



**ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ**  
**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ**  
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**Πτυχιακή εργασία**

Η χρήση των QR Codes σε εφαρμογές AugmentedReality  
για κινητά τηλέφωνα και υλοποίηση ενδεικτικής  
εφαρμογής στα πλαίσια του τμήματος Πληροφορικής του  
ΤΕΙ

Φοιτητής

Κλιάφας Θεόδωρος

Αρ. Μητρώου: 06/3113

Επιβλέπων Καθηγητής

Κουρουπέτρογλου Χρήστος

**Θεσσαλονίκη 2012**

## Περίληψη

Στην παρούσα πτυχιακή εργασία αναπτύχθηκε μια εφαρμογή κινητού τηλεφώνου με χρήση της πλατφόρμας Android από την Google.

Στην εφαρμογή QResence δίνεται η δυνατότητα στον χρήστη να αποκρυσταλλώσει το QR Code μιας συγκεκριμένης αίθουσας της σχολής, να ενημερωθεί για τα μαθήματα που πρόκειται να γίνουν και να τοποθετήσει την παρουσία του. Η εφαρμογή δημιουργήθηκε ώστε να λειτουργεί με τον πιο απλό τρόπο χωρίς ο χρήστης να χρειάζεται να κάνει πολύπλοκες ενέργειες και χωρίς την βοήθεια εγχειριδίου.

Ο κύριος στόχος της πτυχιακής ήταν η ανάπτυξη μια εφαρμογής Augmented Reality που θα έχει πρόσβαση στο διαδίκτυο και θα είναι βασισμένη στην πλατφόρμα Android. Αυτό επιτεύχθηκε με την χρήση των QR Codes που η χρήση τους πολλαπλασιάζεται τα τελευταία χρόνια ανά τον κόσμο. Η χωρητικότητα τους σε δεδομένα, η ταχύτητα αποκωδικοποίησής τους και η απλότητα της χρήσης τους είναι κάποια από τα πλεονεκτήματα που τα χαρακτηρίζουν. Η πλατφόρμα Android επίσης έχει συναρπάσει τα τελευταία χρόνια με τις δυνατότητες της και το χαμηλό κόστος της. Ο συνδυασμός τους για την δημιουργία μιας διαδικτυακής εφαρμογής γίνεται με σκοπό την λειτουργικότητα και την ευκολία που μπορούν να προσθέσουν σε αυτό το έργο. Η χρήση αυτών των τεχνολογιών απαιτεί την γνώση συγκεκριμένων πεδίων και για την υλοποίηση αυτής της εφαρμογής χρησιμοποιήθηκαν διαφορετικές γλώσσες προγραμματισμού.

## Ευρετήριο Περιεχομένων

Περίληψη.....	2
Εισαγωγή.....	7
<b>1. Πλατφόρμα Android.....</b>	<b>9</b>
1.1 Εισαγωγή.....	9
1.2 Τί είναι το Android.....	10
1.3 Αρχιτεκτονική του Android.....	11
1.4 Τί είναι το Android;.....	15
1.5 Επίλογος.....	16
<b>2. QR Codes.....</b>	<b>17</b>
2.1 Εισαγωγή.....	17
2.2 Τί είναι το QR Code;.....	18
2.3 Χαρακτηριστικά των QR Codes.....	19
2.4 Χρήση των QR Codes.....	21
2.5 Παραδείγματα επιτυχημένης χρήσης QR Code.....	22
2.6 Λανθασμένη χρήση των QR Codes.....	25
2.7 Επίλογος.....	26
<b>3. Ανάλυση της εφαρμογής .....</b>	<b>27</b>
3.1 Εισαγωγή.....	27
3.2 Επικοινωνία της συσκευής με τον σέρβερ και με τη βάση δεδομένων.....	28
3.3 Βάση δεδομένων.....	30
3.4 Καταγραφή απαιτήσεων και σχεδιασμός της εφαρμογής και του site.....	32
3.4.1 Υλοποίηση QR Code Scanner με χρήση της βιβλιοθήκης ZXing.....	32
3.4.2 Σύνδεση με τον σέρβερ για έλεγχο των Login δεδομένων.....	34

3.4.3 Αναζήτηση μαθημάτων σύμφωνα με την μέρα, την ώρα και την αίθουσα.....	36
3.4.4 Ηλεκτρονική τοποθέτηση παρουσίας στη βάση δεδομένων.....	38
3.4.5 Δημιουργία ιστοσελίδας για την εμφάνιση των παρουσιών.....	38
3.5 Εργαλεία και τεχνολογίες που χρησιμοποιήθηκαν.....	40
3.6 Επίλογος.....	43
<b>4. Προγραμματισμός της εφαρμογής .....</b>	<b>44</b>
4.1 Εισαγωγή.....	44
4.2 Παρουσίαση των τριών φορμών της εφαρμογής και ανάλυση των πιο σημαντικών γραμμών κώδικα.....	45
4.2.1 1 <sup>η</sup> φόρμα: Εκκίνηση εφαρμογής.....	46
4.2.2 2 <sup>η</sup> φόρμα: Εισαγωγή στοιχείων Login.....	50
4.2.3 3 <sup>η</sup> φόρμα: Αναζήτηση, εμφάνιση μαθημάτων και τοποθέτηση Παρουσίας.....	52
4.3 Εμφάνιση και χειρισμός των στοιχείων της φόρμας.....	70
4.4 Τί είναι το Android manifest;.....	72
4.5 Επίλογος.....	74
<b>5. Προγραμματισμός του site Παρουσίες .....</b>	<b>75</b>
5.1 Εισαγωγή.....	75
5.2 Φόρμα εισαγωγής προσωπικών στοιχείων και έλεγχός τους.....	76
5.3 Εμφάνιση παρουσιών για κάθε μάθημα.....	79
5.4 Έξοδος απο το σύστημα.....	84
5.5 Επίλογος.....	85
<b>6. Συμπεράσματα .....</b>	<b>86</b>
<b>7. Βιβλιογραφία – Αναφορές .....</b>	<b>90</b>

## Ευρετήριο Εικόνων

Εικόνα 1.1: Λογότυπο της πλατφόρμας Android.....	9
Εικόνα 1.2: Τα επίπεδα του λειτουργικού συστήματος Android.....	11
Εικόνα 1.3: Πετυχημένες εφαρμογές στο Android.....	14
Εικόνα 2.1: Παράδειγμα γραμμωτού κώδικα.....	17
Εικόνα 2.2: QR Code, κώδικας δύο διαστάσεων.....	19
Εικόνα 2.3: QR Code format.....	21
Εικόνα 2.4: Διαφημιστική καμπάνια της εταιρίας Tesco στη Ν. Κορέα.....	22
Εικόνα 2.5: Εφαρμογή αγοράς προϊόντων Starbucks.....	23
Εικόνα 2.6: Spotify playlist κωδικοποιημένη σε QR Code.....	24
Εικόνα 3.1: Επικοινωνία συσκευής, σέρβερ και βάσης δεδομένων.....	29
Εικόνα 3.2: Σχεδιάγραμμα βάσης δεδομένων.....	31
Εικόνα 3.3: QR Code Scanner.....	33
Εικόνα 3.4: Φόρμα καταγραφής Login στοιχείων.....	34
Εικόνα 3.5: Εμφάνιση των μαθημάτων για την ενημέρωση του φοιτητή.....	37
Εικόνα 3.6: Λογότυπα απο τις γλώσσες προγραμματισμού που Χρησιμοποιήθηκαν.....	41
Εικόνα 4.1: 1 <sup>η</sup> φόρμα της εφαρμογής QResence.....	46
Εικόνα 4.2: Αδυναμία αποκωδικοποίησης QR Code.....	49
Εικόνα 4.3: Εισαγωγή Username και Password.....	50
Εικόνα 4.4: Φόρμα τοποθέτησης παρουσίας.....	52
Εικόνα 4.5: Εμφάνιση μηνύματος αποτυχίας σύνδεσης στο ίντερνετ.....	54
Εικόνα 4.6: Εμφάνιση μηνύματος αποτυχίας Login.....	56
Εικόνα 4.7: Εμφάνιση μηνύματος για μή αποδεκτές ώρες και μέρες.....	66
Εικόνα 4.8: Εμφάνιση μηνύματος επιτυχούς εισαγωγής παρουσίας.....	70

Πτυχιακή εργασία του Θεόδωρου Κλιάφα

Εικόνα 5.1: Φόρμα εισαγωγής στοιχείων ταυτοποίησης χρήστη.....	76
Εικόνα 5.2: Εμφάνιση μηνύματος λόγω παράληψης του πεδίου password.....	78
Εικόνα 5.3: Εμφάνιση στοιχείων του χρήστη και μαθημάτων που παρακολουθεί.....	79
Εικόνα 5.4: Εμφάνιση πίνακα παρουσιών σύμφωνα με το μάθημα ‘Προγραμματισμός II’ .....	83

# Εισαγωγή

Η χρήση των κινητών τηλεφώνων άλλαξε δραματικά την καθημερινότητα του ανθρώπου. Τα τελευταία χρόνια η ύπαρξη των εφαρμογών στις διάφορες πλατφόρμες έχει βοηθήσει σημαντικά στην επίλυση καθημερινών προβλημάτων με εύκολο και γρήγορο τρόπο. Η ανάπτυξη εφαρμογών για την πλατφόρμα Android είναι κάτι που έχει κερδίσει μεγάλο ποσοστό της βιομηχανίας καθώς πλέον αναγνωρίζεται σαν επάγγελμα. Πάρα πολλές εφαρμογές με λειτουργικό, εκπαιδευτικό ή διασκεδαστικό σκοπό έχουν έρθει στην επιφάνεια με εκατομύρια χρηστών να τις χρησιμοποιούν. Πολλές από αυτές κάνουν χρήση των QR Codes καθώς η λειτουργικότητα τους οδήγησε στην άμεση εκμετάλλευσή τους με διαφορετικά αποτελέσματα.

Ο λόγος που επέλεξα αυτήν την πτυχιακή είναι το ενδιαφέρον που είχε προκύψει για τις συσκευές Android και για την εκμάθηση της δημιουργίας εφαρμογών σε αυτή τη πλατφόρμα. Η καινοτομία των QR Codes και οι διάφοροι τρόποι χρήσης τους στην κοινωνία ήταν επίσης κατασταλτικός παράγοντας για την ενασχόλησή μου με αυτό το έργο. Δύο τεχνολογίες που είναι στη κορυφή της δημοτικότητας δεν μπορούν παρά να έχουν ένα επιτυχημένο αποτέλεσμα.

Για την εκπόνηση αυτής της πτυχιακής απαιτείται γνώση διαφόρων γλωσσών προγραμματισμού όπως Java, PHP και JSON. Σκοπός ήταν η χρήση αυτών των γλωσσών προγραμματισμού με τον πιο αποδοτικό και αποτελεσματικό τρόπο για την δημιουργία μιας λειτουργικής εφαρμογής που μπορεί να την χειριστεί οποιοσδήποτε. Στα 5 κεφάλαια που ακολουθούν περιγράφεται το υπόβαθρο των τεχνολογιών που χρησιμοποιήθηκαν, ο σχεδιασμός της εφαρμογής και ο προγραμματισμός των πιο σημαντικών κομματιών της. Στο πρώτο κεφάλαιο ο αναγνώστης μπορεί να ενημερωθεί για την βασική αρχιτεκτονική της πλατφόρμας Android, τα χαρακτηριστικά της και την ιδιαιτερότητα των εφαρμογών της. Στο επόμενο κεφάλαιο γίνεται ανάλυση των QR Codes εξηγώντας πως αποθηκεύεται

η πληροφορία στον γραμμωτό κώδικα. Επίσης αναφέρονται παραδείγματα επιτυχημένης και αποτυχημένης χρήσης τους. Στο τρίτο κεφάλαιο γίνεται η ανάλυση των απαιτήσεων για την δημιουργία μιας λειτουργικής εφαρμογής, ο διαχωρισμός των υποέργων της και η ανάλυση της διαδικτυακής της ιδιότητας της. Στο τέταρτο κεφάλαιο εξηγούνται τα σημαντικότερα κομμάτια κώδικα τόσο στην εφαρμογή όσο και στον σέρβερ για να γίνει κατανοητό το πως κινείται η πληροφορία μέχρι την οθόνη του χρήστη. Στο τελευταίο κεφάλαιο γίνεται ανάλυση του σχεδιασμού μιας βοηθητικής για την εφαρμογή ιστοσελίδας που συμπληρώνει την επιθυμητή λειτουργία της. Η επεξήγηση γίνεται σε απλό επίπεδο αλλά απαιτείται γνώση βασικών θεμάτων προγραμματισμού.





Εικόνα 1.1: Λογότυπο της πλατφόρμας Android

## **Κεφάλαιο 1: Πλατφόρμα Android**

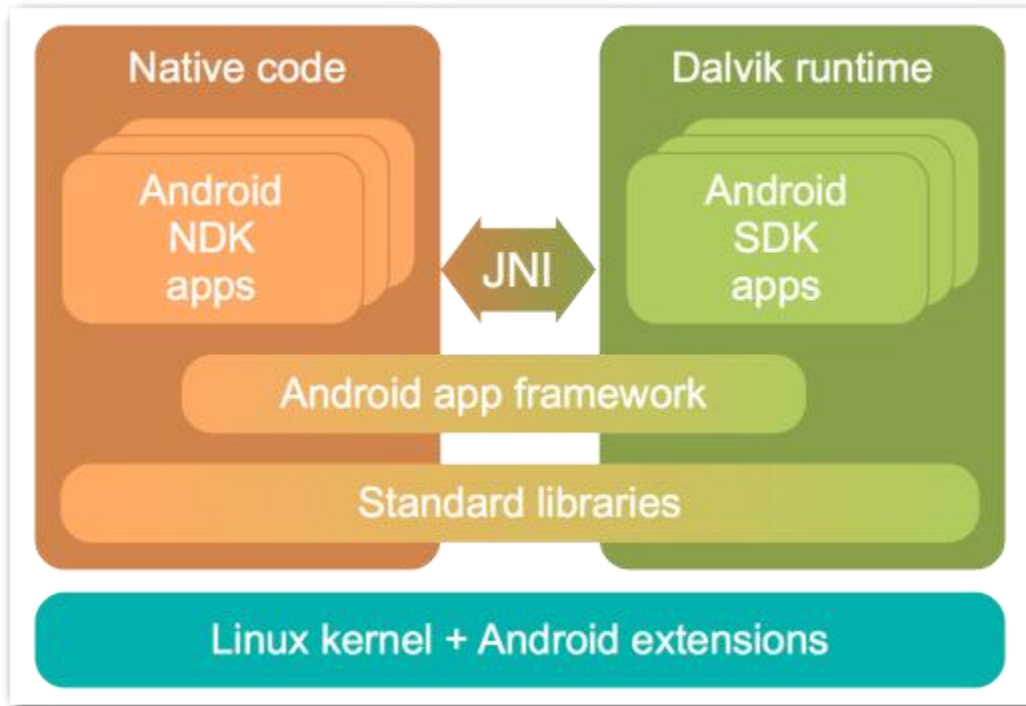
### **1.1 Εισαγωγή**

Στα πλαίσια αυτής της πτυχιακής εργασίας, αναπτύχθηκε μια εφαρμογή τοποθέτησης παρουσιών σε πλατφόρμα Android. Παρακάτω θα δούμε τα βασικά γνωρίσματα αυτής της πλατφόρμας και θα κατανοήσουμε τις δυνατότητες της.

## 1.2 Τι είναι το Android;

Το Android είναι ένα λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας και ταμπλέτες βασισμένο σε πυρήνα Linux. Δημιουργήθηκε αρχικά από την Android Inc και το 2005 αγοράστηκε από τη Google όπου σε συνεργασία με την OpenHandsetAlliance έγινε η ανάπτυξη της πλατφόρμας. Η πρώτη παρουσίαση έγινε τον Νοέμβρη του 2007 ταυτόχρονα με την ανακοίνωση ίδρυσης του OHA, μιας κοινοπραξίας 48 εταιριών που ασχολούνται με την ανάπτυξη λογισμικού και την κατασκευή υλικού με στόχο την εξέλιξη των προτύπων τηλεπικοινωνίας στη κινητή τηλεφωνία. Έγινε εμπορικά διαθέσιμο το φθινόπωρο του 2008 με τη συσκευή HTC Dream, Linux kernel version 2.6 με πληκτρολόγιο που δεν υποστήριζε την οθόνη αφής παρα μόνο τη χρήση σκληρών κουμπιών της συσκευής. Στη συνέχεια ακολούθησαν και άλλες συσκευές με διαφορετικές δυνατότητες με αποκορύφωμα το Nexus one κατασκευασμένο από την HTC σε συνεργασία με την Google που κατείχε καινοτόμα χαρακτηριστικά σε σχέση με τους πρόγονους του.

### 1.3 Αρχιτεκτονική του Android



Εικόνα 1.2: Τα επίπεδα του λειτουργικού συστήματος Android

Το Android τρέχει σε πυρήνα Linux και το λειτουργικό του σύστημα στα ανώτερα στρώματα αποτελείται από βιβλιοθήκες, το πλαίσιο εφαρμογής και απλές εφαρμογές γραμμένες από τους κατασκευαστές λογισμικού. Είναι μια στοίβα λογισμικού στην οποία κάθε επίπεδο διαθέτει συγκεκριμένες υπηρεσίες για το επόμενο επίπεδο, ξεκινώντας από τη διαχείριση των διεργασιών στον πυρήνα σε συνεργασία με τις εγγενείς βιβλιοθήκες λογισμικού και τις εφαρμογές έως το οπτικό αποτέλεσμα στην οθόνη της συσκευής. Πιο αναλυτικά:

### **Πυρήνας Linux:**

Για την ακρίβεια είναι μια απλουστευμένη έκδοση του Linux στην οποία υπολείπονται βασικά κομμάτια του κορμού. Στις ολοκληρωμένες εκδόσεις του Linux το σύστημα περιέχει δικές του βιβλιοθήκες, shell (κέλυφος) και editors με δικό τους γραφικό περιβάλλον κάτι που δε θα βρούμε αυτούσια στο Android. Επομένως ο πυρήνας ενός Android θα χαρακτηρίζοταν ως μια γυμνή εκδοχή ενός συστήματος Linux που σκοπό έχει να χρησιμοποιήσει τα πλεονεκτήματα που του παρέχει. Το βασικότερο είναι η ελευθερία που παρέχει στους προγραμματιστές ως open source δίνοντας τους κίνητρο να πειραματιστούν με αυτό, που έχει ως αποτέλεσμα τη συνεχή ανάπτυξη του συστήματος. Χαρακτηρίζεται για την αξιοπιστία του και την ασφάλεια που διαθέτει στις βελτιωμένες νέες εκδόσεις της πλατφόρμας. Το Android του αναθέτει την διαχείριση της μνήμης, την διαχείριση των διεργασιών και άλλες πρωταρχικές λειτουργίες.

### **Dalvik runtime:**

Σε αυτό το επίπεδο συναντάμε βασικές βιβλιοθήκες της Java και την εικονική μηχανή Dalvik. Η Dalvik είναι η μηχανή διεργασιών που τρέχει τις εφαρμογές στις συσκευές με Android. Οι εφαρμογές συνήθως γράφονται σε java ή C/C++ και μεταγλωτίζονται σε μορφή bytecode. Η εικονική μηχανή μετατρέπει τα αρχεία .class σε αρχεία .dex (DalvikExecutableformat) που είναι ειδικά διαμορφωμένα για συσκευές με περιορισμένη μνήμη και επεξεργαστική ισχύ. Για αυτό τον λόγο όλα τα APIs που χρησιμοποιούν οι εφαρμογές είναι ορισμένα σε κλάσεις, αντικείμενα και μεθόδους της μηχανής Dalvik.

### **Native libraries:**

Αυτό το επίπεδο επιτρέπει στη συσκευή να χειρίζεται διαφορετικούς τύπους δεδομένων. Οι βιβλιοθήκες είναι γραμμένες σε γλώσσα C/C++ και έχουν δημιουργηθεί από τους κατασκευαστές για κάποια συγκεκριμένη συσκευή. Μερικές από αυτές είναι:

- SQLite: Είναι η μηχανή αποθήκευσης δεδομένων που χρησιμοποιεί το Android.
- WebKit: Είναι η μηχανή πλοήγησης που χρησιμοποιείται για την απεικόνιση html εγγράφων.
- OpenGL: Είναι η μηχανή που απεικονίζει τα δισδιάστατα και τρισδιάστατα γραφικά στην οθόνη της συσκευής.

### **Πλαίσιο εφαρμογής:**

Αυτό είναι το επίπεδο με το οποίο οι εφαρμογές έχουν απευθείας αλληλεπίδραση. Εδώ εκτελούνται οι βασικές λειτουργίες της συσκευής τις οποίες μπορεί να χρησιμοποιήσει ένας προγραμματιστής για την εφαρμογή του.

Τα σημαντικότερα στοιχεία αυτού του επιπέδου είναι:

- Activity Manager – Διαχειριστής Δραστηριοτήτων: Είναι υπεύθυνο για τον έλεγχο του χρόνου ζωής των εφαρμογών.
- Content Providers – Διαχειριστής Περιεχομένου: Είναι υπεύθυνο για τις πληροφορίες που ανταλλάσσουν μεταξύ τους οι εφαρμογές.
- Location Manager – Διαχειριστής Τοποθεσίας: Βρίσκει την τοποθεσία χρησιμοποιώντας το GPS ή το δίκτυο.
- Resource Manager – Διαχειριστής Πόρων: Είναι υπεύθυνο για τα δεδομένα που τρέχουν στις εφαρμογές και δεν είναι κώδικας. Για παράδειγμα μπορεί να είναι σχεδιαγράμματα XML ή κωδικοί χρωμάτων.



Εικόνα 1.3: Πετυχημένες εφαρμογές στο Android

### Εφαρμογές:

Στο τελευταίο επίπεδο βρίσκονται οι εφαρμογές που έχουν δημιουργηθεί για απευθείας αλληλεπίδραση με τον χρήστη. Είναι προγράμματα που είτε έχουν εγκατασταθεί από τους κατασκευαστές λογισμικού είτε έχουν αναπτυχθεί από προγραμματιστές για να διευκολύνουν την καθημερινότητα των χρηστών. Είναι συνήθως γραμμένες σε γλώσσα Java από προγραμματιστές που χρησιμοποιούν το Android Software Development Kit ή σε γλώσσα C/C++ με χρήση του Native Development Kit. Οι προεγκατεστημένες εφαρμογές όπως το ημερολόγιο, οι επαφές και το Gmail συνεργάζονται άμεσα με την εικονική μηχανή Dalvik. Οι περισσότερες είναι opensource εκτός από αυτές που διατίθενται απευθείας από τη Google, όπως το Google Maps και το PlayStore. Από το GooglePlay υπάρχει η δυνατότητα αναζήτησης εφαρμογών εκ των οποίων οι περισσότερες διατίθενται

δωρεάν. Επίσης κάποιος μπορεί να βρεί εφαρμογές απο άλλα site χωρίς όμως να φέρουν τον έλεγχο της Google. Τα αρχεία που κατεβάζει ο χρήστης είναι σε μορφή .apk που στην πραγματικότητα είναι ένας συμπιεσμένος .zip φάκελος και επιτρέπουν την εγκατάσταση τους χωρίς να αποσυμπιεστούν.

## **1.4 Γιατί Android;**

Το Android είναι μια μοναδική πλατφόρμα η οποία δίνει την ελευθερία στους προγραμματιστές εφαρμογών να εκμεταλλευτούν πλήρως τις δυνατότητες μιας συσκευής. Παρέχει μια μεγάλη γκάμα απο βιβλιοθήκες για την ανάπτυξη λογισμικού και διαθεσιμότητα πολλών εργαλείων που αυξάνει τη παραγωγικότητα και φαντασία των προγραμματιστών για την δημιουργία μιας πρωτότυπης εφαρμογής. Για παράδειγμα, είναι θέμα απλών εντολών για έναν πεπειραμένο προγραμματιστή να συνδέσει λειτουργίες κάμερας, GPS, φωνητικών εντολών και ίντερνετ σε μια εφαρμογή. Το γεγονός ότι ο πυρήνας του είναι opensource μας διαβεβαιώνει επίσης ότι η ανάπτυξη του θα συνεχίσει και θα υπάρχει συνεχής εξέλιξη καθώς πολλοί προγραμματιστές ασχολούνται με τη δημιουργία και συντήρηση του ελεύθερου λογισμικού.

Απο την πλευρά του χρήστη, μια συσκευή Android είναι σε μεγάλο βαθμό παραμετροποιήσιμη. Στην οθόνη της συσκευής είναι εύκολο κάποιος να προσθέσει ή να αφαιρέσει εφαρμογές και widgets ανάλογα τη συχνότητα που τα χρησιμοποιεί. Αυτό βέβαια δε σημαίνει ότι οι εφαρμογές που δεν βρίσκονται στην οθόνη δεν θα είναι κανονικά εγκατεστημένες. Οι εφαρμογές που κάποιος χρήστης επιθυμεί να εγκαταστήσει μπορούν να ενσωματώνουν λειτουργίες απο τις άλλες βασικές εφαρμογές της συσκευής και να ανταλλάσσουν δεδομένα μεταξύ τους. Οι συσκευές με λογισμικό Android έχουν την δυνατότητα να τρέχουν

ταυτόχρονα διαφορετικές εφαρμογές στο background. Αυτή η ιδιότητα τους ονομάζεται multitasking και είναι πολύ σημαντικό εργαλείο για τον χρήστη καθώς μπορεί με τη χρήση ενός κουμπιού να δει και να επιλέξει την εφαρμογή που θέλει να εκτελέσει. Η Google εκτός από κατασκευάστρια του λογισμικού παρέχει και σημαντική υποστήριξη στις υπηρεσίες της. Τα Android συνεργάζονται καλύτερα από όλες τις άλλες πλατφόρμες με σημαντικές εφαρμογές όπως το Gmail, το GoogleCalendar, το GoogleDocs, το GoogleMaps, τη μηχανή αναζήτησης της Google και το YouTube. Τέλος, συσκευές διαφορετικών δυνατοτήτων έχουν δημιουργηθεί με σκοπό να συνεργαστούν με αυτό το λογισμικό. Το Android δημιουργήθηκε για να λειτουργεί σε συσκευές χαμηλού κόστους με μειωμένες δυνατότητες αλλά και σε συσκευές υψηλού κόστους όπου η πλατφόρμα αναδεικνύει τις δυνατότητες του software (ταχύτητα εκτέλεσης διεργασιών) αλλά και του hardware (ταχύτητα επεξεργασίας).

## 1.5 Επίλογος

Σε αυτό το κεφάλαιο εξηγήσαμε τί είναι το Android, πως εξελίχθηκε ανα τα χρόνια και από ποιά στρώματα αποτελείται. Επίσης είδαμε την χρησιμότητα των εφαρμογών σε συνδυασμό με τα πλεονεκτήματα που παρέχει η πλατφόρμα, τόσο για τους προγραμματιστές όσο και για τους χρήστες.





Εικόνα 2.1: Παράδειγμα γραμμωτού κώδικα

## Κεφάλαιο 2: QR Codes

### 2.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει η ανάλυση του γραμμωτού κώδικα (QR Code) και θα γίνει κατανοητή η σημασία του στις τεχνολογίες του σήμερα. Επίσης θα δούμε σωστές και λανθασμένες πρακτικές απο σύγχρονα παραδείγματα χρήσης τους.

## 2.2 Τί είναι το QR Code;

Το QR Code είναι ένας γραμμωτός κώδικας δύο διαστάσεων που πρωτοσχεδιάστηκε το 1994 στην Ιαπωνία από την Denso Wave για την αυτοκινητοβιομηχανία της Toyota. Το QR προέρχεται από τα αρχικά QuickResponse καθώς έχει δημιουργηθεί από τους σχεδιαστές του για γρήγορη αποκωδικοποίηση δεδομένων. Χρησιμοποιήθηκε αρχικά στη γραμμή παραγωγής για την παρακολούθηση ανταλλακτικών και στη συνέχεια οι εταιρίες είδαν ότι μπορούν να το χρησιμοποιήσουν και με άλλες εφαρμογές. Σήμερα η χρήση του είναι εμφανής στις έξυπνες συσκευές με ιδιαίτερο ενδιαφέρον από διαφορετικούς κλάδους όπως αυτός της διαφήμισης, του εμπορίου ή και ακόμα της διασκέδασης.

Οι γραμμωτοί κώδικες (barcodes) φάνηκαν χρήσιμοι τις προηγούμενες δεκαετίες στην κωδικοποίηση πληροφορίας. Λόγω της διάδοσης τους η βιομηχανία έψαχνε να βρεί ένα νέο τρόπο κωδικοποίησης που να ενσωματώνει περισσότερη πληροφορία σε μικρότερο χώρο. Πειραματικά δημιουργήθηκαν διάφορες ενοποιήσεις των barcode με πιο επιτυχημένο αποτέλεσμα αυτό των QR Code. Αυτό το μοτίβο φάνηκε να κυριαρχεί για το μέγεθος της πληροφορίας που αυξήθηκε συγκριτικά με το εμβαδό που καταλαμβάνει και είναι σχεδόν ίδιο ή και μικρότερο από αυτό των barcodes. Για την ακρίβεια το QR Code είναι ένας συνδιασμός δισδιάστατου barcode που περιέχει πληροφορίες στην κάθετη και στην οριζόντια κατεύθυνση.



Εικόνα 2.2: QR Code, κώδικας δύο διαστάσεων

Το barcode είχε σχεδιαστεί για να σκανάρεται από μία δέσμη φωτός. Περιέχει μικρότερη πληροφορία και σκανάρεται πιο αργά από ένα QR Code. Το δεύτερο αναγνωρίζεται ως μια δισδιάστατη εικόνα από ένα αισθητήρα εικόνων και στη συνέχεια αναλύεται ψηφιακά από συγκεκριμένο επεξεργαστή. Ο επεξεργαστής εντοπίζει τα τετράγωνα στις τρεις γωνίες του QR Code και χρησιμοποιεί ένα μικρότερο τετράγωνο στη τέταρτη γωνία για να επεξεργαστεί το μέγεθος της εικόνας, το προσανατολισμό της και τη γωνία θέασης. Στη συνέχεια οι κουκίδες στην εικόνα αποκρυπτογραφούνται σε δυαδικούς αριθμούς ενώ υπάρχει και δυνατότητα εντοπισμού λάθους μέσω ειδικού αλγορίθμου.

### 2.3 Χαρακτηριστικά QRCode

Τα QR Codes μπορούν να χειριστούν δεδομένα οποιoδήποτε τύπου. Βέβαια το μέγεθος πληροφορίας που μπορεί να αποθηκευτεί σε ένα QR Code εξαρτάται από τον τύπο των δεδομένων, τις διαστάσεις και τον αλγόριθμο διόρθωσης λαθών. Πιο συγκεκριμένα:

## QR code Data Capacity

Numeric only Max. 7,089 characters

Alphanumeric Max. 4296 characters

Binary (8bits) Max. 2,953 characters

Kanji, full-width Kana Max. 1,817 characters

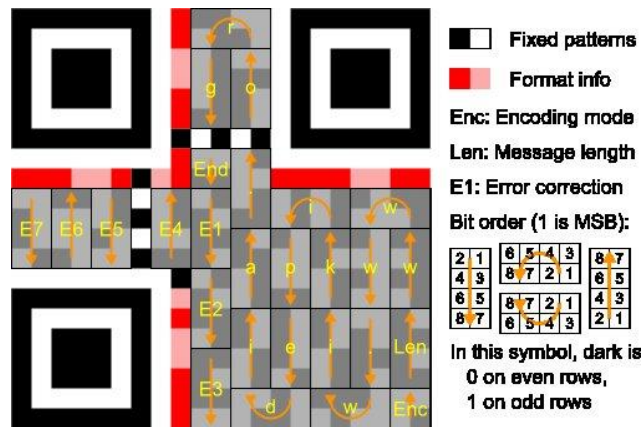
Σχήμα 2.1: Τύποι δεδομένων στον γραμμωτό κώδικα

### Διόρθωση λαθών

Τα QR Codes διαθέτουν έναν αλγόριθμο διόρθωσης λαθών (Reed-Solomon) που κυμαίνεται σε 4 επίπεδα. Με χρήση διόρθωσης λαθών υψηλότερου επιπέδου μειώνουμε άμεσα τη δυνατότητα αποθήκευσης περισσότερων δεδομένων στον κώδικα μας. Με αφορμή το ενδεχόμενο ότι ένα QR Code μπορεί να αποκρυπτογραφηθεί ενώ περιέχει κάποια λάθη εμφάνισης, πολλοί πρόσθεσαν λογότυπα και χρώματα στο μοτίβο για να τα κάνουν πιο ελκυστικά στους χρήστες.

### Κωδικοποίηση

Τα δεδομένα τοποθετούνται από τα δεξιά προς τα αριστερά στις περιοχές όπου δεν υπάρχουν τα τρία τετράγωνα. Τα τρία τετράγωνα παίζουν σημαντικό ρόλο καθώς δίνουν πληροφορίες σχετικά με τον κώδικα, δηλαδή το επίπεδο διόρθωσης λαθών αλλά και το πως διαχωρίζονται τα δεδομένα στο κεντρικό κομμάτι. Το μορφότυπο (format), περιμετρικά των τετραγώνων, έχει δικό του αλγόριθμο διόρθωσης λάθους και περιέχει ένα επιπλέον αντίγραφο του μέσα στον γραμμωτό κώδικα για κάθε ενδεχόμενο.



Εικόνα 2.3: QR Code format

## 2.4 Χρήση των QR Codes

Τα τελευταία χρόνια η χρήση των QR Codes έχει αυξηθεί καθώς έχουν μπει στην καθημερινότητα μας. Θα τα παρατηρήσουμε σε περιοδικά, τοίχους, επαγγελματικές κάρτες και σε πολλά άλλα σημεία στον κόσμο. Η ευκολία που παρέχουν στον να συνδέουν τους ανθρώπους μεταξύ τους με multimedia τρόπο είναι πολύ σημαντική τόσο για τις επιχειρήσεις όσο και για τους καταναλωτές. Είναι ένα νέο βήμα στον τομέα του μάρκετινγκ – διαφήμισης παρέχοντας μόνο πλεονεκτήματα για αυτούς που τα χρησιμοποιούν. Η χρήση τους δεν γίνεται αποκλειστικά για εμπορικούς σκοπούς. Ο καθένας μπορεί να κωδικοποιήσει ένα σύνδεσμο ο οποίος περιέχει ένα βίντεο, ένα άρθρο, μια συνταγή μαγειρέματος ή ακόμη και το μενού κάποιου εστιατορίου. Η ιδιότητα τους να δίνουν αλάνθαστα αποτελέσματα με γρήγορη και εύκολη ανάγνωση βοήθησε στην επέκτασή τους εκτός της Ιαπωνίας όπου και δημιουργήθηκαν. Έχουν ιδιαίτερη απήχηση στο κοινό καθώς δε χρειάζονται κάποια ιδιαίτερη εμπειρία και γνώση για τη χρήση τους και το βασικότερο είναι ότι η δημιουργία τους είναι δωρεάν. Το μόνο που χρειάζεται κάποιος είναι να εγκαταστήσει μια εφαρμογή στη συσκευή του και να είναι συνδεδεμένος στο ίντερνετ.

## 2.5 Παραδείγματα επιτυχημένης χρήσης QR Code

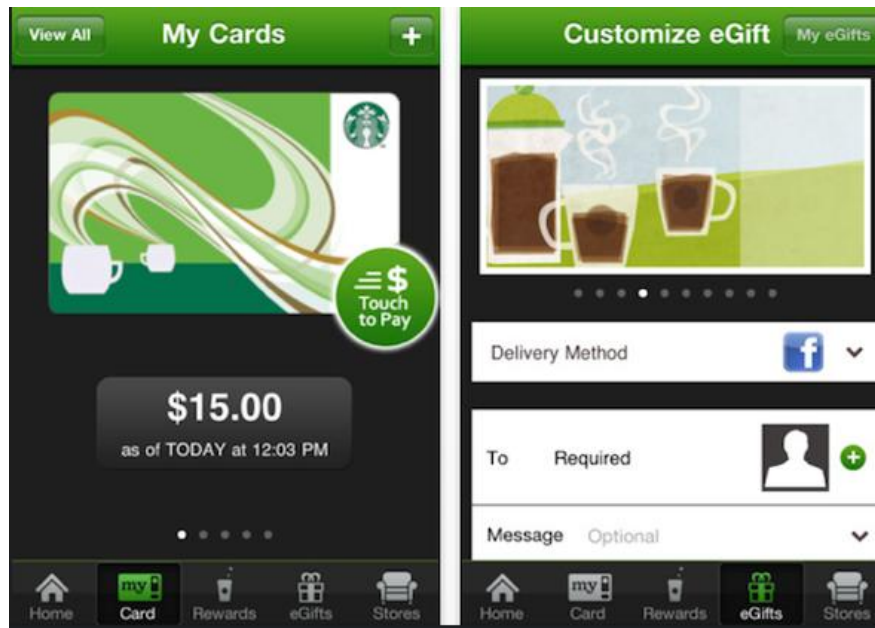
Το παράρτημα της διεθνούς εταιρίας Tesco στη Νότιο Κορέα με σκοπό την αύξηση των πωλήσεων του ηλεκτρονικού της καταστήματος (HomePlus) εκμεταλλεύτηκε τη παρουσία των QR Codes και διευκόλυνε την ζωή των καταναλωτών της. Πιο συγκεκριμένα, η εταιρία πρόσθεσε εικόνες των προϊόντων της συνοδευόμενα από QR Codes στις στάσεις του μετρό ώστε να ενημερώνονται οι καταναλωτές κατά τη χρονική διάρκεια που περιμένουν τον συρμό. Οι εικόνες προσομοιάζουν τα πραγματικά ράφια ενός καταστήματος με είδη καθημερινής χρήσης. Έτσι ο καταναλωτής μπορεί να σκανάρει τα QR Codes των προϊόντων με το Smartphone του, να ενημερωθεί με μία σύντομη περιγραφή και να τα προσθέσει στο 'καλάθι' του για αγορά. Η εταιρία στη συνέχεια αποστέλλει τα προϊόντα απευθείας στο σπίτι του καταναλωτή μέσα σε λίγες ώρες από τη στιγμή της παραγγελίας. Αυτή η ιδέα φάνηκε πολύ χρήσιμη στους εργαζόμενους με περιορισμένο ελεύθερο χρόνο βοηθώντας τους να εξοικονομήσουν πολύτιμες ώρες από τη μέρα τους. Επίσης αύξησε τις online πωλήσεις κατά 130% μέσα σε λίγους μήνες βοηθώντας την Tesco να κατακτήσει τη πρώτη θέση στον ανταγωνισμό των ηλεκτρονικών πωλήσεων.



Εικόνα 2.4: Διαφημιστική καμπάνια της εταιρίας Tesco στη Ν. Κορέα

## Πτυχιακή εργασία του Θεόδωρου Κλιάφα

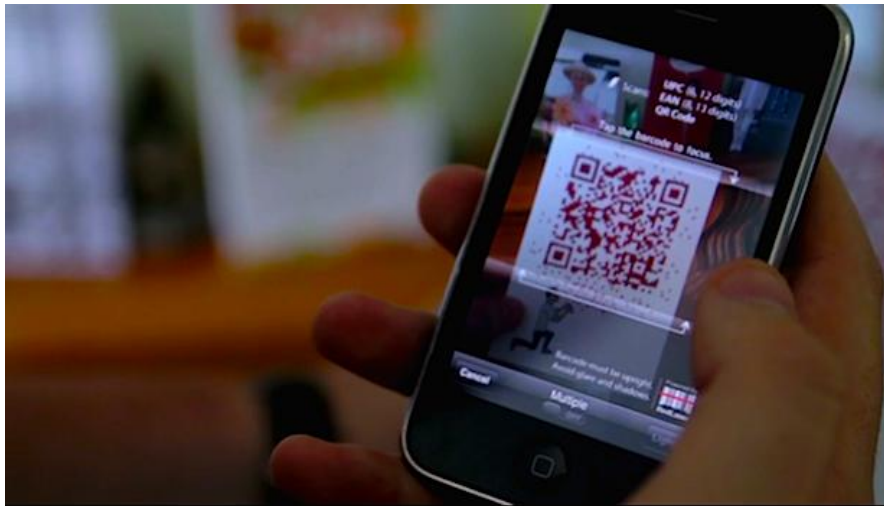
Με παρόμοιο σκεπτικό, άλλα καταστήματα όπως τα Starbucks, χρησιμοποιούν QR Codes για την γρήγορη πληρωμή των προϊόντων τους αποφεύγοντας έτσι τις μεγάλες ουρές στα ταμεία. Το μόνο που χρειάζεται να κάνουν οι πελάτες τους, είναι να ενσωματώσουν την προπληρωμένη κάρτα τους στην εφαρμογή στο κινητό τους που σκανάρει τα QR Codes. Με αυτό τον τρόπο εκτός του ότι θα πληρώνουν πιο γρήγορα, θα ενημερώνονται για τα προϊόντα και τα διάφορα υποκαταστήματα. Σε κάποιες περιπτώσεις μάλιστα, θα πετύχετε και QR Code πάνω στη συσκευασία του προϊόντος η οποία θα παραπέμπει σε πληροφορίες σχετικά με τα συστατικά του, τον τρόπο παραγωγής του και τους τρόπους παραγγελίας του.



Εικόνα 2.5: Εφαρμογή αγοράς προϊόντων Starbucks

Εκτός των υπηρεσιών πληρωμής, διάφορες επιχειρήσεις πρόσθεσαν τα QR Codes για να αυξήσουν την δημοφιλή τους. MountainDew και TacoBell (αλυσίδες fast-food) συνεργάζονται, τοποθετώντας QR Codes στα κύπελλα των αναψυκτικών που πουλάνε. Όταν ο χρήστης σκανάρει το QR Code του δίνεται η δυνατότητα να κατεβάσει ένα από τα δημοφιλή τραγούδια δωρεάν. Όπως καταλαβαίνετε αυτές οι εταιρίες έψαξαν καλά το ηλικιακό κοινό τους και κατάφεραν με αυτόν τον τρόπο να διαφημίσουν το όνομα τους ανέξοδα.

Κάποιοι άλλοι θέλοντας να πρωτοτυπήσουν τη περίοδο των χριστουγέννων έδωσαν την εναλλακτική στους πελάτες τους να τυπώσουν ένα QR Code στη συσκευασία ενός δώρου, το λεγόμενο “SantaTag”. Αυτός που έκανε την αγορά του προϊόντος είχε τη δυνατότητα να σκανάρει το SantaTag και να ηχογραφήσει ένα φωνητικό μήνυμα για τον παραλήπτη. Ο δεύτερος με τη παραλαβή του δώρου, σκανάρωντας το QR Code μπορούσε να ακούσει το μήνυμα σαν ευχετήρια κάρτα.



Εικόνα 2.6: Spotifyplaylist κωδικοποιημένη σε QR Code



Κάτι αντίστοιχο διέθεσε στην αγορά και η εταιρία Spotify. Έχοντας δημιουργήσει λογαριασμό στο μουσικό site υπάρχει η δυνατότητα επιλογής τραγουδιών για τη δημιουργία λίστας (playlist). Στη συνέχεια ο χρήστης κωδικοποιεί τη λίστα σε QR Code και την αποστέλλει σε κάποια επαφή του σε μορφή κάρτας.

## 2.6 Λανθασμένη χρήση των QR Codes

Η αυξημένη χρήση των QR Codes σε άλλες χώρες οδήγησε σε τακτικές αποτυχημένες και χωρίς λειτουργικότητα. Συχνά οι χρήστες ανταλλάσσουν QR Codes χωρίς να σκεφτούν αν το περιεχόμενο της σελίδας που στέλνουν είναι φιλικό προς εμφάνιση σε κινητές συσκευές. Σε κάποιες περιπτώσεις, λόγω ενθουσιασμού της χρήσης νέων τεχνολογιών, αποστέλλουν τον γραμμωτό κώδικα σε μορφή mail ενώ κάλλιστα θα μπορούσαν να στείλουν τον σύνδεσμο (URL) της σελίδας χωρίς να χρειάζεται η χρήση της κάμερας του κινητού για σκανάρισμα και εμφάνιση της.

Στην άλλη πλευρά, οι εταιρίες μάρκετινγκ που έχουν τοποθετήσει σε πολλούς χώρους διαφημίσεις με γραμμωτούς κώδικες χωρίς να υπάρχει δυνατότητα σύνδεσης στο ίντερνετ. Τέτοια παραδείγματα θα βρούμε μέσα σε αεροπλάνα και σταθμούς του μετρό όπου η σύνδεση στο διαδίκτυο είναι συνήθως αδύνατη. Επίσης είναι πολλές οι περιπτώσεις όπου τα QR Codes είναι τοποθετημένα σε μέρη που δεν είναι δυνατό να τα σκανάρει ο χρήστης. Σε κάποιους τοίχους κτιρίων τα QR Codes τοποθετούνται σε ψηλά σημεία που είναι δύσκολο για τη συσκευή να εστιάσει και να τα αποκρυπτογραφήσει. Ο γραμμωτός κώδικας για να αποκρυπτογραφηθεί, εκτός από τις διαστάσεις χρειάζεται και μια ελάχιστη χρονική διάρκεια εμφάνισης ώστε η συσκευή να μπορέσει να εστιάσει σε αυτό. Επομένως κινούμενα QR Codes τυπωμένα σε λεωφορεία και στις πλευρές των αυτοκινήτων παρουσιάζουν δυσλειτουργικότητα αποκρυπτογράφησης. Τέλος, κάθε QR Code πρέπει να συνοδεύεται από μια μικρή περιγραφή ώστε ο χρήστης να κρίνει αν τον ενδιαφέρει η πληροφορία που κρύβεται από πίσω.

## 2.7 Επίλογος

Η χρήση των QR Codes έχει διευκολύνει πολύ τη ζωή των ανθρώπων. Η αυξημένη χρησιμότητα τους οδήγησε στην εισαγωγή τους σε διάφορες επιχειρήσεις με διαφορετικά αποτελέσματα. Απο τα παραπάνω παραδείγματα εύκολα κάποιος μπορεί να καταλάβει ότι ο γραμμωτός κώδικας θα φέρει επανάσταση στον κόσμο της multimedia επικοινωνίας. Ο συνδυασμός των τεχνολογιών του Android και του QR Code υπόσχεται πολλές χρήσιμες και λειτουργικές εφαρμογές. Παρακάτω θα περιγράψουμε μια απο αυτές.

## Κεφάλαιο 3: Ανάλυση της εφαρμογής

### 3.1 Εισαγωγή

Η εφαρμογή QResence είναι μια ιδέα που προέκυψε από την ανάγκη τοποθέτησης παρουσιών στα εργαστηριακά μαθήματα του τμήματος μας. Δημιουργήθηκε στα πλαίσια της σχολής με σκοπό να βοηθήσει στο πολυσυζητημένο αυτό θέμα.

Είναι συνήθες το φαινόμενο της παράληψης τοποθέτησης παρουσιών τόσο από τους φοιτητές όσο και από τους καθηγητές στη διάρκεια ενός μαθήματος. Με αυτή την εφαρμογή, δίνετε η δυνατότητα στους κατόχους έξυπνων κινητών να σκανάρουν τον γραμμωτό κώδικα έξω από κάθε αίθουσα και να τοποθετούν την παρουσία τους στο μάθημα που επιθυμούν. Το σκεπτικό είναι απλό. Θα τοποθετήσουμε έξω από κάθε εργαστήριο ένα χαρακτηριστικό QR Code του οποίου η αποκρυπτογράφηση θα δίνει πληροφορίες που σχετίζονται με τα μαθήματα που γίνονται σε εκείνη την αίθουσα. Ο φοιτητής θα μπορεί να σκανάρει με τη συσκευή του το QR Code μιας συγκεκριμένης αίθουσας και θα ενημερώνεται για τα μαθήματα που γίνονται σε ένα πλαίσιο χρόνου που δε ξεπερνά τις τέσσερις ώρες. Στη συνέχεια θα επιλέγει ένα από τα δύο πιθανά μαθήματα που εμφανίζονται στην οθόνη της συσκευής του και θα τοποθετεί την παρουσία του σε ηλεκτρονική μορφή. Για να ενημερωθεί σχετικά με τις παραδόσεις, δημιουργήθηκε το site 'Παρουσίες' στο οποίο μπορεί να δει σε ποιά μαθήματα ήταν παρών και ποιές μέρες και ώρες τοποθέτησε τις παρουσίες του.

Η εφαρμογή είναι ευέλικτη στα πλαίσια του χρόνου. Όποτε μια αίθουσα δύνατε να φιλοξενήσει δύο συνεχόμενα μαθήματα σε διάστημα τεσσάρων ωρών, σε συγκεκριμένα χρονικά 'παράθυρα' θα εμφανίζονται και τα δύο μαθήματα ώστε να επιλέξει ο φοιτητής σε ποιο θέλει να τοποθετήσει παρουσία. Με λίγα λόγια δίνετε

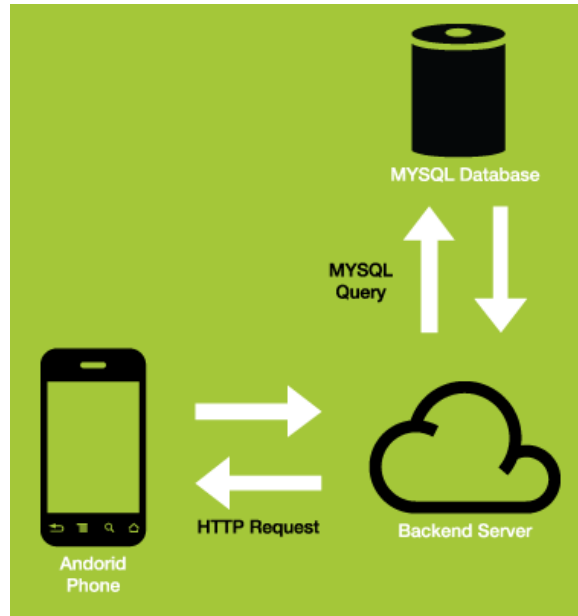
ένα χρονικό περιθώριο 30 λεπτών όπου ο φοιτητής θα ενημερώνετε για το μάθημα που γίνετε ή που θα γίνει την επόμενη ώρα.

Για παράδειγμα ας δούμε το σενάριο μιας αίθουσας στην οποία διδάσκονται 2 μαθήματα. Το πρώτο ξεκινάει στις 10 το πρωί και τελειώνει στις 12 όπου αρχίζει το δεύτερο μάθημα το οποίο τελειώνει στις 2 το μεσημέρι. Ο φοιτητής απο τις 10 έως και τις 11:44 όταν σκανάρει το QR Code θα του εμφανίζεται μόνο το πρώτο μάθημα. Απο τις 11:45 έως και τις 12:15 θα του εμφανίζονται και τα δύο μαθήματα με σκοπό να επιλέξει σε ποιό θέλει να βάλει παρουσία ή για να ενημερωθεί για το επόμενο μάθημα το οποίο θα διδαχθεί στην αίθουσα. Απο τις 12:16 και μετά θα εμφανίζεται αποκλειστικά το δεύτερο μάθημα.

Με τα τωρινά δεδομένα, είναι πολλές οι φορές στις οποίες οι φοιτητές φεύγουν απο την τάξη στο τέλος του μαθήματος χωρίς να έχουν τοποθετήσει παρουσία. Με το QResence τους δίνεται η ευκαιρία με το που θυμηθούν 15 λεπτά μετά το τέλος του μαθήματος να επιστρέψουν και να τοποθετήσουν παρουσία ηλεκτρονικά μέσω της εφαρμογής. Αυτό θα δώσει λύση στην απροσεξία καθηγητών και φοιτητών στο θέμα του παρουσιολογίου.

### **3.2 Επικοινωνία της συσκευής με τον σέρβερ και με τη βάση δεδομένων**

Η εφαρμογή QResence είναι μια διαδικτυακή υπηρεσία η οποία για να λειτουργήσει σε μια συσκευή απαιτεί σύνδεση στο ίντερνετ. Συνεργάζεται απόλυτα με τον σέρβερ του TEI (erodios) ο οποίος εκτελεί ερωτήματα στη βάση δεδομένων που είναι αποθηκευμένη σε αυτόν.



Εικόνα3.1: Επικοινωνία συσκευής, σέρβερ και βάσης δεδομένων

Για τη λήψη αποτελεσμάτων στην οθόνη του κινητού εκτελούνται κάποια συγκεκριμένα βήματα στην εφαρμογή. Δεδομένου ότι η συσκευή έχει εγκαταστήσει σύνδεση με το διαδίκτυο, με την ενεργοποίηση κάποιας λειτουργίας της εκτελούνται εντολές επικοινωνίας με τον σέρβερ. Πιο συγκεκριμένα, η εφαρμογή καλεί μια διεύθυνση URL η οποία οδηγεί σε ένα script (πρόγραμμα) αποθηκευμένο στον σέρβερ. Τη στιγμή που καλείτε το script ξεκινάει η εκτέλεση του. Οι πρώτες γραμμές σε αυτό το πρόγραμμα επιδιώκουν τη δημιουργία σύνδεσης του σέρβερ με τη βάση δεδομένων που χρησιμοποιούμε. Αφού καθιερωθεί αυτή η σύνδεση, οι επόμενες εντολές διατυπώνουν συγκεκριμένα ερωτήματα στους πίνακες της βάσης και στη συνέχεια το πρόγραμμα λαμβάνει ένα πλήθος δεδομένων. Τα αποτελέσματα κάτω από επεξεργασία αλλάζουν μορφή ώστε να είναι συμβατά με την εφαρμογή. Στις τελευταίες γραμμές του προγράμματος βρίσκονται εντολές που αποστέλλουν τα μορφοποιημένα δεδομένα στην εφαρμογή η οποία τα εμφανίζει στην οθόνη της συσκευής. Από την στιγμή που ο χρήστης θα ενεργοποιήσει μια λειτουργία που απαιτεί

επικοινωνία με τον σέρβερ, τα αποτελέσματα θα εμφανιστούν σε χρόνο που καθορίζεται απο τη σύνδεση στο ίντερνετ, το ποσοστό χρήσης του σέρβερ και τις δυνατότητες του υλικού της συσκευής.

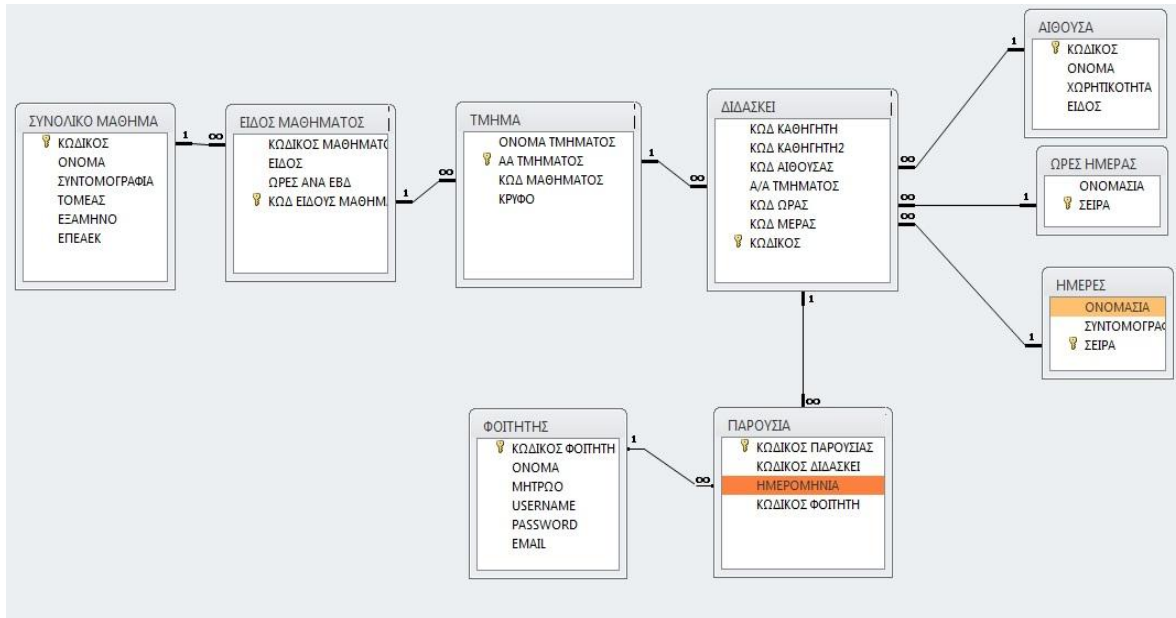
### **3.3 Βάση δεδομένων**

Για να δημιουργήσουμε τη βάση δεδομένων που χρησιμοποιούμε στην εφαρμογή μας βασιστήκαμε στην ήδη υπάρχουσα βάση που δημιουργήθηκε στη πτυχιακή του Γαβριηλίδη Σέργιου «Υλοποίηση διαδικτυακής εφαρμογής διαχείρισης του προγράμματος μαθημάτων του τμήματος Πληροφορικής» με επιβλέποντα τον κύριο Κεραμόπουλο Ευκλείδη. Αυτή η βάση δεδομένων αναπτύχθηκε σε τεχνολογία IBM DB2 και περιλαμβάνει ένα πλήθος πινάκων με στοιχεία και πληροφορίες που αφορούν την Σχολή Πληροφορικής. Πρόθεση μας ήταν να εργαστούμε πάνω σε αυτή τη βάση δεδομένων αλλά για λόγους ασφάλειας δεδομένων της σχολής μας δόθηκε ένα αντίγραφο του σχήματος της.

Ο σέρβερ που φιλοξενεί τη βάση του QResence είναι ο erodios και η εφαρμογή για τη διαχείριση της είναι το phpMyAdmin.

Παρατηρώντας τη βάση δεδομένων που μας δόθηκε, καταλήξαμε στο συμπέρασμα ότι δε θα χρειαστούμε αρκετούς απο τους πίνακες που αναφέρονται σε δεδομένα μη χρήσιμα για την εφαρμογή μας. Αντιγράψαμε τον σκελετό εφτά πινάκων με τις συνδέσεις τους και προσθέσαμε ακόμη δύο πίνακες.

## Πτυχιακή εργασία του Θεόδωρου Κλιάφα



Εικόνα 3.2: Σχεδιάγραμμα βάσης δεδομένων

Στο σχεδιάγραμμα οι πίνακες ‘Συνολικό μάθημα’, ‘Είδος μαθήματος’, ‘Τμήμα’, ‘Διδάσκει’, ‘Αίθουσα’, ‘Ωρες ημέρας’ και ‘Ημέρες’ προυπήρχαν από τη προηγούμενη βάση δεδομένων και οι πίνακες ‘Παρουσία’ και ‘Φοιτητής’ προστέθηκαν ώστε να χρησιμοποιηθούν για την εφαρμογή μας.

Στους πίνακες ‘Συνολικό μάθημα’ και ‘Είδος μαθήματος’ χρησιμοποιούμε πληροφορίες σχετικά με τα ονόματα και τους κωδικούς των μαθημάτων. Ο πίνακας ‘Διδάσκει’ λειτουργεί ουσιαστικά σαν μια σύζευξη μεταξύ των πινάκων ‘Τμήμα’, ‘Αίθουσα’, ‘Ωρες ημέρας’, ‘Ημέρες’ και ‘Παρουσία’. Μέσα από αυτή τη σύζευξη των πινάκων διατυπώνονται συνδιαστικά ερωτήματα από τα οποία προκύπτουν πληροφορίες σχετικά με τα τμήματα των μαθημάτων, τις αίθουσες, τις ώρες και τις μέρες που διδάσκονται. Στον πίνακα ‘Παρουσία’ καταγράφεται η παρουσία με τον κωδικό του φοιτητή, την ημερομηνία που τοποθετήθηκε και τον κωδικό ‘διδάσκει’ που παραπέμπει σε πληροφορίες σχετικά με το μάθημα. Στον πίνακα ‘Φοιτητής’ καταγράφονται πληροφορίες σχετικά με τον φοιτητή με πιο σημαντικά πεδία αυτά του ‘username’ και ‘password’ για την ταυτοποίηση των στοιχείων του.

### **3.4 Καταγραφή απαιτήσεων και σχεδιασμός της εφαρμογής και του site**

Για να φτάσουμε στο τελικό αποτέλεσμα της εφαρμογής προυπήρξαν αρκετά στάδια σχεδιασμού. Για να δημιουργήσουμε μια λειτουργική για τον χρήστη εφαρμογή έπρεπε να γίνει καταγραφή των βασικών απαιτήσεων. Έτσι, για να φτιάξουμε την εφαρμογή όσο πιο απλή γίνεται έγινε διαχωρισμός της σε 5 κομμάτια που στην συνέχεια συνδιάστηκαν για να καταλήξουμε στο τελικό αποτέλεσμα. Ο κεντρικός στόχος για την εφαρμογή είναι η γρήγορη χρήση της με μικρό αριθμό επιλογών και ελάχιστη τοποθέτηση δεδομένων από τον χρήστη.

#### **3.4.1 Υλοποίηση QR Code Scanner με χρήση της βιβλιοθήκης ZXing**

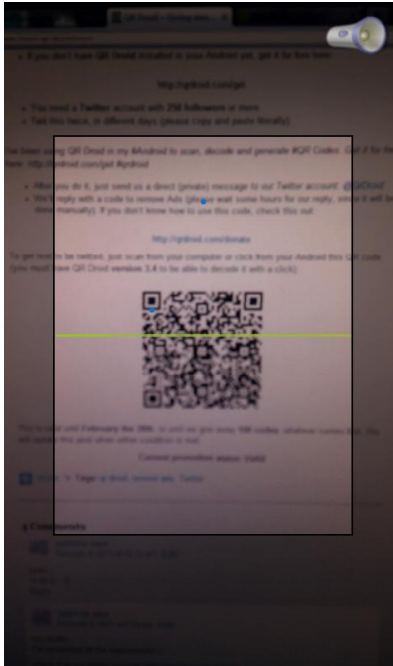
Ένα από τα βασικότερα κομμάτια στην εφαρμογή QResence είναι η χρήση ενός QR code Scanner για την αποκωδικοποίηση του γραμμωτού κώδικα έξω από κάθε αίθουσα. Με μια απλή αναζήτηση στο διαδίκτυο κάποιος μπορεί να βρεί πλήθος εφαρμογών που αποκωδικοποιούν τους γραμμωτούς κώδικες. Κάποιες από αυτές είναι το QuickMark, το QRDroid και το QRBARCODESCANNER. Στα πλαίσια της δικιάς μας εφαρμογής όμως είναι απαραίτητο μετά την αποκωδικοποίηση του QR Code να συνδιαστούν δεδομένα, να γίνουν αιτήματα στον σέρβερ και να εκτελεστούν ερωτήματα σε συγκεκριμένα πεδία στη βάση δεδομένων. Αρκετές από αυτές τις εφαρμογές δίνουν τη δυνατότητα συνεργασίας με άλλες εφαρμογές αποστέλλοντας πληροφορίες όπως τα αποκωδικοποιημένα δεδομένα ενός QR Code. Βασική προπόθεση όμως είναι να γίνει ξεχωριστή εγκατάσταση αυτών των εφαρμογών πέραν της εγκατάστασης της δικιάς μας εφαρμογής.

Έτσι, ήταν επιτακτική η ανάγκη δημιουργίας ενός QR Code Scanner από την αρχή. Με την έναρξη της εφαρμογής υπάρχει η δυνατότητα επιλογής ανάγνωσης ενός QR Code με το πάτημα ενός κουμπιού. Αυτό παραπέμπει στη χρήση της



## Πτυχιακή εργασία του Θεόδωρου Κλιάφα

κάμερας ώστε να αναγνωρισθεί το επιθυμητό QR Code και μετά απο επεξεργασία να επιστραφεί η αποκωδικοποιημένη πληροφορία για μετέπειτα χρήση.



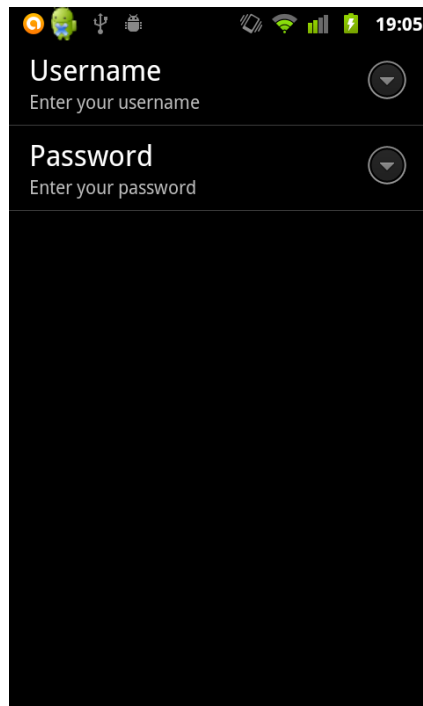
Εικόνα 3.3: QR Code Scanner

Ένα πολύ σημαντικό εργαλείο για την αποκωδικοποίηση του στιγμιότυπου είναι η βιβλιοθήκη ZXing η οποία είναι διαθέσιμη απο την Google. Με την εισαγωγή του .jar αρχείου μέσα στο project και την χρήση των κατάλληλων εντολών, ο καθένας έχει την δυνατότητα να χειριστεί αυτή την βιβλιοθήκη και τις λειτουργίες της. Συνοπτικά, μετά τη λήψη της φωτογραφίας η εικόνα μετατρέπεται σε δεδομένα φωτεινότητας και διαστάσεων για να είναι συμβατά με τη λειτουργία αποκρυπτογράφησης της βιβλιοθήκης και μετά απο επεξεργασία επιστρέφει το αποτέλεσμα για να το χειριστεί ο προγραμματιστής όπως εκείνος θέλει.

Μέσα απο αυτή τη διαδικασία κατανόησα καλύτερα τη λειτουργία της αποκωδικοποίησης του γραμμωτού κώδικα και ολοκληρώθηκε με επιτυχία το πρώτο κομμάτι υλοποίησης της εφαρμογής.

### 3.4.2 Σύνδεση με τον σέρβερ για έλεγχο των Login δεδομένων

Όπως κάθε διαδικτυακή εφαρμογή που χρησιμοποιείται για ιδιωτική χρήση, έτσι και στο QResence πρέπει να υπάρχει έλεγχος των προσωπικών στοιχείων για την ολοκληρωμένη λειτουργία του. Ο χρήστης εισάγει τα στοιχεία του σε μια φόρμα και μετά την αποκρυπτογράφηση του γραμμωτού κώδικα γίνεται ο έλεγχος τους σε σύνδεση με τον σέρβερ δεδομένου ότι η συσκευή είναι συνδεδεμένη στο διαδίκτυο. Σε περίπτωση που τα στοιχεία είναι λανθασμένα δεν είναι δυνατή η επιλογή μαθήματος για τοποθέτηση παρουσίας και γίνεται η εμφάνιση ενός σχετικού μηνύματος που παραπέμπει τον χρήστη στην αλλαγή των στοιχείων του.



Εικόνα 3.4: Φόρμα καταγραφής Login στοιχείων

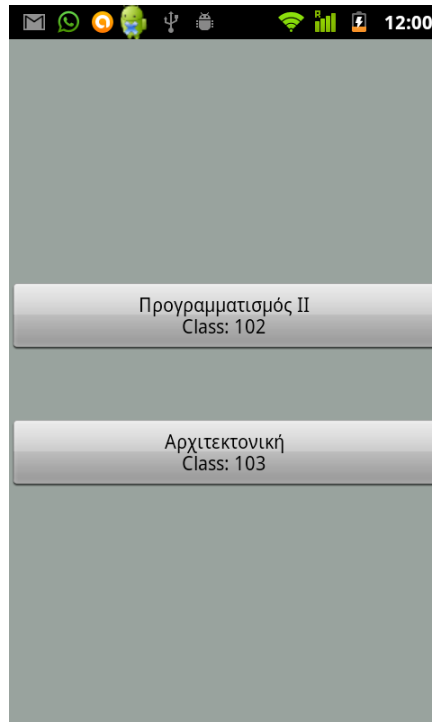
Σε αυτό το κομμάτι αρχικά έγιναν δοκιμές για τη σύνδεση της εφαρμογής με τον erodio και με τη βάση δεδομένων που φιλοξενεί. Η εφαρμογή καλεί συγκεκριμένα scripts που είναι αποθηκευμένα στον σέρβερ και εκτελούν συγκεκριμένες λειτουργίες. Για τον έλεγχο των στοιχείων username και password καλείτε ένα συγκεκριμένο πρόγραμμα το οποίο παραλαμβάνει τις παραπάνω πληροφορίες απο την εφαρμογή και εκτελεί ερωτήματα στη βάση δεδομένων. Στον πίνακα 'Φοιτητής' υπάρχουν τα πεδία username και password όπου και γίνεται η αναζήτηση. Αν βρεθούν τα στοιχεία που έδωσε ο χρήστης τότε η βάση επιστρέφει τα αποτελέσματα στο script και απο εκεί αποστέλλονται πίσω στην εφαρμογή. Εκεί γίνεται και δεύτερος έλεγχος ώστε να διαβεβαιωθεί οτι τα στοιχεία είναι τα σωστά. Αν τα στοιχεία που έδωσε ο χρήστης είναι έγκυρα, η εφαρμογή κυλάει ομαλά χωρίς να παρουσιάζεται κάποιο σχετικό μήνυμα. Αν βρεθεί κάποιο λάθος τότε εμφανίζεται ένα μήνυμα στην οθόνη του κινητού που ενημερώνει τον χρήστη ότι τα στοιχεία που δόθηκαν είναι λάθος και τον προτρέπει ώστε να ξαναγίνει η εισαγωγή τους σωστά.

Ακόμα και στην περίπτωση της χρήσης λανθασμένων στοιχείων αναγνώρισης, με τη χρήση της αποκωδικοποιημένης διεύθυνσης URL εμφανίζονται τα μαθήματα που διδάσκονται χωρίς να είναι δυνατή η τοποθέτηση παρουσίας. Έτσι ο φοιτητής μπορεί να ενημερωθεί σχετικά με το πρόγραμμα μαθημάτων της αίθουσας τη συγκεκριμένη ώρα.

### **3.4.3 Αναζήτηση μαθημάτων σύμφωνα με την μέρα, την ώρα και την αίθουσα**

Με την αποκωδικοποίηση ενός QR Code της σχολής, μια απο τις πληροφορίες που λαμβάνουμε είναι ο αριθμός της αίθουσας που ουσιαστικά είναι και η ταυτότητα της στην εφαρμογή. Αυτός ο αριθμός αποθηκεύεται προσωρινά και χρησιμοποιείται σε συνδιασμό με άλλα στοιχεία για να κληθεί στη συνέχεια ένας αριθμός προγραμμάτων αποθηκευμένα στον σέρβερ.

Δεδομένου ότι η αποκρυπτογράφηση έχει γίνει επιτυχώς, η εφαρμογή οδηγεί τον χρήστη στην επόμενη φόρμα όπου γίνεται η αναζήτηση των μαθημάτων. Μετά την μετάβαση αυτή, εκτελούνται αυτόματα κάποιες λειτουργίες. Πιο συγκεκριμένα, η πρώτη λειτουργία είναι ο έλεγχος της σύνδεσης στο διαδίκτυο. Σε περίπτωση που ο χρήστης έχει ξεχάσει να συνδεθεί είτε μέσω δικτύου τηλεφωνίας είτε μέσω ασύρματου δικτύου βγαίνει για δύο δευτερόλεπτα ένα προειδοποιητικό μήνυμα που υπενθυμίζει στον χρήστη να ενεργοποιήσει τη σύνδεση για λήψη δεδομένων. Αν η σύνδεση είναι επιτυχής δεν εμφανίζεται κάποιο μήνυμα και ο χρήστης μπορεί να συνεχίσει κανονικά τη χρήση της εφαρμογής. Στη συνέχεια υπάρχουν λειτουργίες αναγνώρισης της ώρας και της μέρας. Μια απο τις βιβλιοθήκες του Android SDK μπορεί να μας ενημερώσει για αυτά τα δεδομένα με χρήση των κατάλληλων εντολών. Το σύστημα επιστρέφει αυτά τα δύο στοιχεία και κάτω απο επεξεργασία μετατρέπονται στους επιθυμητούς τύπους για αναζήτηση στη βάση δεδομένων. Έχοντας αυτές τις τρεις πληροφορίες που έχουν μεταβληθεί σύμφωνα με τα στάνταρ της βάσης, πραγματοποιείται κλήση στον σέρβερ. Το script που καλείται, διαβάζει τον αριθμό της αίθουσας την ημέρα και την ώρα και ύστερα απο έλεγχο πραγματοποιεί ερώτημα στη βάση με τις παραπάνω μεταβλητές.



Εικόνα 3.5 Εμφάνιση των μαθημάτων για την ενημέρωση του φοιτητή

Η εφαρμογή για να μπορέσει να χρησιμοποιήσει αυτά τα αποτελέσματα πρέπει να τα λάβει σε συγκεκριμένη μορφή. Ως εκ τούτου, το πρόγραμμα στον σέρβερ λαμβάνει τα αποτελέσματα από τη βάση και με χρήση εντολών της PHP τα εμφανίζει ως ένα μορφοποιημένο πίνακα δεδομένων και τα αποστέλλει στην εφαρμογή. Εκεί γίνεται η παραλαβή αυτών των δεδομένων και έλεγχος για το πλήθος τους. Δηλαδή αν έχουν επιστραφεί ένα ή δύο μαθήματα το οποίο εξαρτάται από την ώρα που έγινε το ερώτημα, την αίθουσα και κατ'επέκταση το πρόγραμμα της σχολής. Αναλόγως του αριθμού των αποτελεσμάτων εμφανίζονται στην εφαρμογή ένα ή δύο μαθήματα σε μορφή κουμπιών (button). Ο χρήστης ενημερώνεται για τα μαθήματα που πραγματοποιούνται και έχει την επιλογή να τοποθετήσει παρουσία σε κάποιο από αυτά. Να σημειωθεί ότι η εφαρμογή επιστρέφει αποτελέσματα μεταξύ των ωρών 07:45 έως και 18:00. Πριν και μετά αυτού του χρονικού διαστήματος δεν είναι δυνατή η εμφάνιση δεδομένων.

#### **3.4.4 Ηλεκτρονική τοποθέτηση παρουσίας στη βάση δεδομένων**

Μετά την εμφάνιση των μαθημάτων ο χρήστης έχει τη δυνατότητα επιλογής τοποθέτησης της παρουσίας του σε ένα απο τα διαθέσιμα κουμπιά - μαθήματα στη οθόνη της συσκευής του. Με τη διαδικασία αφής του επιλεγμένου button στην οθόνη του κινητού, γίνεται πάλι έλεγχος για την ορθότητα του username και password όπως αναφέρθηκε προηγουμένως. Αν όλα έχουν κυλήσει ομαλά ως τώρα, χωρίς την εμφάνιση μηνυμάτων, με την ενεργοποίηση του button μαθήματος καλείται ένα άλλο script για την τοποθέτηση της παρουσίας. Στο πρόγραμμα αυτό μεταφέρονται το username του σπουδαστή και το τμήμα το οποίο επέλεξε να τοποθετήσει την παρουσία του. Με την σειρά του ο σέρβερ παραλαμβάνει αυτές τις πληροφορίες, διατυπώνει ερωτήματα στη βάση για την μετατροπή αυτών των στοιχείων σε συμβατά δεδομένα για τον πίνακα των παρουσιών, και με μία εντολή εισαγωγής δεδομένων προσθέτει στη βάση ακόμα ένα στοιχείο. Κατα την διάρκεια εισαγωγής της παρουσίας γίνεται αυτόματος έλεγχος στον σέρβερ για την ημερομηνία και την ώρα, δεδομένα που δε χρειάζεται να προσθέσουμε εμείς στην εντολή εισαγωγής δεδομένων. Η παρουσία με τον κωδικό (id) του φοιτητή, τον κωδικό του μαθήματος, την ημερομηνία και τον κωδικό της παρουσίας έχει καταχωρηθεί στο σύστημα.

#### **3.4.5 Δημιουργία ιστοσελίδας για την εμφάνιση των παρουσιών**

Με τη χρήση του QResence, ο φοιτητής πρέπει να έχει την δυνατότητα να ελέγχει τα μαθήματα που παρακολουθεί και να ενημερωθεί για τις ημέρες και τις ώρες που τοποθέτησε τις παρουσίες του σε αυτά. Στα πλαίσια αυτών των απαιτήσεων δημιουργήθηκε η ιστοσελίδα 'Παρουσίες' για την ενημέρωση των χρηστών της εφαρμογής πάνω σε αυτό το ζήτημα.

Η χρήση της είναι απλή. Με την εισαγωγή των ίδιων στοιχείων αναγνώρισης (username και password) που χρησιμοποιούνται στην εφαρμογή, ο φοιτητής έχει τη δυνατότητα να μεταβεί στη βασική φόρμα της ιστοσελίδας. Εκεί θα του εμφανιστούν στην αριστερή πλευρά της σελίδας τα στοιχεία του και τα μαθήματα τα οποία παρακολούθησε. Με την επιλογή ενός εκ των διαθέσιμων μαθημάτων, εμφανίζεται στο κέντρο της σελίδας ένας πίνακας με όλες τις παρουσίες σε αυτό το μάθημα με συγκεκριμένες ημέρες και ώρες που υποδηλώνουν την στιγμή που τοποθετήθηκαν ηλεκτρονικά. Όπως σε όλες τις ιστοσελίδες, υπάρχει και επιλογή Logout ώστε ο χρήστης να βγει απο το σύστημα αφότου έχει ενημερωθεί.

Για την ανάπτυξη αυτού του εγχειρίματος προγραμματίστηκαν τρία προγράμματα τα οποία και αποθηκεύτηκαν στον σέρβερ με σκοπό να τα καλεί ο χρήστης μέσω του περιηγητή του. Στο πρώτο προγραμματίστηκε η λειτουργία ελέγχου των προσωπικών στοιχείων του χρήστη. Το πρόγραμμα κάνει αναζήτηση στα πεδία username και password του πίνακα 'Φοιτητής' και συγκρίνει τα αποτελέσματα με τα στοιχεία που έδωσε ο χρήστης. Κάθε πιθανότητα εισαγωγής λανθασμένων στοιχείων οδηγεί σε ένα μήνυμα που παραπέμπει στη διόρθωσή του σημείου που δημιουργεί το πρόβλημα της μετάβασης στη κεντρική σελίδα. Εφόσον τα στοιχεία είναι σωστά καλείται το δεύτερο πρόγραμμα η λειτουργία του οποίου είναι να αναζητά στο πίνακα 'Παρουσίες' τα μαθήματα τα οποία παρακολούθησε ο συγκεκριμένος χρήστης. Με την επιλογή κάποιου μαθήματος το πρόγραμμα αυτό αναζητά στη βάση δεδομένων τις παρουσίες που τοποθετήθηκαν εμφανίζοντας τις μέρες και τις ώρες αυτών με χρονική ταξινόμηση. Τέλος με την επιλογή Logout, καλείται και το τρίτο πρόγραμμα το οποίο αποχαιρετά τον χρήστη με ένα μήνυμα και η βασική του λειτουργία είναι να σβήσει το id του χρήστη για τη παρούσα συνεδρία. Έτσι στον χρήστη δε του επιτρέπεται να ξαναεμφανίσει την κεντρική φόρμα παρα μόνο αν ξαναπροσπαθήσει να κάνει τον έλεγχο Login.

### **3.5 Εργαλεία και τεχνολογίες που χρησιμοποιήθηκαν**

Για τη δημιουργία της εφαρμογής έγινε η εγκατάσταση του προγράμματος Eclipse Indigo και του Android SDK. Το SDK δίνει τη δυνατότητα δίνει τη δυνατότητα στους προγραμματιστές να δημιουργήσουν ένα εξομοιωτή (emulator) ενός Android συστήματος μέσω του Android manager. Ένας emulator έχει κάποια κοινά χαρακτηριστικά με ένα simulator (προσομοιωτής) αλλά εκείνο που το κάνει πραγματικά να διαφέρει είναι ότι κατά το compilation του προγράμματος δημιουργείται μία πραγματική μηχανή με όλες τις λειτουργίες ενός τηλεφώνου. Δίνει πολλές επιλογές στον προγραμματιστή καθώς μπορεί να περιηγηθεί στο διαδίκτυο μέσω αυτού του εικονικού τηλεφώνου, να κάνει εγκατάσταση εφαρμογών απο το GooglePlay ή να δει πως λειτουργεί η εφαρμογή που δημιουργεί. Όλα αυτά γίνονται στην οθόνη του υπολογιστή του προγραμματιστή. Μια εικονική ‘συσσκευή’ που με το κλείσιμο του προγράμματος εγγραφής κώδικα συνεχίζει να υπάρχει. Το μόνο μειονέκτημα της είναι ότι δε μπορεί να γίνει χρήση κάμερας επομένως για την εφαρμογή μας πραγματοποιήθηκε η σύνδεση του Eclipse με μια Android συσκευή. Για να γίνει αυτή η σύνδεση πρέπει αρχικά να εγκατασταθούν οι κατάλληλοι οδηγό λογισμικού (drivers) για να αναγνωριστεί η συσκευή απο τα Windows, να δηλωθεί η σύνδεση στις επιλογές του Eclipse Indigo και να επιτραπεί στο λογισμικό του Android μέσω ρυθμίσεων debugging η εμφάνιση της εφαρμογής στην οθόνη του κινητού.





Εικόνα 3.5: Λογότυπα απο τις γλώσσες προγραμματισμού που χρησιμοποιήθηκαν

Με την ενεργοποίηση της εκκίνησης της εφαρμογής στο Eclipse εμφανίζεται ένα παράθυρο για να επιλέξει ο προγραμματιστής αν θέλει να γίνει εγκατάσταση της εφαρμογής του στον emulator ή στη συσκευή του. Το Eclipse δίνει επίσης τη δυνατότητα πολλών εργαλείων στον χρήστη, στο προγραμματιστικό κομμάτι μέχρι και την εμφάνιση της εφαρμογής στη οθόνη του κινητού. Για τη λειτουργικότητα της εφαρμογής χρησιμοποιήθηκε η γλώσσα προγραμματισμού Java, όπου το Eclipse διαθέτει έναν κειμενογράφο (editor) με διάφορες αυτόματες ιδιότητες φιλικές στον προγραμματιστή. Για την εμφάνιση της εφαρμογής υπάρχει γραφικό και προγραμματιστικό περιβάλλον στη γλώσσα σήμανσης κειμένου XML. Ο προγραμματιστής μπορεί να κάνει drag and drop διάφορα στοιχεία απο τη παλέτα αριστερά και να τα τοποθετήσει με διαφορετική βαρύτητα στην επιφάνεια. Στο προγραμματιστικό κομμάτι δίνονται κάποιες πληροφορίες αναγνώρισης του κάθε στοιχείου, πληροφορίες στοίχισης κ.α.. με σκοπό να τα χειριστεί ο χρήστης στο πρόγραμμα του με τη χρήση εντολών της Java. Με λίγα λόγια τα στοιχεία της XML και ο κώδικας της Java συνεργάζονται.

Στη μεριά του σέρβερ έπρεπε να γίνει η συγγραφή πολλαπλών PHP script και να αποθηκευθούν στο συστημά του. Για τη ανάπτυξη αυτών των προγραμμάτων έγινε χρήση του phpDesigner 8. Ιδιαιτερότητες υπάρχουν και σε αυτόν τον κειμενογράφο καθώς τα εργαλεία που χρησιμοποιούνται απο τους

προγραμματιστές του είναι διαφορετικά απο το Eclipse. Τα script δεν έχουν γραφικό περιβάλλον καθώς το μόνο που επιστρέφουν συνήθως είναι αποτελέσματα απο τη βάση δεδομένων. Ο έλεγχος τους γίνεται μέσω ενός προγράμματος περιήγησης και χρήση απλής HTML. Τα προγράμματα PHP περιέχουν κώδικα σε γλώσσα Javascript, κάτι που υποστηρίζει αυτός ο κειμενογράφος.

Για να γίνει αποθήκευση των script στον σέρβερ έγινε χρήση του Filezilla. Το γραφικό περιβάλλον αυτού του προγράμματος είναι ιδιαίτερα απλό, δίνοντας τη δυνατότητα στον χρήστη να δει ένα κατάλογο αρχείων στον σκληρό δίσκο του υπολογιστή του και ένα κατάλογο αρχείων αποθηκευμένα στον σέρβερ. Με την επιλογή συγκεκριμένων PHP αρχείων εύκολα γίνεται η μεταφορά και αποθήκευση τους στους καταλόγους του σέρβερ. Απο τη στιγμή που είναι αποθηκευμένα εκεί, γίνεται εύκολα η κλήση τους τόσο απο ένα πρόγραμμα περιήγησης είτε μέσω κάποιας διαδικτυακής εφαρμογής που ζητάει κάποια δεδομένα.

Για τη δημιουργία της βάσης έγινε χρήση του προγράμματος phpMyAdmin που φιλοξενείται στον erodio. Εκεί επαναδημιουργήθηκαν όλοι οι πίνακες με όλα τα απαραίτητα πεδία και κάποια δεδομένα για δοκιμαστική χρήση. Η χρήση της γλώσσας βάσεων δεδομένων MySQL είναι απαραίτητη για έργα μικρού βεληνεκούς. Η εύκολη σύνταξη της βοήθησε στη δημιουργία των κατάλληλων ερωτημάτων για την αναζήτηση δεδομένων απο την εφαρμογή και απο την ιστοσελίδα.

Αυτά τα τέσσερα προγράμματα χρησιμοποιήθηκαν κυρίως σε συνδιασμό καθώς οι γλώσσες προγραμματισμού συνεργάζονται σε όλα τα προγράμματα που χρειάζονται για την υλοποίηση αυτής της πτυχιακής

### 3.6 Επίλογος

Η δημιουργία μιας εφαρμογής για την καταγραφή των παρουσιών με ηλεκτρονικό τρόπο περιέχει πολλά ερωτήματα για τον τρόπο κατασκευής της. Το γεγονός ότι διαφορετικές τεχνολογίες πρέπει να συνεργαστούν για το τελικό αποτέλεσμα μας οδήγησε στην κατάτμηση του έργου σε μικρά παραδοτέα με αυτοτελή λειτουργία. Ο διαχωρισμός τους έγινε με υποκειμενικά κριτήρια κάτω απο ώριμη σκέψη με στόχο την απλή χρήση της εφαρμογής και της ιστοσελίδας απο την πλευρά του χρήστη. Ο συνδιασμός των συγκεκριμένων γλωσσών προγραμματισμού δίνει σε κάθε κομμάτι την απαιτούμενη λειτουργικότητα με τον πιο αποδοτικό τρόπο. Με το τέλος της κατασκευής όλων των υποέργων έγινε η ένωση τους με κάποιες μικρές τροποποιήσεις, οδηγώντας στην ολοκληρωμένη λειτουργία της εφαρμογής και της ιστοσελίδας. Όλα τα υποέργα περιέχουν κάποια κοινά και κάποια διαφορετικά στοιχεία προγραμματισμού. Στο επόμενο κεφάλαιο θα εξηγήσουμε τα σημαντικότερα κομμάτια κώδικα που απαιτούνται για τη δημιουργία αυτής της εφαρμογής.

# Κεφάλαιο 4: Προγραμματισμός της εφαρμογής

## 4.1 Εισαγωγή

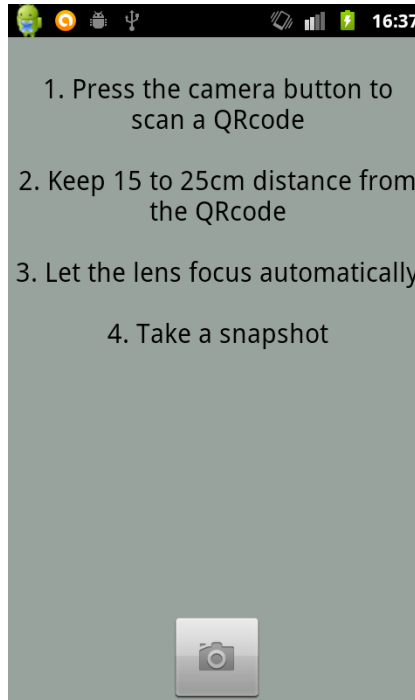
Το QResence είναι μια διαδικτυακή εφαρμογή η υλοποίηση της οποίας απαιτεί τον συνδυασμό γλωσσών προγραμματισμού για την εμφάνιση, την λειτουργικότητα και την επικοινωνία της με τον σέρβερ. Όπως στις περισσότερες εφαρμογές που αναπτύσσονται για την πλατφόρμα Android, βασική γλώσσα προγραμματισμού είναι η Java. Όλη η λειτουργία στην εφαρμογή μας είναι προγραμματισμένη σε αυτή τη γλώσσα η οποία βασίζεται σε γεγονότα που ενεργοποιεί ο χρήστης από την οθόνη της συσκευής του. Επίσης γίνεται χρήση της γλώσσας σήμανσης κειμένου XML που κυρίως αφορά τη μορφοποίηση της εμφάνισης των διαφορετικών στοιχείων της εφαρμογής στην οθόνη του κινητού. Η συνεργασία αυτών των δύο γλωσσών προγραμματισμού απαιτείται για τη δημιουργία μιας απλής έως και της πιο σύνθετης εφαρμογής. Εκτός από αυτές τις γλώσσες προγραμματισμού χρησιμοποιούνται εκτενώς και οι PHP, MySQL και JSON. Αυτές οι γλώσσες, πέρα από το κομμάτι κατασκευής της εφαρμογής, υλοποιούν την λειτουργικότητα που αφορά την ανταλλαγή δεδομένων με τον σέρβερ και τη βάση δεδομένων, απαραίτητα για τη διαδικτυακή επικοινωνία τους. Τα προγράμματα της PHP εκτελούνται κάτω από αιτήματα της εφαρμογής για αναζήτηση και εισαγωγή δεδομένων στη βάση δεδομένων. Η λειτουργία τους είναι η διατύπωση ερωτημάτων σε γλώσσα διαχείρισης δεδομένων της MySQL στη βάση που φιλοξενείται στον σέρβερ. Με τη λήψη των αποτελεσμάτων γίνεται η μετατροπή τους σε μορφότυπο ανταλλαγής δεδομένων JSON ώστε να είναι συμβατά με την εφαρμογή. Η απομακρυσμένη επικοινωνία επιτυγχάνεται χρησιμοποιώντας τους πόρους του ίντερνετ.

Παρακάτω θα αναλύσουμε τα κομμάτια του κώδικα που αφορούν τις σημαντικότερες λειτουργίες της εφαρμογής και των προγραμμάτων που εκτελούνται στον σέρβερ.

#### **4.2 Παρουσίαση των τριών φορμών της εφαρμογής και ανάλυση των πιο σημαντικών γραμμών κώδικα**

Η εφαρμογή QResence μετά απο κάποιες απλοποιήσεις για την αποτελεσματικότερη χρήση της κατέλειξε να αποτελείται απο 4 φόρμες. Κάθε μια απο τις φόρμες εκτελεί ξεχωριστή λειτουργία επομένως και ο κώδικας που υπάρχει πίσω απο αυτήν αποτελείται απο διαφορετικές εντολές. Για το τελικό αποτέλεσμα όλες οι φόρμες συνεργάζονται, κάθεμια παρέχοντας στην επόμενη σημαντικά δεδομένα για την ολοκλήρωση της λειτουργίας της. Όλη η λειτουργία της εφαρμογής εκτελείται σε τρία αρχεία .java τα οποία για να επεξεργαστούν και να εμφανίσουν τα απαιτούμενα δεδομένα πρέπει να συνεργαστούν με την λειτουργία της κάμερας και της σύνδεσης στο διαδίκτυο. Θα παρουσιάσουμε κάθε φόρμα και θα εξηγήσουμε τη ροή των δεδομένων μέσα απο συγκεκριμένες γραμμές κώδικα.

#### 4.2.1 1<sup>η</sup> φόρμα: Εκκίνηση εφαρμογής



Εικόνα 4.1: 1<sup>η</sup> φόρμα της εφαρμογής QRResence

Με την έναρξη της εφαρμογής εμφανίζονται στην οθόνη της συσκευής οι οδηγίες για το σκανάρισμα ενός QR Code με χρήση της κάμερας. Στο χαμηλότερο κομμάτι της φόρμας υπάρχει ένα κουμπί με την ένδειξη μιας φωτογραφικής μηχανής που παραπέμπει στην έναρξη λειτουργίας της κάμερας της συσκευής. Με την εκκίνηση της κάμερας, ο χρήστης εστιάζει πάνω στον γραμμωτό κώδικα μιας αίθουσας και τραβάει ένα στιγμιότυπο του. Η λειτουργία της κάμερας ενεργοποιεί κάποιες επιλογές που προτείνουν στον χρήστη την απαθανάτιση ενός στιγμιότυπου εκ νέου ή την χρήση του ήδη υπάρχοντος. Η εικόνα επιστρέφει στην εφαρμογή και ξεκινάει η επεξεργασία της για την αποκρυπτογράφηση των δεδομένων. Η εντολές που επεξεργάζοντε την εικόνα βρίσκονται στις κλάσεις QRcodeActivity και RGBLuminanceSource. Η τελευταία συνεργάζεται άμεσα με την βιβλιοθήκη ZXing για την αποκρυπτογράφηση της εικόνας.

Μετά την αρχικοποίηση των μεταβλητών στην κλάση QRcodeActivity υπάρχει ένας αριθμός απο μεθόδους που καλύπτουν τα διάφορα γεγονότα που συμβαίνουν στην οθόνη της συσκευής. Με την ενεργοποίηση του camerabutton εκτελείται ο παρακάτω κώδικας.

```
i = new Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);  
i.putExtra(MediaStore.EXTRA_OUTPUT, "");  
startActivityForResult(i, RequestCode);
```

Στην πρώτη γραμμή δημιουργείται ένα αντικείμενο τύπου *Intent* που χρησιμοποιείται για την εκκίνηση διάφορων λειτουργιών και στη συγκεκριμένη περίπτωση για τη χρήση της φωτογραφικής μηχανής. Η βιβλιοθήκη *MediaStore* που χρησιμοποιείται στη πρώτη και στη δεύτερη γραμμή περιέχει τις κλάσεις για όλα τα διαθέσιμα media και τους χώρους αποθήκευσης της συσκευής. Στη δεύτερη γραμμή δηλώνουμε ότι το στιγμιότυπο της εικόνας δε θέλουμε να αποθηκευθεί σε κάποιον αποθηκευτικό χώρο και για αυτό η δεύτερη παράμετρος της εντολής είναι το κενό. Η τελευταία εντολή ξεκινάει την εκτέλεση της φωτογραφικής μηχανής με χρήση του αντικειμένου *Intent* και της σταθεράς *RequestCode* που θα χρησιμοποιηθεί μετέπειτα για τον έλεγχο του αποτελέσματος.

Μετά την λήψη της εικόνας απο την φωτογραφική μηχανή της συσκευής η λειτουργία επιστρέφει και πάλι στην κλάση QRcodeActivity. Αφού γίνει έλεγχος για το αποτέλεσμα που παραλαμβάνει απο την προηγούμενη δραστηριότητα γίνεται μετατροπή της εικόνας σε μορφή Bitmap για να είναι συμβατή στη δημιουργία αντικειμένου *RGBLuminanceSource*.

```
bmp = (Bitmap) extras.get("img");  
LuminanceSource source = new RGBLuminanceSource(bmp);
```

Η δημιουργία του αντικειμένου *source* γίνεται απο τον δομητή της κλάσης *RGBLuminanceSource* ο οποίος παραλαμβάνει την εικόνα σε μορφή *Bitmap* και την μετατρέπει σε δεδομένα που την περιγράφουν σε κλίμακες του γκρι. Τα δεδομένα επιστρέφουν στην κλάση *QRcodeActivity* και μετατρέπονται πάλι σε αντικείμενο τύπου *Bitmap*.

Στην συνέχεια γίνεται χρήση της βιβλιοθήκης *ZXing* για τη δημιουργία του αντικειμένου *reader* της κλάσης *MultiFormatReader* που βρίσκεται σε αυτή.

```
Reader reader = new MultiFormatReader();  
com.google.zxing.Result result;  
result = reader.decode(bitmap);
```

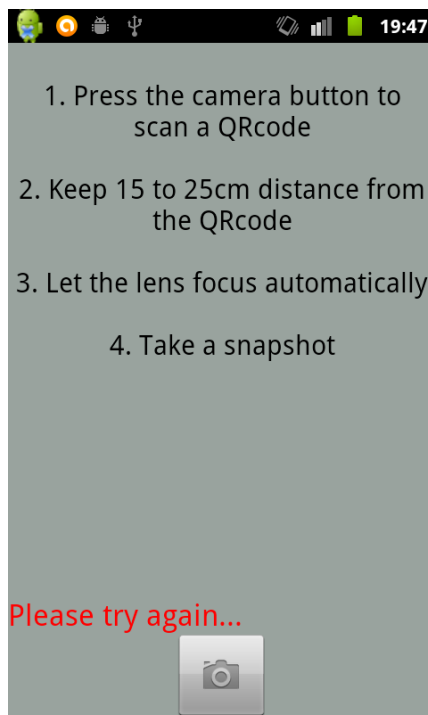
Χρησιμοποιώντας την κλάση *Result* της βιβλιοθήκης δηλώνουμε την αναφορά *result* στην οποία αποθηκεύουμε το αποτέλεσμα της μεθόδου *decode()* του αντικειμένου *reader* που αποκωδικοποιεί την εικόνα (αντικείμενο *Bitmap*).

Έχοντας αποκωδικοποιήσει την εικόνα του γραμμωτού κώδικα της αίθουσας έχουμε λάβει την διεύθυνση που θα χρησιμοποιηθεί αργότερα για τη κλήση ενός συγκεκριμένου *script* στον σέρβερ με την επιπλέον πληροφορία του αριθμού της αίθουσας. Ένα παράδειγμα αποκωδικοποιημένης διεύθυνσης της αίθουσας 201 είναι:

[http://erodios.it.teithe.gr/kourou\\_projects/QRcodes/android/android\\_mathima.php?aitousa=201](http://erodios.it.teithe.gr/kourou_projects/QRcodes/android/android_mathima.php?aitousa=201)

Οι παραπάνω εντολές περιστοιχίζονται απο κώδικα *try-catch* για των χειρισμό εξαιρέσεων. Σε περίπτωση που η κάμερα δεν εστιάσει σωστά στο QR Code, οι εντολές που ακολουθούν θα σταματήσουν να εκτελούνται στο σημείο που καλούνται οι μέθοδοι των κλάσεων της βιβλιοθήκης *ZXing* καθώς δε θα μπορέσουν να επεξεργαστούν ξένα για αυτές δεδομένα. Η εξαίρεση αυτή συλλαμβάνεται παρακάτω και εκτελείται η εμφάνιση του μηνύματος “Please try again...” στην οθόνη του κινητού.



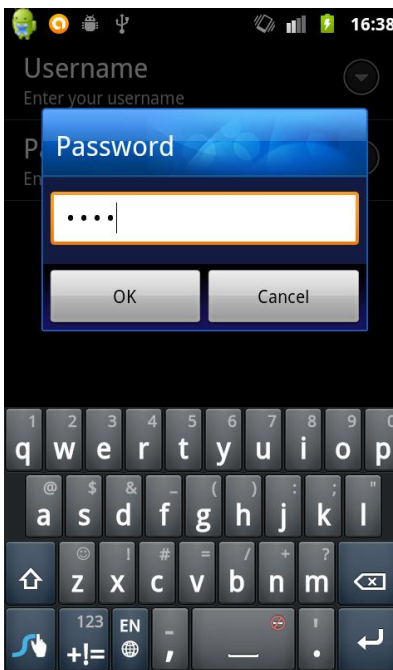


Εικόνα 4.2: Αδυναμία αποκωδικοποίησης QR Code

Συνοπτικά αυτή είναι η λειτουργία αποκωδικοποίησης του γραμμωτού κώδικα. Πολλές εντολές, απαραίτητες για τη ροή και τον έλεγχο των δεδομένων, παραλήφθηκαν για να εστιάσουμε σε αυτές που δείχνουν τον τρόπο μετατροπής των δεδομένων σε πληροφορία. Παραλαμβάνοντας την διεύθυνση με την πληροφορία της αίθουσας μπορούμε να συνεχίσουμε στην επόμενη φόρμα όπου θα γίνει η αναζήτηση των μαθημάτων. Πρώτα όμως ο χρήστης πρέπει να δηλώσει σε μια ξεχωριστή φόρμα τα προσωπικά του στοιχεία (username και password).

#### 4.2.2 2<sup>η</sup> φόρμα: Εισαγωγή στοιχείων Login

Για την πλήρη λειτουργία της εφαρμογής είναι απαραίτητη η εισαγωγή των προσωπικών στοιχείων του χρήστη. Τα δεδομένα αυτά αποθηκεύονται σε μια ξεχωριστή φόρμα που εμφανίζεται με το πάτημα του κουμπιού ρυθμίσεων της συσκευής.



Εικόνα 4.3: Εισαγωγή Username και Password

Με το γεγονός της αφής του κουμπιού εκτελείται μια μέθοδος της κλάσης `QRcodeActivity` που μας οδηγεί στην κλάση `PrefsActivity`. Στον παρακάτω κώδικα βλέπουμε την αναγνώριση του κουμπιού ρυθμίσεων με `id` την τιμή `'item_settings'`. Σε αυτή τη περίπτωση δημιουργείται ένα αντικείμενο `Intent` που μας οδηγεί στην κλάση που θέλουμε με χρήση της μεθόδου `startActivity()`.

```
case R.id.item_settings:  
    startActivity(new Intent(PrefsActivity.class));  
    return true;
```

Η λειτουργία της κλάσης `PrefsActivity` είναι να εμφανίσει την φόρμα εισαγωγής των στοιχείων και να ενώσει αυτά τα δεδομένα με τον κώδικα για την μετέπειτα επεξεργασία τους.

Εδώ χρησιμοποιούμε λειτουργίες της κλάσης `SharedPreferences` που είναι διαθέσιμη από το `Android SDK` και μας επιτρέπει την αποθήκευση και την ανάκτηση προσωπικών δεδομένων σε μια εφαρμογή. Αυτού του είδους η μορφή αποθήκευσης κάνει χρήση ενός 'κλειδιού' για κάθε δεδομένο καθώς με το όνομα αυτού του 'κλειδιού' μπορούμε να ανακτήσουμε την τιμή του. Τα δεδομένα αφού αρχικοποιηθούν παραμένουν αποθηκευμένα και μετά το κλείσιμο της εφαρμογής γεγονός που βοηθά τον χρήστη στο να μη τα εισάγει εκ νέου κάθε φορά που κάνει χρήση της. Τα στοιχεία που αποθηκεύονται είναι αυτά που εισήχθησαν μετά την τελευταία μεταβολή των πεδίων `username` και `password`. Αφού οριστικοποιηθούν από τον χρήστη, η ανάκτηση τους στον κώδικα γίνεται με τον παρακάτω τρόπο.

```
SharedPreferences pr = PreferenceManager.getDefaultSharedPreferences();  
String username = pr.getString("username", "");  
String password = pr.getString("password", "");
```

Δημιουργείται ένα αντικείμενο της κλάσης `SharedPreferences` το οποίο περιέχει όλα τα δεδομένα που έχουν αποθηκευθεί στην φόρμα. Μετά την δήλωση των μεταβλητών `username` και `password` κάνουμε ανάκτηση των δεδομένων από το αντικείμενο με χρήση του διακριτικού τους 'κλειδιού'. Αυτές οι μεταβλητές περιέχουν τις τιμές που έδωσε ο χρήστης και θα χρησιμοποιηθούν αργότερα για τον έλεγχο της ταυτότητας του.

#### 4.2.3 3<sup>η</sup> φόρμα: Αναζήτηση, εμφάνιση μαθημάτων και τοποθέτηση παρουσίας

Η λειτουργία αυτής της φόρμας είναι να εμφανίζει τα μαθήματα που γίνονται εκείνη την ώρα σύμφωνα με το QR Code της αίθουσας που επέλεξε ο χρήστης και η δυνατότητα επιλογής ενός από αυτά για την ηλεκτρονική τοποθέτηση της παρουσίας του. Αυτή η φόρμα εμφανίζεται λόγω της κλάσης AithousaActivity η οποία περιέχει το μεγαλύτερο ποσοστό λειτουργικότητας της εφαρμογής. Εκτός από την εμφάνιση της φόρμας, η κλάση δημιουργεί σύνδεση με τον σέρβερ για έλεγχο των στοιχείων ταυτοποίησης και αναζήτηση δεδομένων, ανακτά δεδομένα από τις άλλες κλάσεις, βρίσκει την παρούσα μέρα και ώρα, και εκτελεί την προσθήκη παρουσίας απομακρυσμένα στη βάση δεδομένων. Μερικές από αυτές τις λειτουργίες της κλάσης θα εξηγήσουμε παρακάτω.



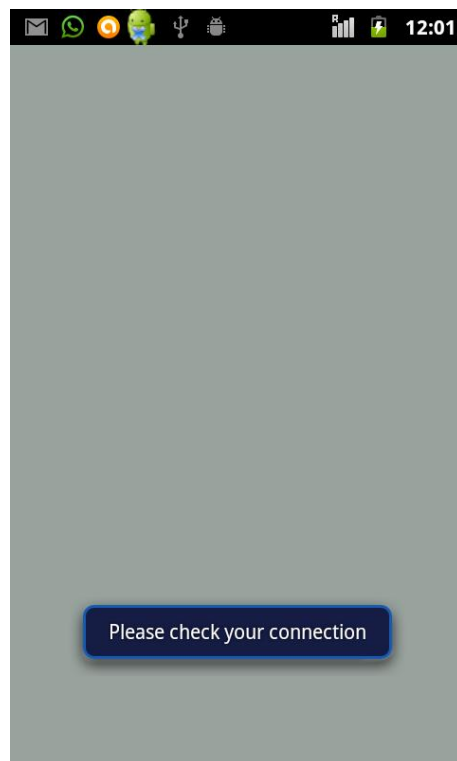
Εικόνα 4.4 Φόρμα τοποθέτησης παρουσίας

## Έλεγχος υλοποίησης της σύνδεσης στο διαδίκτυο και έλεγχος των στοιχείων ταυτοποίησης του χρήστη

Με το που ολοκληρωθεί η μετάβαση στη κλάση `AithousaActivity` φορτώνεται το γραφικό περιβάλλον με τα στατικά στοιχεία της φόρμας στην οθόνη της συσκευής. Αμέσως μετά γίνεται η κλήση της μεθόδου `internetConnection()` η οποία ελέγχει αν έχει επιτευχθεί σύνδεση της συσκευής στο διαδίκτυο, απαραίτητη λειτουργία για την αναζήτηση δεδομένων στην απομακρυσμένη βάση. Στη περίπτωση που η σύνδεση είναι ενεργή καλείται η μέθοδος `testLoginData()` για τον έλεγχο των στοιχείων ταυτοποίησης του χρήστη. Ειδάλλως εκτελείτε η εμφάνιση του μηνύματος "Please check your connection".

```
if (activeNetwork != null&& activeNetwork.isConnectedOrConnecting()) {
    correct_data = testLoginData();
} else {
    Toast.makeText(getBaseContext(), "Please check your connection",
        Toast.LENGTH_SHORT).show();
}
```

Μέσα στην παρένθεση εκτελείται η συνθήκη ελέγχου για το αν η σύνδεση υπάρχει και αν είναι σε διαδικασία ολοκλήρωσης. Η σύνδεση μπορεί να πραγματοποιηθεί μέσω Wifi ή του δικτύου τηλεφωνίας. Το προειδοποιητικό μήνυμα εμφανίζεται με τη χρήση της εντολής `Toast` με τη δεύτερη παράμετρο να καθορίζει τον χρόνο εμφάνισης του (`LENGTH_SHORT=2 sec`).



Εικόνα 4.5 Εμφάνιση μηνύματος αποτυχίας σύνδεσης στο ιντερνετ

Για τη λειτουργία της μεθόδου *testLoginData()* απαιτείται σύνδεση με τον σέρβερ και αναζήτηση δεδομένων στην βάση. Για την υλοποίηση της μεθόδου δηλώνονται στην αρχή της κλάσης κάποιες μεταβλητές που αφορούν λειτουργίες του πρωτοκόλλου HTTP. Μέσα στη μέθοδο αρχικοποιούνται ως εξής:

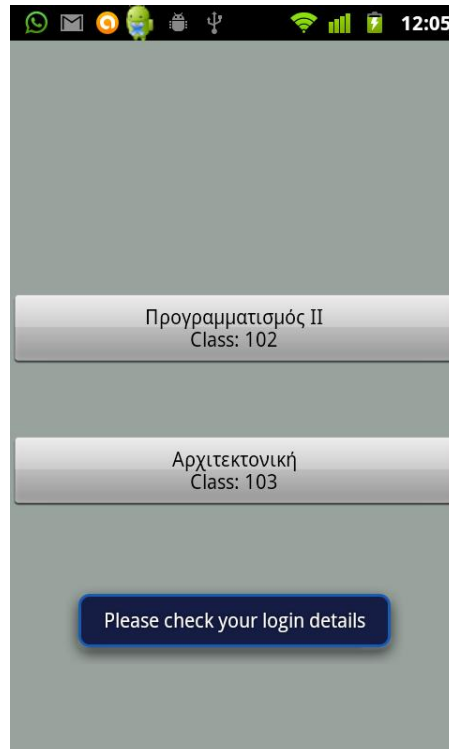
```
httpClient = newDefaultHttpClient();  
httpPost =  
newHttpPost("http://erodios.it.teithe.gr/kourou_projects/QRCodes/androi  
d/android_login.php");  
response = httpClient.execute(httpPost);
```

Στη πρώτη μεταβλητή αναθέτεται το αντικείμενο ενός `HTTPClient` που θα υλοποιήσει στη συνέχεια τη μέθοδο `POST`. Στη δεύτερη μεταβλητή αναθέτεται ένα αντικείμενο τύπου `HttpPost` που έχει ως παράμετρο τη διεύθυνση του `script` που είναι αρμόδιο για τον έλεγχο των στοιχείων του χρήστη. Το αντικείμενο αυτό μεταφέρει τα `username` και `password` του χρήστη στο `script` με τη μέθοδο `POST`.

Μετά την αποστολή των στοιχείων, το πρόγραμμα που εκτελείται στον σέρβερ κάνει αναζήτηση στη βάση ώστε να βρεθεί η οντότητα που έχει στα πεδία της το συγκεκριμένο `username` και `password`. Τα πεδία της οντότητας αποστέλλονται πίσω στην εφαρμογή σε μορφή `JSON`, λειτουργία που θα εξηγήσουμε σε επόμενη ενότητα. Η μέθοδος κάνει έλεγχο για το αν υπάρχουν δεδομένα για να συνεχίσει στη σύγκριση των στοιχείων που έδωσε ο χρήστης με τα αντίστοιχα πεδία των ληφθέντων δεδομένων από τον σέρβερ. Δηλαδή γίνεται διπλός έλεγχος, με την αναζήτηση του φοιτητή στη βάση δεδομένων και με τη σύγκριση αυτών των δεδομένων με τα στοιχεία εισόδου που έδωσε στην εφαρμογή.

```
entity = response.getEntity();
if (entity != null) {
    if (username_str.equals(retUser)&&password_str.equals(retPass)){
        return true;
    }
}
```

Στον παραπάνω κώδικα γίνεται η ανάθεση των ληφθέντων δεδομένων στη αντικείμενο `entity`. Πιο κάτω ελέγχεται η κατάσταση του αντικειμένου για το αν περιέχει δεδομένα και στη συνέχεια γίνεται η σύγκριση των στοιχείων της βάσης με τα στοιχεία που έδωσε ο χρήστης στην εφαρμογή μέσω της μεθόδου συγκρίσεων συμβολοσειρών `equals()`. Στη περίπτωση που αποτύχει κάποιος από τους ελέγχους, γίνεται σύλληψη της εξαίρεσης έξω από το μπλόκ της συνθήκης και εμφανίζεται το μήνυμα “Please check your login details”.



Εικόνα 4.6 Εμφάνιση μηνύματος αποτυχίας Login

## Μεταφορά δεδομένων απο φόρμα σε φόρμα

Η απαραίτητη πληροφορία που πρέπει να περάσει απο την κλάση QRcodeActivity στην κλάση AithousaActivity είναι το αποκωδικοποιημένο URL με την ταυτότητα της αίθουσας. Η λειτουργία αυτή γίνεται με τη χρήση 'καλαθιού' που μας παρέχει η κλάση Bundle. Εδώ αντιστοιχούμε κάθε δεδομένο με ένα αναγνωριστικό, όπως ακριβώς γίνεται και στην χρήση των αντικειμένων SharedPreferences. Το παρακάτω κομμάτι κώδικα βρίσκεται στην κλάση QRcodeActivity.

```
Bundlebasket = newBundle();  
basket.putString("URL", result.getText());  
Intent c = new Intent(QRcodeActivity.this, AithousaActivity.class);  
c.putExtras(basket);
```



Μετά την δημιουργία του αντικειμένου *basket* της κλάσης *Bundle*, εισάγεται σε αυτό το αποτέλεσμα της αποκωδικοποίησης του *QRCode* με το διακριτικό "URL". Στη συνέχεια δημιουργείται ένα αντικείμενο τύπου *Intent* με το οποίο θα γίνει η μετάβαση στην επόμενη κλάση. Η τελευταία εντολή προσθέτει στο αντικείμενο τύπου *Intent* το αντικείμενο *basket* το οποίο 'κουβαλά' τη πληροφορία που θα χρειαστούμε στη συνέχεια.

Στην κλάση *AithousaActivity* υπάρχει η μέθοδος *basket()* η οποία αναλαμβάνει την εξαγωγή των δεδομένων από το 'καλάθι'. Η κλήση της μεθόδου γίνεται με το που ολοκληρωθεί η μετάβαση στην κλάση *AithousaActivity* ώστε το URL να είναι άμεσα διαθέσιμο για την επεξεργασία που ακολουθεί. Παρακάτω βρίσκεται ο κώδικας της μεθόδου.

```
BundlegetBasket = getIntent().getExtras();  
String link = getBasket.getString("URL").toString();
```

Ξαναδημιουργούμε ένα αντικείμενο τύπου *Bundle* στο οποίο γίνεται ανάθεση των πληροφοριών που προστέθηκαν στο αντικείμενο τύπου *Intent* το οποίο μας βοήθησε στην μετάβαση στην κλάση *AithousaActivity*. Στην μεταβλητή *link* τοποθετείται το περιεχόμενο του 'καλαθιού' με το διακριτικό 'URL' σε μορφή συμβολοσειράς. Μόλις μεταφέραμε επιτυχώς την αποκωδικοποιημένη διεύθυνση της αίθουσας που θα χρησιμοποιήσουμε για την αναζήτηση των μαθημάτων με κλήση του προγράμματος που είναι αποθηκευμένο στον σέρβερ.

## Αναζήτηση ώρας και μέρας

Αμέσως μετά τον έλεγχο των στοιχείων του χρήστη και την παραλαβή του URL απο την προηγούμενη φόρμα καλούνται οι μέθοδοι *imera()* και *ora()*. Όπως γίνεται αντιληπτό απο τα ονόματα των μεθόδων οι λειτουργίες τους αφορούν την ενημέρωση της κλάσης με την παρούσα μέρα και ώρα. Αυτές οι πληροφορίες θα φανούν χρήσιμες στην αναζήτηση των αντίστοιχων πεδίων στους πίνακες της βάσης για την εμφάνιση των μαθημάτων που γίνονται στην αίθουσα.

Για την αναζήτηση της ημέρας γίνεται χρήση της υποκλάσης *GregorianCalendar* που κληρονομεί-υλοποιεί τις μεθόδους της αφηρημένης (abstract) κλάσης *Calendar*. Οι μέθοδοι αυτής της κλάσης βοηθάνε με πολλούς τρόπους τον προγραμματιστή να αναζητήσει την ημέρα, τον μήνα και το έτος στη παρούσα στιγμή και να τα παραλάβει σε διάφορες μορφές για να τα χρησιμοποιήσει όπως εκείνος νομίζει. Στην μέθοδο *imera()* γίνεται εμφάνιση του αριθμού που αντιπροσωπεύει την ημέρα σε σχέση με τις επτά ημέρες της εβδομάδας.

```
GregorianCalendar calendar = new GregorianCalendar();
int testDay = calendar.get(Calendar.DAY_OF_WEEK);

if (testDay == 2) {
    dayOfTheWeek = "Deutera";
}
```

Η πρώτη εντολή στη μέθοδο δημιουργεί το αντικείμενο *calendar* της κλάσης *GregorianCalendar*. Η μεταβλητή *testDay* λαμβάνει την τιμή του αριθμού της ημέρας μέσω της κλάσης *Calendar* που αλλάζει τη μορφή της ημερομηνίας σε μορφή 'ημέρα της βδομάδας'. Εφόσον κάθε βδομάδα ξεκινάει απο την Κυριακή, ακολουθεί σύγκριση της μεταβλητής με τον αριθμό 2 συνεχίζοντας έως και τον 6 που υποδηλώνει ότι είναι Παρασκευή. Δεν υπάρχει έλεγχος για το Σάββατο και την Κυριακή καθώς δεν γίνονται μαθήματα εκείνες τις ημέρες. Ο λόγος που μετατρέπουμε τον αριθμό σε ημέρα είναι για να περάσουμε την μεταβλητή *dayOfTheWeek* στο πρόγραμμα στον σέρβερ για τους κατάλληλους ελέγχους.

Στην μέθοδο *ora()* γίνεται αναζήτηση για δύο δεδομένα, τα λεπτά και την ώρα. Κάνουμε χρήση και των δύο αυτών στοιχείων λόγω της ιδιομορφίας της εφαρμογής να επιτρέπει την εμφάνιση μαθημάτων στα χρονικά παράθυρα που βρίσκονται ανάμεσα σε συγκεκριμένες ώρες. Να υπενθυμίσουμε ότι στα τελευταία 15 λεπτά του μαθήματος που γίνεται και στα πρώτα 15 λεπτά του μαθήματος που θα γίνει, το QResence θέλουμε να μας εμφανίζει και τα δυο μαθήματα.

```
Calendar c = Calendar.getInstance();  
int hour = c.get(Calendar.HOUR_OF_DAY);  
int minute_inst = c.get(Calendar.MINUTE);
```

Με την δημιουργία του αντικειμένου της κλάσης Calendar αποθηκεύουμε στις μεταβλητές *hour* και *minute\_inst* την ώρα και το λεπτά αντίστοιχα.

```
if (hour < 10) {  
    if (hour == 7 && minute_inst > 44) {  
        return 1;  
    } elseif (hour == 8) {  
        return 1;  
    } elseif (hour == 9 && minute_inst < 45) {  
        return 1;  
    } elseif (hour == 9 && minute_inst > 44) {  
        return 2;  
    }  
}
```

Ο παραπάνω έλεγχος αφορά τα μαθήματα που γίνονται από τις 8 μέχρι τις 10 το πρωί. Πρώτη προϋπόθεση για αυτά τα μαθήματα, η ώρα που διδάσκονται να είναι πριν τις δέκα. Στη συνέχεια γίνονται διαδοχικοί έλεγχοι της ώρας και των λεπτών με σκοπό να επιστραφεί ένας αριθμός. Αυτός ο αριθμός σε συνδυασμό με την μεταβλητή της ημέρας θα χρησιμοποιηθεί για την αναζήτηση των μαθημάτων. Οι επιστρεφόμενοι αριθμοί της μεθόδου *ora()* κυμαίνονται από το 1 έως και το 9 με τους ζυγούς αριθμούς να αντιστοιχούν σε δυο μαθήματα καθώς αντιπροσωπεύουν τα ενδιάμεσα χρονικά παράθυρα της μισής ώρας. Οι μονοί

## Πτυχιακή εργασία του Θεόδωρου Κλιάφα

αριθμοί μας επιστρέφουν ένα μάθημα και το 0 που αντιστοιχεί στις ώρες πριν τις 8 το πρωί και μετά τις 6 το απόγευμα δεν μας επιστρέφει τίποτα. Πιο αναλυτικά:

Επιστρεφόμενες τιμές της μεθόδου ora()

τιμή 1->7:45 - 9:44

τιμή 2->9:45 - 10:15 //Δύο μαθήματα

τιμή 3->10:16 - 11:44

τιμή 4->11:45 - 12:15 //Δύο μαθήματα

τιμή 5->12:16 - 13:44

τιμή 6->13:45 - 14:15 //Δύο μαθήματα

τιμή 7->14:16 - 15:44

τιμή 8->15:45 - 16:15 // Δύο μαθήματα

τιμή 9->16:16 - 18:00

### **Αναζήτηση μαθημάτων σύμφωνα με την μέρα, την ώρα και την αίθουσα**

Εφόσον έχουμε φτάσει στο σημείο όπου γνωρίζουμε τον αριθμό της αίθουσας, την ημέρα και την ώρα μπορούμε να καλέσουμε το κατάλληλο πρόγραμμα στον σέρβερ για να κάνει την αναζήτηση των μαθημάτων στην βάση δεδομένων. Εδώ ο έλεγχος των δεδομένων γίνεται περισσότερο στη μεριά του σέρβερ για αυτό θα αναλυθεί και κομμάτι του κώδικα του αντίστοιχου PHP προγράμματος ώστε να γίνει αντιληπτό στο πως καταλήγουμε να έχουμε αποτελέσματα. Στην εφαρμογή, όπως και με τον έλεγχο των στοιχείων ταυτοποίησης, η διαδικασία είναι παρόμοια. Η διαφορά είναι πως διαχωρίζει τα δεδομένα που λαμβάνει και πως τα χειρίζεται στην συνέχεια.

Οι μεταβλητές για την ανταλλαγή δεδομένων με τον σέρβερ προϋπάρχουν απο τον έλεγχο των στοιχείων ταυτοποίησης. Στην περίπτωση της αναζήτησης των μαθημάτων αλλάζει η ανάθεση του αντικειμένου *httppost* όπου εισάγεται η διεύθυνση με τον αριθμό της αίθουσας που αποκρυπτογραφήσαμε και παραπέμπει στο αρμόδιο πρόγραμμα που εκτελείται στον σέρβερ. Ο αριθμός της αίθουσας βρίσκεται στο τέλος της διεύθυνσης και γίνεται αντιληπτός απο το πρόγραμμα στον σέρβερ μέσω της μεθόδου GET. Οι μεταβλητές *day* και *time* προσθέτονται στο αντικείμενο *httppost* όπως θα δείξουμε παρακάτω και η λήψη τους απο το script γίνεται μέσω της μεθόδου POST.

```
nameValuePairs = new ArrayList<NameValuePair>();
nameValuePairs.add(new BasicNameValuePair("mera", day));
nameValuePairs.add(new BasicNameValuePair("ora", time));

httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs));
```

Για την μεταφορά μεταβλητών απο την εφαρμογή στον σέρβερ γίνεται χρήση της κλάσης *NameValuePair* το αντικείμενο-λίστα της οποίας ενσωματώνει δεδομένα με το αντίστοιχο διακριτικό τους. Στην συγκεκριμένη περίπτωση προσθέτουμε στο αντικείμενο τις μεταβλητές *day* και *time* με τα διακριτικά *mera* και *ora* αντίστοιχα. Στην συνέχεια το αντικείμενο-λίστα *nameValuePairs* παραδίδεται στο αντικείμενο *httppost* μέσω της μεθόδου *setEntity()*. Είμαστε έτοιμοι να κάνουμε κλήση στο πρόγραμμα του σέρβερ για την αναζήτηση των μαθημάτων σύμφωνα με τις μεταβλητές μας.

Στο script με όνομα *android\_mathima.php* οι πρώτες γραμμές του κώδικα υλοποιούν την σύνδεση με την βάση δεδομένων όπως γίνεται άλλωστε και με τα περισσότερα scripts. Στη συνέχεια γίνεται η χρήση των μεθόδων της PHP, *\_Get* και *\_POST*, για την ανάθεση των μεταβλητών *\$aithousa*, *\$mera* και *\$ora*. Τα δεδομένα αυτά γίνονται αντιληπτά απο τα διακριτικά τους στη διεύθυνση και απο την κλήση που πραγματοποιούμε όπου είναι εσώκλειστα.

```
$aithousa= $_GET['aithousa'];
```

```
$mera= $_POST['mera'];
```

```
$ora= $_POST['ora'];
```

Ακολουθεί ο έλεγχος της ώρας για να καταλήξουμε στο αν θα πρέπει να γίνει αναζήτηση για ένα ή για δύο μαθήματα. Στην περίπτωση των δύο μαθημάτων εκτελείται επαναληπτικός βρόγχος για την αναζήτηση του κάθε μαθήματος με τα αποτελέσματα να αποθηκεύονται σε ένα πίνακα για μετέπειτα χρήση. Για να καταλήξουμε στην εμφάνιση του ονόματος των μαθημάτων υλοποιούμε δύο αναζητήσεις. Στην πρώτη βασιζόμαστε στον πίνακα-σύζευξη 'Διδάσκει' ο οποίος συνδέει του πίνακες 'Αίθουσα', 'Ωρες', 'Ημέρες' και 'Τμήμα'. Σύμφωνα με τις μεταβλητές μας θα μπορέσουμε να βρούμε τον κωδικό του τμήματος τον οποίο θα χρησιμοποιήσουμε στην δεύτερη αναζήτηση για το όνομα του μαθήματος που συνδέεται με αυτό το τμήμα. Στο παράδειγμά μας θα αναφερθούμε στη λήψη αποτελεσμάτων για δύο μαθήματα.

```
$query = mysql_query(")
```

```
SELECT tmima_id
```

```
FROM teaches te, room ro, teaching_days td, teaching_hours th
```

```
WHERE te.room_id = ro.id
```

```
AND ro.name = '$aithousa'
```

```
AND te.day_id = td.id
```

```
AND td.onomasia = '$mera'
```

```
AND te.hour_id = th.id
```

```
AND th.id = '$i'");
```

Αποθηκεύουμε στην μεταβλητή *\$query* το αποτέλεσμα της αναζήτησης των εντολών της MySQL. Η PHP διαθέτει πολλές μεθόδους που συνεργάζονται στενά με την MySQL. Μια από αυτές είναι και η *mysql\_query()* που χρησιμοποιήθηκε πολλές φορές για τις ανάγκες αυτού του project. Στη πρώτη γραμμή των εντολών αναζήτησης, θέτουμε ως επιθυμητό δεδομένο τον κωδικό του τμήματος.

Ακολουθούν οι περιορισμοί της αίθουσας, της μέρας και της ώρας στους αντίστοιχους πίνακες με χρήση των μεταβλητών που λήφθηκαν απο την εφαρμογή. Στον πίνακα 'Αίθουσα' αναζητούμε τον κωδικό της αίθουσας σύμφωνα με την ονομασία της και το ίδιο κάνουμε στην αναζήτηση του κωδικού της μέρας. Στην αναζήτηση του κωδικού της ώρας, η μεταβλητή  $\$i$  προέκυψε απο την μετατροπή της μεταβλητής  $\$ora$  για τις ανάγκες του επαναληπτικού βρόγχου. Οι παραπάνω περιορισμοί συνδιαζόμενοι θα μας επιστρέψουν τον κωδικό ενός τμήματος που θα μπορέσουμε να χρησιμοποιήσουμε για την αναζήτηση του ονόματος του μαθήματος.

Μέσα στον ίδιο επαναληπτικό βρόγχο και μετά την λήψη του αποτελέσματος του κωδικού του τμήματος γίνεται αναζήτηση με βάση τον πίνακα 'Συνολικό μάθημα'. Για αυτήν την αναζήτηση χρειαζόμαστε την σύνδεση των πινάκων 'Είδος μαθήματος' και 'Τμήμα' όπως φαίνεται παρακάτω.

```
$query = mysql_query("
SELECT onoma
FROM lesson le, lesson_type lety, tmima tmi
WHERE le.id = lety.lesson_id
AND lety.type_id = tmi.kodikos_mathimatos
AND tmi.id = '$tmima'");
```

Πάλι με χρήση της εντολής `mysql_query()` υλοποιούμε την αναζήτηση για το όνομα του μαθήματος με μόνο περιορισμό τον κωδικό του τμήματος που βρήκαμε στο προηγούμενο ερώτημα στη βάση. Η αναφορά του πίνακα 'Είδος μαθήματος' είναι απαραίτητη για τη λήψη του αποτελέσματος καθώς συνδέει τους άλλους δύο πίνακες που είναι ξένοι μεταξύ τους.

Έχοντας λάβει τα αποτελέσματα του κωδικού του τμήματος και του ονόματος του μαθήματος, τα αποθηκεύουμε σε ένα πίνακα με σκοπό να γίνει η επανάληψη του βρόγχου για τη λήψη των δεύτερων αποτελεσμάτων. Όταν ολοκληρωθεί και η δεύτερη επανάληψη και όλα τα δεδομένα είναι αποθηκευμένα στον πίνακα το πρόγραμμα βγαίνει απο τον επαναληπτικό βρόγχο και εκτυπώνει τα αποτελέσματα σε μορφή JSON.

```
echo '{"result":' . json_encode($result) . "};"
```

Η εντολή echo είναι αυτή που εμφανίζει τα δεδομένα που θέλουμε μέσα στα αποσοιοποιητικά. Επίσης γίνεται χρήση της εντολής *json\_encode()* της PHP που κωδικοποιεί τα δεδομένα σε μορφή JSON. Αυτή η μετατροπή των δεδομένων είναι απαραίτητη για την συμβατότητας τους με την εφαρμογή. Για να έχουμε το επιθυμητό αποτέλεσμα ώστε η εφαρμογή να αντιληφθεί την διαφορά μεταξύ των δύο κελιών του πίνακα προστέθηκαν προγραμματιστικά οι αγκύλες και οι άνω-κάτω τελείες πριν και μετά το όνομα του πίνακα προς εμφάνιση. Το αποτέλεσμα σε έναν περιηγητή φαίνεται ως εξής:

```
"result": [{
    "tmima_id":110,
    "onoma": Δομές II
  },
  {
    "tmima_id": 113,
    "onoma": ΔίκτυαII,
  }
]
```



Η διαδικασία εκτέλεσης του προγράμματος στον σέρβερ έφτασε στο τέλος της. Τα επιθυμητά αποτελέσματα εστάλησαν στην εφαρμογή που με την σειρά της πρέπει να τα λάβει και να τα εμφανίσει.

```
InputStream instream = entity.getContent();
JSONObject jsonResponse = new
JSONObject(convertStreamToString(instream));
JSONArray result = jsonResponse.getJSONArray("result");
```

Στην πρώτη γραμμή γίνεται η λήψη των αποτελεσμάτων στην μεταβλητή *instream* στη μορφή της ροής δεδομένων. Η δημιουργία του αντικειμένου *jsonResponse* αποκωδικοποιεί τα δεδομένα αφότου έχουν μετατραπεί σε αλφαριθμητική σειρά χαρακτήρων απο την μέθοδο *convertStreamToString()*. Ακολουθεί η δημιουργία του πίνακα *result* της κλάσης *JSONArray* που δέχεται τα δεδομένα του πίνακα που απέστειλε ο σέρβερ με την χρήση της μεθόδου *getJSONArray()*.

Η εφαρμογή έχει λάβει τα δεδομένα και τώρα πρέπει να τα εμφανίσει. Πρώτα γίνεται έλεγχος του αριθμού των κελιών στον πίνακα ώστε να γίνει η εμφάνιση του αντίστοιχου αριθμού κουμπιών στην οθόνη της συσκευής. Μετά γίνεται η εξόρυξη τους απο τον πίνακα με την χρήση των διακριτικών τους.

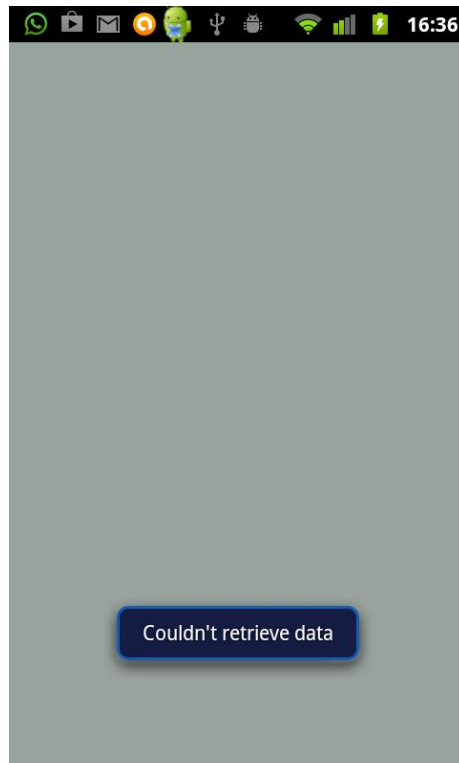
```
JSONObject cell = result.getJSONObject(i);

tmima_php = cell.getString("tmima_id");
mathima_php = cell.getString("onoma");
```

Η δημιουργία του αντικειμένου *cell* λαμβάνει τα δεδομένα ενός κελιού του πίνακα *result* με την μέθοδο *getJSONObject()*. Οι μεταβλητές *tmima\_php* και *mathima\_php* παίρνουν τις τιμές τους σε αλφαριθμητική μορφή απο το κελί του πίνακα σύμφωνα με τα διακριτικά *tmima\_id* και *onoma* αντίστοιχα. Αυτές οι μεταβλητές χρησιμοποιούνται στην εμφάνιση του κουμπιού στην οθόνη της συσκευής.

## Πτυχιακή εργασία του Θεόδωρου Κλιάφα

Όλος ο κώδικας που εξηγήθηκε παραπάνω και ανήκει στην εφαρμογή βρίσκεται σε ένα μπλόκ try-catch. Σε περίπτωση που η μέρα είναι Σάββατο ή Κυριακή αλλά και τις καθημερινές τις ώρες που είναι πριν τις 7:45 το πρωί ή μετά της 6 το απόγευμα το πρόγραμμα στον σέρβερ δεν θα μπορέσει να επιστρέψει αποτελέσματα. Η εφαρμογή δεν θα μπορέσει να αποκωδικοποιήσει το τίποτα και θα προκύψει εξαίρεση. Με την σύλληψη της γίνεται η εμφάνιση του μηνύματος “Couldn't retrieve data”.



Εικόνα 4.7 Εμφάνιση μηνύματος για μη αποδεκτές ώρες και μέρες

## Τοποθέτηση παρουσίας στο επιλεγμένο μάθημα

Ο φοιτητής έχει πλέον τα αποτελέσματα στην οθόνη της συσκευής του. Τα μαθήματα με τα τμήματα τους εμφανίζονται σε μορφή κουμπιών και μπορεί να επιλέξει ένα ώστε να τοποθετήσει ηλεκτρονικά την παρουσία του. Η διαδικασία είναι παρόμοια με τις προηγούμενες. Απαιτείται η κλήση του προγράμματος `android_presence.php` που εκτελείται στον σέρβερ. Επομένως το μόνο που αλλάζει στον κομμάτι της εφαρμογής είναι η διεύθυνση που καλούμε και οι πληροφορίες που προσθέτουμε στην κλήση για να γίνει η τοποθέτηση της παρουσίας.

Το κομμάτι του κώδικα που αναλαμβάνει αυτό το έργο εμπεριέχεται μέσα στη μέθοδο `putPresence(Stringtmima)` με παράμετρο το τμήμα που επιλέγει ο χρήστης. Η διεύθυνση εισάγεται στην μεταβλητή `httpPost` και με τον ίδιο τρόπο που δείξαμε στη προηγούμενη ενότητα, προσθέτουμε στην οντότητα της κλήσης τον κωδικό του τμήματος και το `username` του χρήστη. Γίνεται η κλήση στην διεύθυνση:

[http://erodios.it.teithe.gr/kourou\\_projects/QRcodes/android/android\\_presence.php](http://erodios.it.teithe.gr/kourou_projects/QRcodes/android/android_presence.php)

Στην πλευρά του σέρβερ, στο πρόγραμμα `android_presence.php`, οι πρώτες γραμμές του κώδικα εκτελούν την σύνδεση με την βάση δεδομένων. Στην συνέχεια οι μεταβλητές `$tmima_id` και `$username` παίρνουν την τιμή τους μέσω της μεθόδου POST με τα δεδομένα που προσθέσαμε στην κλήση από την εφαρμογή.

Αυτό το πρόγραμμα είναι αρμόδιο για την τοποθέτηση της παρουσίας. Τα δεδομένα που θα εισάγει στον πίνακα 'Παρουσία' πρέπει να είναι συμβατά με τα πεδία του πίνακα. Αν παρατηρήσουμε τον πίνακα θα καταλάβουμε ότι δεν χρειάζεται το `username` αλλά το `id` του χρήστη και αντί του κωδικού του τμήματος χρειάζεται τον κωδικό του πίνακα 'Διδάσκει' που είναι και το κύριο κλειδί του. Για να τοποθετήσουμε αυτά τα δεδομένα ακολουθούν οι αντίστοιχες αναζητήσεις.

```
$query = mysql_query("
SELECT id_foititis FROM foititis WHERE username='$username'");
```

```
$query = mysql_query("
SELECT kodikos FROM teaches WHERE tmima_id='$tmima_id'");
```

Στην πρώτη αναζήτηση λαμβάνουμε το αποτέλεσμα απο τον πίνακα 'Φοιτητής' σύμφωνα με τον περιορισμό του usernameπου δηλώθηκε στην εφαρμογή.

Στην δεύτερη αναζήτηση στον πίνακα-σύζευξη 'Διδάσκει', βρίσκουμε το αποτέλεσμα του κύριου κλειδιού του πίνακα με μόνο περιορισμό τον κωδικό του τμήματος που είναι ένα απο τα πεδία του.

Έχοντας αποθηκεύσει τα δεδομένα στις κατάλληλες μεταβλητές, το τελευταίο βήμα που απομένει είναι η εισαγωγή ενός στοιχείου με αυτά τα δεδομένα στον πίνακα 'Παρουσία'. Η εντολή εισαγωγής δεδομένων είναι η εξής:

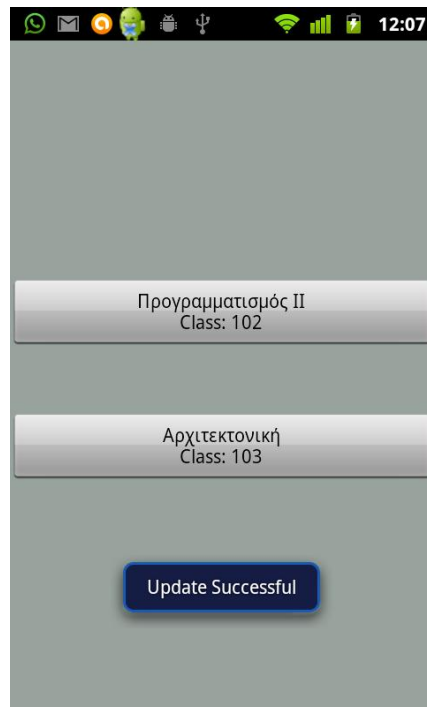
```
$query = mysql_query("
INSERT INTO parousies (id, id_teaches, imerominia, id_foititis) VALUES (NULL,
$id_teaches, NULL, $id_foititis)");
```

Στην MySQL όπως η εντολή SELECT χρησιμοποιείται για την αναζήτηση δεδομένων, έτσι και η εντολή INSERT χρησιμοποιείται για την εισαγωγή δεδομένων σε έναν πίνακα. Ακολουθεί το όνομα του πίνακα και μέσα στην πρώτη παρένθεση αναφέρονται τα ονόματα των πεδίων του. Η λέξη VALUES είναι κομμάτι της εντολής που περιγράφει τα περιεχόμενα της δεύτερης παρένθεσης. Τα δεδομένα προς εισαγωγή καταγράφονται ακριβώς στην ίδια σειρά με αυτή των πεδίων στην πρώτη παρένθεση ώστε να υπάρχει αντιστοιχία.

Εδώ τα μοναδικά δεδομένα που εισάγονται είναι αυτά που βρήκαμε στις προηγούμενες αναζητήσεις μας και είναι στην μορφή μεταβλητών της PHP.

Στα πεδία id και imerominia εισάγουμε τις τιμές NULL καθώς θα προστεθούν τα δεδομένα τους με αυτόματες λειτουργίες του σέρβερ. Το id που είναι και κύριο κλειδί του πίνακα έχει δηλωθεί ως auto-increment που σημαίνει ότι εισάγεται αυξανόμενο αυτόματα με κάθε εισαγωγή νέου στοιχείου. Το πεδίο imerominia που είναι τύπου Timestamp (DD/MM/YY, HH:MM) τοποθετείται αυτόματα με τα στοιχεία που έχει ο σέρβερ της παρούσα στιγμή.

Τα δεδομένα εισήχθησαν στην βάση, η παρουσία τοποθετήθηκε ηλεκτρονικά, αλλά πίσω στην εφαρμογή γίνεται πάλι έλεγχος για την ολοκλήρωση της διαδικασίας. Σε περίπτωση που τοποθετήθηκε επιτυχώς, γίνεται εμφάνιση του μηνύματος "Update Successful" που ενημερώνει τον χρήστη πως κατοχύρωσε την παρουσία του στο μάθημα. Αν κάτι πήγε στραβά, το κομμάτι του κώδικα βρίσκεται ανάμεσα σε μπλόκ try-catch, και η εξαίρεση που προκύπτει εκτελεί την εμφάνιση του μηνύματος "Update Unsuccessful".



Εικόνα 4.8 Εμφάνιση μηνύματος επιτυχούς εισαγωγής παρουσίας

### 4.3 Εμφάνιση και χειρισμός των στοιχείων της φόρμας

Στις προηγούμενες ενότητες αναφερθήκαμε περισσότερο στο κομμάτι του προγραμματισμού που υλοποιεί όλη την λειτουργία της εφαρμογής πίσω απο την οθόνη της συσκευής. Σημαντικό ρόλο όμως παίζουν και τα στοιχεία που τοποθετούνται στις φόρμες για την εμφάνιση της και για τον χειρισμό των γεγονόνων που προκαλεί ο χρήστης. Τα στοιχεία αυτά καταγράφονται σε αρχεία τύπου XML με κάθε τέτοιο αρχείο να αντιστοιχεί σε κάποια απο τις φόρμες που εμφανίζονται με την βοήθεια του κώδικα σε γλώσσα Java. Στην εφαρμογή QResence δεν έγινε χρήση πολλών στοιχείων καθώς οι επιθυμητές λειτουργίες της μας περιόρισαν στη χρήση κουμπιών και πεδίων κειμένου.

```
<Button  
android:id="@+id/buttonMathima2"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_gravity="center"  
android:text=""  
android:visibility="gone"/>
```

Στο παραπάνω κομμάτι κώδικα δηλώνουμε ένα κουμπί με id το `buttonMathima2` που θα χρησιμοποιήσουμε για την εμφάνιση του δεύτερου μαθήματος. Το πλάτος καθορίζεται στο πλάτος της φόρμας και το ύψος προσαρμόζεται στις ανάγκες του κειμένου προς εμφάνιση. Η εμφάνιση του στοιχείου θα γίνει στο κέντρο της φόρμας. Η προεπιλογή του κειμένου που θα εμφανίζεται είναι το κενό και της προβολής του στοιχείου που αντιστοιχεί στο αόρατο. Αυτές οι ιδιότητες μπορούν να αλλάξουν προγραμματιστικά μέσα στον κώδικα των κλάσεων Java.

```
Button subj2 = (Button) findViewById(R.id.buttonMathima2);
```

```
View b = findViewById(R.id.buttonMathima2);  
b.setVisibility(View.VISIBLE);  
subj2.setText(mathima_php2 + "\nClass: " + tmima_php2);
```

Οι παραπάνω γραμμές κώδικα εκτελούνται στην κλάση `AithousaActivity` και βρίσκονται σε διαφορετικές μεθόδους. Η πρώτη εντολή εκτελείται με το που γίνει η μετάβαση σε αυτή την κλάση και εισάγει στην μεταβλητή `subj2` τύπου `Button` το στοιχείο της φόρμας με κωδικό το `buttonMathima2`. Η μέθοδος που χρησιμοποιείται είναι η `findViewById()` που αναζητεί κάποιο στοιχείο με βάση το

id του και είναι μια απο τις πολλές μεθόδους που ενώνουν της κλάσεις της Java με τα αρχεία XML. Αφού έχει γίνει η δήλωση του κουμπιού και αφότου λάβουμε το αποτέλεσμα του δεύτερου προς εμφάνιση μαθήματος πρέπει να αλλάξουμε την ορατότητα του κουμπιού που εξ'αρχής είναι ορισμένη στο 'αόρατο'.

Έτσι δημιουργούμε μια μεταβλητή τύπου View στην οποία αντιστοιχούμε τον κωδικό του κουμπιού και με την χρήση της μεθόδου *setVisibility()* αλλάζουμε την ορατότητα του σε 'ορατό'. Στην τελευταία γραμμή εισάγουμε στο κουμπί με την μέθοδο *setText()* τις μεταβλητές που περιέχουν το όνομα του δεύτερου μαθήματος και του τμήματος που θέλουμε να εμφανίζονται σε αυτό.

#### 4.4 Τι είναι το Android manifest;

Ένα απο τα σημαντικότερα αρχεία στις εφαρμογές που δημιουργούνται για πλατφόρμες Android είναι το αρχείο AndroidManifest.xml. Αυτό το αρχείο παρέχει σημαντικές πληροφορίες της εφαρμογής που ενημερώνουν το σύστημα για τις απαιτούμενες λειτουργίες της πριν ακόμη εκτελέσει τον κώδικα της. Ουσιαστικά υλοποιεί έναν 'χάρτη' για την εφαρμογή που περιγράφει τις κλάσεις που ανήκουν στο πακέτο της. Εκτός αυτού, δίνει όνομα σε αυτό το πακέτο όπως επίσης και στις κλάσεις που για να εκτελεστούν πρέπει να δηλωθούν όπως φαίνεται παρακάτω.

```
<activityandroid:name=".QRcodeActivity">
<intent-filter>
    <actionandroid:name="android.intent.action.MAIN"/>
    <categoryandroid:name="android.intent.category.LAUNCHER"/>
</intent-filter>
</activity>

<activityandroid:name=".AithousaActivity"></activity>
```



Το αρχείο όπως διακρίνετε είναι τύπου XML οπότε και η συγγραφή του κώδικα γίνεται με χρήση ετικετών (tags). Τα ονόματα των κλάσεων δηλώνονται μέσα στις ιδιότητες του tag <activity>. Για τη κλάση που θα ξεκινήσει η εφαρμογή πρέπει να γίνει ιδιαίτερη δήλωση στο αρχείο αυτο. Στην δικιά μας περίπτωση δηλώνουμε ότι η QRcodeActivity θέλουμε να είναι η κύρια κλάση και αυτή που θα εκτελεστεί πρώτη με την εκκίνηση της εφαρμογής.

Επίσης σε αυτό το αρχείο δηλώνουμε το χαμηλότερο επίπεδο του Android API που απαιτείται για την λειτουργία της εφαρμογής. Αυτό συνεπάγεται στην έκδοση της πλατφόρμας Android που είναι εγκατεστημένη στη συσκευή. Ίσως από τις σημαντικότερες λεπτομέρειες στην υλοποίηση μιας εφαρμογής όμως είναι η δήλωση των δικαιωμάτων της στο Android manifest. Εδώ δηλώνουμε στο σύστημα τα εργαλεία της συσκευής που χρειάζεται να κάνει χρήση η εφαρμογή και απαιτούνται για την ορθή λειτουργία της.

```
<uses-sdkandroid:minSdkVersion="8"/>
```

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

```
<uses-
```

```
permissionandroid:name="android.permission.ACCESS_NETWORK_STATE"/>
```

Όπως φαίνεται στην πρώτη γραμμή δηλώνουμε το SDK της εφαρμογής που αντιστοιχεί στην έκδοση 2.2 της πλατφόρμας Android (Froyo). Πιο κάτω γίνεται η δήλωση των δικαιωμάτων της εφαρμογής. Η λειτουργία της ως διαδικτυακή απαιτεί την χρήση ιντερνετ και ο έλεγχος της συνδεσιμότητας της στο διαδίκτυο δηλώνεται ώστε να γίνει η εμφάνιση του προειδοποιητικού μηνύματος στον χρήστη.

## 4.5 Επίλογος

Σε αυτό το κεφάλαιο δείξαμε το λειτουργικό κομμάτι της εφαρμογής. Αναλύσαμε τον τρόπο που προκύπτουν τα δεδομένα και το πως συνεργάζονται για την τελική έκβαση του αποτελέσματος. Το σκεπτικό της εφαρμογής QResence είναι απλό. Για να το υλοποιήσουμε όμως, απαιτούνται πιο περίπλοκες διαδικασίες που εκτελούνται με άγνοια του χρήστη με σκοπό την γρήγορη και ποιοτική εξυπηρέτηση του. Η λειτουργία της είναι επικεντρωμένη στις αδυναμίες του χρήστη, παρέχοντας πολλούς εξειδικευμένους ελέγχους για πιθανά ατοπήματα του και ενημερώνοντας τον με τα κατάλληλα μηνύματα. Κάθε φοιτητής του τμήματος που είναι κάτοχος συσκευής με πλατφόρμα Android μπορεί να κάνει χρήση της εφαρμογής ώστε να τοποθετήσει την παρουσία του στο μάθημα που επιθυμεί. Πρέπει όμως να ενημερωθεί και για το ιστορικό των παρουσιών του, ενέργεια που μπορεί να κάνει μέσα απο την ιστοσελίδα 'Παρουσίες'. Στο επόμενο κεφάλαιο θα δούμε το προγραμματιστικό κομμάτι της ιστοσελίδας για να γίνει κατανοητό πως η επεξεργασία των δεδομένων μας οδηγεί στο τελικό αποτέλεσμα της εμφάνισης των παρουσιών που τοποθετήθηκαν με ηλεκτρονικό τρόπο.

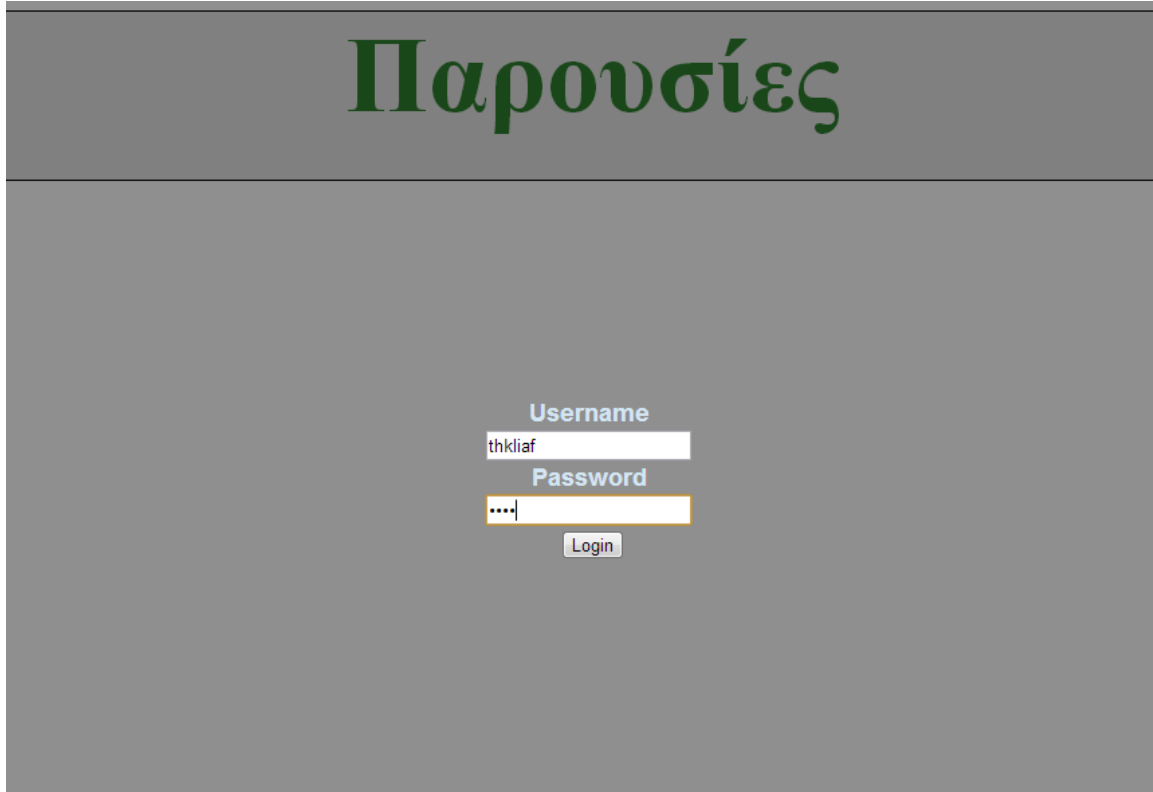
# Κεφάλαιο 5: Προγραμματισμός του site

## Παρουσίες

### 5.1 Εισαγωγή

Η ιστοσελίδα 'Παρουσίες' είναι ένα βοηθητικό κομμάτι της πτυχιακής για την παρουσίαση των αποτελεσμάτων της εφαρμογής QResence. Η λειτουργία και ο σχεδιασμός της είναι απλός καθώς η μόνη απαίτηση που υπάρχει είναι να εμφανίζει τις παρουσίες που τοποθέτησε ηλεκτρονικά ο χρήστης. Αποτελείται από 4 αρχεία τύπου PHP που αφορούν την λειτουργικότητα της ιστοσελίδας και ένα αρχείο τύπου CSS που αφορά τη εμφάνιση και την στοίχιση των HTML στοιχείων της σελίδας. Αυτά τα πέντε αρχεία είναι αποθηκευμένα στον σέρβερ και εκτελούνται με κλήση των κατάλληλων διευθύνσεων μέσω ενός προγράμματος περιήγησης ιστού (browser). Ο διαχωρισμός των απαιτήσεων της σελίδας γίνεται σε τρία κομμάτια. Πρώτον, θέλουμε να γίνεται έλεγχος της ταυτοποίησης των στοιχείων του χρήστη ώστε κάποιος να ενημερώνεται αποκλειστικά για τις κινήσεις του. Δεύτερον, ο χρήστης με τα αληθή στοιχεία που έδωσε μπορεί να ενημερωθεί για τις παρουσίες που τοποθέτησε ηλεκτρονικά ως αναφορά τα μαθήματα που παρακολούθησε. Τέλος, με την ολοκλήρωση της διαδικασίας ο χρήστης πρέπει να κλείσει την σελίδα όπου και γίνεται έξοδος από το σύστημα. Η ιστοσελίδα 'Παρουσίες' χρησιμοποιεί την ίδια βάση δεδομένων που είναι αποθηκευμένη στον σέρβερ του ερωδιού. Για την αναζήτηση και την εμφάνιση των δεδομένων γίνεται χρήση όλων των πινάκων της βάσης. Στις επόμενες ενότητες θα αναλύσουμε την εκτέλεση των βασικών εντολών κώδικα που οδηγούν στην ολοκληρωμένη λειτουργία της σελίδας.

## 5.2 Φόρμα εισαγωγής προσωπικών στοιχείων και έλεγχος τους

The image shows a login form on a grey background. At the top, the word "Παρουσίες" is written in a large, green, serif font. Below this, there are two input fields. The first is labeled "Username" and contains the text "thkliaf". The second is labeled "Password" and contains four dots. Below the password field is a "Login" button.

Εικόνα 5.1 Φόρμα εισαγωγής στοιχείων ταυτοποίησης χρήστη

Η φόρμα εισαγωγής στοιχείων του χρήστη είναι απλή στην εμφάνιση καθώς η μόνη λειτουργία της είναι ο έλεγχος των δεδομένων. Ο φοιτητής πρέπει να τοποθετήσει στα αντίστοιχα πεδία το username και password του και να πατήσει το κουμπί 'Login' για να μεταβεί στην επόμενη σελίδα. Ο έλεγχος των στοιχείων και η εμφάνιση της φόρμας γίνεται από την εκτέλεση εντολών στο αρχείο site\_login.php.

Στο αρχείο αυτό η πρώτη εντολή δημιουργούν την σύνδεση με την βάση δεδομένων για την διατύπωση των ερωτημάτων. Μετά την υλοποίηση της σύνδεσης υπάρχουν εντολές που κάνουν εκκίνηση μιας συνεδρίας (session) και

αποθηκεύουν σε αυτή δεδομένα που απαιτούνται στην λειτουργία των επόμενων σελίδων.

```
session_start();
```

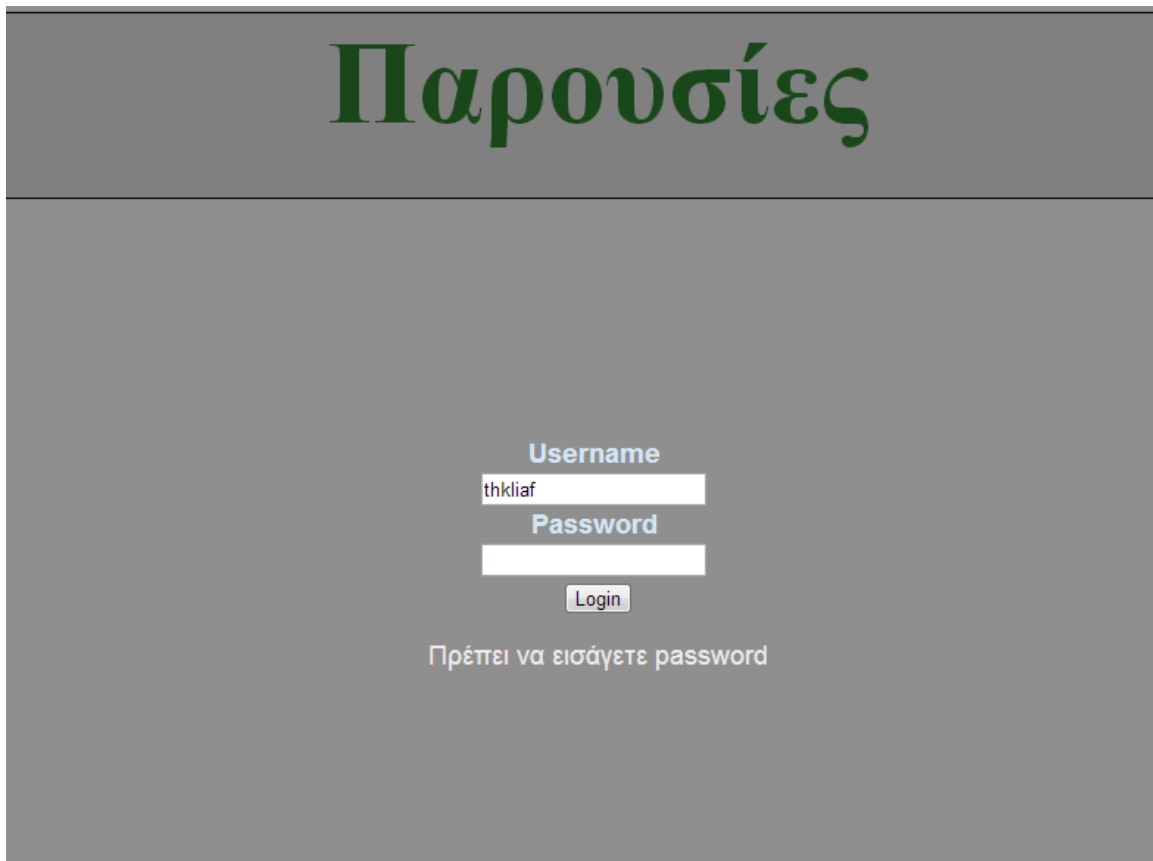
Η χρήση εντολών συνεδρίας είναι ιδιαίτερα σημαντικές στην λειτουργία μιας ιστοσελίδας καθώς επιτρέπουν στον χρήστη να πλοηγηθεί σε όλες της φόρμες της χωρίς να του ζητήτε κάθε φορά η εισαγωγή των στοιχείων του.

Για να οριστικοποιηθούν οι μεταβλητές session πρέπει να γίνει ο έλεγχος των στοιχείων εισόδου. Πρίν απο αυτό τον έλεγχο δύο μεταβλητές λαμβάνουν με την μέθοδο POST τα δεδομένα που τοποθέτησε ο χρήστης στα πεδία username και password της φόρμας. Στην συνέχεια αυτές οι μεταβλητές χρησιμοποιούνται για αναζήτηση δεδομένων στον πίνακα 'Φοιτητής'. Πιο συγκεκριμένα με χρήση του username του χρήστη λαμβάνουμε όλα τα δεδομένα των πεδίων του πίνακα που αντιστοιχούν στον φοιτητή. Σε περίπτωση που δεν επιστραφούν δεδομένα απο την βάση, σημαίνει ότι το username που δόθηκε είναι λανθασμένο και μετά απο έλεγχο γίνεται η εμφάνιση του μηνύματος 'Το username που εισάγατε δεν βρέθηκε'. Αν επιστραφούν δεδομένα που αντιστοιχούν στον χρήστη γίνεται σύγκριση του password που τοποθέτησε στην φόρμα με το password που μας επέστρεψε η βάση δεδομένων. Με το επιτυχές αποτέλεσμα της σύγκρισης αποθηκεύονται στις session μεταβλητές τα δεδομένα των πεδίων username και id του φοιτητή που βρήκαμε στην πρώτη αναζήτηση και οδηγούμαστε στην επόμενη φόρμα. Αν η σύγκριση βγάλει αναληθές αποτέλεσμα τότε γίνεται η εμφάνιση του μηνύματος 'Παρακαλώ εισάγεται το σωστό password'.

```
$_SESSION['username'] = $dbuser;  
$_SESSION['id_foititis'] = $dbid;  
header("Location: ./site_index.php");
```

Η εντολή `header()` της PHP κάνει την μετάβαση στο αρχείο που της δίνεται, στην περίπτωση μας στο `site_index.php` που είναι αποθηκευμένο στον ίδιο φάκελο.

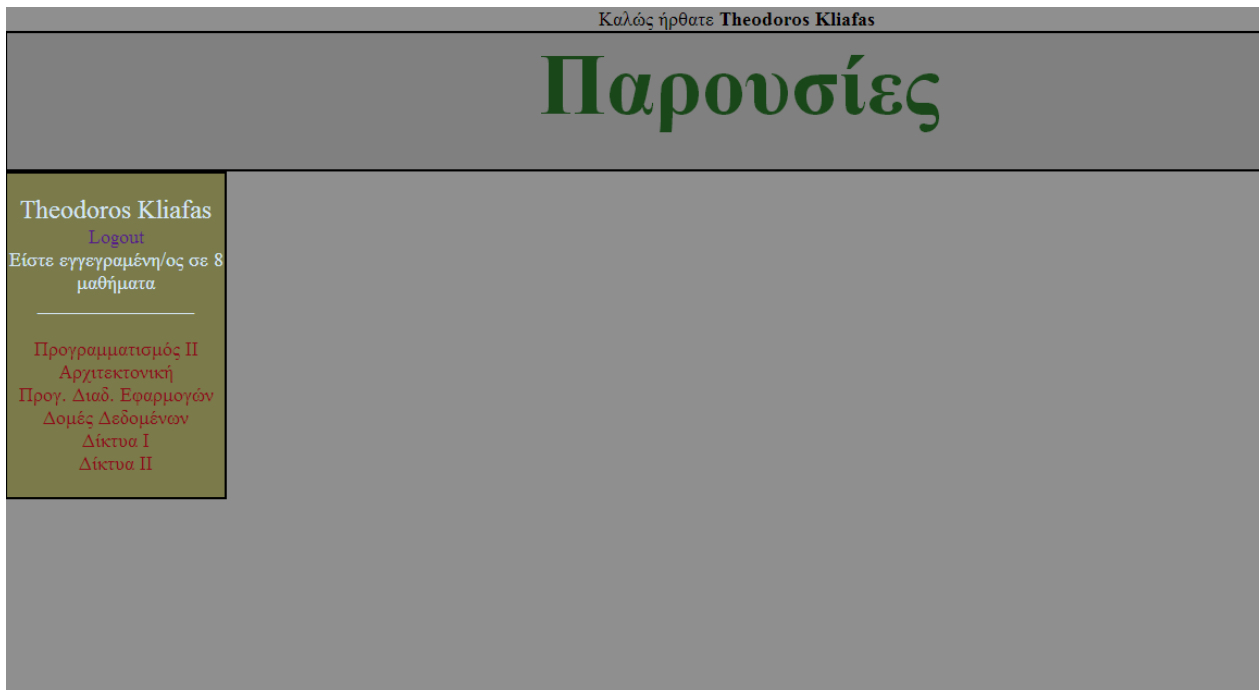
Υπάρχει βέβαια και τον ενδεχόμενο όπου ο χρήστης λόγω βιασύνης μπορεί να έχει παραλήψει την τοποθέτηση των στοιχείων ταυτοποίησης. Υπάρχει έλεγχος και για αυτές τις περιπτώσεις με τα αντίστοιχα προειδοποιητικά μηνύματα. Αν το πεδίο του `username` είναι κενό εμφανίζεται στην οθόνη του υπολογιστή το μήνυμα 'Πρέπει να εισάγεται username'. Αν έχει παραλήψει το `password`, γίνεται η εμφάνιση του μηνύματος 'Πρέπει να εισάγεται password'. Οι δύο αυτοί έλεγχοι γίνονται πρίν την αναζήτηση στην βάση δεδομένων για προφανής λόγους.



Εικόνα 5.2 Εμφάνιση μηνύματος λόγω παράληψης του πεδίου password

### 5.3 Εμφάνιση παρουσιών για κάθε μάθημα

Το βασικότερο κομμάτι της λειτουργίας του site εκτελείται στο αρχείο site\_index.php. Αφού γίνει επιτυχής αναγνώριση στοιχείων στην προηγούμενη φόρμα, γίνεται η μετάβαση σε αυτό το αρχείο που εμφανίζει στην σελίδα μαθήματα που παρακολουθεί ο συγκεκριμένος χρήστης.



Εικόνα 5.3 Εμφάνιση στοιχείων του χρήστη και μαθημάτων που παρακολουθεί

Με το που φορτωθεί το έγγραφο φορτώνονται οι session μεταβλητές username και userid που αρχικοποιήσαμε αμέσως μετά τον έλεγχο των στοιχείων. Πάνω σε αυτές τις μεταβλητές θα βασιστούν οι επόμενες αναζητήσεις μας.

Πρώτα γίνεται ο έλεγχος του αν έχουν λάβει κάποια τιμή ή αν είναι κενές. Σε περίπτωση που κάποιος χρήστης καλέσει το scriptindex.php χωρίς να έχει προηγηθεί έλεγχος των στοιχείων ταυτοποίησης του, οι μεταβλητές τύπου session δεν θα έχουν λάβει τιμές. Ο έλεγχος δεν θα επιτρέψει τη λειτουργία της σελίδας και θα εμφανίσει στην οθόνη το μήνυμα 'Πατήστε εδώ για να κάνετε Login' παραπέμποντας στην αρχική σελίδα ταυτοποίησης στοιχείων. Στην περίπτωση που έχουν αρχικοποιηθεί, γίνεται αναζήτηση του ονόματος του χρήστη σύμφωνα με το id του και το αποτέλεσμα εμφανίζεται στο ανώτερο κομμάτι της σελίδας με ένα σχετικό σχόλιο καλωσορίσματος.

Στο αριστερό κομμάτι της σελίδας εμφανίζεται το σύνολο των παρουσιών που έχει πραγματοποιήσει ο χρήστης και απο κάτω τα μαθήματα στα οποία ήταν παρών. Για τις εμφανίσεις των παραπάνω δεδομένων απαιτούνται δυο διαφορετικές αναζητήσεις. Για το σύνολο των παρουσιών του χρήστη χρησιμοποιούμε το id του για την αναζήτηση του πλήθους των οντοτήτων που είναι αποθηκευμένες στον πίνακα 'Παρουσία'. Το ερώτημα διατυπωμένο σε MySQL μας επιστρέφει τα αποτελέσματα τα οποία φιλτράρονται με την μέθοδο count() της PHP που μας δίνει το πλήθος τους. Για τον φοιτητή 'TheodorosKliafas' με id=1 εμφανίζεται στην σελίδα το μήνυμα 'Είστε εγγεγραμμένος σε 8 μαθήματα'.

Το πλήθος των μαθημάτων το χρησιμοποιούμε επίσης και για την εμφάνιση των ονομάτων τους. Ουσιαστικά δημιουργούμε ένα επαναληπτικό βρόγχο με όριο επαναλήψεων αυτόν τον αριθμό. Σε κάθε επανάληψη εμφανίζουμε το αποτέλεσμα της αναζήτησης δηλαδή το όνομα του κάθε μαθήματος με μόνη συνθήκη το id του χρήστη.

```
echo "<a href='site_index.php?subj=" . urlencode($mathima[$i]) . ">$mathima[$i]</a>";
```

Η παραπάνω εντολή βρίσκεται μέσα στον βρόγχο επανάληψης. Με την εντολή echo εμφανίζουμε στο έγγραφο ότι υπάρχει μέσα στα αποσιωπητικά. Το όνομα του μαθήματος που εμφανίζεται σε κάθε επανάληψη βρίσκεται αποθηκευμένο στον πίνακα \$mathima[] με το \$i να επιλέγει κάθε φορά νέο κελί του πίνακα. Για



να μπορεί ο χρήστης να επιλέξει ένα μάθημα και να δει τις παρουσίες που πραγματοποίησε σε αυτό, πρέπει το αποτέλεσμα να είναι σε μορφή υπερσυνδέσμου ώστε με το που το επιλέξει να επαναφορτωθεί η σελίδα με τα επιθυμητά δεδομένα. Για την δημιουργία του υπερσυνδέσμου χρησιμοποιούμε το tag `<a href=' '></a>` με την ιδιότητα href να λαμβάνει την τιμή της διεύθυνσης της σελίδας με πρόσθετη πληροφορία το μάθημα που επιλέγει ο χρήστης για να ενημερωθεί για τις παρουσίες του (πχ `href='site_index.php?subj=Δίκτυα+II'` ). Με το πάτημα του υπερσυνδέσμου γίνεται κλήση της διεύθυνσης που εμπεριέχεται σε αυτόν. Το αρχείο `site_index.php` ξανακαλείται με την διαφορά ότι αυτήν την φορά η διεύθυνση του περιέχει την πληροφορία του επιλεγμένου μαθήματος απο τον χρήστη. Με την μέθοδο GET λαμβάνουμε το επιλεγμένο μάθημα ως εξής:

```
if(isset($_GET['subj'])){  
    $sel_subject = $_GET['subj'];  
}else{  
    $sel_subject = NULL;  
}
```

Ο παραπάνω έλεγχος πραγματοποιείται στην αρχή του προγράμματος για να ληφθεί η μεταβλητή του επιλεγμένου μαθήματος `$sel_subject` σε περίπτωση που έχει αρχικοποιηθεί. Με την μέθοδο `isset()` της PHP ελέγχουμε εαν κάποια μεταβλητή έχει πάρει τιμή.

Με το αληθές αποτέλεσμα της μεθόδου δημιουργείται ένας πίνακας με τρεις στήλες. Η πρώτη στήλη αφορά το όνομα του μαθήματος, η δεύτερη την ημερομηνία τοποθέτησης κάθε παρουσίας και η τρίτη την ώρα που πραγματοποιήθηκε. Το σύνολο των γραμμών του πίνακα είναι δυναμικό καθώς εξαρτάται απο τις παρουσίες που τοποθέτησε ο φοιτητής. Για τον αριθμό των παρουσιών ανά μάθημα δημιουργείται ερώτημα στη βάση με περιορισμούς το όνομα του μαθήματος και το id του χρήστη.

Το ερώτημα αυτό μας επιστρέφει το πεδίο της ημερομηνίας κάθε παρουσίας και είναι τύπου Timestamp (DD/MM/YY, HH:MM). Ο τύπος Timestamp μας δίνει την ακριβή ημερομηνία και ώρα της παρουσίας διαχωρίζοντας τα δυο δεδομένα με ένα κόμμα. Το πλήθος των επιστρεφόμενων αποτελεσμάτων ορίζει τις γραμμές που θα έχει ο πίνακας. Το αποτέλεσμα τύπου Timestamp περιέχει και τα δυο δεδομένα όμως η ημερομηνία και η ώρα πρέπει να εμφανιστούν σε διαφορετική στήλη. Για τον διαχωρισμό των δεδομένων και την μορφοποίηση τους για την κατανοητή εμφάνιση τους στην σελίδα καλούνται δυο ξεχωριστοί μέθοδοι. Η μέθοδος `reverse_imerominia($imerominia)` λαμβάνει ως παράμετρο το δεδομένο τύπου Timestamp, διαχωρίζει την ημερομηνία από την ώρα και τοποθετεί κάθετο ανάμεσα στην μέρα, τον μήνα και το έτος έναντι της παύλας. Η μέθοδος `ora_imerominia($imerominia)` λαμβάνει ως παράμετρο το ίδιο δεδομένο όμως διαχωρίζει την ώρα από την ημερομηνία και την επιστρέφει.

```
$pieces=explode(",",$imerominia);  
$arr=$pieces[1];  
return $str;
```

Ο παραπάνω κώδικας βρίσκεται στην μέθοδο `ora_imerominia()`. Εδώ γίνεται κατά κύριο λόγο χρήση της μεθόδου `explode()` της PHP όπου η πρώτη εντολή τοποθετεί στον πίνακα `$pieces` τα δυο δεδομένα (ημερομηνία και ώρα) που διαχωρίζονται με το κόμμα. Η επιστρεφόμενη μεταβλητή `$arr` δέχεται το δεύτερο κελί του πίνακα `$pieces` που αντιστοιχεί στο δεδομένο της ώρας.

Έχοντας το όνομα του μαθήματος, τον αριθμό των παρουσιών στο συγκεκριμένο μάθημα και την ημερομηνία και ώρα διαχωρισμένες για κάθε παρουσία μπορούμε να εμφανίσουμε τον πίνακα σύμφωνα με αυτά τα δεδομένα.

The screenshot shows a web interface for a user named Theodoros Klifafas. At the top, it says 'Καλώς ήρθατε Theodoros Klifafas'. The main heading is 'Παρουσίες' (Presentations). On the left, there is a sidebar with the user's name, a 'Logout' link, and a message: 'Είστε εγγεγραμμένος σε 8 μαθήματα'. Below this, a list of courses is shown: 'Προγραμματισμός II', 'Αρχιτεκτονική', 'Προγ. Διαδ. Εφαρμογών', 'Δομές Δεδομένων', 'Δίκτυα I', and 'Δίκτυα II'. The main content area displays a table for the selected course, 'Προγραμματισμός II', with columns for 'Μάθημα', 'Ημερομηνία', and 'Ώρα'. The table contains two rows of data.

Μάθημα	Ημερομηνία	Ώρα
Προγραμματισμός II	05/07/2012	10:00
Προγραμματισμός II	12/07/2012	10:00

Εικόνα 5.4 Εμφάνιση πίνακα παρουσιών σύμφωνα με το μάθημα 'Προγραμματισμός II'

Ο χρήστης ενημερώνεται για τις ημερομηνίες και τις ώρες που τοποθέτησε τις παρουσίες του για κάποιο συγκεκριμένο μάθημα ενώ μπορεί να επιλέξει κάποιο άλλο από το μενού στα αριστερά. Με την επιλογή ενός άλλου μαθήματος ξανακαλείται η σελίδα αλλά αυτή τη φορά ο κεντρικός πίνακας περιέχει τα δεδομένα που αφορούν το συγκεκριμένο μάθημα.

## 5.4 Έξοδος απο το σύστημα

Με το που ολοκληρώσει ο χρήστης την ενημέρωση του, μπορεί να επιλέξει τον υπερσύνδεσμο 'Logout' για να κάνει έξοδο απο το σύστημα. Η ιδιότητα href του υπερσυνδέσμου έχει ως τιμή την διεύθυνση του αρχείου site\_logout.php. Με την κλήση αυτού του αρχείου γίνεται πάλι έλεγχος για το αν έχουν οριστεί οι session μεταβλητές. Σε περίπτωση που δεν έχουν οριστεί εμφανίζεται στην σελίδα το μήνυμα 'Πρέπει να συνδεθείτε για να εμφανιστεί η σελίδα. Login' με τον υπερσύνδεσμο Login να οδηγεί στην φόρμα εισαγωγής των στοιχείων ταυτοποίησης. Στην περίπτωση που όλα βαίνουν καλώς, εμφανίζεται το μήνυμα 'Επιτυχής αποσύνδεση.Login' και γίνεται λήξη της συνεδρίας με την εντολή session\_destroy() της PHP. Ο χρήστης έχει κάνει επιτυχή έξοδο απο το σύστημα έχοντας καταστρέψει τις κατάλληλες μεταβλητές που μπορεί να χρησιμοποιηθούν απο τρίτους για την υποκλοπή πληροφοριών που τον αφορούν.

## 5.5 Επίλογος

Στο κεφάλαιο αυτό δείξαμε την λειτουργία του βοηθητικής ιστοσελίδας 'Παρουσίες' που είναι απαραίτητη για την ενημέρωση του φοιτητή στο θέμα των παρουσιών. Η ιστοσελίδα συνεργάζεται απόλυτα με την εφαρμογή QResence καθώς μοιράζονται την ίδια βάση δεδομένων. Τα δεδομένα που αποθηκεύει στη βάση ο χρήστης μέσω της εφαρμογής έχει τη δυνατότητα να τα εμφανίσει στην ιστοσελίδα. Το σκεπτικό πίσω απο το site ήταν απλό και η λειτουργία του προσαρμόστηκε μόνο στην ενημέρωση του χρήστη. Η χρήση του είναι ιδιαίτερα εύκολη αφού με ελάχιστες κινήσεις οποιοσδήποτε μπορεί να εμφανίσει τα δεδομένα που τον ενδιαφέρουν. Η λειτουργία ελέγχου των στοιχείων ταυτοποίησης διαφυλάσσει την διαρροή των προσωπικών δεδομένων κάθε χρήστη. Με την δημιουργία της ιστοσελίδας ολοκληρώνεται η ανάγκη ηλεκτρονικής τοποθέτησης και εμφάνισης των παρουσιών για τη σχολή Πληροφορικής.

## Συμπεράσματα

Το βασικό κομμάτι της πτυχιακής ήταν η δημιουργία μιας εφαρμογής για την πλατφόρμα Android που σκοπό έχει την δυνατότητα τοποθέτησης παρουσίας για τα εργαστηριακά τμήματα της σχολής. Για την ολοκλήρωση αυτού του έργου απαιτούνται γνώσεις που αφορούν την ανάπτυξη εφαρμογών για συσκευές Android, την υλοποίηση προγραμμάτων που εκτελούνται στον σέρβερ και την μετατροπή των δεδομένων που αποθηκεύονται ή ανακτώνται απο την βάση δεδομένων. Οι βασικές γλώσσες προγραμματισμού που χρησιμοποιήθηκαν για την εφαρμογή είναι η Java και η XML, για τα προγράμματα στον σέρβερ η PHP, η MySQL και η JSON και για την ιστοσελίδα η HTML και η CSS. Οι περισσότερες απο αυτές της γλώσσες διδάσκονται στο τμήμα Πληροφορικής. Όμως στα πλαίσια της δημιουργίας μιας εφαρμογής για την πλατφόρμα Android η μέθοδος που χρησιμοποιείται είναι αρκετά διαφορετική απο τον τρόπο που μάθαμε στην σχολή. Εξαιρετη πηγή πληροφόρησης είναι το διαδίκτυο καθώς κάθε σημαντικό βήμα για την επίλυση αποριών γίνεται με τις κατάλληλες αναζητήσεις σε ιστοτόπους που εξειδικεύονται σε αυτού του είδους τα θέματα. Επίσης τα περιβάλλοντα ανάπτυξης του κώδικα βοήθησαν πολύ καθώς με την χρήση συγκεκριμένων εργαλείων γίνεται πολύ πιο εύκολη η δουλειά του προγραμματιστή.

Η εφαρμογή QRsence είναι μια αρχή για την υλοποίηση ενός προγράμματος ηλεκτρονικής καταγραφής παρουσιών που είναι απαραίτητη για το τμήμα Πληροφορικής. Ο συνδυασμός των QR Codes σε αυτό το έργο προσδίδει μεγάλη λειτουργικότητα και ευκολία καθώς με την γρήγορη αποκωδικοποίηση τους λαμβάνουμε μεγάλο αριθμό δεδομένων που διαφορετικά θα έπρεπε να εισάγει ο χρήστης στο πρόγραμμα. Οι περισσότερες συσκευές δεν έχουν δέκτη για λήψη RFID δεδομένων και το πλεονέκτημα των QR Codes είναι οτι πολύ απλά μπορούν να τυπωθούν σε ένα χαρτί και να σκαναριστούν απο την κάμερα οποιουδήποτε smartphone. Η απόφαση να αναπτυχθεί σε πλατφόρμα Android

είναι δικαιολογημένη καθώς τα τελευταία χρόνια αυξάνονται ραγδαία οι χρήστες που χρησιμοποιούν έξυπνα κινητά με αυτό το λειτουργικό σύστημα. Αυτό όμως συνεπάγεται ότι θα υπάρχει και ένας σχετικά μεγάλος αριθμός φοιτητών που δε θα μπορεί να κάνει χρήση της καθώς είναι κάτοχοι συσκευών που λειτουργούν σε διαφορετικές πλατφόρμες (πχ iOS, BlackBerry).

Η εφαρμογή μπορεί να φανεί ιδιαίτερα χρήσιμη σε αρχικό στάδιο για την ένταξη των καθηγητών και των φοιτητών στο ηλεκτρονικό παρουσιολόγιο, μια ιδέα που θα λύσει πολλά απο τα τωρινά προβλήματα που συναντάμε στην χρήση των πρόχειρων εγγράφων που τοποθετούν οι φοιτητές την παρουσία τους. Με το ηλεκτρονικό παρουσιολόγιο εξασφαλίζουμε ασφάλεια αποθήκευσης των επιθυμητών δεδομένων με πληρέστερη περιγραφή για την ώρα, την μέρα και το μάθημα. Με τη κατάλληλη μελέτη για την ασφάλεια αυτού του εγχειρίματος διαφυλάσσουμε την χρήση του μέσω διαδικτύου για την συνεχή ενημέρωση των καθηγητων και των φοιτητών και για την δημιουργία ιστορικού αρχείου για την επίλυση κάθε πιθανού προβλήματος.

Η εφαρμογή μπορεί να λειτουργήσει πειραματικά στη σχολή στα πλαίσια ενός εξαμήνου για να γίνουν αντιληπτοί οι ακριβείς λόγοι χρήσης της και οι αδυναμίες της. Στην συνέχεια μπορεί να γίνει ανάπτυξη της αντίστοιχης εφαρμογής στις επικρατέστερες πλατφόρμες έξυπνων κινητών. Στις επόμενες εκδόσεις της εφαρμογής μπορεί να γίνεται αρχικός έλεγχος των στοιχείων του χρήστη απο κάποιον administrator στο τμήμα για να συνδέεται η κάθε συσκευή με τα στοιχεία του κατόχου της. Έτσι θα είναι δύσκολο κάποιος φοιτητής να τοποθετεί παρουσίες μέσω του κινητού του για λογαριασμό τρίτων καθώς η φυσική διεύθυνση της συσκευής του θα είναι άμεσα συνδεδεμένη με αυτόν.

Σε αυτή τη πτυχιακή για λόγους ασφάλειας χρησιμοποιήθηκε διαφορετική βάση δεδομένων απο την κεντρική βάση δεδομένων της σχολής. Για την σωστή λειτουργία της εφαρμογής θα πρέπει οι δυο βάσεις δεδομένων να επικοινωνούν μεταξύ τους ή ακόμη καλύτερα η εφαρμογή να επικοινωνεί απευθείας με την κεντρική βάση. Έτσι η εφαρμογή θα είναι συνδεδεμένη με το πρόγραμμα

μαθημάτων κάθε φοιτητή και θα γνωρίζει τυχόν αλλαγές στο πρόγραμμα όπως αναπληρώσεις ή επιμήκυνση του εξαμήνου.

Ίσως η πιο σωστή λύση να είναι ο προγραμματισμός μιας κεντρικής εφαρμογής που θα είναι εγκατεστημένη στους υπολογιστές των εργαστηρίων ώστε να αποφύγουμε την μή συμβατότητα με τις διαφορετικές συσκευές που έχει κάθε χρήστης. Για την λειτουργία αυτού του προγράμματος θα πρέπει να γίνει σχεδίαση σε βάθος για τις ανάγκες της σχολής. Τα δεδομένα που θα χρησιμοποιεί θα πρέπει να είναι αποθηκευμένα στην κεντρική βάση δεδομένων της σχολής και η λειτουργία της να βασίζεται στον κεντρικό σέρβερ. Αυτό θα οφελήσει στην άμεση επικοινωνία της εφαρμογής με το πρόγραμμα των μαθημάτων και των αιθουσών που ανανεώνεται κάθε νέο εξάμηνο και με πολλές άλλες πληροφορίες που βρίσκονται αποθηκευμένες εκεί. Η εγκατάσταση του προγράμματος στους υπολογιστές τους τμήματος θα εξασφαλίσει μεγαλύτερη ασφάλεια στο σύστημα. Η υποκλοπή στοιχείων και η μετατροπή τους θα είναι μια δύσκολη διαδικασία αμα γίνει σωστός προγραμματισμός της ασφάλειας που αρμόζει στα προγράμματα της σχολής. Το επίπεδο ασφάλειας σε μια εφαρμογή που λειτουργεί σε έξυπνες συσκευές θα ήταν πιθανώς χαμηλότερο. Επίσης περιορίζει την τοποθέτηση παρουσιών στον χώρο της αίθουσας που είναι ένα σημαντικό θέμα καθώς με την εφαρμογή στα κινητά μπορεί να τοποθετηθεί παρουσία χωρίς να παρευρίσκεται ο φοιτητής στο μάθημα.

Όποια και αν είναι η συνέχεια του εγχειρήματος πολλά θετικά θα προκύψουν απο την δημιουργία ηλεκτρονικού παρουσιολογίου. Είτε μέσω εφαρμογών σε κινητά είτε μέσω προγραμμάτων στους υπολογιστές των εργαστηρίων τα δεδομένα που θα αποθηκεύονται στην βάση θα είναι διαθέσιμα για έλεγχο απο τους καθηγητές και απο τους φοιτητές για κάθε πιθανό ενδεχόμενο. Επίσης θα μπορεί να ξεκινήσει έρευνα για τα μαθήματα που παρατηρείται μεγαλύτερη καθυστέρηση στην τοποθέτηση παρουσίας, για τα μαθήματα που πραγματοποιείται ο μεγαλύτερος αριθμός παρουσιών και διάφορα άλλα θέματα προς αναζήτηση. Οι πληροφορίες που θα προκύψουν απο αυτά τα δεδομένα θα βοηθήσουν στην



βελτιστοποίηση των προβληματικών κομματιών της σχολής τόσο στην εκπαιδευτική διαδικασία όσο και στα διοικητικά θέματα.

Η εφαρμογή QResence είναι η αρχή για την δημιουργία του ηλεκτρονικού παρουσιολογίου. Για να θεωρηθεί πετυχημένο το εν λόγω εγχείρημα χρησιμοποιήθηκαν οι πιο πολυχρησιμοποιημένες και προσιτές τεχνολογίες της εποχής μας, η πλατφόρμα Android και τα QR Codes. Το αποτέλεσμα ελπίζω να δώσει έναυσμα ώστε να συνεχίσει η έρευνα πάνω σε αυτήν την ιδέα που θα λύσει πολλά απο τα προβλήματα της σχολής.

# Βιβλιογραφία

- 1) P. Deitel, H. Deitel, A. Deitel, M. Morgano (2012). Android για προγραμματιστές  
<http://www.deitel.com/Books/Android/AndroidforProgrammers/tabid/3606/Default.aspx>
- 2) Shane Conder, Lauren Darcey (2011). Android Application Development in 24 Hours  
<http://www.pdfbook.co.ke/details.php?title=Sams%20Teach%20Yourself%20Android%20Application%20Development%20in%2024%20Hours&author=Shane%20Conder,%20Lauren%20Darcey&category=Computers&eid=333&type=Book>
- 3) Πτυχιακή Εργασία του Σέργιου Γαβριηλίδη (2012)
- 4) Android Developers: What is Android?  
<http://developer.android.com/about/index.html>
- 5) Android Developers: Managing Virtual Devices  
<http://developer.android.com/tools/devices/index.html>
- 6) Working with XML on Android  
<http://www.ibm.com/developerworks/opensource/library/x-android/>
- 7) Android Developers: Debugging  
<http://developer.android.com/tools/debugging/index.html>

## Αναφορές

- 1) <http://www.eclipse.org/>
- 2) <http://www.mpsoftware.dk/phpdesigner.php>
- 3) <http://light.mindtalk.com/post/4f118fd5f7b7303a3d001787>
- 4) [http://en.wikipedia.org/wiki/QR\\_code](http://en.wikipedia.org/wiki/QR_code)
- 5) <http://php.net/>
- 6) <http://stackoverflow.com/>
- 7) <http://mysql.net/>
- 8) <http://www.youtube.com/watch?v=Xpkbu2GrJpE>
- 9) [http://www.youtube.com/watch?v=5RHtKIo\\_KDI&list=SPE08A97D36D5A255F](http://www.youtube.com/watch?v=5RHtKIo_KDI&list=SPE08A97D36D5A255F)
- 10) <http://www.youtube.com/watch?v=-c4jERnki9w>