443 Type A και επιτρέπει λειτουργίες ανάγνωσης και εγγραφής δεδομένων. Τα δεδομένα που είναι αποθηκευμένα στο Tag μπορούν να τροποποιηθούν εκτός αν το Tag είναι σε λειτουργία μόνο ανάγνωσης (read-only) . Το μέγεθος της μνήμης του συγκεκριμένου τύπου Tag περιορίζεται σε χωρητικότητες από 48bytes μέχρι και 2KB. Η ταχύτητα μεταφοράς δεδομένων είναι ίδια με του Type 1 στα 106 kbps . [4]
- Type 3: Το NFC Tag Type 3 βασίζεται στην Sony Felica (contactless smart card) και έχει μεταβλητή χωρητικότητα με μέγιστη το 1MB. Προσφέρει μεγαλύτερη ταχύτητα μεταφοράς δεδομένων στα 212 και 424 kbps. Ο συγκεκριμένος τύπος Tag προορίζεται για πιο σύνθετες εφαρμογές και είναι αρκετά πιο ακριβός από τους υπόλοιπους τύπους. [4]
- Type 4: Το NFC Tag Type 4 είναι συμβατό με το πρότυπο ISO/IEC 14443 Type A και Type B. Η διαφορά του από τα υπόλοιπα είναι ότι είναι διαμορφωμένο εργοστασιακά. Κατά την κατασκευή του προγραμματίζεται το αν προορίζεται για εγγραφή (write mode) ή για ανάγνωση μόνο (read-only mode). Έχει μεγαλύτερη χωρητικότητα 32KB και μεγαλύτερη ταχύτητα τα 424 kbps. [4]

#### **2.4.5 Έξυπνες Κάρτες Χωρίς Επαφή (Contactless Smart Cards)**

Οι “έξυπνες” κάρτες χωρίς επαφή (contactless smart cards) (Εικόνα 7), είναι κάρτες οι οποίες λειτουργούν χωρίς να έρθουν σε επαφή (απόσταση ανάλογη του τύπου καρτών) με κάποια εξωτερική συσκευή. Αποτελούνται από ένα microchip και μια κεραία μέσω της οποίας πραγματοποιείται η ανταλλαγή δεδομένων μεταξύ κάρτας και αναγνώστη (reader). Στις contactless smart cards οι πληροφορίες αποθηκεύονται στο microchip το οποίο διαθέτει έναν μικροεπεξεργαστή με τους απαραίτητους μηχανισμούς

ασφαλείας και τη δικιά του εσωτερική μνήμη. Η ενέργεια που χρειάζεται η κάρτα για να ενεργοποιηθεί παράγεται από την κεραία μέσω της δημιουργίας ηλεκτρομαγνητικού πεδίου. Η κεραία είναι υπεύθυνη για την αποστολή και λήψη των δεδομένων μεταξύ της εξωτερικής συσκευής και της κάρτας. Οι readers κάνοντας χρήση της τεχνολογίας RFID δημιουργούν το ηλεκτρομαγνητικό πεδίο έτσι ώστε όταν πλησιάσει μία κάρτα η ενέργεια του πεδίου θα μεταφερθεί μέσω της κεραία στο microchip. Μέσω της διαδικασίας αυτής έχει δημιουργηθεί ασύρματη διασύνδεση και είναι εφικτή η μεταφορά και αποθήκευση δεδομένων από την εξωτερική συσκευή στην κάρτα.



Εικόνα 7: Δομή Contactless Smartcard [6]

Οι contactless smart cards βρίσκουν εφαρμογή σε διαδικασίες που απαιτούν την ύπαρξη απορρήτου όπως οι πληροφορίες για την υγεία κάποιου ή στοιχεία ταυτοποίησης. Επίσης μπορούν να χρησιμοποιηθούν για γρήγορες συναλλαγές και ως ηλεκτρονικό διαβατήριο παρέχοντας υψηλή ασφάλεια στα δεδομένα που ανταλλάσσονται και αποθηκεύονται στην κάρτα κάνοντας χρήση ισχυρών κρυπτογραφικών κλειδιών.

Οι contactless smart cards είναι διαθέσιμες σε πλαστικές κάρτες, ρολόγια, μπρελόκ και φυσικά σε mobile συσκευές διότι το NFC δίνει τη δυνατότητα να λειτουργούν ως εξομοιωτές έξυπνης κάρτας. Μέχρι σήμερα υπάρχουν 3 πρότυπα καρτών .

- **ISO/IEC 10563 – Closing Coupling Smart Cards**

Το συγκεκριμένο πρότυπο έχει τίτλο “Identification Cards-Contactless Integrated Circuit Cards” και ορίζει τη δομή και τη λειτουργία των καρτών μικρής εμβέλειας. Οι συγκεκριμένες κάρτες λειτουργούν σε απόσταση μικρότερη του 1 εκατοστού[3]. [3]

- **ISO/IEC 14443-Proximity Coupling Smart Cards**

Το συγκεκριμένο πρότυπο έχει τίτλο “Identification Cards-Proximity Integrated Circuit Cards” ορίζοντας την δομή και τη λειτουργία των καρτών εγγυημένης ζεύξης. Σύμφωνα με το πρότυπο οι έξυπνες κάρτες λειτουργούν σε απόσταση μικρότερη των 5 εκατοστών στη συχνότητα των 13.56 MHZ και είναι ο τύπος καρτών που χρησιμοποιείται στο NFC. [3]

- **ISO/IEC 15693-Vicinity Coupling Smart Cards**

Το συγκεκριμένο πρότυπο έχει τίτλο “Identification Cards-Contactless Integrated Circuit Cards-Vicinity Cards” το οποίο ορίζει τη δομή και τη λειτουργία των καρτών ευρείας περιοχής. Οι κάρτες έχουν εμβέλεια 1 μέτρο και λειτουργούν στη συχνότητα των 13.56 MHZ. Το συγκεκριμένο πρότυπο έχει 4 μέρη: τα φυσικά χαρακτηριστικά, την προετοιμασία για την ασύρματη σύνδεση, το πρωτόκολλο για την αποφυγή συγκρούσεων και το πρωτόκολλο μεταφοράς. [3]

### **ISO/IEC 14443 Smart Cards (NFC Smart Cards)**

1)MIFARE: Η MIFARE κατασκευάστηκε από την NXP και υπόκειται στο πρότυπο ISO/IEC 14443 Type A. Από το 2011 η MIFARE κατέχει το 80% του συνολικού αριθμού contactless smart cards που χρησιμοποιούνται σε παγκόσμιο επίπεδο. Λειτουργεί στη συχνότητα των 13,56 MHz και έχει διάφορα μεγέθη μνήμης. Η οικογένεια καρτών MIFARE περιέχει διαφόρους τύπους καρτών όπως Ultralight, Standard, Desfire, Classic, Plus και SmartMX. Οι σημαντικότερες, έως τώρα, εφαρμογές της είναι σε συστήματα εισιτηρίων Μέσων Μαζικής Μεταφοράς, στις ηλεκτρονικές πληρωμές, σε έλεγχο πρόσβασης σε χώρους που απαιτούν έλεγχο, σταθμούς διοδίων καθώς και σε εφαρμογές loyalty.

2)Felica: Κατασκευάστηκε από τη Sony και λειτουργεί στη συχνότητα των 13.56 Mhz όπως προβλέπει και το πρότυπο ISO/IEC 18092 του NFC. Η Sony έκανε αίτηση για να



εγκριθεί η Felica υπό το ISO/IEC 14443, αλλά απορρίφθηκε. Η Felica προσφέρει ταχύτητα μεταφοράς δεδομένων στα 212 kbps.

#### **2.4.6 Σύγκριση RFID Tag με Contactless Smart Card**

Στις μέρες μας πολλές εφαρμογές κάνουν χρήση της RF τεχνολογίας για την αυτόματη ταυτοποίηση ανθρώπων και αντικειμένων. Οι εφαρμογές αυτές ποικίλουν από τον εντοπισμό ζώων μέχρι την εκτέλεση ασφαλών και γρήγορων συναλλαγών. Όλες αυτές οι εφαρμογές χρησιμοποιούν ραδιοκύματα για την ασύρματη επικοινωνία και τη μεταφορά δεδομένων [3].

Τα RFID Tags είναι σημαντικό κομμάτι των RFID συστημάτων. Είναι απλά και φθηνά. Χρησιμοποιούνται σε προγράμματα όπως για παράδειγμα αναγνώρισης ζώων και ήρθαν για να αντικαταστήσουν τα εκτυπωμένα barcodes. Όταν ένα tag εισέλθει στην εμβέλεια του RFID reader τα δεδομένα μεταφέρονται χωρίς την προστασία κάποιου μέσου ασφαλείας ή ελέγχου αυθεντικότητας. Το αποτέλεσμα είναι ότι οποιοσδήποτε reader μπορεί να λάβει τα δεδομένα από το tag.

Οι contactless smart cards επίσης λειτουργούν βάση της RF τεχνολογίας. Χρησιμοποιούνται όμως σε εφαρμογές που απαιτούν υψηλό επίπεδο ασφαλείας για τη μεταφορά δεδομένων. Όπως προαναφέρθηκε, διαθέτουν στο εσωτερικό τους ασφαλή μικροελεγκτή και εσωτερική μνήμη, εκτελώντας έτσι με ασφάλεια τις ενέργειες που απαιτεί η εφαρμογή. Η απόσταση που απαιτείται για την λειτουργία των καρτών είναι το πολύ 5 εκατοστά.

Συμπερασματικά, οι contactless smart cards έναντι των RFID tags, πρέπει να χρησιμοποιούνται όταν απαιτείται υψηλή διασφάλιση δεδομένων.

#### **2.4.7 NFC Αναγνώστης (NFC Reader)**

Ένας NFC Reader (NFC αναγνώστης) είναι μια ενεργή συσκευή NFC η οποία μπορεί να είναι εσωτερική ή εξωτερική. Ένας εσωτερικός αναγνώστης μπορεί να ενσωματωθεί σε ένα κινητό τηλέφωνο NFC ώστε να εκτελεί διαδικασίες ανάγνωσης/εγγραφής από και προς το Tag. Ένα κινητό τηλέφωνο με εσωτερικό αναγνώστη είναι πάντα ενεργό ώστε να παράγει το δικό του μαγνητικό πεδίο το οποίο είναι απαραίτητο για τη διασύνδεση μεταξύ ενός Tag ή ενός δευτέρου τηλεφώνου. Τέλος υπάρχει και ο εξωτερικός

NFC reader (Εικόνα 8) που χρησιμοποιείται για ανάγνωση/εγγραφή καρτών όπως επίσης για την πραγματοποίηση χρηματικών συναλλαγών.



Εικόνα 8: NFC Reader [7]

## **2.5 Εφαρμογές NFC**

### **2.5.1 Πληρωμές**

Το NFC έχει όλες τις προδιαγραφές ώστε να αποτελέσει σημαντικό κομμάτι στις πληρωμές. Κάθε κάτοχος κινητού τηλεφώνου NFC μπορεί με ένα άγγιγμα να δώσει το προβλεπόμενο αντίτιμο για τις υπηρεσίες που του προσφέρονται. Το NFC έχει τη δυνατότητα να εγγυηθεί εξοικονόμηση χρόνου διότι ο χρόνος που χρειάζεται για τη συναλλαγή είναι ελάχιστος - της τάξης του δευτερολέπτου – και παράλληλα, ασφάλεια διότι γίνονται γρήγορα από την κοντινή απόσταση των 5 εκατοστών.

### **2.5.2 Μεταφορές**

Η εφαρμογή του NFC στα Μέσα Μαζικής Μεταφοράς είναι μία καινοτομία που προσφέρει πολλά προνόμια στον επιβάτη. Το NFC μπορεί να υποστηρίξει την αγορά εισιτηρίων ή καρτών μακράς διάρκειας με ένα απλό άγγιγμα στα σημεία πώλησης τους. Ο έλεγχος επιβίβασης του επιβάτη πραγματοποιείται μέσω του κινητού τηλεφώνου σε τρένα, λεωφορεία και αεροπλάνα. Επίσης, μέσα στα Μέσα Μαζικής Μεταφοράς ο επιβάτης θα μπορεί με ένα άγγιγμα στα «έξυπνα» πόστερ που είναι αναρτημένα, να κατεβάσει αυτόματα πληροφορίες όπως δρομολόγια και χάρτες καθώς θα μπορεί να ενημερώνεται αυτόματα για προσφορές εισιτηρίων και καινούργιες εκπτώσεις.

### **2.5.3 Υγεία**

Το NFC σε νοσοκομειακές μονάδες χρησιμοποιείται για το ιστορικό των ασθενών προσφέροντας την απαραίτητη ασφάλεια ως προς το απόρρητο των δεδομένων. Κάθε ασθενής θα διαθέτει το δικό του NFC Tag στο οποίο αποθηκεύεται το ιατρικό ιστορικό του καθώς και οι επισκέψεις του σε νοσοκομειακές μονάδες. Οι γιατροί, με την τεχνολογία NFC και με ένα άγγιγμα με το NFC κινητό τους θα μπορούν να δουν αμέσως το ιστορικό του ασθενούς και να είναι σε θέση να καταγράψουν όλες τις ενέργειες που θα πραγματοποιηθούν για την νοσηλεία του. Με το NFC λοιπόν εύκολα θα υπάρχει μια πλήρης εικόνα για τον γιατρό σε σχέση με την πορεία της υγείας των ασθενών.

### **2.5.4 Επιχειρήσεις**

Οι επιχειρήσεις μέσω της τεχνολογίας NFC μπορούν να δημιουργήσουν ένα σύστημα ελέγχου και εποπτείας των εργαζομένων. Κάθε υπάλληλος θα δηλώνει την ώρα έ-

λευσης και αποχώρησης του από την επιχείρηση με ένα απλό σκανάρισμα της έξυπνης κάρτας του στον reader εισόδου/εξόδου. Επομένως, θα υπάρχει πλήρης εποπτεία των ωρών εργασίας κάθε υπαλλήλου. Παράλληλα, ο εργοδότης έχει τη δυνατότητα, μέσω της έξυπνης κάρτας, να ορίσει τα μέρη στα οποία τα μέλη του προσωπικού θα έχουν ελεύθερη πρόσβαση.

### **2.5.5 Εκπαίδευση**

Στην εκπαίδευση, χάρις στην τεχνολογία του NFC, είναι εφικτός ο αυτοματισμός πολλών διαδικασιών που πραγματοποιούνται σε ένα σχολικό ή ακαδημαϊκό συγκρότημα. Κάθε μαθητής/φοιτητής διαθέτει την προσωπική του NFC ταυτότητα την οποία χρησιμοποιεί ώστε να δηλώσει την παρουσία του στους χώρους του συγκροτήματος (τάξεις, αμφιθέατρα, βιβλιοθήκη, γυμναστήριο, κ.ά.), απλά και εύκολα, φέρνοντάς την σε επαφή με έναν NFC αναγνώστη. Κατά αυτόν τον τρόπο υπάρχει έλεγχος και ασφάλεια στους χώρους που κινούνται οι μαθητές/φοιτητές, δίνοντας τη δυνατότητα καταγραφής και ενημέρωσης των γονέων στην περίπτωση μαθητών μικρής ηλικίας. Είναι επίσης εφικτές, χάρις στο NFC, μικρό-συναλλαγές εντός του συγκροτήματος ώστε να μην είναι απαραίτητη η χρήση μετρητών από τους μαθητές/φοιτητές.

## **2.6 Ασφάλεια στο NFC**

### **2.6.1 Network Sniffing**

Όπως σε κάθε ασύρματη μετάδοση, έτσι και στο NFC, υπάρχει ο κίνδυνος της υποκλοπής πακέτων κατά τη μετάδοση των δεδομένων. Όπως προαναφέρθηκε η μετάδοση στο NFC γίνεται με την εκπομπή ηλεκτρομαγνητικών κυμάτων ανάμεσα στις δύο συσκευές. Θεωρητικά λοιπόν η επίθεση μπορεί να γίνει με χρήση μιας κεραίας ώστε ο εισβολέας να λάβει το ηλεκτρομαγνητικό κύμα και με τον κατάλληλο εξοπλισμό για αποκρυπτογράφηση δεδομένων να υποκλέψει τα πακέτα. Επειδή στο NFC η απόσταση που απαιτείται για να πραγματοποιηθεί η σύνδεση είναι 5 εκατοστά, σε αυτήν την απόσταση μόνο μπορεί να γίνει η υποκλοπή του ραδιοκύματος. Οπότε είναι προφανές πως με την τεχνολογία NFC είναι δύσκολο έως αδύνατο να πραγματοποιηθεί υποκλοπή πακέτων χωρίς να γίνει αντιληπτός ο υποκλοπέας.

### **2.6.2 Αλλοίωση Δεδομένων**

Ένα άλλο είδος επίθεσης που ενδέχεται να υποστεί μια επικοινωνία NFC είναι η αλλοίωση των δεδομένων. Σε αυτή την περίπτωση σκοπός του εισβολέα δεν είναι μόνο να δει την κίνηση των πακέτων αλλά να επέμβει στην μετάδοση. Ο εισβολέας μπορεί να πετύχει το σκοπό του αν μεταδίδει τα δικά του δεδομένα στη σωστή συχνότητα και στο σωστό χρόνο, ώστε ο παραλήπτης να μην το αντιληφθεί και να λάβει αλλοιωμένα τα δεδομένα. Όπως και στην περίπτωση του Network Sniffing είναι σχεδόν αδύνατο να γίνει υποκλοπή λόγω της μικρής απόστασης που ορίζει το NFC, δηλαδή τα πέντε εκατοστά.

## 2.7 Πλεονεκτήματα NFC

### Ευκολία στη χρήση

Το NFC έχει το πλεονέκτημα ότι είναι απλό στη χρήση του και προορίζεται για όλες τις ηλικίες. Δεν χρειάζεται τίποτα περισσότερο από τα να ακουμπήσει κάποιος, από μικρά παιδιά μέχρι ηλικιωμένους ανθρώπους, το κινητό του τηλέφωνο σε ένα tag ή σε έναν αναγνώστη. Χάρη στην αυτοματοποιημένη διαδικασία στην οποία έχει προγραμματιστεί η κάθε εφαρμογή, ανάλογα με την κατάσταση, δρα μόνη της. Για παράδειγμα στα σενάρια πληρωμών δεν χρειάζεται να έχει μαζί του κάποιος όλες του τις κάρτες παρά μόνο το κινητό του τηλέφωνο. Όλοι θα μπορούν με ένα άγγιγμα σε κάποιο NFC πόστερ να λάβουν αυτόματα χρήσιμες πληροφορίες για αυτό που απλά κοιτάζουν χωρίς περεταίρω διαδικασία.

### Ασφάλεια

Το NFC προσφέρει το απαιτούμενο επίπεδο ασφαλείας στη μεταφορά των δεδομένων διότι κάνει χρήση ασφαλούς καναλιού μεταφοράς. Όλες οι συναλλαγές γίνονται σε απόσταση 5 εκατοστών έτσι είναι σχεδόν απίθανο κάποιος να παρέμβει για να υποκλέψει δεδομένα. Το NFC διασφαλίζει το απόρρητο των πληροφοριών, όπως καταστάσεις υγείας και κωδικούς καρτών. Στην περίπτωση για παράδειγμα του ιατρικού ιστορικού τα δεδομένα χάρις στη λειτουργία έξυπνης κάρτας του κινητού τηλεφώνου δεν μπορούν να χαθούν και να τα διαβάσει κάποιος. Το ίδιο και με τις πιστωτικές κάρτες. Εάν χαθεί μια πιστωτική κάρτα μπορεί εύκολα κάποιος να τη χρησιμοποιήσει και να κάνει αγορές, ενώ εάν είναι όλες οι κάρτες στο NFC κινητό τηλέφωνο ακόμα και να χαθεί το κινητό θα πρέπει να είναι γνωστός ο κωδικός πρόσβασης της συσκευής.

### Ευρεία Χρήση

Με το NFC μπορούν να αυτοματοποιηθούν πολλές διαδικασίες της καθημερινότητας του ανθρώπου. Έχοντας ο καθένας ένα κινητό τηλέφωνο NFC θα μπορεί να κάνει εύκολα πληρωμές, να έχει όλα του τα εισιτήρια για λεωφορεία, μετρό, τρένα, αεροπλάνα, κινηματογράφο, γήπεδα, μουσεία μέσα στο κινητό του και απλώς με ένα άγγιγμα να παίρνει το ελεύθερο για τις υπηρεσίες που θέλει να χρησιμοποιήσει. Όλες του οι ενέργειες θα πραγματοποιούνται πιο γρήγορα με το NFC χάρη στην ευρεία χρήση σε πολλούς τομείς.







## 3 Android

Το Android είναι το λειτουργικό σύστημα που έφτιαξε η Google βασισμένο στον πυρήνα του Linux για «έξυπνα» κινητά τηλέφωνα (smartphones). Από την αρχή της κυκλοφορίας του μέχρι και σήμερα έχει αναπτυχθεί ραγδαία τόσο η εξέλιξή του τεχνολογικά όσο και ο αριθμός των χρηστών που το επιλέγουν παγκοσμίως. Στη συνέχεια θα γίνει αναφορά στο Android και στην ανάπτυξη εφαρμογής για την πλατφόρμα.



Εικόνα 9: Λογότυπο Android [8]

### 3.1 Ιστορική Αναδρομή

Το Android είναι λειτουργικό σύστημα για «έξυπνα» κινητά τηλέφωνα (smartphones). Η ανάπτυξή του ξεκίνησε το 2003 από την Android Inc. Το 2005 η Google εξαγόρασε την Android Inc και προσέλαβε τους ιδρυτές της. Το 2007 έγινε η πρώτη παρουσίαση του λειτουργικού συστήματος Android και το ίδιο έτος ιδρύθηκε ο οργανισμός Open Handset Alliance ο οποίος ήταν μια κοινοπραξία 48 εταιριών κινητής τηλεφωνίας και εταιριών λογισμικού και υλικού. Ένα χρόνο αργότερα από την παρουσίαση του Android ήρθε στην κυκλοφορία και η πρώτη συσκευή Android από την HTC με όνομα HTC Dream. Στα επόμενα χρόνια το Android έχει βελτιωθεί και αναπτυχθεί

στο χώρο των κινητών τηλεφώνων και αποτελεί βασική προτίμηση των προγραμματιστών στην επιλογή τους.

## 3.2 Εκδόσεις Android

Οι εκδόσεις του Android ξεκίνησαν με την επίσημη ονομασία “Cupcake” η οποία βασίστηκε στον πυρήνα του Linux 2.6.27 τον Μάιο του 2009. Ήταν

η έκδοση 1.5 η οποία είχε τα εξής χαρακτηριστικά :

- Εγγραφή και αναπαραγωγή βίντεο μέσα από την εφαρμογή της φωτογραφικής μηχανής
- Δυνατότητα μεταφόρτωσης βίντεο στο Youtube και φωτογραφιών στο Picasa
- Υποστήριξη περισσότερων πληκτρολογίων με δυνατότητα πρόβλεψης λέξεων
- Συνδεσιμότητα μέσω Bluetooth
- Λειτουργίες αντιγραφής-επικόλλησης στον περιηγητή διαδικτύου
- Εγκατάσταση widgets στην αρχική οθόνη και υποστήριξη ύπαρξης πολλαπλών οθονών

Λίγο αργότερα τον Σεπτέμβριο του 2009 ήρθε η έκδοση 1.6 βασισμένη στον πυρήνα 2.6.29 του Linux με την ονομασία “Donut” το οποίο έφερε με τη σειρά του τις εξής αναβαθμίσεις :

- Δημιουργία εφαρμογής κάμερας, βιντεοκάμερας και προβολής φωτογραφιών (gallery) η οποία επιτρέπει την πολλαπλή επιλογή φωτογραφιών προς διαγραφή.
- Υποστήριξη για οθόνες WVGA.
- Ανανεωμένη αναζήτηση με τη βοήθεια προτεινόμενων λέξεων και αναβάθμιση της φωνητικής αναζήτησης.
- Βελτιωμένο Android Market.
- Αναβάθμιση τεχνολογιών Bluetooth, CDMA/EVDO, 802.1x, VPN .

Τον Οκτώβριο του 2009 ήρθε η βελτιωμένη έκδοση 2.0 που ονομάστηκε “Eclair” φέρνοντας τις εξής αλλαγές:

- Δυνατότητα συγχρονισμού παραπάνω από έναν λογαριασμό Gmail και δυνατότητα συγχρονισμού επαφών.
- Υποστήριξη Bluetooth 2.1

- Δυνατότητα αποστολής sms, email ή κλήσης μιας επαφής μέσω παρατεταμένης επιλογής σε μια επαφή
- Νέα χαρακτηριστικά στη κάμερα όπως: υποστήριξη φωτογραφικού φλας, ψηφιακό ζουμ, λειτουργία σκηνής, ισορροπία λευκού, εφέ χρώματος και η δυνατότητα macro εστίασης
- Υποστήριξη περισσότερων μεγεθών οθόνης και αναλύσεων
- Ανανεωμένα Google Maps
- Προσθήκη Live Wallpaper (κινούμενη ταπετσαρία οθόνης)
- Ανανεωμένη διεπαφή χρήστη (UI) στο πρόγραμμα περιήγησης διαδικτύου με μικρο-γραφίες σελιδοδεικτών, μεγέθυνση διπλού αγγίγματος και υποστήριξη HTML5.

Τον Μάιο του 2010 ήρθε η νέα έκδοση του Android (Android 2.2) με όνομα “Froyo” η οποία βασίζεται στον πυρήνα του Linux . Οι νέες επιλογές που προσφέρει είναι οι εξής:

- Βελτιστοποίηση στην ταχύτητα του λειτουργικού συστήματος, στη μνήμη και στην απόδοσή του.
- Βελτιστοποίηση της ταχύτητας των εφαρμογών μέσω του JIT Compilation (Just In Time Compilation).
- Υποστήριξη του C2DM (Android Cloud to Device Messaging)
- Υποστήριξη USB tethering και Wi-Fi Hotspot
- Προσθήκη επιλογής για την απενεργοποίηση της πρόσβασης δεδομένων μέσω κινητής τηλεφωνίας
- Ανανεωμένο Android Market με υποστήριξη αυτόματης ενημέρωσης των εφαρμογών του κινητού
- Υποστήριξη Adobe Flash

Έπειτα ήρθαν οι αναβαθμίσεις του Froyo (2.2.1-2.2.3) οι οποίες διόρθωσαν σφάλματα στην αποστολή SMS ,βελτίωσαν την επίδοση του λειτουργικού συστήματος και έφεραν δυο ανανεώσεις για την ασφάλεια του λειτουργικού.

Το Δεκέμβριο του 2010 ήρθε η έκδοση 2.3 με όνομα Gingerbread η οποία είχε τις εξής επιδράσεις στο λειτουργικό σύστημα :

- Ανανεωμένη διεπαφή χρήστη (User Interface) η οποία βελτίωσε την ταχύτητα
- Υποστήριξη NFC (Near-Field-Communication)
- Υποστήριξη πολύ μεγάλων μεγεθών οθονών και ανάλυσης (WXGA και μεγαλύτερες)

- Προεγκατεστημένη υποστήριξη για πολλαπλές κάμερες στη συσκευή
- Αλλαγή από YAFFS (Yet Another Flash File System) σε σύστημα αρχείου ext4
  - Ανανεωμένος Garbage Collector (συλλέκτης σκουπιδιών μνήμης) ώστε να διαγράφονται από τη μνήμη τα αντικείμενα που δεν χρησιμοποιούνται

Το Φεβρουάριο του 2011 ήρθε η έκδοση 3.0 του Android με ονομασία “Honeycomb” η οποία ήταν η πρώτη αναβάθμιση του λειτουργικού συστήματος που απευθυνόταν σε ταμπλέτες (Tablet Pc). Βασισμένη στον πυρήνα 2.6.36 του Linux. Μερικές από τις αναβαθμίσεις είναι οι εξής :

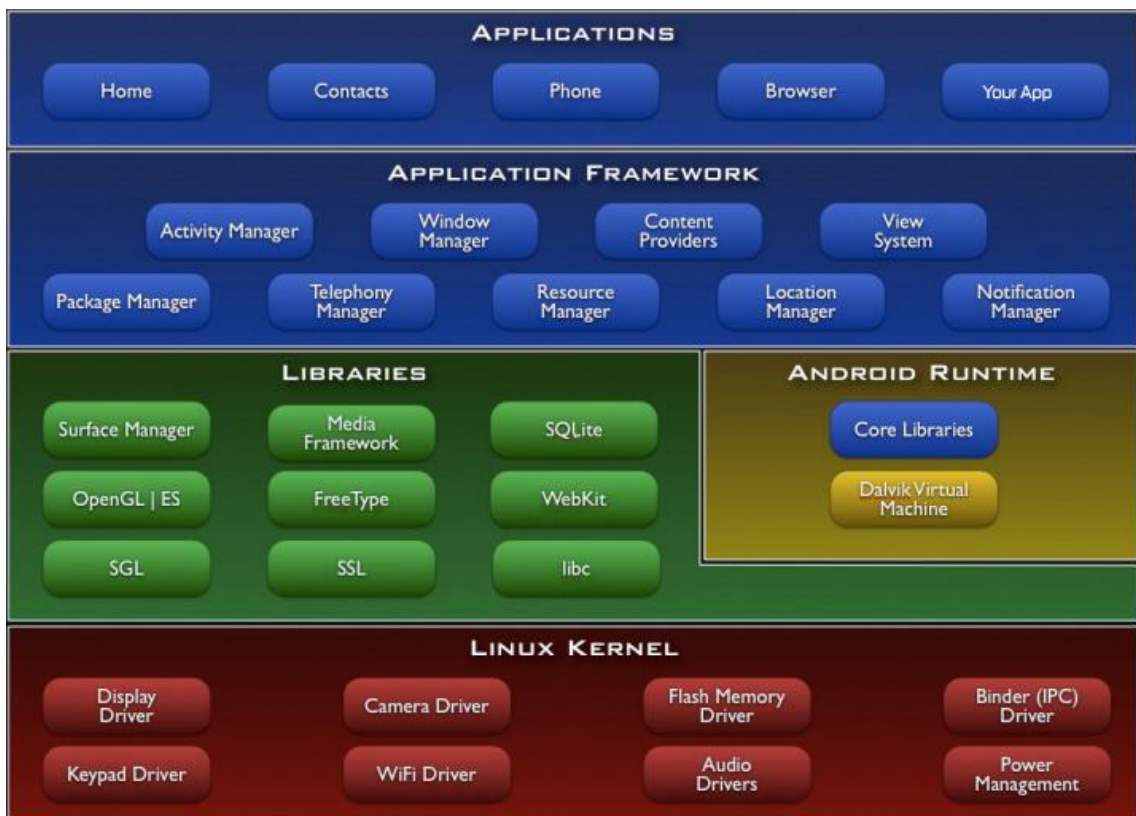
- Υποστήριξη πολυπύρηνων επεξεργαστών
- Ανανεωμένη διεπαφή χρήστη (UI) για την υποστήριξη Tablet
- Αναβάθμιση στις υπάρχουσες εφαρμογές (Περιηγητής διαδικτύου, camera, E-mail) ώστε να υποστηρίζουν τις οθόνες των tablet.

Η επόμενη έκδοση του Android ήταν η 4.0 και κυκλοφόρησε τον Οκτώβριο του 2011 με την ονομασία “Ice Cream Sandwich” βασισμένη στον 3.0.1 πυρήνα του Linux. Μερικά από τα σημαντικά νέα χαρακτηριστικά που έφερε η έκδοση 4.0 είναι τα παρακάτω :

- Νέο βελτιωμένο UI φέρνοντας τα νέα εικονικά κουμπιά στην μπάρα συστήματος
- Δυνατότητα λήψης screenshot
- Προσθήκη εικονικής αναπαράστασης των ανοιχτών εφαρμογών (multitasking) με την παρατεταμένη επιλογή του home button.
- Δυνατότητα δημιουργίας φακέλων στην αρχική οθόνη
- Διόρθωση των σφαλμάτων του πληκτρολογίου
- Μεταφορά δεδομένων μεταξύ συσκευών με τη χρήση της τεχνολογίας NFC (Android Beam)
- Λειτουργία που επιτρέπει δύο συσκευές να συνδεθούν μεταξύ τους μέσω ασυρμάτου δικτύου (Wi-Fi Direct)

### 3.3 Αρχιτεκτονική

Το Android μπορεί να θεωρηθεί ως μια στοίβα λογισμικού η οποία αποτελείται από στρώματα τα οποία συνησφαίρουν στην ομαλή και σωστή λειτουργία του συστήματος. Ξεκινώντας από το χαμηλότερο επίπεδο έχουμε με τη σειρά τον πυρήνα, τις εγγενής βιβλιοθήκες και τον χρόνο εκτέλεσης, το πλαίσιο εφαρμογής και τέλος στο υψηλότερο έχουμε το στρώμα εφαρμογής (Εικόνα 10).



Εικόνα 10: Αρχιτεκτονική Android [9]

#### 3.3.1 Πυρήνας

Στο χαμηλότερο επίπεδο βρίσκεται ο πυρήνας του λειτουργικού συστήματος του Linux ξεκινώντας από τον 2.6. Ευθύνη του πυρήνα είναι η επικοινωνία με το υλικό (hardware) παρέχοντας γι' αυτό το σκοπό τους οδηγούς (drivers). Πιο συγκεκριμένα είναι υπεύθυνος για την διαχείριση μνήμης, τη διαχείριση εργασιών, τη δικτύωση, την επικοινωνία με την κάμερα και την οθόνη της συσκευής. Ο πυρήνας του Linux είναι σταθερός και δοκιμασμένος γεγονός το οποίο φέρνει την απαιτούμενη αξιοπιστία και σταθερότητα που απαιτείται για μία συσκευή και ένα λειτουργικό σύστημα.

### 3.3.2 Βιβλιοθήκες

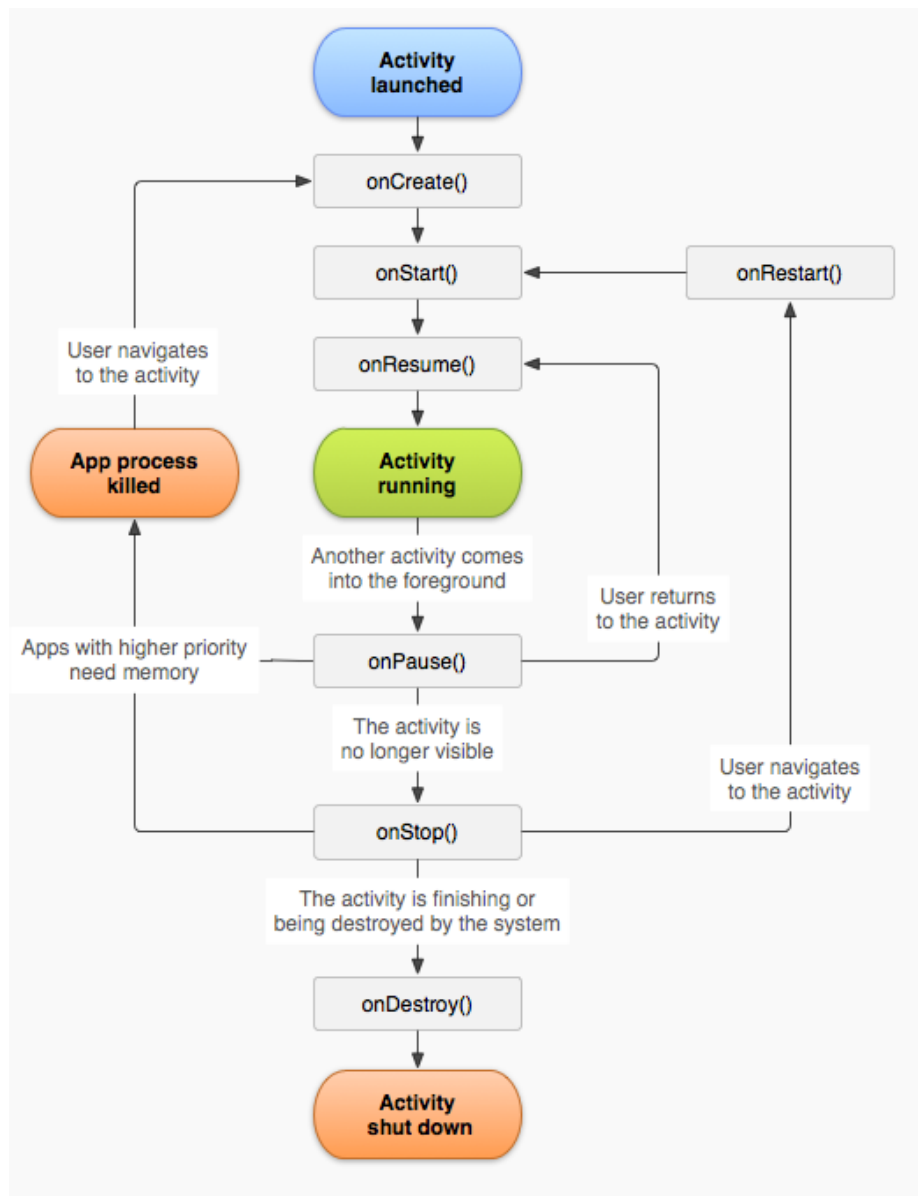
Το επόμενο στρώμα μετά τον πυρήνα είναι οι εγγενείς βιβλιοθήκες (native libraries) του Android. Οι βιβλιοθήκες είναι γραμμένες σε γλώσσα C/C++ και είναι μεταγλωττισμένες για να «τρέχουν» στη συγκεκριμένη αρχιτεκτονική του Android. Οι εγγενείς βιβλιοθήκες γράφτηκαν με σκοπό την κλήση τους από προγράμματα υψηλότερου επιπέδου. Από την έκδοση Donut και έπειτα δόθηκε η δυνατότητα σε όλους τους προγραμματιστές να γράφουν τις δικές τους βιβλιοθήκες με τη βοήθεια της βιβλιοθήκης NDK (Native Development Kit). Μερικές από τις σημαντικότερες βιβλιοθήκες είναι η SQLite η οποία δίνει τη δυνατότητα αποθήκευσης δεδομένων στην συσκευή, η SSL που είναι το κρυπτογραφικό πρωτόκολλο για την ασφαλή επικοινωνία με το διαδίκτυο και η Webkit που παρέχει λειτουργίες για την περιήγηση στο διαδίκτυο.

### 3.3.3 Android Runtime

Στο ίδιο επίπεδο με τις εγγενείς βιβλιοθήκες βρίσκεται και το Android Runtime. Σε αυτή την ομάδα έχουμε όλες τις βιβλιοθήκες της γλώσσας προγραμματισμού Java που επιτρέπουν στους προγραμματιστές να αναπτύξουν εφαρμογές με την Java. Επίσης υπάρχει και η εικονική μηχανή Dalvik η οποία είναι υπεύθυνη για τη δημιουργία εκτελέσεων αρχείων Java με σκοπό να εκτελεστούν στο Android διότι το λειτουργικό δεν μπορεί να χειρισθεί απευθείας την γλώσσα Java. Σε αυτό το σημείο πρέπει να αναφερθεί πως κάθε εφαρμογή που εκτελείται στο Android ακόμα και σε εφαρμογές που εκτελούνται παράλληλα, εκτελούνται στην δικιά τους Dalvik Virtual Machine.

### 3.3.4 Πλαίσιο Εφαρμογής

Πλαίσιο Εφαρμογής (Application Framework) είναι το στρώμα υψηλού επιπέδου που προσφέρει στους προγραμματιστές την δυνατότητα να χρησιμοποιούν τις έτοιμες δομικές μονάδες του Android ώστε να φτιάξουν τις δικές τους εφαρμογές. Μερικές από αυτές είναι ο View Manager ο οποίος προσφέρει τη χρήση εικονιδίων, πλαισίων κειμένου, λίστες κουμπιών κλπ, ο Διαχειριστής Τοποθεσίας (Location Manager) ο οποίος δίνει την ακριβή τοποθεσία του κινητού τηλεφώνου. Αξίζει να αναφερθεί και ο Διαχειριστής Δραστηριοτήτων (Activity Manager) ο οποίος διαχειρίζεται τον κύκλο ζωής κάθε εφαρμογής (Εικόνα 11).



Εικόνα 11: Κύκλος Ζωής Activity [10]

### 3.3.5 Εφαρμογές

Στο τελευταίο και υψηλότερο επίπεδο βρίσκονται οι κοινές εφαρμογές που βλέπει και χρησιμοποιεί κάθε χρήστης του Android όπως για παράδειγμα οι επαφές, η εφαρμογή για κλήση αριθμού τηλεφώνου, τα μηνύματα και η μουσική. Είναι γραμμένες με τη γλώσσα προγραμματισμού Java. Το επίπεδο εφαρμογής μπορεί να θεωρηθεί το πιο σημαντικό διότι αυτό είναι το επίπεδο που βλέπει ο τελικός χρήστης.

## **3.4 Εργαλεία για την ανάπτυξη εφαρμογών Android**

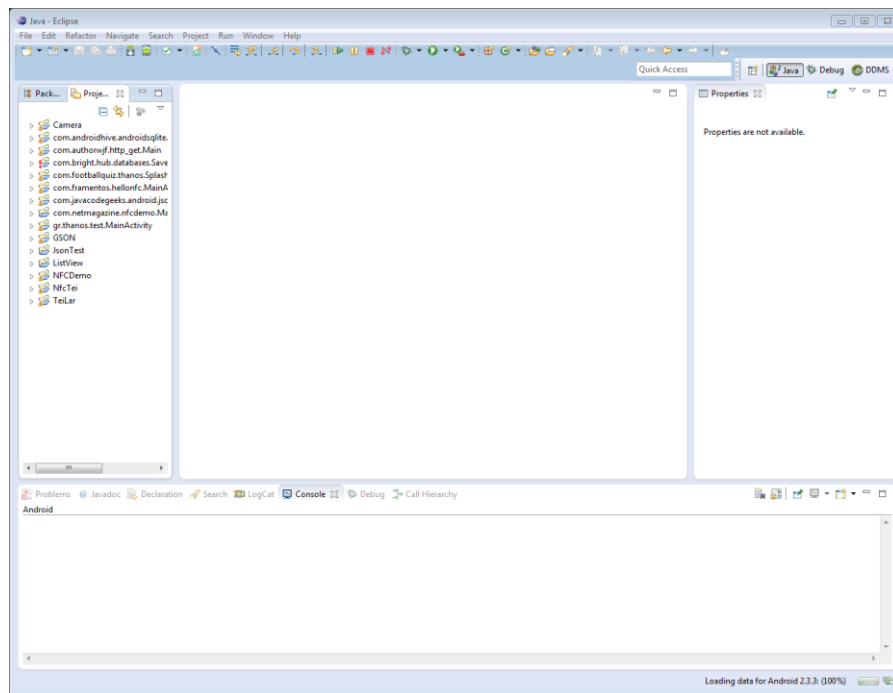
### **3.4.1 Γλώσσα Προγραμματισμού JAVA**

Η γλώσσα προγραμματισμού Java είναι κατά κύριο λόγο η γλώσσα που χρησιμοποιείται για την ανάπτυξη εφαρμογών για το Android. Η Java αναπτύχθηκε από την εταιρία Sun Microsystems από τον James Gosling και κυκλοφόρησε το 1995. Είναι μια αντικειμενοστραφής γλώσσα και οι εφαρμογές της τρέχουν σε οποιοδήποτε λειτουργικό σύστημα (Windows, Linux, Unix, Mac OSX) χωρίς να χρειάζονται μεταγλώττιση χάρη στην εικονική μηχανή της Java (Java Virtual Machine-JVM). Στην περίπτωση του Android, όπως προαναφέρθηκε, οι εφαρμογές εκτελούνται στην εικονική μηχανή Dalvik μετατρέποντας τα αρχεία σε εκτελέσιμα αρχεία .dex. Η Java διαθέτει και συλλέκτη απορριμμάτων (Garbage Collector) ο οποίος αναλαμβάνει την αποδέσμευση μνήμης από δεδομένα τα οποία δεν χρησιμοποιούνται.

### **3.4.2 Eclipse IDE**

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





Εικόνα 12 : Eclipse IDE

### 3.4.3 Android SDK

Το Android SDK είναι το βασικότερο εργαλείο για τους προγραμματιστές οι οποίοι αναπτύσσουν εφαρμογές για το Android. Περιλαμβάνει βιβλιοθήκες (API Libraries), μηχανισμό αποσφαλμάτωσης (debugger) καθώς και τον εικονικό εξομοιωτή συσκευής Android ο οποίος λειτουργεί όπως κάθε συσκευή Android. Το Android SDK υποστηρίζει όλα τα λειτουργικά συστήματα και πολλά περιβάλλοντα ανάπτυξης κώδικα (IDE) αλλά ο βασικότερος είναι το Eclipse IDE με την επέκταση (plugin) ADT Android.

Το ADT παρέχει τον AVD (Android Virtual Device) ο οποίος είναι ο προσομοιωτής της συσκευής τόσο στο λειτουργικό σύστημα όσο και στο υλικό μιας πραγματικής συσκευής Android. Επίσης ένα ακόμη εργαλείο απαραίτητο για τους προγραμματιστές είναι το Dalvik Debug Monitor Service (DDMS) μέσω του οποίου είναι δυνατή η διαχείριση των εργασιών στην εικονική μηχανή Dalvik και πιο συγκεκριμένα η λήψη στιγμιότυπου μιας εφαρμογής, εμφάνιση πληροφοριών για τα νήματα, τα μηνύματα ημερολογίου (log) και πολλές άλλες χρήσιμες πληροφορίες.

### 3.4.4 Δομή μιας Android Εφαρμογής

Κάθε Android εφαρμογή αποτελεί και ένα project στο Eclipse IDE και αποτελείται απαραίτητα από τα παρακάτω στοιχεία :

#### 3.4.4.1 Φάκελος src

Ο φάκελος src (source) περιέχει τα πακέτα στα οποία υπάρχει ο πηγαίος κώδικας της εφαρμογής ο οποίος περιέχει αρχεία με όνομα ActivityName.java, όπου ActivityName είναι το όνομα του Activity έτσι ακριβώς έχει δηλωθεί στο AndroidManifest αρχείο, και άλλα αρχεία πηγαίου κώδικα με κατάληξη .java .

#### 3.4.4.2 Φάκελος bin

Ο Φάκελος bin περιέχει τα μεταγλωττισμένα αρχεία καθώς και το Android Application Package αρχείο (.apk) της εφαρμογής.

#### 3.4.4.3 Φάκελος gen

Ο φάκελος gen περιέχει τα αρχεία που αποτελούν τους πόρους της εφαρμογής τα οποία παράγονται αυτόματα.

#### 3.4.4.4 Φάκελος libs

Περιέχει τις βιβλιοθήκες που χρησιμοποιούνται στην εφαρμογή.

#### 3.4.4.5 Φάκελος res

Ο φάκελος res περιέχει τους πόρους της εφαρμογής στους εξής υπο-φακέλους:

- Φάκελος drawable: περιέχει αρχεία bitmap (PNG, JPEG ή GIF) και εικόνες τα οποία περιγράφουν εικόνες ή σχέδια στην εφαρμογή. Ο φάκελος res περιέχεται τέσσερις φορές με τα ονόματα drawable-hdpi, drawable-mdpi, drawable-ldpi και drawable-xhdpi παρέχοντας τα γραφικά σε διάφορες αναλύσεις.
- Φάκελος layout : απαρτίζεται από τρεις φακέλους με ονόματα layout-small, layout-long και layout-large περιέχοντας όλα τα xml αρχεία τα οποία αντιπροσωπεύουν το UI (User Interface) της εφαρμογής.
- Φάκελος menu: περιέχει τα xml αρχεία για κάθε menu της εφαρμογής.

#### 3.4.4.6 AndroidManifest.xml

Κάθε εφαρμογή Android είναι απαραίτητο να περιέχει το αρχείο AndroidManifest.xml μέσα στον root φάκελό της. Το αρχείο αυτό περιέχει πληροφορίες που χρειάζεται να ξέρει το Android πριν εκτελεστεί ο κώδικας της εκάστοτε εφαρμογής. Κάποιες από τις σημαντικότερες πληροφορίες που παρέχει είναι οι εξής:

- Ορίζει το όνομα του πακέτου Java της εφαρμογής το οποίο είναι μοναδικό αναγνωριστικό.
- Περιγράφει τα μέρη της εφαρμογής (components) όπως activities, broadcast receivers κλπ και ορίζει τα ονόματα των κλάσεων.
- Ορίζει τα δικαιώματα που πρέπει να έχει η εφαρμογή για την σωστή λειτουργία της.
- Ορίζει την ελάχιστη έκδοση του Android που απαιτείται για τρέξει η εφαρμογή. [9]



# 4 Εφαρμογή

## 4.1 Περιγραφή Πλατφόρμας

Η πλατφόρμα NFC University αναπτύχθηκε με στόχο την αυτοματοποίηση καθημερινών διεργασιών που γίνονται σε ένα ίδρυμα τριτοβάθμιας εκπαίδευσης. Οι ρουτίνες που καλείται να αυτοματοποιήσει είναι η καταγραφή των παρουσιών κάθε φοιτητή στα εργαστηριακά τμήματα με ένα άγγιγμα της κάρτας του φοιτητή στο NFC κινητό τηλέφωνο του καθηγητή. Καταγράφοντας την παρουσία κάθε σπουδαστή στη βάση δεδομένων, είναι εφικτή η ύπαρξη αρχείου παρουσιών καθώς και καταστάσεων με τους σπουδαστές που παρακολούθησαν επαρκώς κάποιο εργαστηριακό μάθημα.

Επίσης, η πλατφόρμα NFC University επιτρέπει τον έλεγχο της κάρτας σίτισης του σπουδαστή στο εστιατόριο του εκπαιδευτικού ιδρύματος, με την απλή διαδικασία του «Σκαναρίσματος» της φοιτητικής του ταυτότητας στο NFC κινητό τηλέφωνο του εστιατορίου.

Για την λειτουργία της πλατφόρμας δημιουργήθηκε μια βάση δεδομένων για την αποθήκευση των απαραίτητων πληροφοριών, μια Desktop εφαρμογή για την δημιουργία των εργαστηριακών ομάδων στην αρχή κάθε εξαμήνου καθώς και για την παραγωγή αναφορών (παρουσιολόγιο, καταστάσεις επιτυχούς παρακολούθησης, ημερομηνίες εργαστηριακών ομάδων, κλπ).

Η πλατφόρμα αποτελείται από δύο Android εφαρμογές που αποτελούν το πιο σημαντικό κομμάτι της.

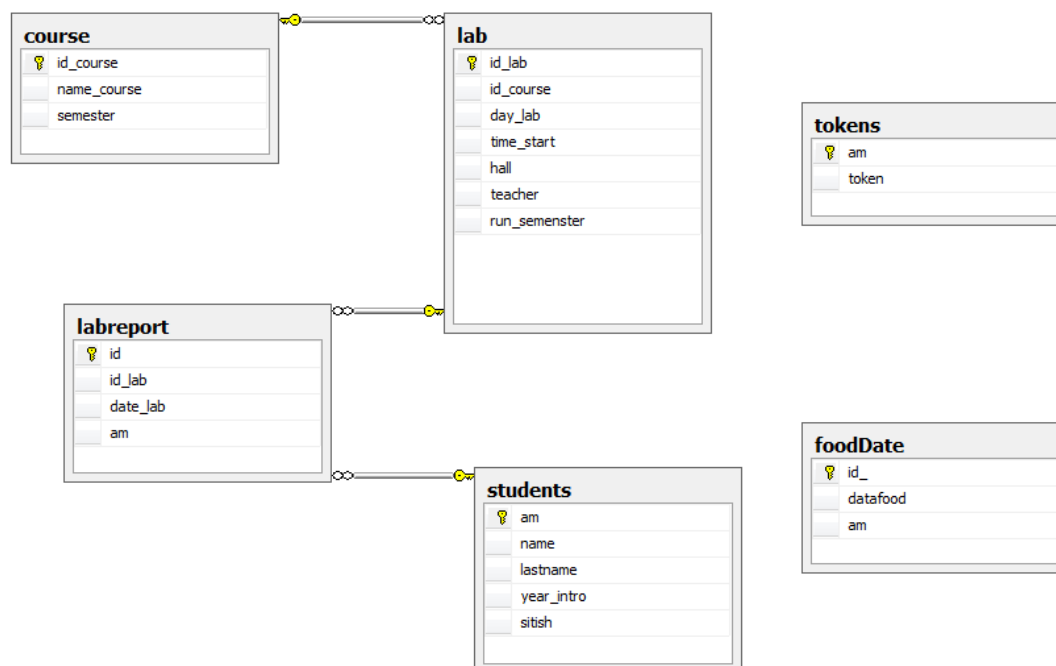
Η πρώτη φορητή εφαρμογή που αναπτύχθηκε εξυπηρετεί την γραμματεία του ιδρύματος προγραμματίζοντας με ένα NFC κινητό τηλέφωνο τα στοιχεία του σπουδαστή (κατά την εγγραφή του φοιτητή στο ίδρυμα) στην NFC κάρτα του. Επίσης υπάρχει η επιλογή στην φοιτητική λέσχη να γίνει έλεγχος της κάρτας σίτισης του σπουδαστή, επιτρέποντας ή όχι την είσοδο του στο εστιατόριο, και να αγοράσει γεύματα έναντι χρημάτων που προγραμματίζονται στην κάρτα (στην περίπτωση που κάποιος δεν διαθέτει κάρτα σίτισης).

Η δεύτερη εφαρμογή Android αφορά στην καταγραφή των παρουσιών των σπουδαστών σε κάθε εργαστήριο κάνοντας χρήση της τεχνολογίας NFC. Η άντληση και απο-

στολή δεδομένων της εφαρμογής Android από και προς την κεντρική βάση δεδομένων πραγματοποιήθηκε με την ανάπτυξη υπηρεσιών διαδικτύου (Web Services).

## 4.2 SQL Server Database

Οι περισσότερες εφαρμογές χρειάζονται μια βάση δεδομένων για την αποθήκευση των απαραίτητων πληροφοριών. Στην εφαρμογή NFC University χρησιμοποιήθηκε η Microsoft SQL Server Database για την αποθήκευση των δεδομένων. Η βάση δεδομένων με όνομα university αποτελείται από τους πίνακες όπως φαίνεται στην Εικόνα 13



Εικόνα 13: University Database

### Πίνακας course

Ο πίνακας course δημιουργήθηκε για την αποθήκευση των μαθημάτων του ιδρύματος και περιέχει τρία πεδία:

- **id\_course** (int, not null) είναι το πρωτεύον κλειδί και περιέχει τον κωδικό του μαθήματος
- **name\_course** (nvarchar(50), not null), περιέχει το όνομα του μαθήματος
- **semester** (nvarchar(2)) είναι το εξάμηνο στο οποίο βρίσκεται το μάθημα.

### Πίνακας lab

Ο πίνακας lab περιέχει τις εργαστηριακά τμήματα του κάθε εξαμήνου όπως αυτά αποφασίζονται από τα μέλη του ιδρύματος. Περιέχει τα εξής πεδία:

- **id\_lab** (int, not null) είναι το πρωτεύον κλειδί
- **id\_course** που είναι το ξένο κλειδί και αναφέρεται στο id\_course του πίνακα course
- **day\_lab** (nvarchar(10), not null) με την ημέρα διεξαγωγής του εργαστηρίου
- **time\_start** (varchar(5), not null) με την ώρα έναρξης του εργαστηρίου
- **hall** (nvarchar(10), not null) η αίθουσα διεξαγωγής του εργαστηρίου
- **teacher** (nvarchar(25), not null) με το όνομα του διδάσκοντα
- **run\_semenster** (nvarchar(7), not null) με το τρέχον εξάμηνο διεξαγωγής

### Πίνακας labreport

Ο πίνακας labreport είναι ο πίνακας στον οποίο καταγράφονται οι παρουσίες των φοιτητών και περιέχει τα εξής πεδία:

- **id** (int, not null) το οποίο είναι το πρωτεύον κλειδί
- **id\_lab** που είναι το ξένο κλειδί και αναφέρεται στο id\_lab του πίνακα lab
- **date\_lab** (nvarchar(10), not null) το οποίο περιέχει την ημερομηνία διεξαγωγής του εργαστηρίου
- **am** το οποίο είναι ξένο κλειδί και αναφέρεται στο am του πίνακα students

### Πίνακας students

Στον πίνακα students βρίσκονται τα στοιχεία του κάθε φοιτητή και είναι τα εξής:

- **am** (int, not null) το οποίο είναι το πρωτεύον κλειδί και περιέχει τον αριθμό μητρώου του κάθε φοιτητή
- **name** (nvarchar(20), not null) με το όνομα του σπουδαστή
- **lastname** (nvarchar(20), not null) με το επώνυμο του
- **year\_intro** (nvarchar(7), not null) με το εξάμηνο εισαγωγής του
- **sitish** (varchar(1), null) για το αν κατέχει κάρτα σίτισης

## Πίνακας tokens

Ο πίνακας tokens δημιουργήθηκε για το εστιατόριο και για να αποθηκευτούν τα γεύματα τα οποία έχει αγοράσει ο σπουδαστής που δεν έχει κάρτα σίτισης, με τα εξής δύο πεδία:

- **am** (varchar (15), not null) με τον αριθμό μητρώου του σπουδαστή
- **tokens** (int, null) με τον αριθμό των γευμάτων που έχει αγοράσει έναντι χρημάτων

## Πίνακας foodDate

Ο πίνακας foodDate δημιουργήθηκε για το εστιατόριο, για την αποθήκευση της ημερομηνίας που έκανε χρήση της κάρτας σίτισης του κάθε σπουδαστής:

- **id** (int, not null) το οποίο είναι το πρωτεύον κλειδί του πίνακα
- **am** (varchar(15), null) με τον αριθμό μητρώου του φοιτητή με κάρτα σίτισης
- **datafood** (nvarchar(10), null) με την ημερομηνία στην οποία έκανε χρήση της κάρτας σίτισής του ο σπουδαστής

## 4.3 Web Services

Το Server Side κομμάτι της εφαρμογής αναπτύχθηκε με τη γλώσσα C# και είναι τα Web Services τα οποία κάνει χρήση η Android εφαρμογή για την επικοινωνία με τη κεντρική βάση δεδομένων university. Είναι Rest Services και η απάντηση που στέλνουν στον Android Client είναι σε μορφή JSON.

### Μέθοδος GetCollection

Η πρώτη ανάγκη για την εφαρμογή NFC Attendance η οποία καταγράφει την παρουσία του σπουδαστή σε κάθε εργαστήριο, με το άγγιγμα της κάρτας του, ήταν να στείλει ο server τα εργαστήρια που διδάσκονται την συγκεκριμένη ημέρα της εβδομάδας. Η μέθοδος που το υλοποιεί είναι η *GetCollection*. Η κλήση της μεθόδου από τον client γίνεται με το URL:

<http://localhost/service1/Service1/GetData/{day}>

όπου το *day* είναι η μεταβλητή που δέχεται ως είσοδό της η μέθοδος *GetCollection*. Η μεταβλητή *day* είναι string και παίρνει τιμές απο 1 έως 5 ανάλογα με την ημέρα της εβδομάδας. Όταν ο server λάβει το αίτημα αμέσως αντιστοιχεί τον αριθμό 1 έως 5 με την αντίστοιχη ημέρα της εβδομάδας ώστε να εκτελέσει το query στη βάση και να πάρει τα σωστά αποτελέσματα.



```

switch (day_to_int)
{
    case 1: daytostring = "ΔΕΥΤΕΡΑ"; //mapping imeron me arithmous
        break;
    case 2: daytostring = "ΤΡΙΤΗ";
        break;
    case 3: daytostring = "ΤΕΤΑΡΤΗ";
        break;
    case 4: daytostring = "ΠΕΜΠΤΗ";
        break;
    case 5: daytostring = "ΠΑΡΑΣΚΕΥΗ";
        break;
}

```

Αυτόματα λοιπόν, ο server στέλνει στον Client μία λίστα με τα εργαστήρια που διδάσκονται στο τρέχων εξάμηνο και την συγκεκριμένη ημέρα στο κινητό τηλέφωνο.

```

try
{
    using (SqlConnection conn = new SqlConnection("Data Source=(local);Initial Catalog=university;Integrated Security=True"))
    {
        conn.Open();
        //string to opoio periexei to query gia na ferei apo th vash ta ergasthria pou ginontai thn sigkekrimenh mera kai to sigkekrimeno eksamino
        string cmdStr = String.Format("select lab_id_lab,course.name_course,lab.time_start,dbo.lab.hall from course join lab on (course.id_course=lab.id_course and lab.day_lab='{0}' and run_semenster='{1}');",
            daytostring, eksamino);
        SqlCommand cmd = new SqlCommand(cmdStr, conn);
        SqlDataReader rd = cmd.ExecuteReader();
        if (rd.HasRows)
        {
            while (rd.Read())
            {
                courses.Add(new Course(rd.GetInt32(0), rd.GetString(1), rd.GetString(2), rd.GetString(3)));
            }
        }
        conn.Open();
    }
}
catch (Exception e)
{
    Console.WriteLine(e);
}
finally
{
}
return courses;
}

```

## Μέθοδος InsertStudent

Η μέθοδος InsertStudent είναι εκείνη που καλείται από τον Android Client με σκοπό να αποθηκεύσει στον πίνακα students της βάσης university τα στοιχεία του σπουδαστή. Είναι μια void μέθοδος και δέχεται τα εξής ορίσματα: string am, string name, string lastname, string year\_intro, string sitisi. Το URL που επιτρέπει την κλήση της από τον Android Client είναι το εξής:

[http://localhost/Service1/Service1/PostName/{am}/{name}/{lastname}/{year\\_intro}/{sitisi}](http://localhost/Service1/Service1/PostName/{am}/{name}/{lastname}/{year_intro}/{sitisi}).

```
[WebGet(BodyStyle = WebMessageBodyStyle.Wrapped, ResponseFormat = WebMessageFormat.Json,
UriTemplate = "PostName/{am}/{name}/{lastname}/{year_intro}/{sitisi}")]
public void InsertStudent (string am, string name, string lastname, string year_intro, string sitisi)
{
    //methodos pou grafei sth vash ton neo foithth
    sc.Open();
    try
    {
        cmd = new SqlCommand("Insert into students values ('" +
            am.ToString() + "','" + name.ToString() + "','" +
            lastname.ToString() + "','" + year_intro.ToString() + "','" + sitisi.ToString() + "')", sc);
        cmd.ExecuteNonQuery();
    }
    catch(Exception ex)
    {
        Console.WriteLine(ex.Message);
    }
    finally
    {
        sc.Close();
    }
}
```

## Μέθοδος createReport

Η μέθοδος createReport είναι μια void μέθοδος η οποία καλείται από τον Android Client με σκοπό την καταγραφή της παρουσίας του φοιτητή στο εργαστήριο (query στη βάση university). Έχει ως ορίσματα δύο string το id\_lab (ο κωδικός του εργαστηριακού τμήματος) και το am (αριθμός μητρώου σπουδαστή). Το URL που επιτρέπει την κλήση της είναι το: [http://localhost/Service1/Service1/DayLab/{id\\_lab}/{am}](http://localhost/Service1/Service1/DayLab/{id_lab}/{am}).

```

[WebGet(BodyStyle = WebMessageBodyStyle.Wrapped, ResponseFormat = WebMessageFormat.Json, UriTemplate = "DayLab/{id_lab}/{am}")]
public void createReport(string id_lab, string am)
{
    //methodos pou kataxorei tin parousia tou foithth : eisodos einai to id_lab kai to am
    int flag = 0;
    SqlDataAdapter da2 = new SqlDataAdapter("select id_lab,date_lab,am from dbo.labreport", sc);
    DataTable dt2 = new DataTable();
    da2.Fill(dt2);
    sc.Open();
    string date = System.DateTime.Today.ToString("dd/MM/yyyy");
    try
    {
        for (int i = 0; i < dt2.Rows.Count; i++)
        {
            if ((string)dt2.Rows[i]["am"] == am)
            {
                if ((Int32)dt2.Rows[i]["id_lab"] == Int32.Parse(id_lab))
                {
                    flag = 1;
                }
            }
        }
        if (flag == 0)
        {
            cmd = new SqlCommand("Insert into labreport (id_lab,date_lab,am) values ('" +
                Int32.Parse(id_lab) + "', '" + date + "', '" + am.ToString() + "')", sc);
            cmd.ExecuteNonQuery();
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
    }
    finally
    {
        sc.Close();
    }
}
}

```

## Μέθοδος foodEligable

Η μέθοδος foodEligable είναι η μέθοδος που δέχεται σαν όρισμα τον αριθμό μη-τρώου του σπουδαστή και επιστρέφει στον Android Client μια Boolean τιμή για το αν ο σπουδαστής με κάρτα σίτισης έχει πάρει το γεύμα του. Η μέθοδος αντιμετωπίζει την πολλαπλή χρήση μιας κάρτας σίτισης. δεδομένου ότι η κάρτα σίτισης προσφέρει ένα μόνο γεύμα ημερησίως. Το URL της είναι:

<http://localhost/service1/Service1/Food/{am}>

```
[WebGet(BodyStyle = WebMessageBodyStyle.Bare, ResponseFormat = WebMessageFormat.Json, UriTemplate = "Food/{am}/")]
public bool foodEligible(string am)
{
    //methodos pou elegxei an o foititis pou exei karta sitishs exei faei se mia mera
    bool food=true;
    int flag = 0;
    SqlDataAdapter da2 = new SqlDataAdapter("select datafood,am from dbo.foodDate", sc);
    DataTable dt2 = new DataTable();
    da2.Fill(dt2);
    sc.Open();
    string date = System.DateTime.Today.ToString("dd/MM/yyyy");
    try
    {
        for (int i = 0; i < dt2.Rows.Count; i++)
        {
            if ((string)dt2.Rows[i]["am"] == am && (string)dt2.Rows[i]["datafood"]==date)
            {
                flag = 1;
                food= false;
            }
        }

        if (flag == 0)
        {
            cmd = new SqlCommand("Insert into foodDate (datafood,am) values ('" + date + "','" + am.ToString() + "')", sc);
            cmd.ExecuteNonQuery();
            food = true;
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
    }
    finally
    {
        sc.Close();
    }
    return food;
}
```

## Μέθοδος insertToken

Η μέθοδος insertToken είναι μια void μέθοδος η οποία επιτρέπει την αγορά γευμάτων από τους φοιτητές που δεν έχουν κάρτα σίτισης. Η μέθοδος αυτή καταγράφει τις συναλλαγές, δηλαδή τις αγορές γευμάτων που πραγματοποιεί ο σπουδαστής, και ενημερώνει αυτόματα το λογαριασμό του με τα διαθέσιμα γεύματα που έχει. Η μέθοδος δέχεται λοιπόν ως ορίσματα τον αριθμό μητρώου του σπουδαστή και τα tokens (γεύματα) τα οποία θέλει να πιστώσει το λογαριασμό του. Η κλήση της γίνεται με το URL:

<http://localhost/service1/Service1/Token/{am}/{tokens}>. Μόλις λοιπόν κληθεί δημιουργεί το query για τη βάση αποθηκεύοντας έτσι τα δεδομένα.

```
[WebGet(BodyStyle = WebMessageBodyStyle.Bare, ResponseFormat = WebMessageFormat.Json, UriTemplate = "Token/{am}/{tokens}")]
public void insertToken(string am, string tokens)
{
    int flag = 0;
    try
    {
        sc.Open();
        int tokensInt = Int32.Parse(tokens);
        int tokensYet;
        SqlDataAdapter da2 = new SqlDataAdapter("select am from dbo.tokens", sc);
        DataTable dt2 = new DataTable();
        da2.Fill(dt2);
        for (int i = 0; i < dt2.Rows.Count; i++)
        {
            if ((string)dt2.Rows[i]["am"] == am)
            {
                cmd = new SqlCommand("select token from tokens where am like '" + am.ToString() + "'", sc);
                tokensYet = ((int)cmd.ExecuteScalar());
                tokensInt += tokensYet;
                cmd = new SqlCommand("update tokens set token='" + tokensInt + "' where am like '" + am.ToString() + "'", sc);
                cmd.ExecuteNonQuery();
                flag = 1;
            }
        }
        if (flag == 0)
        {
            cmd = new SqlCommand("Insert into tokens (am,token) values ('" + am.ToString() + "','" + Int32.Parse(tokens) + "')", sc);
            cmd.ExecuteNonQuery();
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
    }
    finally
    {
        sc.Close();
    }
}
```

## Μέθοδος updateToken

Η μέθοδος updateToken είναι εκείνη που μειώνει τα tokens (γεύματα) από το λογαριασμό του σπουδαστή, όταν αυτός καταναλώνει γεύματα στη λέσχη. Η μέθοδος δέχεται ένα όρισμα τον αριθμό μητρώου του σπουδαστή και εκτελεί το κατάλληλο query στον πίνακα tokens. Η κλήση της γίνεται μέσω του URL:

<http://localhost/service1/service1/TokenUpdate/{am}>.

```

public void updateToken(string am)
{
    //methodos pou meionei kata ena ta token tou foititi o opoios agorazei geuma dexomenh to am tou
    try
    {
        sc.Open();
        int tokensminus;
        SqlDataAdapter da2 = new SqlDataAdapter("select am from dbo.tokens", sc);
        DataTable dt2 = new DataTable();
        da2.Fill(dt2);

        cmd = new SqlCommand("select token from tokens where am like '" + am.ToString() + "'", sc);
        tokensminus = ((int)cmd.ExecuteScalar());
        tokensminus--;
        for (int i = 0; i < dt2.Rows.Count; i++)
        {
            if ((string)dt2.Rows[i]["am"] == am)
            {
                cmd = new SqlCommand("update tokens set token='" + tokensminus + "'where am like '" + am.ToString() + "' ", sc);
                cmd.ExecuteNonQuery();
            }
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
    }
    finally
    {
        sc.Close();
    }
}

```

## Μέθοδος GetTokens

Η μέθοδος GetTokens είναι μια int μέθοδος η οποία δέχεται σαν όρισμα τον αριθμό μητρώου του φοιτητή και επιστρέφει τα διαθέσιμα token (γεύματα) που έχει. Η κλήση της από τον Android Client γίνεται με το URL:

<http://localhost/Service1/service1/GetToken/{am}>. Η μέθοδος εκτελεί το query στον πίνακα tokens ώστε να λάβει τα διαθέσιμα γεύματα.

```

public int GetTokens(string am)
{
    //methodos pou epistrefei ta token tou foithth dexomeno to am tou
    int tokens=0;
    try
    {
        sc.Open();
        cmd = new SqlCommand("select token from tokens where am like '" + am.ToString() + "'", sc);
        tokens = ((int)cmd.ExecuteScalar());
    }
    catch(Exception ex)
    {
        Console.WriteLine(ex.Message);
    }
    finally
    {
        sc.Close();
    }
    return tokens;
}

```

## 4.4 Εφαρμογή Android

Η Android εφαρμογή που αναπτύχθηκε όπως προαναφέρθηκε χωρίζεται σε δύο εφαρμογές πελάτη, για να εξυπηρετήσει τις ανάγκες του εκπαιδευτικού ιδρύματος. Παρακάτω θα αναφερθούν πλήρως και τα δύο μέρη της εφαρμογής.

### 4.4.1 TEI LARISSA NFC APPLICATION

Η πρώτη εφαρμογή με τίτλο TEI LARISSA NFC APPLICATION εξυπηρετεί την γραμματεία και το εστιατόριο. Ξεκινώντας την εφαρμογή εμφανίζεται η αρχική οθόνη (Εικόνα 14):



Εικόνα 14: Αρχική Οθόνη Εφαρμογής

Σε αυτή την οθόνη ο χρήστης είναι σε θέση να επιλέξει την τοποθεσία στην οποία βρίσκεται. Στην οθόνη βρίσκονται δύο κουμπια: Γραμματεία (btnGram) και Εστιατόριο (btnEst). Ο κώδικας μέσω της setOnClickListener αποφασίζει ποιο Activity θα ξεκινήσει.

```

btnGram.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        // TODO Auto-generated method stub
        Intent i = new Intent(StartPageActivity.this, StartActivity.class);
        startActivity(i);
    }
});
btnEst.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        // TODO Auto-generated method stub
        Intent j=new Intent(StartPageActivity.this,estiatorio.class);
        startActivity(j);
    }
});

```

#### 4.4.1.1 Γραμματεία

Ο χρήστης επιλέγει την τοποθεσία «Γραμματεία».

Η αρχική οθόνη της Γραμματείας παρουσιάζεται στην Εικόνα 15:



Εικόνα 15: Scan Page Γραμματείας



Η κλάση που διαχειρίζεται την οθόνη είναι η StartActivity.java. Η εφαρμογή περιμένει τον χρήστη να φέρει σε επαφή την κινητή συσκευή με την NFC κάρτα και η μέθοδος onNewIntent() εξετάζει αν είναι άδεια ή γεμάτη, ή ακόμα πιο συγκεκριμένα αν είναι συμβατή με το NDEF Format ή όχι.

```
52 protected void onNewIntent(Intent intent) {
53     if (NfcAdapter.ACTION_TAG_DISCOVERED.equals(intent.getAction())) {
54         mytag = intent.getParcelableExtra(NfcAdapter.EXTRA_TAG);
55         try {
56             Ndef ndef = Ndef.get(mytag);
57             if (ndef != null)
58             {
59                 Parcelable[] rawMsgs =intent.getParcelableArrayExtra(NfcAdapter.EXTRA_NDEF_MESSAGES);
60                 NdefRecord relayRecord = ((NdefMessage)rawMsgs[0]).getRecords()[0];
61                 try {
62                     String gsonPayload=new String(relayRecord.getPayload(),"UTF-8");
63                     SRecordPayload record=new Gson().fromJson(gsonPayload, SRecordPayload.class);
64                     Intent i = new Intent(StartActivity.this, StudentInfo.class);
65                     i.putExtra("name", record.getFirstName().toString());
66                     i.putExtra("lastname",record.getLastName().toString());
67                     i.putExtra("am",record.getAm().toString());
68                     i.putExtra("food",record.isFoodEligible());
69                     i.putExtra("year_intro", record.getSemester().toString());
70                     startActivity(i);
71                 } catch (UnsupportedEncodingException e) {
72                     e.printStackTrace();
73                 }
74             } else {
75                 NdefFormatable format = NdefFormatable.get(mytag);
76                 if (format != null)
77                 {
78                     try {
79                         format.connect();
80                         Intent i = new Intent(StartActivity.this, MainActivity.class);
81                         startActivity(i);
82                     } catch (IOException e)
83                     {
84                         e.printStackTrace();
85                     }
86                 }
87             }
88         } catch (Exception e) {
89             e.printStackTrace();
90         }
91     }
92 }
```

Εικόνα 16: StartActivity.java

Εξετάζοντας αρχικά το κομμάτι του κώδικα που περιέχει το else, δηλαδή το γεγονός ότι η NFC κάρτα είναι άδεια με σκοπό τον αρχικό προγραμματισμό της με τα στοιχεία του φοιτητή. Σε αυτό το κομμάτι του κώδικα, η εφαρμογή ξεκινάει το επόμενο Activity, δηλαδή την κλάση MainActivity.java, η οποία προγραμματίζει τις κάρτες με τα στοιχεία του φοιτητή και τα στέλνει στη βάση δεδομένων.

Στην περίπτωση που η κάρτα του σπουδαστή είναι άδεια εμφανίζεται η παρακάτω οθόνη (Εικόνα 17):

ψ Android 100 NFC 14:05

**NfcTei**

**Όνομα**

**Επιθετο**

**Εξαμηνο Εισαγωγης**

**ΑΜ**

☐ **Σίτιση**

**ΕΓΓΡΑΦΗ**

Εικόνα 17: Οθόνη Εγγραφής Φοιτητή

Η κλάση που αντιστοιχεί στο συγκεκριμένο layout είναι όπως προαναφέρθηκε η Main-Activity.java.

Όπως βλέπουμε υπάρχουν 4 πεδία κειμένου (EditText) στα οποία ο χρήστης θα εισάγει τα στοιχεία του φοιτητή και ένα πεδίο ελέγχου (checkbox) στο οποίο επιλέγεται η ύπαρξη κάρτας σίτισης. Η κύρια ενέργεια που θα αναλυθεί είναι ο κώδικας που εκτελείται πατώντας το κουμπί ΕΓΓΡΑΦΗ (btnWrite).

```

64 btnWrite.setOnClickListener(new OnClickListener() {
65
66     public void onClick(View v) {
67         // TODO Auto-generated method stub
68         if (am.getText().toString().length() != 0 && name.getText().toString().length() != 0
69             && lastname.getText().toString().length() != 0 && name.getText().toString().length() != 0
70             && chb != null) {
71             try {
72                 if (mytag == null) {
73                     Toast.makeText(ctx,
74                         ctx.getString(R.string.error_detected),
75                         Toast.LENGTH_LONG).show();
76                 } else {
77                     // 192.168.0.93/Service1/Service1/PostName/3666/TEST/TEST/2002E/Y
78                     if (chb.isChecked()) {
79                         food = "Y";
80                     } else {
81                         food = "N";
82                     }
83                     String url = "http://192.168.0.92/Service1/Service1/PostName/"
84                         + am.getText().toString().toUpperCase()
85                         + "/"
86                         + name.getText().toString().toUpperCase()
87                         + "/"
88                         + lastname.getText().toString().toUpperCase()
89                         + "/"
90                         + year_intro.getText().toString().toUpperCase()
91                         + "/"
92                         + food.toString().toUpperCase();
93                     PostStudent data = new PostStudent();
94                     data.execute(url);
95                     write(am.getText().toString().toUpperCase(),
96                         name.getText().toString().toUpperCase(), lastname.getText().toString().toUpperCase(),
97                         chb.isChecked(), year_intro.getText().toString().toUpperCase(), mytag);

```

Εικόνα 18: MainActivity.java

Αν έχουν συμπληρωθεί όλα τα πεδία δημιουργείται το string με το url που θα καλέσει το WebService insertStudent (Μέθοδος InsertStudent). Η κλήση του WebService γίνεται μέσω της δημιουργίας AsyncTask το οποίο επιτρέπει την εκτέλεση αυτής της ενέργειας στο background και όχι στο κυρίως νήμα της εφαρμογής. Το AsyncTask φαίνεται στην παρακάτω εικόνα:

```

140 private class PostStudent extends AsyncTask<String, Void, Void> {
141
142     @Override
143     protected Void doInBackground(String... params) {
144         // TODO Auto-generated method stub
145         try {
146             HttpClient client = new DefaultHttpClient();
147             URI uri = new URI(params[0]);
148             HttpGet httpget = new HttpGet(uri);
149             client.execute(httpget);
150         } catch (Exception e) {
151             e.printStackTrace();
152         }
153         return null;
154     }
155
156 }

```

Το AsyncTask δέχεται ως όρισμα το URL(String) και καλεί το service ώστε να αποθηκευτούν τα στοιχεία στη βάση δεδομένων.

Η επόμενη ενέργεια που εκτελείται με το πάτημα του btnWrite είναι η εγγραφή των στοιχείων στην κάρτα. Για τον σκοπό αυτό πρέπει να δημιουργηθεί το NDEF Message στο οποίο θα ενθυλακωθεί το NDEF Record που θα γραφτεί στη NFC κάρτα.

Σε αυτό το σημείο πρέπει να αναφερθεί ο τύπος δεδομένων (αντικείμενο) και η κλάση που αντιπροσωπεύει τα στοιχεία που θα γραφτούν στην NFC κάρτα. Η κλάση SRecordPayload.java είναι το object με τα στοιχεία του φοιτητή.

```
public class SRecordPayload {

    public static final String RECORD_TYPE = "gr.thanos.nfctei:S";

    @Expose
    @SerializedName("am")
    private String am;
    @Expose
    @SerializedName("fn")
    private String firstName;
    @Expose
    @SerializedName("ln")
    private String lastName;
    @Expose
    @SerializedName("fe")
    private boolean foodEligible;
    @Expose
    @SerializedName("sm")
    private String semenster;
    /**
     *
     */
    public SRecordPayload() {

    }

    public SRecordPayload(String am, String firstName, String lastName,
        boolean isFoodEligible,String semenster) {
        super();
        this.am = am;
        this.firstName = firstName;
        this.lastName = lastName;
        this.foodEligible = isFoodEligible;
        this.semenster=semenster;
    }
}
```

Εικόνα 19: SRecordPayload.java

Η μέθοδος write() είναι εκείνη που γράφει το NDEF Message στην κάρτα.

```

private void write(String am, String firstName,String lastName,boolean isLunchEligible,String semester,
    Tag tag)
    throws IOException,
    FormatException {

    NdefRecord[] records = { createRecord(am,firstName,lastName,isLunchEligible,semester) };

    NdefMessage message = new NdefMessage(records);
    try {
        // Get an instance of Ndef for the tag.
        Ndef ndef = Ndef.get(tag);
        if (ndef != null) {
            ndef.connect();

            // Write the message
            ndef.writeNdefMessage(message);
        } else {
            NdefFormatable format = NdefFormatable.get(tag);
            if (format != null) {
                try {
                    format.connect();
                    format.format(message);

                } catch (IOException e) {

                }
            }
        }
        ndef.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

Εικόνα 20: Μέθοδος Write

Όπως βλέπουμε στον κώδικα (Εικόνα 21) καλείται η μέθοδος createRecord() η οποία δημιουργεί το NDEF Record και το επιστρέφει στην μέθοδο write(). Το record που θα γραφτεί στην κάρτα είναι σε μορφή JSON string και το type του record είναι TNF\_EXTERNAL\_TYPE το οποίο είναι μοναδικό ώστε να μην μπορεί οποιαδήποτε εφαρμογή να χρησιμοποιήσει την κάρτα παρα μόνο η συγκεκριμένη.

```

private NdefRecord createRecord(String am, String firstName,
    String lastName, boolean isLunchEligible,String semester) {

    String recordType = SRecordPayload.RECORD_TYPE;
    byte[] recordTypeBytes = recordType.getBytes();

    SRecordPayload recordPayload = new SRecordPayload(am, firstName,
        lastName, isLunchEligible,semester);

    GsonBuilder builder = new GsonBuilder();
    Gson gson = builder.create();

    String jsonPayload = gson.toJson(recordPayload);

    byte[] payloadBytes = null;
    try {
        payloadBytes = jsonPayload.getBytes("UTF-8");
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }

    if (payloadBytes == null)
        return null;

    return new NdefRecord(NdefRecord.TNF_EXTERNAL_TYPE, recordTypeBytes,
        new byte[0], payloadBytes);
}

```

Εικόνα 21: Η μέθοδος createRecord

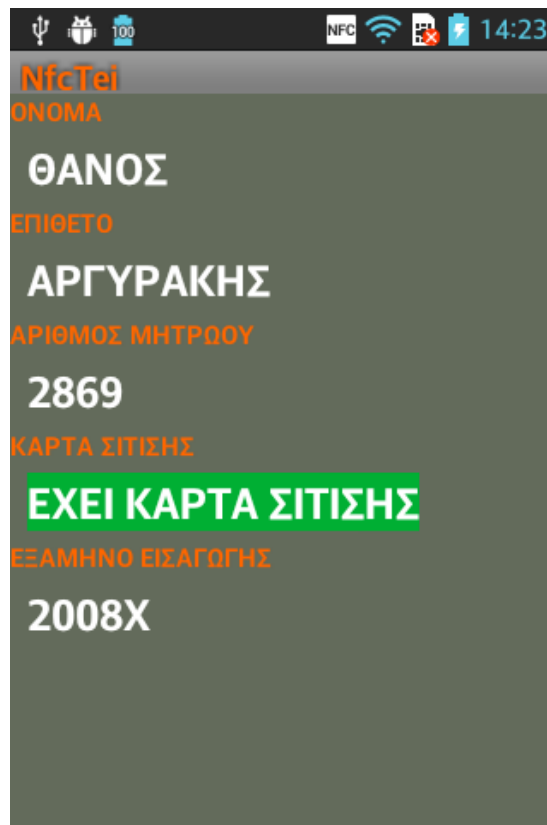
Η εφαρμογή της Γραμματείας, όπως προαναφέρθηκε, στην αρχική της οθόνη (Εικόνα 15) όταν έρθει σε επαφή με μια γεμάτη κάρτα (έχει ήδη το NDEF message με τα στοιχεία του φοιτητή) ξεκινάει ένα activity με σκοπό να δείξει στον χρήστη τα περιεχόμενά της.

```

Parcelable[] rawMsgs =intent.getParcelableArrayExtra(NfcAdapter.EXTRA_NDEF_MESSAGES);
NdefRecord relayRecord = ((NdefMessage)rawMsgs[0]).getRecords()[0];
try {
    String gsonPayload=new String(relayRecord.getPayload(),"UTF-8");
    SRecordPayload record=new Gson().fromJson(gsonPayload, SRecordPayload.class);
    Intent i = new Intent(StartActivity.this, StudentInfo.class);
    i.putExtra("name", record.getFirstName().toString());
    i.putExtra("lastname",record.getLastName().toString());
    i.putExtra("am",record.getAm().toString());
    i.putExtra("food",record.isFoodEligible());
    i.putExtra("year_intro", record.getSemester().toString());
    startActivity(i);
}

```

Σε αυτό το σημείο εάν υπάρχει NDEF Message στην κάρτα το διαβάζει και αποθηκεύει το πρώτο record του. Έπειτα με τη βιβλιοθήκη GSON (αφού το record είναι σε μορφή JSON) δημιουργείται το αντικείμενο SRecordPayload και πλέον μέσω των Getter της κλάσης SRecordPayload.java παίρνει τα στοιχεία της κάρτας και τα εισάγει ως μεταβλητές στο επόμενο activity με όνομα StudentInfo.java με σκοπό την προβολή τους στον χρήστη. Η οθόνη προβολής των στοιχείων είναι η παρακάτω (Εικόνα 22):



Εικόνα 22: Οθόνη προβολής περιεχομένου κάρτας

Η κλάση που εμφανίζει τα στοιχεία του φοιτητή στην οθόνη είναι η StudentInfo.java.

```
passedName = getIntent().getExtras().getString("name");
passedLastName=getIntent().getExtras().getString("lastname");
passedAm=getIntent().getExtras().getString("am");
passedFood=getIntent().getExtras().getBoolean("food");
passedSm=getIntent().getExtras().getString("year_intro");
name.setText(passedName);
lastname.setText(passedLastName);
am.setText(passedAm);
year.setText(passedSm);
if(passedFood){
    food.setBackgroundResource(R.color.green);
    food.setText("ΕΧΕΙ ΚΑΡΤΑ ΣΙΤΙΣΗΣ");
}else{
    food.setText("ΔΕΝ ΕΧΕΙ ΚΑΡΤΑ ΣΙΤΙΣΗΣ");
}
```

Η κλάση StudentInfo δέχεται τις τιμές από το προηγούμενο activity και τις παρουσιάζει στην οθόνη του κινητού.

#### 4.4.1.2 Εστιατόριο

Ο χρήστης επιλέγει την τοποθεσία «ΕΣΤΙΑΤΟΡΙΟ».

Η αρχική οθόνη παρουσιάζεται στην Εικόνα 23 και εδώ πραγματοποιείται ο έλεγχος της κάρτας σίτισης για την είσοδο του σπουδαστή στο εστιατόριο όπως επίσης και για τους φοιτητές που δεν έχουν κάρτα σίτισης και θέλουν να γεμίσουν γεύματα την κάρτα τους έναντι χρηματικού ποσού.



Εικόνα 23: ScanPage Εστιατορίου

Όταν έρθει σε επαφή η NFC κάρτα με το κινητό τηλέφωνο, η εφαρμογή ελέγχει την ύπαρξη κάρτας σίτισης :

```
if (record.isFoodEligible()) {  
    img.setBackgroundResource(R.drawable.tikgreen);  
    img.setVisibility(View.VISIBLE);  
    chkfood.execute("http://192.168.0.92/service1/Service1/Food/"+record.getAm().toString());  
}
```

Η κλάση που διαχειρίζεται το εστιατόριο είναι η estiatorio.java. Εάν λοιπόν υπάρχει κάρτα σίτισης εμφανίζεται κατάλληλη ένδειξη στην οθόνη (Εικόνα 25) και γίνεται η κλήση στο web service Food (Μέθοδος foodEligable) για να καταγραφεί στη βάση δεδομένων ότι ο εν λόγω φοιτητής πήρε το γεύμα του και δεν δικαιούται δεύτερο την ίδια ημέρα. Η κλήση στο Webservice γίνεται με τη χρήση της κλάσης AsyncTask με όνομα CheckoodEligable (Εικόνα 24)



```

private class CheckFoodEligable extends AsyncTask<String, Void, String>{
    BufferedReader reader = null;

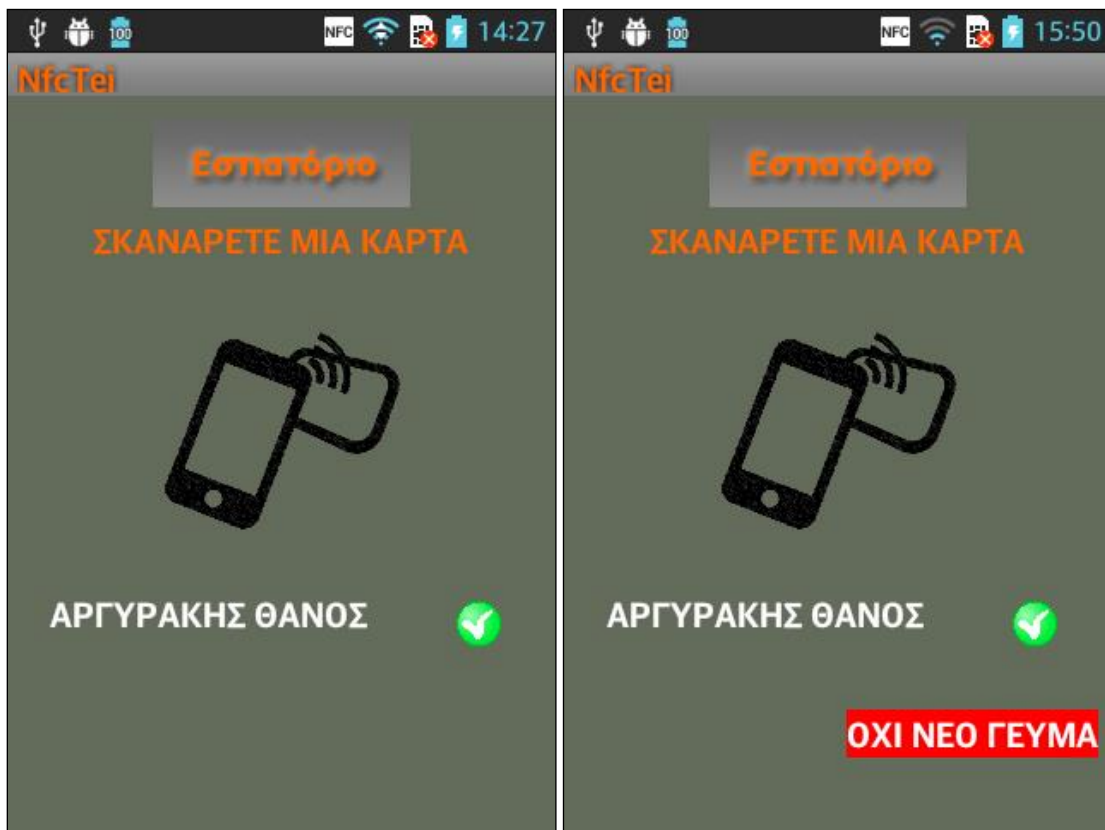
    @Override
    protected String doInBackground(String... params) {
        // TODO Auto-generated method stub
        try{
            HttpClient client = new DefaultHttpClient();
            URI uri = new URI(params[0]);
            HttpGet httpget = new HttpGet(uri);
            HttpResponse response = client.execute(httpget);
            InputStream stream = response.getEntity().getContent();
            reader = new BufferedReader(new InputStreamReader(stream));
            StringBuffer buffer = new StringBuffer("");
            String line = "";
            String newline = System.getProperty("line.separator");
            while ((line = reader.readLine()) != null) {
                buffer.append(line + newline);
            }
            reader.close();
            data = buffer.toString();
        }catch(Exception e){
            e.printStackTrace();
        }
        return data;
    }

    @Override
    protected void onPostExecute(String result) {
        // TODO Auto-generated method stub
        super.onPostExecute(result);
        if(data.contains("false"))
            txtStatus.setText("OXI NEO ΓΕΥΜΑ");
    }
}
}

```

Εικόνα 24: AsyncTask CheckFoodEligable

Στην doInBackground γίνεται η κλήση στο Web Service Food και αν η απάντηση είναι false τότε στην onPostExecute, όταν πλέον η διεργασία επιστρέψει στο κυρίως νήμα του UI (User Interface), εμφανίζεται το μήνυμα «OXI NEO ΓΕΥΜΑ» όπως φαίνεται στην Εικόνα 25



Εικόνα 25: Φοιτητής Με καρτα σίτισης

Στην κλάση `estiatorio.java`, υπάρχει ο παρακάτω κώδικας ο οποίος διαχειρίζεται την περίπτωση φοιτητών που δεν έχουν καρτα σίτισης.

```

if (record.isFoodEligible()) {
    img.setBackgroundResource(R.drawable.tikgreen);
    img.setVisibility(View.VISIBLE);
    chkfood.execute("http://192.168.0.79/service1/Service1/Food/"+record.getAm().toString());
}
else{
    HaveToken havToken=new HaveToken();
    havToken.execute("http://192.168.0.79/service1/Service1/GetToken/"+record.getAm().toString(),
        record.getAm().toString());
    img.setBackgroundResource(R.drawable.tikred);
    img.setVisibility(View.VISIBLE);
    img.setOnClickListener(new OnClickListener() {

        public void onClick(View v) {
            AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(ctx);
            alertDialogBuilder.setTitle("ΥΠΗΡΕΣΙΑ ΕΣΤΙΑΤΟΡΙΟΥ");
            alertDialogBuilder.setView(getLayoutInflater().inflate(R.layout.dialogtoken, null));
            alertDialogBuilder.setMessage("ΦΟΡΤΩΣΗ ΓΕΥΜΑΤΩΝ");
            alertDialogBuilder.setCancelable(false);
            alertDialogBuilder.setNegativeButton("Άκυρο",new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog,int id) {
                }
            });
            alertDialogBuilder.setPositiveButton("Αποθήκευση",new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog,int id) {
                    Dialog d=(Dialog) dialog;
                    try{
                        token=(EditText)d.findViewById(R.id.token);
                        String test=token.getText().toString();
                        InsertToken insToken=new InsertToken();
                        insToken.execute("http://192.168.0.79/service1/service1/Token/"+record.getAm().toString()+
                            "/" +token.getText().toString());
                        Integer oldtokens=Integer.parseInt(txtStatus.getText().toString());
                        Integer newToken=Integer.parseInt(test);
                        oldtokens+=newToken;
                        txtYpoloipo.setText("NEO ΥΠΟΛΟΙΠΟ");
                        txtStatus.setText(oldtokens.toString());
                    }catch(Exception e){
                        e.printStackTrace();
                    }
                }
            });

            AlertDialog alertDialog = alertDialogBuilder.create();
            alertDialog.show();
        }
    });
}
};

```

Στην περίπτωση του else (που ο φοιτητής δεν έχει κάρτα σίτισης) ελέγχεται αν έχει διαθέσιμα γεύματα μέσω της κλήσης στο Web Service GetTokens (Μέθοδος GetTokens):

```

HaveToken havToken=new HaveToken();
havToken.execute("http://192.168.0.92/service1/Service1/GetToken/"+record.getAm().toString());

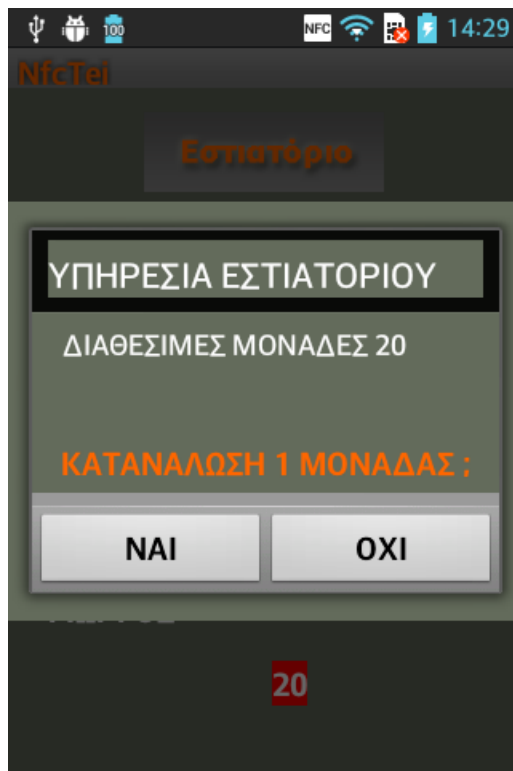
```

Ο έλεγχος γίνεται μέσα στο AsyncTask HaveToken. Εάν ο φοιτητής δεν έχει γεύματα εμφανίζεται στην οθόνη η Εικόνα 26

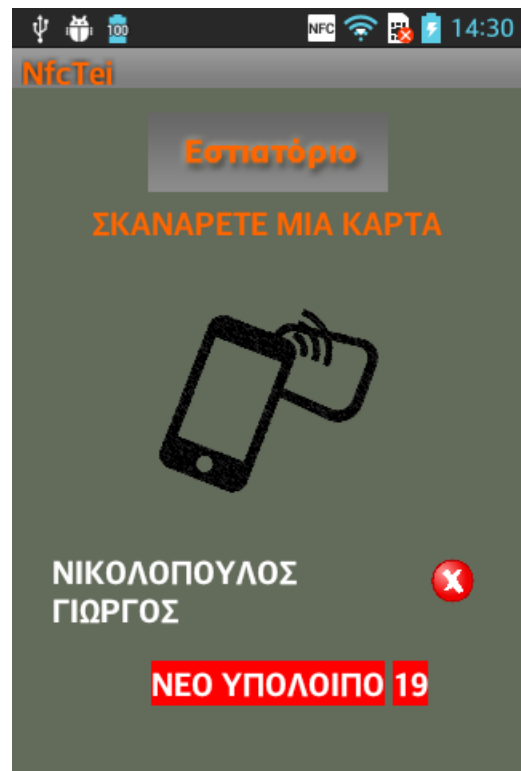


Εικόνα 26

Η επόμενη κατάσταση που διαχειρίζεται η εφαρμογή του εστιατορίου είναι όταν ο φοιτητής που έχει ήδη γεύματα στην κάρτα του θέλει να καταναλώσει μία μονάδα για να πάρει το γεύμα του. Εάν λοιπόν η κλήση στο service HaveToken επιστρέψει ότι έχει διαθέσιμα γεύματα μέσω της `onPostExecute` στο `AsyncTask` της κλάσης `HaveToken` βγαίνει το κατάλληλο `AlertDialog` το οποίο ρωτάει αν θέλει να καταναλώσει μια μονάδα φαγητού ώστε να εισέλθει στο εστιατόριο (**Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**):



Εικόνα 27: Ερώτηση Κατανάλωσης

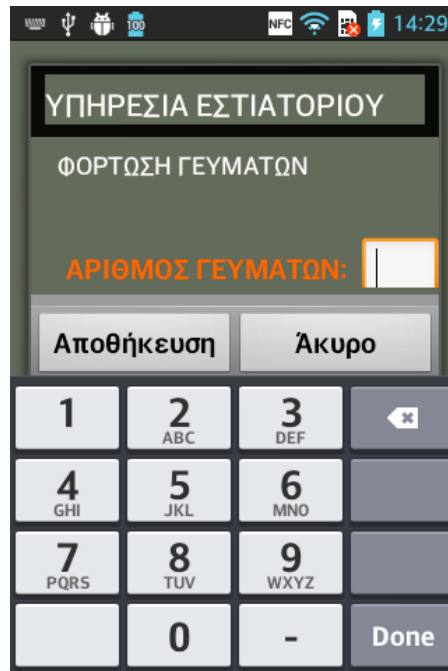


Εικόνα 28: Οθόνη μετά από κατανάλωση

Στην περίπτωση που πατήσει ΝΑΙ για να μειωθεί κατά μία μονάδα το υπόλοιπο των γευμάτων καλείται το service TokenUpdate

```
TokenMinus tkMinus=new TokenMinus();
tkMinus.execute("http://192.168.0.92/service1/service1/TokenUpdate/"+amOnPost.toString());
//dokimh
Integer pontoι=Integer.parseInt(txtStatus.getText().toString());
pontoι--;
txtYpoloipo.setText("ΝΕΟ ΥΠΟΛΟΙΠΟ");
txtStatus.setText(pontoι.toString());
.....
```

Τώρα εάν ο σπουδαστής επιθυμεί να γεμίσει τον λογαριασμό του με γεύματα, πατώντας ο χρήστης της εφαρμογής το κόκκινο εικονίδιο (Εικόνα 26), του εμφανίζεται το εξής AlertDialog:



Όταν πατήσει το κουμπί αποθήκευση καλείται το Webservice insertToken(Μέθοδος insertToken) το οποίο προσθέτει τα γεύματα στο λογαριασμό του φοιτητή

```
InsertToken insToken=new InsertToken();
insToken.execute("http://192.168.0.92/service1/service1/Token/"+record.getAm().toString()+"/"+token.getText().toString());
```

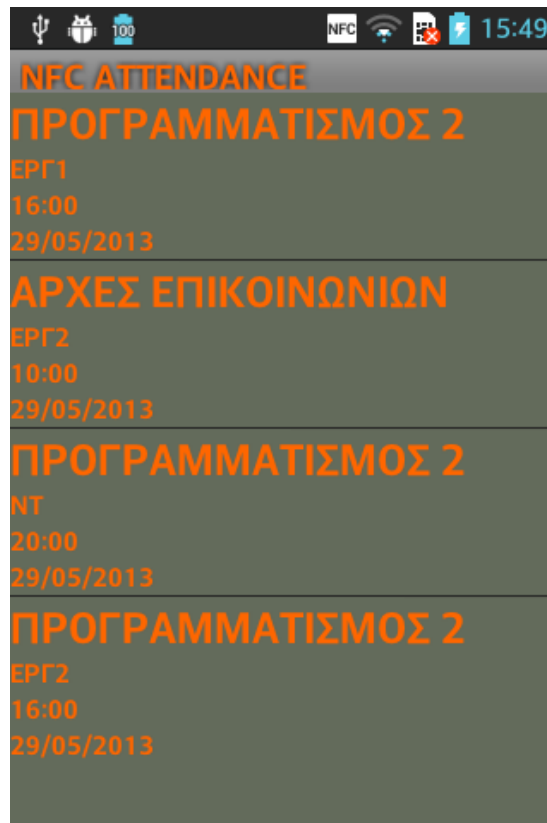
Για την κλήση του service στο background της εφαρμογής δημιουργήθηκε ένα AsyncTask με όνομα InsertToken το οποίο ενημερώνει τη βάση δεδομένων για τη συναλλαγή του φοιτητή.

```
private class InsertToken extends AsyncTask<String, Void, Void>{

    @Override
    protected Void doInBackground(String... params) {
        // TODO Auto-generated method stub
        try{
            HttpClient client = new DefaultHttpClient();
            URI uri = new URI(params[0]);
            HttpGet httpget = new HttpGet(uri);
            HttpResponse response = client.execute(httpget);
        }catch(Exception ex)
        {
            ex.printStackTrace();
        }
        return null;
    }
}
```

#### 4.4.2 NFC Attendance

Η δεύτερη εφαρμογή αναπτύχθηκε για να αυτοματοποιήσει τη διαδικασία της παρουσίας των φοιτητών στα εργαστήρια. Στην αρχή κάθε εργαστηρίου ο καθηγητής ξεκινάει την εφαρμογή NFC Attendance και αμέσως εμφανίζεται στην οθόνη λίστα με τα εργαστήρια της εκάστοτε ημέρας, από τα οποία πρέπει κάνει την επιλογή του (Εικόνα 29)



Εικόνα 29: Αρχική Οθόνη NFC Attendance

Η κλάση που αντιστοιχεί στο activity είναι η TeilarMain.java. Η εφαρμογή στο ξεκίνημά της καταγράφει την ημέρα αυτόματα ώστε να κληθεί το Web Service που θα επι-

στρέψει τη λίστα με τα εργαστήρια που διδάσκονται την εκάστοτε ημέρα.

```
SimpleDateFormat sdf = new SimpleDateFormat("EEEE");
String day = sdf.format(d);
if (day.contentEquals("Monday")) {
    daypost = "1";
}
if (day.contentEquals("Wednesday")) {
    daypost = "3";
}
if (day.contentEquals("Tuesday")) {
    daypost = "2";
}
if (day.contentEquals("Thursday")) {
    daypost = "4";
}
if (day.contentEquals("Friday")) {
    daypost = "5";
}
```

Η λίστα περιέχει αντικείμενα της κλάσης Collection.java

```
public class Collection {
    private List<Course> GetCollectionResult;
    public List<Course> getCollection(){
        return GetCollectionResult;
    }
    public void setCollection(List<Course> GetCollectionResult){
        this.GetCollectionResult=GetCollectionResult;
    }
}
```

Όπως φαίνεται έχουμε μια λίστα απο Course αντικείμενα. Η κλάση Course.java αντιπροσωπεύει το κάθε μάθημα με ιδιότητες όπως το όνομα, το id και την ώρα έναρξης.

Ο παρακάτω κώδικας δημιουργεί την λίστα απο τα αντικείμενα που επιστρέφει το Web Service. Η απάντηση του service είναι ένα JSON string όπου με τη βιβλιοθήκη GSON γίνεται το parse σε αντικείμενα τύπου Collection.class.

```
Collection collections = new Gson().fromJson(data,Collection.class);
for (Course course : collections.getCollection()) {
    fetch.add(course);
}
return fetch;
} catch (Exception e) {
    e.printStackTrace();
}
```

Αφού λοιπόν φορτωθεί η λίστα, μέσω του onItemClickListener ξεκινάει το activity στο οποίο θα πραγματοποιηθεί η διαδικασία του παρουσιολογίου.



```
private AdapterView.OnItemClickListener onListener = new AdapterView.OnItemClickListener() {

    public void onItemClick(AdapterView<?> parent, View view, int position,
        long id) {
        // TODO Auto-generated method stub
        Intent i = new Intent(TeilarMain.this, Nfc.class);
        Course cn = new Course();
        cn = fetch.get(position);
        i.putExtra("name", cn.getName().toString());
        i.putExtra("id", cn.getId());
        i.putExtra("hall", cn.getHall().toString());
        i.putExtra("time", cn.getTime().toString());
        // i.putExtra("address", cn.getAddress().toString());
        startActivity(i);
    }
};
```

Εδώ λοιπόν, όταν ο καθηγητής επιλέξει το εργαστήριο που διδάσκει, αμέσως ξεκινάει το activity Nfc.java και η εφαρμογή δείχνει την εξής οθόνη (Εικόνα 30):



Εικόνα 30: Οθόνη Σκαναρίσματος

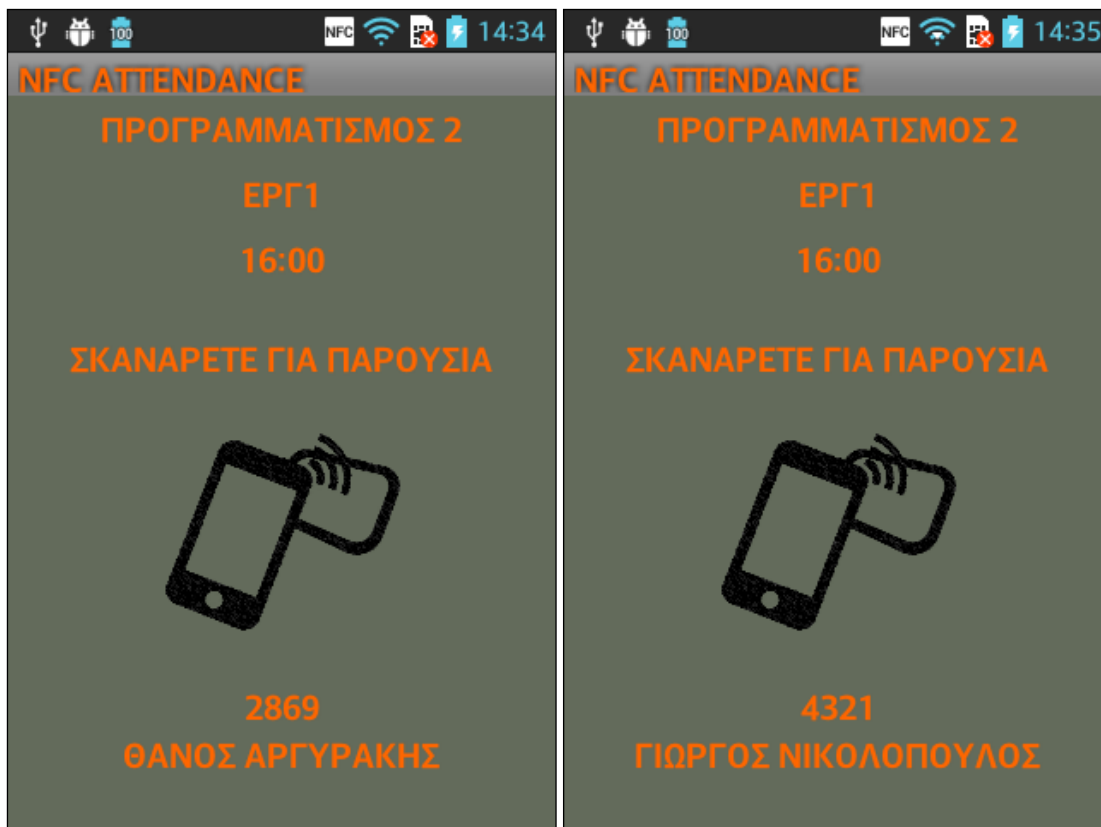
Η κλάση Nfc.java περιέχει την μέθοδο onNewIntent στην οποία όταν έρθει σε επαφή το κινητό τηλέφωνο με την NFC κάρτα παίρνει τα δεδομένα της κάρτας και καλεί το service για να αποθηκεύσει στη βάση δεδομένων την παρουσία του φοιτητή.

```

protected void onNewIntent(Intent intent) {
    if (NfcAdapter.ACTION_TAG_DISCOVERED.equals(intent.getAction())) {
        String am = null;
        mytag = intent.getParcelableExtra(NfcAdapter.EXTRA_TAG);
        Parcelable[] rawMsgs = intent.getParcelableArrayExtra(NfcAdapter.EXTRA_NDEF_MESSAGES);
        NdefMessage[] msgs = new NdefMessage[rawMsgs.length];
        System.arraycopy(rawMsgs, 0, msgs, 0, msgs.length);
        NdefRecord relayRecord = msgs[0].getRecords()[0];
        String recType = new String(relayRecord.getType());
        if (recType.equals(SRecordPayload.RECORD_TYPE)) {
            SRecordPayload s = getSRecordPayload(relayRecord.getPayload());
            am = s.getAm();
            Postdata data = new Postdata();
            data.execute("http://192.168.0.92/service1/Service1/DayLab/" + passedInt + "/" + am.toString());
            txtAM.setText(am);
            txtName.setText(s.getFirstName().toString() + " " + s.getLastName().toString());
        }
    }
}

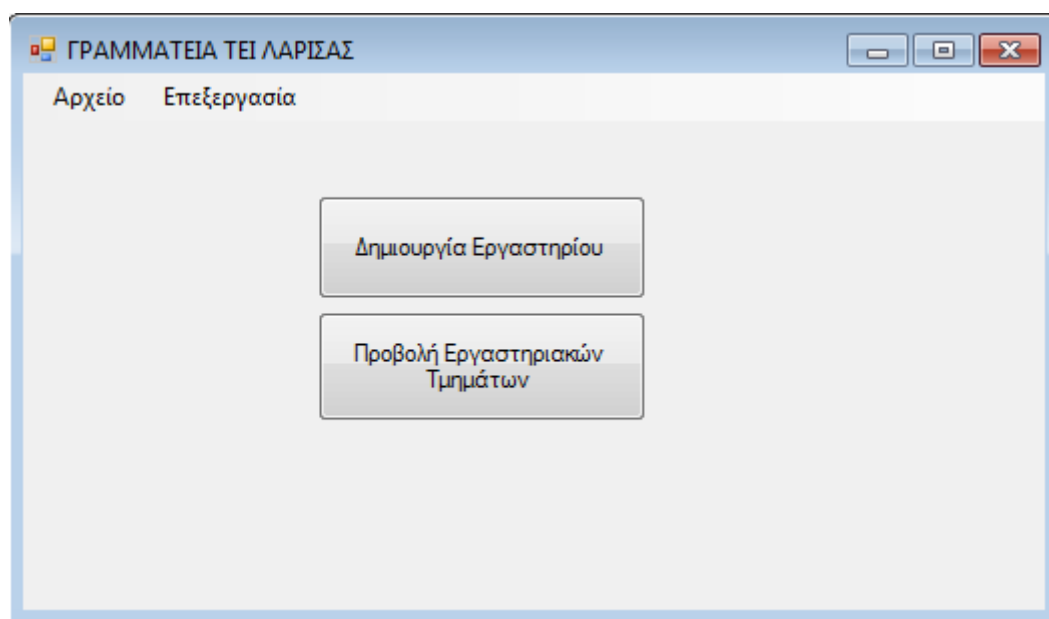
```

Όταν ο καθηγητής «σκανάρει» την κάρτα του φοιτητή βλέπει στην οθόνη το όνομα του φοιτητή που «σκάνανε» τελευταίο.



## 4.5 Desktop Εφαρμογή

Η Desktop εφαρμογή αναπτύχθηκε με σκοπό την αυτόματης εξαγωγή παρουσιολογίου καθώς και καταστάσεων επαρκούς παρακολούθησης. Η εφαρμογή επίσης τροφοδοτεί την NFC Attendance με τα εργαστήρια καθώς στην αρχή του εξαμήνου η γραμματοδεία θα είναι σε θέση να αποθηκεύσει τα εργαστήρια τα οποία έχουν προγραμματιστεί για το εκάστοτε εξάμηνο. Η αρχική οθόνη της εφαρμογής είναι η Εικόνα 31:



Εικόνα 31: Αρχική Οθόνη Εφαρμογής

Επιλέγοντας το «Δημιουργία Εργαστηρίου» εμφανίζεται η φόρμα δημιουργίας του εργαστηριακού τμήματος (Εικόνα 32):

The image shows a form window titled 'Δημιουργία Εργαστηριακού Τμήματος'. It contains several input fields and dropdown menus. The labels on the left are: 'Όνομα Εργαστηρίου', 'Ημέρα Διαξαγωγής', 'Ωρα Έναρξης', 'Αίθουσα', 'Διδάσκων', and 'Εξάμηνο Διεξαγωγής'. Each label is followed by a control element: a dropdown menu for the first four, a text box for 'Διδάσκων', and a text box followed by a small dropdown arrow for 'Εξάμηνο Διεξαγωγής'. At the bottom center of the form is a button labeled 'Εγγραφή'.

Εικόνα 32: Φόρμα Δημιουργίας Εργαστηρίου

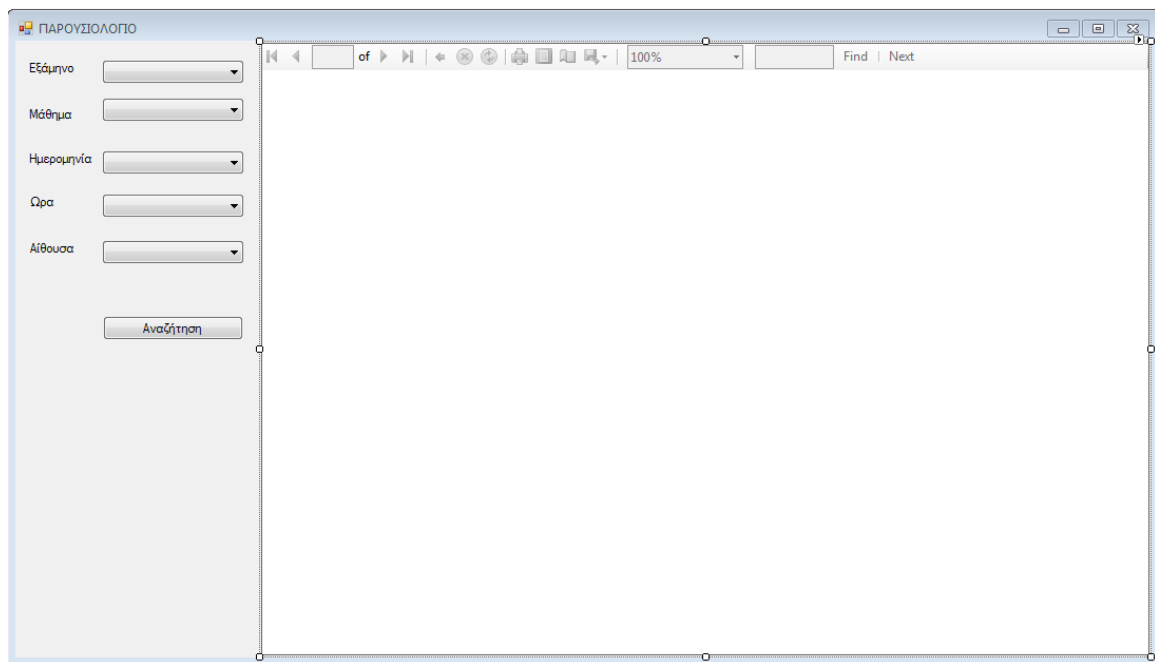
```

private void btnSave_Click(object sender, EventArgs e)
{
    try
    {
        int flag = 0;
        SqlDataAdapter da2 = new SqlDataAdapter("select day_lab,time_start,hall from dbo.lab", sc);
        DataTable dt2 = new DataTable();
        da2.Fill(dt2);
        sc.Open();
        int id_course;
        string namecourse = cmbNames.Text;
        for (int i = 0; i < dt2.Rows.Count; i++)
        {
            if (((string)dt2.Rows[i]["day_lab"] == cmbDay.Text) && ((string)dt2.Rows[i]["time_start"] == cmbTime.Text)
                && ((string)dt2.Rows[i]["hall"] == cmbHall.Text))
            {
                flag = 1;
            }
        }
        if (flag == 1)
        {
            MessageBox.Show("Αδύνατη η δημιουργία Τμήματος την συγκεκριμένη ώρα και στην συγκεκριμένη αίθουσα");
        }
        if (flag != 1)
        {
            cmd2 = new SqlCommand("select id_course from course where name_course like '" + namecourse + "'", sc);
            id_course = ((int)cmd2.ExecuteScalar());
            string runsem=txtRunSem.Text.ToString()+cmbsem.Text.ToString();
            if(id_course!=0 && cmbNames.Text != "" && cmbTime.Text!=" " && cmbDay.Text!=" " && cmbsem.Text!=" "
                && txtRunSem.Text!=" " && txtTeach.Text!=" " && cmbHall.Text!=" "){
                cmd = new SqlCommand("Insert into lab (id_course,day_lab,time_start,hall,teacher,run_semenster) values('"
                    + id_course + "','" + cmbDay.Text.ToString() + "','" + cmbTime.Text.ToString() + "','" + cmbHall.Text.ToString() + "','"
                    + txtTeach.Text.ToString() + "','" + runsem.ToString() + "')", sc);
                cmd.ExecuteNonQuery();
                MessageBox.Show("Η εγγραφή καταχωρήθηκε επιτυχώς");
            }else{
                MessageBox.Show("Παρακαλώ συμπληρώστε όλα τα στοιχεία");
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        sc.Close();
    }
}

```

Στη φόρμα δημιουργίας, εφόσον συμπληρωθούν όλα τα πεδία, γίνονται οι απαραίτητοι έλεγχοι για να διασφαλιστεί ότι οι εγγραφές θα είναι σωστές και δεν θα υπάρχουν εργαστήρια την ίδια ημέρα και ώρα στην ίδια αίθουσα. Όταν ο χρήστης από τη γραμματοτεία πατήσει το κουμπί «Εγγραφή» εκτελείται το query στη βάση και αποθηκεύονται τα δεδομένα στον πίνακα lab της βάσης university.

Το επόμενο στοιχείο που προσφέρει η Desktop εφαρμογή της γραμματείας είναι τα παρουσιολόγια για κάθε ημέρα που διεξήχθη εργαστήριο. Επιλέγοντας «Αρχείο» στην αρχική οθόνη (Εικόνα 31) εμφανίζεται ένα μενού με πρώτη επιλογή «Παρουσιολόγιο». Όταν ο χρήστης επιλέξει το παρουσιολόγιο ανοίγει η εξής φόρμα:



Εικόνα 33: Φόρμα Παρουσιολογίου

Σε αυτή τη φόρμα οι προεπιλεγμένες τιμές συμπληρώνονται δυναμικά ανάλογα με τα «Σκαναρίσματα» της εφαρμογής NFC Attendance. Πιο συγκεκριμένα, με τις καθημερινές παρουσίες δημιουργούνται τα Μαθήματα. Επιλέγοντας συγκεκριμένο μάθημα γεμίζει η λίστα με τις ημερομηνίες που διεξήχθη το συγκεκριμένο μάθημα καθώς και οι ώρες, αίθουσες. Η κλάση που αναπτύχθηκε για αυτό το σκοπό είναι η FormParousies.cs.

Όταν ο καθηγητής επιλέξει το εργαστήριο που θέλει να δει τους παρόντες φοιτητές και πατήσει το κουμπί αναζήτηση θα δει στην οθόνη του την Εικόνα 34 με το παρουσιολόγιο.

ΠΑΡΟΥΣΙΟΛΟΓΙΟ

Εξάμηνο: 2013Ε

Μάθημα: ΑΡΧΕΣ ΕΠΙΚΟΙΝΩΝΙΩΝ

Ημερομηνία: 29/05/2013

Ωρα: 10:00

Αίθουσα: ΕΡΓ2

Αναζήτηση

ΠΑΡΟΝΤΕΣ ΦΟΙΤΗΤΕΣ ΣΤΟ ΕΡΓΑΣΤΗΡΙΟ:  
ΑΡΧΕΣ ΕΠΙΚΟΙΝΩΝΙΩΝ  
ΗΜ/ΝΙΑ: 29/05/2013  
ΩΡΑ: 10:00  
ΑΙΘΟΥΣΑ: ΕΡΓ2

Όνομα	Επώνυμο	ΑΜ	Εξάμηνο Εισαγωγής	Μάθημα	Αίθουσα
ΘΑΝΟΣ	ΑΡΓΥΡΑΚΗΣ	2869	2008Χ	ΑΡΧΕΣ ΕΠΙΚΟΙΝΩΝΙΩΝ	ΕΡΓ2
ΓΙΩΡΓΟΣ	ΝΙΚΟΛΟΠΟΥΛΟΣ	4321	2010Ε	ΑΡΧΕΣ ΕΠΙΚΟΙΝΩΝΙΩΝ	ΕΡΓ2

Εικόνα 34: Φόρμα Παρουσιολογίου

Η παραπάνω εικόνα είναι ένα πλήρες report το οποίο μπορεί να εξαχθεί σε μορφή pdf, excel ή word. Τη δυνατότητα αυτή την προσφέρει το Rerport System της Microsoft (RDLC).

Μια ακόμη δυνατότητα που προσφέρει η εφαρμογή είναι η εξαγωγή των φοιτητών με επαρκή παρακολούθηση σε συγκεκριμένο τμήμα. Ο καθηγητής επιλέγει απο την αρχική οθόνη την επιλογή «Αρχείο» και επιλέγει «Καταστάσεις Επαρκής Παρακολούθησης». Η Φόρμα που θα δει στη οθόνη του είναι η παρακάτω (Εικόνα 35):

Εικόνα 35: Φόρμα Επαρκούς Παρακολούθησης

Το κριτήριο επαρκούς παρακολούθησης είναι ο σπουδαστής να έχει παρακολουθήσει τουλάχιστον το 80% των εργαστηριακών μαθημάτων που πραγματοποιήθηκαν κατά τη διάρκεια όλου του εξαμήνου.

Όταν ο χρήστης επιλέξει «Αναζήτηση Φοιτητών με επαρκή παρακολούθηση» και εφόσον έχει δώσει τις πληροφορίες για το τμήμα που ενδιαφέρεται, θα εμφανιστεί η εξής πλήρης αναφορά:

Ονομα	Επώνυμο	ΑΜ	ΒΑΘΜΟΣ
ΓΙΑΝΝΗΣ	ΤΖΩΡΤΖΗΣ	1111	
ΘΑΝΟΣ	ΑΡΓΥΡΑΚΗΣ	2900	
ΙΑΣΩΝΑΣ	ΣΕΡΙΑΤΟΣ	4444	

## 4.6 Merodroid

Το Merodroid είναι ένας ιστότοπος (Website) και πιο συγκεκριμένα μία μηχανή αναζήτησης για Android components. Είναι βασισμένο στο merobase, το οποίο αποτελεί παρομοίως μηχανή αναζήτησης για software components (κομμάτια κώδικα) γραμμένον σε διάφορες γλώσσες προγραμματισμού. Αξίζει να σημειωθεί ότι το πρώτο συνθετικό "mero" της λέξης "merodroid" προέρχεται από την ελληνική λέξη "μέρος", εννοώντας part. Το Merodroid απευθύνεται σε Android προγραμματιστές παρέχοντάς τους τη δυνατότητα να αναζητήσουν έτοιμα κομμάτια κώδικα (κλάσεις της java) όπως Activities, Fragments, Intent Filters, Broadcas Receives κλπ. Όπως αναφέρεται και στο site περιέχει πάνω από 50.000 Android components.

Σκοπός του είναι να διευκολύνει τους προγραμματιστές ώστε να χρησιμοποιούν έτοιμα κομμάτια κώδικα. Κάθε χρήστης του Merodroid έχει τη δυνατότητα να αναρτήσει κι αυτός με τη σειρά του τον δικό του κώδικα, τον οποίο κρίνει ως επανα-χρησιμοποιήσιμο.

Όπως κάθε πληροφοριακό σύστημα έτσι και το Merodroid έχει περιθώρια βελτίωσης. Χρησιμοποιώντας το Merodroid, εντόπισα κάποια σημεία, τα οποία κατά τη γνώμη μου χρειάζονται βελτίωση. Το site χρήζει βελτίωσης ως προς την πλοήγηση του χρήστη. Το σημείο που υστερεί είναι όταν ο χρήστης πληκτρολογεί τη λέξη κλειδί, για τη οποία αναζητεί component. Αμέσως του εμφανίζεται η λίστα με τα αποτελέσματα. Επιλέγοντας λοιπόν το επιθυμητό component βλέπει τον κώδικα και όταν θελήσει να επιστρέψει πίσω στη λίστα το site δεν ανταποκρίνεται ορθώς και δεν τον μεταφέρει στη λίστα αποτελεσμάτων. Συνεπώς, ο χρήστης χάνει τη λίστα με τα αποτελέσματα μέσα στα οποία αναζητεί τον κώδικα που τον ενδιαφέρει.

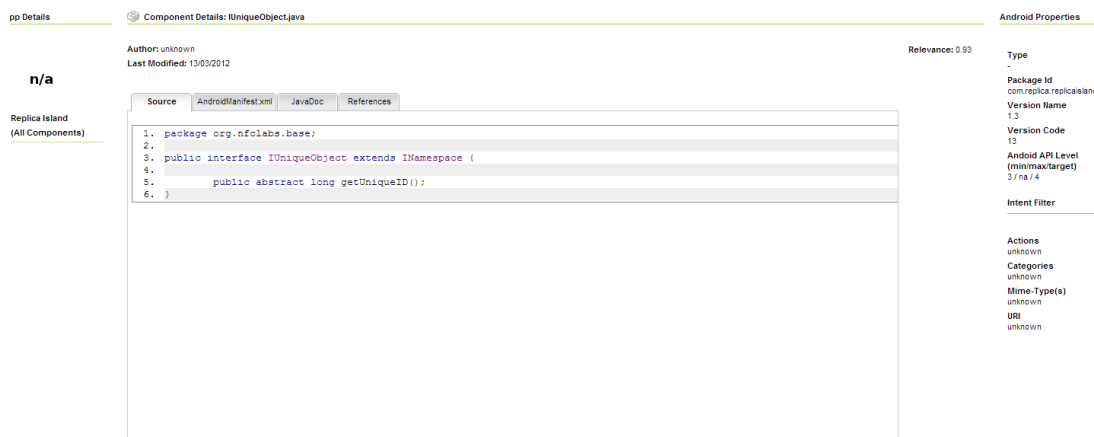
Ας δούμε τώρα ένα παράδειγμα αναζήτησης component στο Merodroid. Ο χρήστης πληκτρολογεί τη λέξη NFC στο πεδίο αναζήτησης, όπως φένεται στην Εικόνα 36 και στην Εικόνα 37





Εικόνα 36: Αναζήτηση component

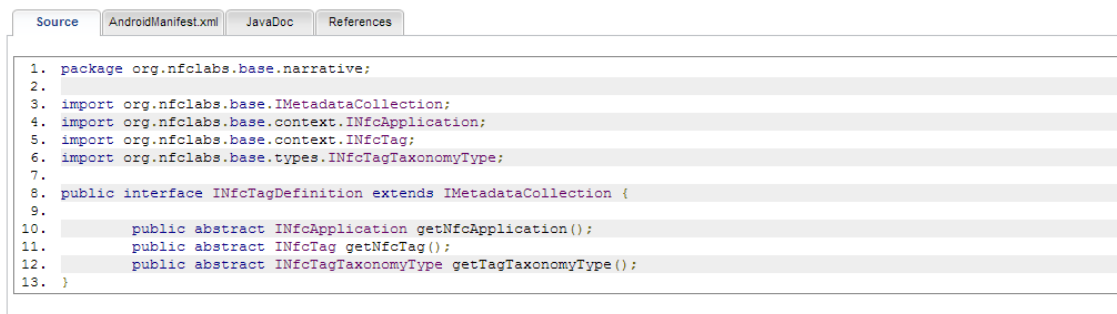
Όπως φέεται στις εικόνες τα ονόματα των component δεν ταιριάζουν με την λέξη κλειδί που δόθηκε για αναζήτηση γεγονός το οποίο δεν διευκολύνει τον προγραμματιστή να εντοπίσει το επιθυμητό component. Πατώντας στο πρώτο αποτέλεσμα ο χρήστης βλέπει τον πηγαίο κώδικα του component όπως φέεται στην



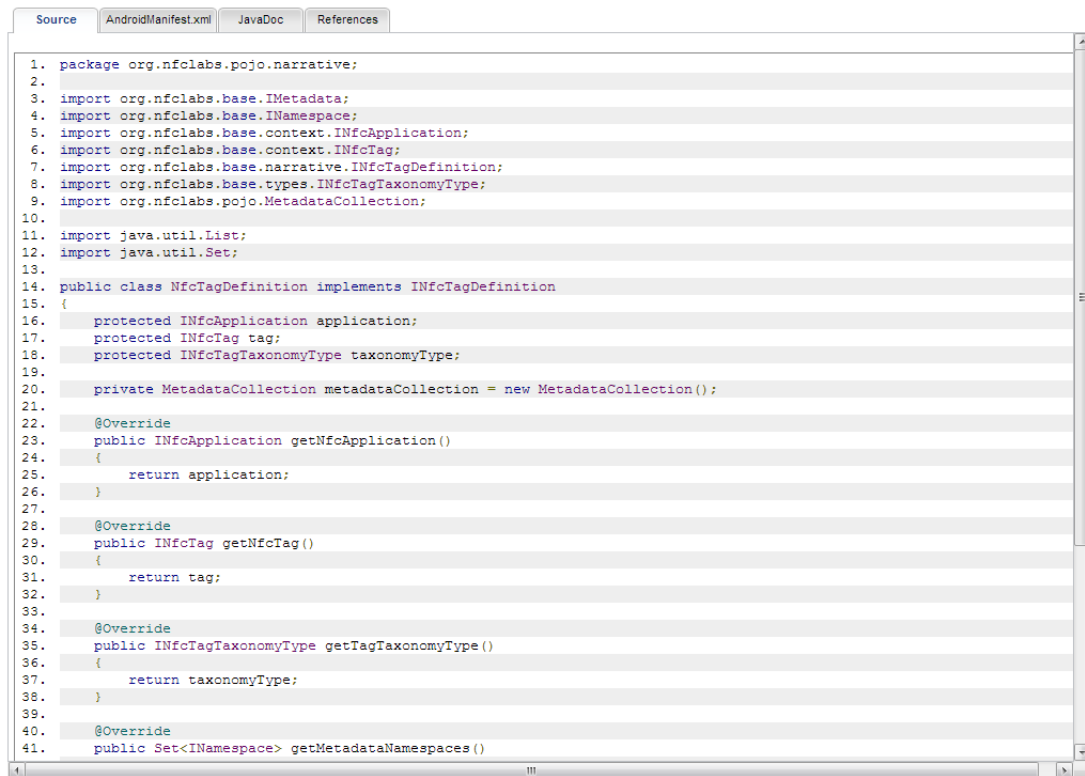
Εικόνα 38: Πηγαίος κώδικας component

Εάν ο χρήστης πατήσει την επιλογή πίσω στον περιηγητή, η ιστοσελίδα δεν ανταποκρίνεται σωστά και δεν μεταβαίνει στην λίστα με τα αποτελέσματα.

Το φαινόμενο της δύσκολης πλοήγησης στην ιστοσελίδα δεν βοήθησε στο να βρεθεί κάποιο έτοιμο component για την πτυχιακή εργασία. Μετά από αναζητήσεις με λέξεις κλειδιά όπως “nfc tag”, “ndef message”, “ndef record”, βρέθηκαν κάποια component, κλάσεις java, τα οποία υλοποιούν κάποια interfaces και τα περισσότερα είτε είχαν μόνο τις δηλώσεις μεθόδων είτε ελλιπή κώδικα (μόνο το return της μεθόδου). Κάποια παραδείγματα φέρονται στις παρακάτω εικόνες.



Εικόνα 39: Android Component



The screenshot shows an IDE window with tabs for 'Source', 'AndroidManifest.xml', 'JavaDoc', and 'References'. The 'Source' tab is active, displaying the following Java code:

```
1. package org.nfcclabs.pojo.narrative;
2.
3. import org.nfcclabs.base.IMetadata;
4. import org.nfcclabs.base.INamespace;
5. import org.nfcclabs.base.context.INfcApplication;
6. import org.nfcclabs.base.context.INfcTag;
7. import org.nfcclabs.base.narrative.INfcTagDefinition;
8. import org.nfcclabs.base.types.INfcTagTaxonomyType;
9. import org.nfcclabs.pojo.MetadataCollection;
10.
11. import java.util.List;
12. import java.util.Set;
13.
14. public class NfcTagDefinition implements INfcTagDefinition
15. {
16.     protected INfcApplication application;
17.     protected INfcTag tag;
18.     protected INfcTagTaxonomyType taxonomyType;
19.
20.     private MetadataCollection metadataCollection = new MetadataCollection();
21.
22.     @Override
23.     public INfcApplication getNfcApplication()
24.     {
25.         return application;
26.     }
27.
28.     @Override
29.     public INfcTag getNfcTag()
30.     {
31.         return tag;
32.     }
33.
34.     @Override
35.     public INfcTagTaxonomyType getTagTaxonomyType()
36.     {
37.         return taxonomyType;
38.     }
39.
40.     @Override
41.     public Set<INamespace> getMetadataNamespaces()
```

Εικόνα 40: Android Component



## 5 Συμπεράσματα

Συμπερασματικά, η τεχνολογία NFC είναι μια καινοτομία η οποία αναπτύσσεται τα τελευταία χρόνια σε συνδυασμό με την είσοδο των έξυπνων κινητών τηλεφώνων στην αγορά όπως και επίσης με τη ραγδαία εξέλιξη του λειτουργικού συστήματος για smartphones το Android. Για αυτό το λόγο επιλέξαμε και το Android SDK για την ανάπτυξη της εφαρμογής διότι υποστηρίζει την τεχνολογία NFC.

Με την ολοκλήρωση της παρούσας πτυχιακής εργασίας έγινε κατανοητή η τεχνολογία NFC και πως μπορεί να συμβάλει στην αυτοματοποίηση κάποιων εργασιών σε ένα εκπαιδευτικό ίδρυμα. Είδαμε πως το λειτουργικό σύστημα Android και στο SDK που προσφέρει δίνει τη δυνατότητα να αναπτύξει κανείς εφαρμογές που κάνουν χρήση της τεχνολογίας NFC μέσω ενός «έξυπνου» κινητού τηλεφώνου (smartphone).



# Βιβλιογραφία

- [1] [www.nfc-forum.org](http://www.nfc-forum.org)
- [2] [www.google.com](http://www.google.com)
- [3] Near Field Communication (NFC): From Theory to Practice -Vedat Coskun | Kerem Ok | Busra Ozdenizci , Εκδόσεις Wiley
- [4] NFC Forum Type Tags White Paper V1.0, 1 April 2009- PUBLIC
- [5] NFC Data Exchange Format (NDEF), NFC Forum NDEF 1.0
- [6] NFC Record Type Definition (RTD), NFC Forum
- [7] NFC URI Record Type Definition, NFC Forum
- [8] Programming Android, O'REILLY
- [9] [www.android.com](http://www.android.com)

# Εικόνες

- [1] Εικόνα 1: Αρχιτεκτονική RFID συστήματος: Near Field Communication (NFC): From Theory to Practice
- [2] Εικόνα 2: RFID tag:  
[http://www.greenerpackage.com/sites/default/files/RFID\\_Tags.jpg](http://www.greenerpackage.com/sites/default/files/RFID_Tags.jpg)
- [3] Εικόνα 3: Λογότυπο NFC: <http://img.talkandroid.com/uploads/2013/03/NFC-logo-forum.jpg>
- [4] Εικόνα 4: Λειτουργία Android Beam για peer-to-peer επικοινωνία:  
<http://www.android.com/images/mr1/beam.png>
- [5] Εικόνα 6:  
[http://developer.nokia.com/Community/Wiki/Understanding\\_NFC\\_Data\\_Exchange\\_Format\\_\(NDEF\)\\_messages](http://developer.nokia.com/Community/Wiki/Understanding_NFC_Data_Exchange_Format_(NDEF)_messages)
- [6] Εικόνα 7: Δομή Contactless Smartcard:  
<http://www.idwholesaler.com/images/learning-center/contactless-card.gif>

- [7] Εικόνα 8: NFC Reader: <http://www.nfcworld.com/wp-content/uploads/2013/06/nfc-porter.jpg>, <http://imgusr.tradekey.com/p-B2221329-20081027063323/acr122-nfc-contactless-smart-card-reader.gif>
- [8] Εικόνα 9: Λογότυπο Android: [http://www.realmdigital.co.za/images/uploaded\\_images/android-logo.png](http://www.realmdigital.co.za/images/uploaded_images/android-logo.png)
- [9] Εικόνα 10: Αρχιτεκτονική Android: <http://ist-music.berlios.de/site/images/music-android-architecture.jpg>
- [10] Εικόνα 11: Κύκλος Ζωής Activity: <http://www.android-app-market.com/wp-content/uploads/2012/03/Android-Activity-Lifecycle.png>
- [11] Εικόνα 36: Αναζήτηση component: [www.merodroid.com](http://www.merodroid.com)
- [12] Εικόνα 37: Αναζήτηση component: [www.merodroid.com](http://www.merodroid.com)
- [13] Εικόνα 38: Πηγαίος κώδικας component: [www.merodroid.com](http://www.merodroid.com)
- [14] Εικόνα 39: Android Component: [www.merodroid.com](http://www.merodroid.com)
- [15] Εικόνα 40: Android Component: [www.merodroid.com](http://www.merodroid.com)







