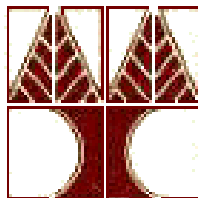


**Ατομική διπλωματική εργασία**

**ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΠΤΥΞΗ ANDROID ΕΦΑΡΜΟΓΗΣ  
ΓΙΑ ΑΛΛΗΛΕΠΙΔΡΑΣΗ ΜΕΤΑΞΥ ΦΟΙΤΗΤΩΝ ΚΑΙ  
ΕΠΙΒΛΕΠΟΝΤΩΝ ΚΑΘΗΓΗΤΩΝ**

Φειδίας Ιωαννίδης

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2013

# **ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ**

## **ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

### **Σχεδίαση και ανάπτυξη Android εφαρμογής για αλληλεπίδραση μεταξύ φοιτητών και επιβλεπόντων καθηγητών**

**Φειδίας Ιωαννίδης**

Επιβλέπων Καθηγήτρια

Γεωργία Καπιτσάκη

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου.

Μάιος 2013

# Ευχαριστίες

Θα ήθελα να εκφράσω τις ευχαριστίες μου στην επιβλέπων καθηγήτρια μου Δρ. Γεωργία Καπιτσάκη, για την βοήθεια της, την υποστήριξη της και την πολύ καλή συνεργασία που είχαμε καθ' όλο το χρονικό διάστημα εκπόνησης της διπλωματικής αυτής εργασίας. Οι συμβουλές της, οδηγίες αλλά και κατευθύνσεις της, συνέβαλαν στο μέγιστο, για την επιτυχή κατά την άποψη μου ολοκλήρωση της εργασίας αυτής.

Επίσης θα ήθελα να ευχαριστήσω όλους τους καθηγητές του τμήματος και τους συμφοιτητές μου που προσφέρθηκαν να δοκιμάσουν την εφαρμογή και όσους συνέβαλαν στον καθορισμό των απαιτήσεων. Καθώς επίσης και τον κύριο Νέαρχο Πασπαλλή για την βοήθειά του.

Τέλος, θα ήθελα να εκφράσω ένα μεγάλο ευχαριστώ στην οικογένεια μου και τους φίλους μου, που με στήριξαν ηθικά και ψυχολογικά σε ολόκληρη την πορεία μου κατά τη διάρκεια φοίτησης μου στο Πανεπιστήμιο Κύπρου αλλά και κατά τη δύσκολη αυτή περίοδο ολοκλήρωσης της διπλωματικής μου εργασίας.

# Περίληψη

Στόχος της παρούσας διπλωματικής εργασίας είναι να χρησιμοποιήσει της δυνατότητες που προσφέρουν οι έξυπνες κινητές συσκευές και συγκεκριμένα το λειτουργικό σύστημα Android, για την ανάπτυξη μιας εφαρμογής για κινητές συσκευές. Η εφαρμογή αυτή σκοπό έχει να βελτιώσει την επικοινωνία και την αλληλεπίδραση μεταξύ των καθηγητών και των φοιτητών του τμήματος. Η εφαρμογή αυτή διαθέτει διάφορες λειτουργίες που εξυπηρετούν τον στόχο αυτό, όπως ανταλλαγή άμεσων μηνυμάτων, οργάνωση συναντήσεων και πολλές άλλες που θα αναλυθούν εκτενώς στην συνέχεια της εργασίας. Πέρα από την εφαρμογή για Android δημιουργήθηκαν επιπλέον μια διαδικτυακή εφαρμογή για δημιουργία και επεξεργασία λογαριασμών καθηγητών και φοιτητών, μια βάση δεδομένων για μόνιμη αποθήκευση δεδομένων που αφορούν τους χρήστες και ενέργειές τους και μια υπηρεσία διαδικτύου η οποία λειτουργεί σαν μεσολαβητής μεταξύ της android εφαρμογής και της βάσης δεδομένων.

Μετά την ολοκλήρωση της υλοποίησης του συστήματος, η εφαρμογή δόθηκε σε καθηγητές και φοιτητές του τμήματος για εκτενείς δοκιμές. Η δοκιμαστική χρήση από τους καθηγητές και φοιτητές του τμήματος σε πραγματικές συνθήκες χρήσης βοήθησε στην διόρθωση και βελτίωση της εφαρμογής. Τα πρώτα μηνύματα ήταν ενθαρρυντικά και λήφθηκαν χρήσιμα σχόλια.

Στην εργασία που ακολουθεί αρχικά υπάρχει μια σύντομη εισαγωγή όπου γίνεται μια πρώτη περιγραφή της εφαρμογής και του λειτουργικού συστήματος Android. Στο επόμενο κεφάλαιο υπάρχει η καταγραφή και ανάλυση των απαιτήσεων στις οποίες βασίζονται και οι λειτουργίες της εφαρμογής. Στο τρίτο κεφάλαιο αναλύονται οι τεχνολογίες και τα πρωτόκολλα που χρησιμοποιήθηκαν κατά την ανάπτυξη του συστήματος. Ακολούθως αναλύεται λεπτομερώς η διαδικασία υλοποίησης όλων των κομματιών του συστήματος. Στο πέμπτο κεφάλαιο περιγράφονται τα αποτελέσματα των δοκιμών και στο τέλος στο έκτο κεφάλαιο υπάρχουν τα συμπεράσματα και μελλοντικές εργασίες που μπορούν να γίνουν στην εφαρμογή.

# Περιεχόμενα

|                                                                        |           |
|------------------------------------------------------------------------|-----------|
| <b>Κεφάλαιο 1. Εισαγωγή .....</b>                                      | <b>1</b>  |
| 1.1 Σκοπός Εργασίας .....                                              | 1         |
| 1.2 Λειτουργικό σύστημα Android.....                                   | 2         |
| 1.3 Περίγραμμα Εργασίας.....                                           | 3         |
| <b>Κεφάλαιο 2. Λειτουργίες και Αρχιτεκτονική εφαρμογής .....</b>       | <b>5</b>  |
| 2.1 Καταγραφή απαιτήσεων.....                                          | 5         |
| 2.2 Ανάλυση απαιτήσεων και περιγραφή βάσης δεδομένων.....              | 19        |
| 2.3 Επικοινωνία μεταξύ εφαρμογής, web service και βάσης δεδομένων..... | 28        |
| <b>Κεφάλαιο 3. Τεχνολογίες/Πρωτόκολλα Ανάπτυξης Εφαρμογής .....</b>    | <b>31</b> |
| 3.1 C# / ASP.NET .....                                                 | 31        |
| 3.2 WCF RESTful web service .....                                      | 33        |
| 3.3 SQL server 2008 .....                                              | 34        |
| 3.4 Really Simple Context Middleware (RSCM).....                       | 35        |
| 3.5 Extensible Messaging and Presence Protocol (XMPP) .....            | 36        |
| <b>Κεφάλαιο 4. Υλοποίηση .....</b>                                     | <b>38</b> |
| 4.1 Υλοποίηση web service.....                                         | 38        |
| 4.2 Υλοποίηση βάσης δεδομένων .....                                    | 40        |
| 4.3 Υλοποίηση εφαρμογής.....                                           | 42        |
| 4.3.1 Γραφική διεπαφή χρήστη .....                                     | 43        |
| 4.3.2 Υλοποίηση λειτουργιών .....                                      | 48        |
| 4.4 Υλοποίηση διαδικτυακής εφαρμογής .....                             | 57        |
| 4.4.1 Αρχική σελίδα .....                                              | 57        |
| 4.4.2 Εγγραφή καθηγητή .....                                           | 58        |
| 4.4.3 Εγγραφή φοιτητή .....                                            | 60        |
| 4.4.4 Επεξεργασία λογαριασμού .....                                    | 61        |
| 4.4.5 Επεξεργασία από τον διαχειριστή .....                            | 64        |

|                                                           |           |
|-----------------------------------------------------------|-----------|
| <b>Κεφάλαιο 5. Δοκιμή και αξιολόγηση .....</b>            | <b>68</b> |
| 5.1 Παρουσίαση αποτελεσμάτων δοκιμών .....                | 68        |
| 5.2 Ανάλυση αποτελεσμάτων.....                            | 70        |
| <b>Κεφάλαιο 6. Συμπεράσματα – Μελλοντική Εργασία.....</b> | <b>73</b> |
| 6.1 Συμπεράσματα.....                                     | 73        |
| 6.2 Μελλοντική Εργασία .....                              | 75        |
| <b>Βιβλιογραφία .....</b>                                 | <b>77</b> |

# Κεφάλαιο 1. Εισαγωγή

---

|                                      |   |
|--------------------------------------|---|
| 1.1 Σκοπός Εργασίας .....            | 1 |
| 1.2 Λειτουργικό σύστημα Android..... | 2 |
| 1.3 Περίγραμμα Εργασίας.....         | 3 |

---

Στο κεφάλαιο αυτό γίνεται μια εισαγωγή στην παρούσα διπλωματική εργασία. Αρχικά γίνεται μια περιγραφή του στόχου της εργασίας και ακολούθως γίνεται αναφορά στο λειτουργικό σύστημα Android και στο τι θα ακολουθήσει στα επόμενα κεφάλαια.

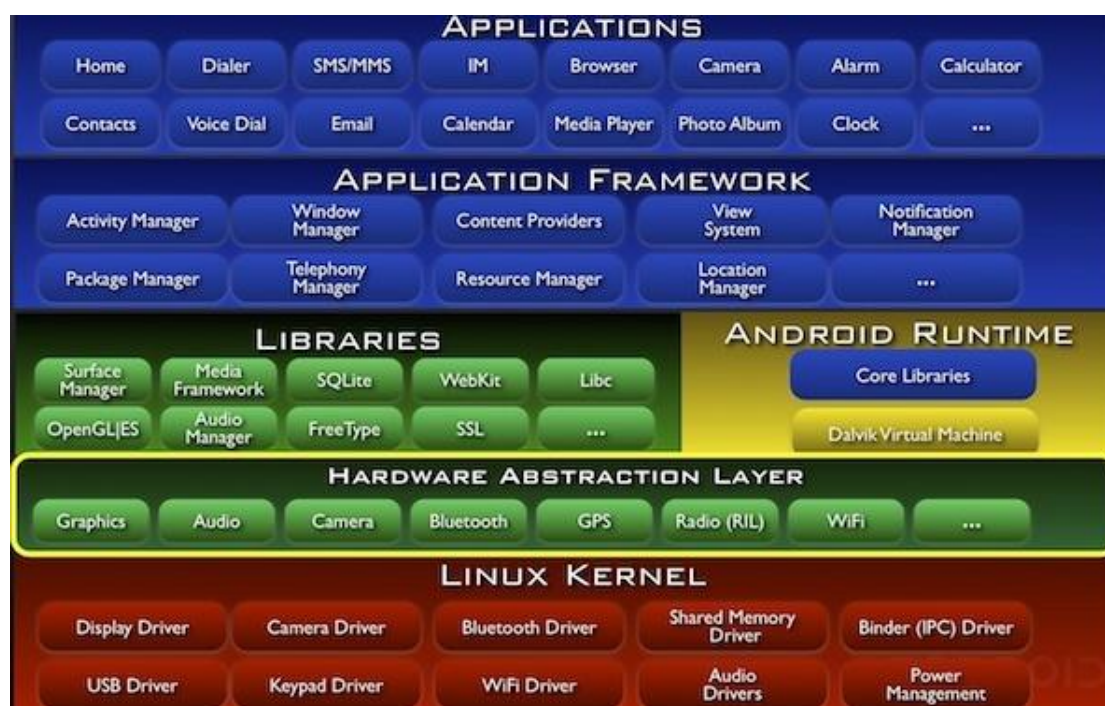
## 1.1 Σκοπός εργασίας

Σκοπός αυτής της διπλωματικής εργασίας είναι η ανάπτυξη μιας εφαρμογής για κινητές συσκευές με λειτουργικό σύστημα Android μέσω της οποίας μπορούν να επικοινωνούν και να αλληλεπιδρούν καθηγητές και φοιτητές. Στόχος της εφαρμογής αποτελεί η βελτίωση της επικοινωνίας μεταξύ Επιβλεπόντων Διπλωματικών/Ακαδημαϊκών συμβούλων και φοιτητών του τμήματος Πληροφορικής με σκοπό την εκπόνηση διπλωματικής εργασίας είτε για την παρακολούθηση της ακαδημαϊκής πορείας κάποιου φοιτητή αντίστοιχα. Μέσω αυτής της εφαρμογής η επικοινωνία μεταξύ καθηγητών γίνεται πιο εύκολη και πιο άμεση, ενώ βελτιώνεται και η παρακολούθηση της πορείας του φοιτητή από τον καθηγητή.

Επιπλέον, μια εφαρμογή που θα είναι εγκατεστημένη σε μια κινητή συσκευή επιτρέπει την χρήση της σε οποιοδήποτε χώρο και οποιαδήποτε στιγμή. Μέσω των αισθητήρων της συσκευής μπορεί να χρησιμοποιηθεί η γνώση συγκεκριμένου που μπορεί να χρησιμοποιηθεί για διάφορες λειτουργίες. Με την χρήση αυτών των χαρακτηριστικών των κινητών συσκευών η επικοινωνία μπορεί να γίνει οπουδήποτε και οποιαδήποτε στιγμή.

## 1.2 Λειτουργικό σύστημα Android

Το Android είναι ένα λειτουργικό σύστημα που τρέχει κυρίως σε κινητές συσκευές με οθόνη αφής όπως κινητά τηλέφωνα και tablets. Το Android είναι λειτουργικό σύστημα ανοιχτού κώδικα που ανήκει στην Google [20]. Λόγω του ότι είναι λογισμικό ανοιχτού κώδικα μπορεί να τροποποιηθεί από οποιοδήποτε όπως για παράδειγμα κατασκευαστές συσκευών, εταιρίες τηλεπικοινωνιών και διάφορους developers. Το λειτουργικό είναι σχεδιασμένο χρησιμοποιώντας της γλώσσες C/C++ και Java, βασίζεται στο Linux και χρησιμοποιεί τον Monolithic kernel (μονολιθικό πυρήνα) που είναι μια τροποποιημένη έκδοση του πυρήνα του λειτουργικού συστήματος Linux (Linux kernel). Ένα επιπλέον πλεονέκτημα του Android είναι πως τρέχει και στις δύο κύριες αρχιτεκτονικές ARM και x86 καθώς επίσης και σε MIPS αρχιτεκτονικές.



Αρχιτεκτονική Android

Οι εφαρμογές που σχεδιάζονται για το λειτουργικό Android είναι γραμμένες στη γλώσσα προγραμματισμού Java. Οι εφαρμογές ακολούθως μεταγλωττίζονται και τρέχουν στην εικονική μηχανή Dalvik, η οποία είναι μια εξειδικευμένη υλοποίηση εικονικής μηχανής, σχεδιασμένη για χρήση από κινητές συσκευές.

Το λειτουργικό σύστημα Android υποστηρίζει τεχνολογίες συνδεσιμότητας όπως GSM, Bluetooth, Wi-Fi, GPS, CDMA, HSDPA, GPRS, EDGE, LTE κτλπ. Επίσης οι περισσότερες συσκευές που χρησιμοποιούν το Android διαθέτουν διάφορους σένσορες όπως accelerometer, gyroscope, proximity, barometer κτλπ.

Τέλος σύμφωνα με τις τελευταίες μελέτες το μεγαλύτερο ποσοστό κινητών συσκευών αυτή τη στιγμή χρησιμοποιεί το λειτουργικό σύστημα Android [21]. Το ποσοστό συσκευών που τρέχουν το λειτουργικό Android σύμφωνα με πρόσφατες μελέτες ξεπερνά το 70%, περισσότερα δηλαδή από 'τι όλα τα άλλα λειτουργικά συστήματα για κινητές συσκευές μαζί, όπως iOS, Windows phone, Blackberry OS κλπ.

Όλα τα προαναφερθέντα χαρακτηριστικά του Android αλλά και η ευκολία χρήσης και ανάπτυξης στο συγκεκριμένο λειτουργικού ήταν σημαντικοί παράγοντες που οδήγησαν στην επιλογή του συγκεκριμένου λειτουργικού συστήματος.

### **1.3 Περίγραμμα εργασίας**

Τελειώνοντας το κεφάλαιο 1, στο οποίο παρουσιάστηκε ο σκοπός της διπλωματικής εργασίας και αναφέρθηκαν κάποια γενικά χαρακτηριστικά και ιδιότητες του λειτουργικού συστήματος Android, ακολουθεί μια σύντομη περιγραφή για το τι θα περιέχουν τα επόμενα κεφάλαια.

Στο κεφάλαιο 2 παρουσιάζονται οι λειτουργίες και η αρχιτεκτονική της εφαρμογής. Όσο αφορά τις λειτουργίες θα παρουσιαστούν έγγραφα για καταγραφή και ανάλυση απαιτήσεων, ενώ όσων αφορά την αρχιτεκτονική θα παρουσιαστεί ο τρόπος επικοινωνίας μεταξύ της εφαρμογής στην κινητή συσκευή, του web service και της βάσης δεδομένων.

Στο κεφάλαιο 3 αναλύονται οι τεχνολογίες και τα πρωτόκολλα τα οποία χρησιμοποιήθηκαν κατά την ανάπτυξη της εφαρμογής. Αυτές είναι, C#/ASP .NET, Windows Communication Foundation (WCF) Representational State Transfer (RESTful) (υπηρεσία διαδικτύου), SQL server 2008, Really Simple Context Middleware (RSCM) και Extensible Messaging and Presence Protocol (XMPP).

Στο κεφάλαιο 4 παρουσιάζονται λεπτομερώς ο τρόπος και οι τεχνικές που χρησιμοποιήθηκαν για την υλοποίηση όλων των μερών που αποτελούν την εργασία (εφαρμογή, διαδικτυακή εφαρμογή, υπηρεσία διαδικτύου, βάση δεδομένων).

Το κεφάλαιο 5 περιέχει τα αποτελέσματα από τις δοκιμές της εφαρμογής καθώς επίσης και ανάλυση τους.

Τέλος στο κεφάλαιο 6 υπάρχουν τα συμπεράσματα και η μελλοντική εργασία που μπορεί να γίνει σε σχέση με αυτήν την εργασία.

## Κεφάλαιο 2. Λειτουργίες και Αρχιτεκτονική εφαρμογής

---

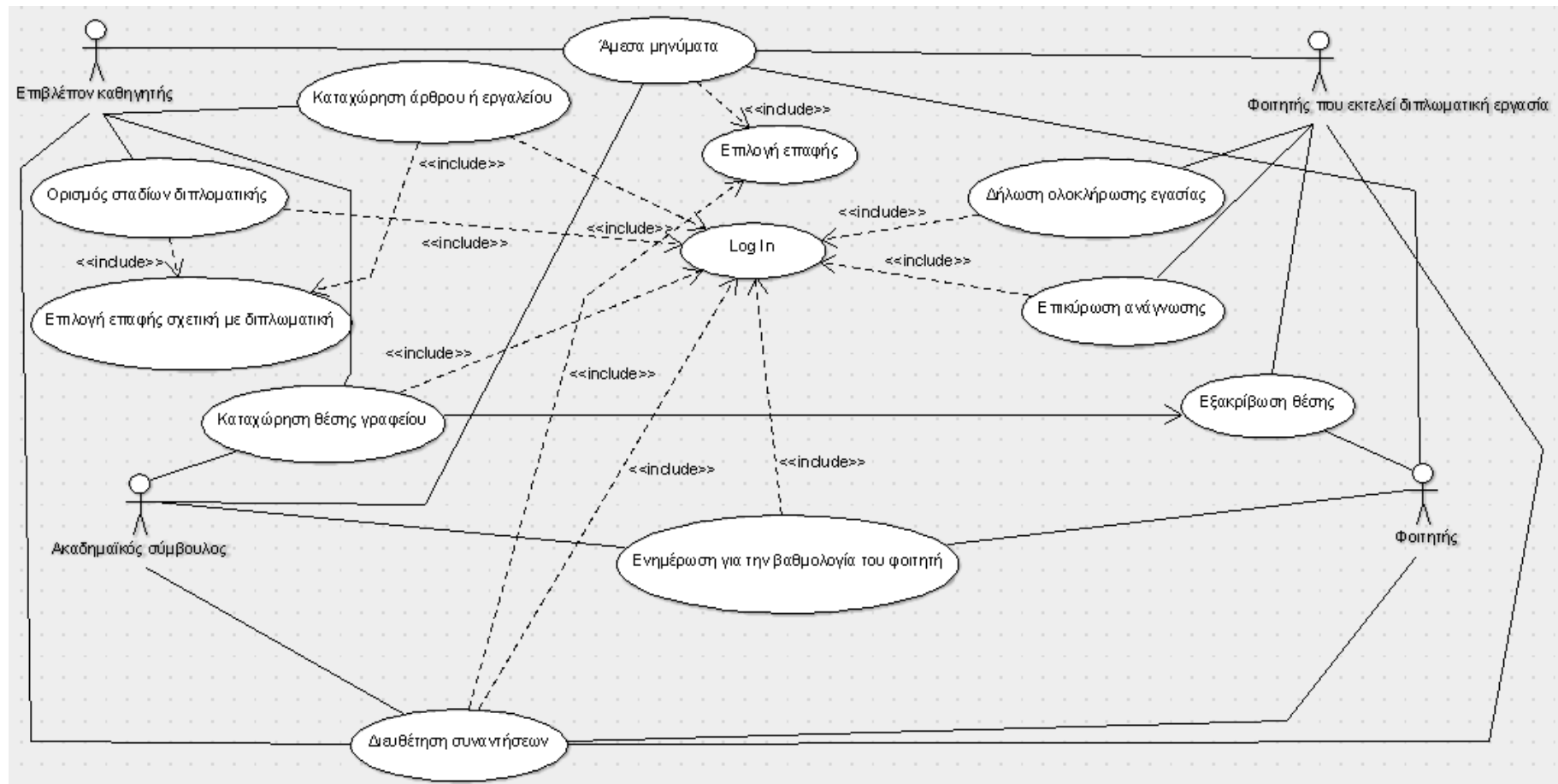
|                                                                        |    |
|------------------------------------------------------------------------|----|
| 2.1 Καταγραφή απαιτήσεων.....                                          | 5  |
| 2.2 Ανάλυση απαιτήσεων και περιγραφή βάσης δεδομένων.....              | 19 |
| 2.3 Επικοινωνία μεταξύ εφαρμογής, web service και βάσης δεδομένων..... | 28 |

---

Στο κεφάλαιο 2 καταγράφονται και αναλύονται λεπτομερώς οι απαιτήσεις, από τις οποίες ακολούθως προκύπτουν οι λειτουργίες της εφαρμογής. Επίσης παρουσιάζεται ο τρόπος επικοινωνίας μεταξύ της εφαρμογής, της υπηρεσίας διαδικτύου(web service) και της βάσης δεδομένων.

### 2.1 Καταγραφή απαιτήσεων

Οι απαιτήσεις έχουν εξαχθεί μετά από εισηγήσεις καθηγητών του τμήματος και από δικές μας διαπιστώσεις για λειτουργίες που θα ήταν χρήσιμες στην εφαρμογή. Στο διάγραμμα 2.1.1 υπάρχει ένα use case διάγραμμα με απεικόνιση όλων των λειτουργιών.



2.1.1 Διάγραμμα Use case

## Βασικές απαιτήσεις

### **1. Άμεσα μηνύματα**

#### **Purpose**

Σκοπός της λειτουργίας αυτής είναι η επικοινωνία επιβλεπόντων διπλωματικών/ακαδημαϊκών συμβούλων και φοιτητών μέσω άμεσων μηνυμάτων. Αυτή την λειτουργία θα μπορούν να την χρησιμοποιήσουν και ο φοιτητής και ο επιβλέπον καθηγητής/ακαδημαϊκός σύμβουλος.

#### **Stimulus/Response Sequence**

Η λειτουργία αυτή θα ενεργοποιείται μόλις ο χρήστης στείλει ένα μήνυμα σε κάποιο άτομο από τις επαφές του.

#### **Associated Functional Requirements**

##### **1. ΕΠΙΛΟΓΗ ΕΠΑΦΗΣ**

#### **Introduction**

Ο χρήστης έχει τη δυνατότητα να επιλέξει το άτομο που επιθυμεί να στείλει μήνυμα από τις δύο λίστες επαφών του, είτε από την λίστα με τις επαφές που σχετίζονται με διπλωματικές εργασίες, είτε από την λίστα που αφορά ακαδημαϊκό σύμβουλο.

#### **Inputs**

Θα πρέπει να επιλεγεί από τον χρήστη ένα άτομο από τις δύο λίστες επαφών του, είτε από την λίστα με τις επαφές που σχετίζονται με διπλωματικές εργασίες, είτε από την λίστα που αφορά ακαδημαϊκό σύμβουλο

#### **Processing**

Η εφαρμογή θα ψάχνει μέσα στη βάση δεδομένων για να βρει ποια άτομα σχετίζονται με τον χρήστη. Για παράδειγμα στην λίστα των επαφών ενός καθηγητή που σχετίζονται με τις διπλωματικές εργασίες που επιβλέπει θα βρίσκονται οι φοιτητές που επιβλέπει και στην λίστα των επαφών που αφορά ακαδημαϊκό σύμβουλο οι φοιτητές των οποίων είναι ακαδημαϊκός σύμβουλος.

### Outputs

Θα εμφανίζεται στην εφαρμογή ένα νέο παράθυρο στο οποίο ο χρήστης μπορεί να ξεκινήσει την συνομιλία με το άτομο το οποίο επέλεξε από μια από τις δύο λίστες.

## 2. ΑΠΟΣΤΟΛΗ ΜΗΝΥΜΑΤΟΣ

### Introduction

Ο χρήστης αφού επιλέξει την επαφή έχει την δυνατότητα να γράψει και να στείλει το μήνυμα.

### Inputs

Ο χρήστης πρέπει να γράψει το μήνυμα και ακολούθως να πατήσει το κουμπί αποστολή.

### Processing

Η εφαρμογή θα πάρει το μήνυμα που έχει γράψει ο χρήστης και θα στείλει το μήνυμα στον παραλήπτη.

### Outputs

Η εφαρμογή θα εμφανίζει το μήνυμα που αποστάληκε ως αποσταλμένο και θα εμφανίζεται στην συνομιλία.

## 2. Ορισμός σταδίων διπλωματικής

### **Purpose**

Σκοπός της λειτουργίας αυτής είναι ο καθηγητής να μπορεί να ορίζει στάδια στη διπλωματική, τα οποία πρέπει να ολοκληρώσει ο φοιτητής σε συγκεκριμένες χρονικές στιγμές. Αυτή την λειτουργία θα μπορούν να χρησιμοποιούν μόνο οι καθηγητές οι οποίοι επιβλέπουν κάποιο φοιτητή, ενώ μόνο οι φοιτητές που εργάζονται σε κάποια διπλωματική εργασία μπορούν να δουν τα στάδια που θέτει ο επιβλέπων καθηγητής τους.

### **Stimulus/Response Sequence**

Η λειτουργία αυτή θα ενεργοποιείται μόλις ο καθηγητής προσπαθήσει να ορίσει ένα στάδιο στην διπλωματική του φοιτητή.

### **Associated Functional Requirements**

#### **1. Login**

##### Introduction

Ο χρήστης για να χρησιμοποιήσει αυτή την λειτουργία απαιτείται πρώτα να έχει κάνει log in χρησιμοποιώντας το όνομα χρήστη (username) και τον κωδικό (password) του.

##### Inputs

Ο χρήστης πρέπει να δώσει το όνομα χρήστη (username) και τον κωδικό (password) του.

##### Processing

Η εφαρμογή θα ψάξει μέσα στην βάση δεδομένων για να βρει αν υπάρχει κάποιο άτομο με το όνομα χρήστη (username) και τον κωδικό (password) που έδωσε ο χρήστης και να εξακριβώσει αν είναι καθηγητής ή φοιτητής.

##### Outputs

Θα εμφανίζονται στην εφαρμογή το κεντρικό παράθυρο αν ο χρήστης έδωσε σωστά στοιχεία, αλλιώς θα του εμφανίζεται μήνυμα λάθους.

##### Error handling

Αν ο χρήστης δώσει λάθος όνομα χρήστη (username) ή κωδικό (password) τότε η εφαρμογή θα τον ενημερώνει με μήνυμα λάθους και θα του ζητά να τα ξαναδώσει.

#### **2. ΕΠΙΛΟΓΗ ΕΠΑΦΗΣ ΠΟΥ ΣΧΕΤΙΖΕΤΑΙ ΜΕ ΑΔΕ**

##### Introduction

Ο καθηγητής έχει τη δυνατότητα να επιλέξει από την λίστα που σχετίζεται με διπλωματικές εργασίες το φοιτητή που επιθυμεί να ορίσει κάποιο στάδιο στην διπλωματική του.

#### Inputs

Θα πρέπει να επιλεγεί από τον χρήστη ένα άτομο από την λίστα με τις επαφές που σχετίζονται με διπλωματικές εργασίες.

#### Processing

Η εφαρμογή θα ψάχνει μέσα στη βάση δεδομένων για να βρει ποια άτομα σχετίζονται με τον χρήστη. Για παράδειγμα στην λίστα των επαφών ενός καθηγητή που σχετίζονται με τις διπλωματικές εργασίες που επιβλέπει θα βρίσκονται οι φοιτητές που επιβλέπει.

#### Outputs

Θα εμφανίζεται στην εφαρμογή ένα νέο παράθυρο στο οποίο ο καθηγητής μπορεί να επιλέξει ημερομηνία και να γράψει μια σύντομη περιγραφή για το στάδιο.

### 3. ΟΡΙΣΜΟΣ ΣΤΑΔΙΩΝ

#### Introduction

Ο καθηγητής αφού επιλέξει την επαφή πρέπει να επιλέξει ημερομηνία και να γράψει μια σύντομη περιγραφή του σταδίου για να ορίσει ένα στάδιο.

#### Inputs

Ο καθηγητής πρέπει να επιλέξει ημερομηνία και να γράψει μια σύντομη περιγραφή του σταδίου και ίσως και μια αναφορά σε κάποιο εργαλείο που σχετίζεται με το στάδιο.

#### Processing

Η εφαρμογή θα πάρει τη περιγραφή ή και το εργαλείο που έχει γράψει ο καθηγητής και την ημερομηνία που έχει δώσει και θα δημιουργήσει μια καταχώρηση στην βάση δεδομένων που θα σχετίζεται με την επαφή που έχει επιλέξει προηγουμένως ο καθηγητής.

## Outputs

Η εφαρμογή θα εμφανίζει μήνυμα επιτυχίας στον καθηγητή αν ολοκληρώθηκε επιτυχώς ο ορισμός του σταδίου και το χρονοδιάγραμμα που έχει ορίσει μέχρι εκείνη την στιγμή. Επίσης όταν ο φοιτητής χρησιμοποιήσει την εφαρμογή θα δει στο ημερολόγιό του το νέο στάδιο που καταχώρησε ο καθηγητής.

### **3. Εξακρίβωση θέσης**

#### **Purpose**

Σκοπός της λειτουργίας αυτής είναι ο φοιτητής να γνωρίζει αν ο καθηγητής βρίσκεται στο γραφείο του χρησιμοποιώντας την θέση του καθηγητή (αν το επιτρέπει). Η λειτουργία αυτή μπορεί να χρησιμοποιηθεί μόνο από τους επιβλέποντες καθηγητές/ακαδημαϊκούς σύμβουλους, ενώ οι φοιτητές θα μπορούν μόνο να δουν αν ο επιβλέπων καθηγητής/ακαδημαϊκός σύμβουλος βρίσκεται στο γραφείο του.

#### **Stimulus/Response Sequence**

Η λειτουργία αυτή θα ενεργοποιείται μόλις ο καθηγητής ανοίξει την εφαρμογή.

#### **Associated Functional Requirements**

##### 1. Login

##### 2. ΚΑΤΑΧΩΡΗΣΗ ΘΕΣΗΣ ΓΡΑΦΕΙΟΥ

#### Introduction

Αποθηκεύεται η θέση του γραφείου του καθηγητή στην βάση δεδομένων, για να συγκριθεί μετέπειτα με την θέση του καθηγητή και να εξακριβωθεί αν ο καθηγητής βρίσκεται στο γραφείο του

#### Processing

Η εφαρμογή θα παίρνει την θέση του καθηγητή και θα την αποθηκεύει στην βάση όταν ο καθηγητής υποδείξει πως βρίσκεται στο γραφείο του.

## Outputs

Θα εμφανίζεται μήνυμα επιτυχίας στον καθηγητή.

### 3. ΕΛΕΓΧΟΣ ΚΑΙ ΠΑΡΟΥΣΙΑΣΗ ΑΝ Ο ΚΑΘΗΓΗΤΗΣ ΒΡΙΣΚΕΤΑΙ ΣΤΟ ΓΡΑΦΕΙΟ ΤΟΥ

#### Introduction

Ο φοιτητής μπορεί να ξέρει αν ο καθηγητής βρίσκεται στο γραφείο του ή όχι βασιζόμενος στο αν η τελευταία γνωστή θέση του καθηγητή ήταν περίπου η ίδια με αυτή που έχει δώσει ο καθηγητής σαν την θέση του γραφείου του.

#### Processing

Η εφαρμογή θα ελέγξει αν η τελευταία γνωστή θέση του καθηγητή ήταν το γραφείο του. Η λειτουργία αυτή θα πραγματοποιηθεί είτε με τη χρήση του μεσισμικού Really Simple Context Middleware (RSCM) είτε χρησιμοποιώντας τον Location Manager του Android.

#### Outputs

Η εφαρμογή θα εμφανίζει στον φοιτητή αν ο καθηγητής βρίσκεται στο γραφείο του ή όχι (ο καθηγητής μπορεί να καθορίσει αν θα εμφανίζεται αυτή η πληροφορία στον φοιτητή).

### 4. Άρθρα/εργαλεία

#### **Purpose**

Σκοπός της λειτουργίας αυτής είναι ο καθηγητής να ανεβάζει κάποια χρήσιμα άρθρα/εργαλεία και ακολούθως ο φοιτητής να μπορεί να διαβάσει αυτά τα άρθρα/εργαλεία. Την λειτουργία αυτή θα μπορούν να χρησιμοποιήσουν μόνο οι επιβλέποντες καθηγητές, ενώ μόνο οι φοιτητές που εργάζονται σε μια διπλωματική εργασία μπορούν να δουν τα άρθρα/εργαλεία που ανεβάζει ο επιβλέπων καθηγητής του.

### **Stimulus/Response Sequence**

Η λειτουργία αυτή θα ενεργοποιείται μόλις ο καθηγητής προσπαθήσει να καταχωρήσει κάποιο άρθρο ή κάποιο χρήσιμο εργαλείο με μια μικρή περιγραφή.

### **Associated Functional Requirements**

#### **1. Login**

#### **2. ΕΠΙΛΟΓΗ ΕΠΑΦΗΣ ΠΟΥ ΣΧΕΤΙΖΕΤΑΙ ΜΕ ΑΔΕ**

##### **Introduction**

Ο καθηγητής έχει τη δυνατότητα να επιλέξει από την λίστα που σχετίζεται με διπλωματικές εργασίες το φοιτητή που το άρθρο που θα ανεβάσει θα σχετίζεται με την διπλωματική του εργασία.

##### **Inputs**

Θα πρέπει να επιλεγεί από τον χρήστη ένα άτομο από την λίστα με τις επαφές που σχετίζονται με διπλωματικές εργασίες.

##### **Processing**

Η εφαρμογή θα ψάχνει μέσα στη βάση δεδομένων για να βρει ποια άτομα σχετίζονται με τον χρήστη. Για παράδειγμα στην λίστα των επαφών ενός καθηγητή που σχετίζονται με τις διπλωματικές εργασίες που επιβλέπει θα βρίσκονται οι φοιτητές που επιβλέπει.

##### **Outputs**

Θα εμφανίζεται στην εφαρμογή ένα νέο παράθυρο στο οποίο ο καθηγητής μπορεί να ανεβάσει το άρθρο.

#### **3. ΚΑΤΑΧΩΡΗΣΗ ΑΡΘΡΟΥ Ή ΕΡΓΑΛΕΙΟΥ**

##### **Introduction**

Ο καθηγητής μπορεί να καταχωρήσει τα άρθρα ή κάποιο χρήσιμο εργαλείο που θέλει να διαβάσει/δεν ο φοιτητής.

### Inputs

Ο καθηγητής γράφει το link για το άρθρο ή το όνομα του εργαλείου με μια σύντομη περιγραφή και το καταχωρεί.

### Processing

Η εφαρμογή θα πάρει το link ή το όνομα του εργαλείου με μια σύντομη περιγραφή και θα το αποθηκεύσει στην βάση δεδομένων

### Outputs

Στον καθηγητή θα εμφανιστεί μήνυμα επιτυχίας. Επίσης όταν ο φοιτητής ανοίξει την εφαρμογή, μετά την καταχώρηση του άρθρου ή του εργαλείου με μια σύντομη περιγραφή από τον καθηγητή, θα μπορεί να δει το εργαλείο και να ανοίξει το link.

## 4. ΕΠΙΚΥΡΩΣΗ ΑΝΑΓΝΩΣΗΣ ΑΠΟ ΦΟΙΤΗΤΗ

### Introduction

Ο φοιτητής μπορεί να επικυρώσει πως έχει διαβάσει ένα άρθρο που του υπόδειξε ο επιβλέπων καθηγητής του ή έχει δει κάποιο εργαλείο.

### Inputs

Ο φοιτητής πατάει ένα κουμπί δίπλα από το εργαλείο ή το άρθρο για να δηλώσει πως το έχει δει/διαβάσει.

### Processing

Η εφαρμογή θα ελέγξει αν ο φοιτητής έχει δηλώσει κάποιο εργαλείο ή άρθρο σαν διαβασμένο και θα το καταχωρήσει στην βάση δεδομένων.

### Outputs

Στον καθηγητή θα εμφανιστεί δίπλα από το άρθρο ή εργαλείο που έδωσε στον φοιτητή πως το έχει διαβάσει.

## **Δευτερεύοντες απαιτήσεις**

### **1. Ενημέρωση για την βαθμολογία του φοιτητή**

#### **Purpose**

Σκοπός της λειτουργίας αυτής είναι να ενημερώνεται ο φοιτητής και ο ακαδημαϊκός σύμβουλος του αν ο μέσος όρος των βαθμών του φοιτητή πέσει κάτω από το 5.0.

#### **Stimulus/Response Sequence**

Η λειτουργία αυτή θα ενεργοποιείται αν ο μέσος όρος των βαθμών του φοιτητή πέσει κάτω από το 5.0.

#### **Associated Functional Requirements**

##### **1. Login**

##### **2. ΕΝΗΜΕΡΩΣΗ ΦΟΙΤΗΤΗ ΚΑΙ ΑΚΑΔΗΜΑΪΚΟΥ ΣΥΜΒΟΥΛΟΥ**

#### **Introduction**

Νοούμενου ότι ο μέσος όρος των βαθμών του φοιτητή έχει πέσει κάτω από το 5.0, η εφαρμογή θα εμφανίσει μήνυμα στον καθηγητή με το οποίο θα τον ενημερώνει και θα τον προτρέπει να διευθετήσει συνάντηση.

#### **Processing**

Η εφαρμογή θα ελέγχει αν ο μέσος όρος των βαθμών του φοιτητή έχει πέσει κάτω από το 5.0 και αν έχει πέσει θα εμφανίζει μήνυμα στον ακαδημαϊκό του σύμβουλο.

#### **Outputs**

Θα εμφανίζεται στην εφαρμογή ένα μήνυμα που θα ενημερώνει τον ακαδημαϊκό του σύμβουλο.

## **2. Εργασίες που ολοκλήρωσε ο φοιτητής**

### **Purpose**

Σκοπός της λειτουργίας αυτής είναι ο φοιτητής να μπορεί να δηλώνει ποια εργασία έχει ολοκληρώσει σε κάποια συγκεκριμένη ημερομηνία. Η λειτουργία αυτή θα μπορεί να χρησιμοποιηθεί από τους φοιτητές που εργάζονται σε μια διπλωματική εργασία για να δηλώσουν τι έχουν κάνει κάποια συγκεκριμένη ημερομηνία, ενώ ο επιβλέπων καθηγητής θα μπορεί να δει τι ολοκλήρωσε ο φοιτητής.

### **Stimulus/Response Sequence**

Η λειτουργία αυτή θα ενεργοποιείται μόλις ο φοιτητής προσπαθήσει να δηλώσει μια εργασία.

### **Associated Functional Requirements**

#### **1. Login**

#### **2. ΔΗΛΩΣΗ ΕΡΓΑΣΙΑΣ**

##### **Introduction**

Ο φοιτητής πρέπει να επιλέξει ημερομηνία και να γράψει μια σύντομη περιγραφή της εργασίας που ολοκλήρωσε για να δηλώσει την ολοκλήρωση μιας εργασίας.

##### **Inputs**

Ο φοιτητής πρέπει να επιλέξει ημερομηνία και να γράψει μια σύντομη περιγραφή της εργασίας.

##### **Processing**

Η εφαρμογή θα πάρει τη περιγραφή που έχει γράψει ο φοιτητής και την ημερομηνία που έχει δώσει και θα δημιουργήσει μια καταχώρηση στην βάση δεδομένων.

##### **Outputs**

Η εφαρμογή θα εμφανίζει μήνυμα επιτυχίας στον φοιτητή αν ολοκληρώθηκε επιτυχώς η δήλωση της εργασίας. Επίσης όταν ο καθηγητής χρησιμοποιήσει την εφαρμογή θα δει στο ημερολόγιό του την εργασία που δήλωσε πως έχει ολοκληρώσει ο φοιτητής.

### **Πρόσθετες απαιτήσεις**

#### **1. Διευθέτηση συναντήσεων**

##### **Purpose**

Σκοπός της λειτουργίας αυτής είναι η διευθέτηση συναντήσεων. Αυτή η λειτουργία θα μπορεί να την χρησιμοποιήσει και ο ακαδημαϊκός σύμβουλος/επιβλέπον καθηγητής και ο φοιτητής.

##### **Stimulus/Response Sequence**

Η λειτουργία αυτή θα ενεργοποιείται μόλις ο χρήστης προσπαθήσει να διευθετήσει μια συνάντηση.

##### **Associated Functional Requirements**

#### **1. Login**

#### **2. ΕΠΙΛΟΓΗ ΕΠΑΦΗΣ**

##### **Introduction**

Ο χρήστης έχει τη δυνατότητα να επιλέξει, από τις δύο λίστες επαφών του, είτε από την λίστα με τις επαφές που σχετίζονται με διπλωματικές εργασίες, είτε από την λίστα που αφορά ακαδημαϊκό σύμβουλο, το άτομο που επιθυμεί να διευθετήσει συνάντηση.

##### **Inputs**

Θα πρέπει να επιλεγεί από τον χρήστη ένα άτομο από τις δύο λίστες επαφών του, είτε από την λίστα με τις επαφές που σχετίζονται με διπλωματικές εργασίες, είτε από την λίστα που αφορά ακαδημαϊκό σύμβουλο

### Processing

Η εφαρμογή θα ψάχνει μέσα στη βάση δεδομένων για να βρει ποια άτομα σχετίζονται με τον χρήστη. Για παράδειγμα στην λίστα των επαφών ενός καθηγητή που σχετίζονται με τις διπλωματικές εργασίες που επιβλέπει θα βρίσκονται οι φοιτητές που επιβλέπει και στην λίστα των επαφών που αφορά ακαδημαϊκό σύμβουλο οι φοιτητές των οποίων είναι ακαδημαϊκός σύμβουλος.

### Outputs

Θα εμφανίζεται στην εφαρμογή ένα νέο παράθυρο στο οποίο ο χρήστης μπορεί να επιλέξει ημερομηνίες και ώρες για την συνάντηση με το άτομο που επέλεξε από τη μια από τις δύο λίστες επαφών.

## 3. ΕΠΙΛΟΓΗ ΩΡΑΣ ΚΑΙ ΗΜΕΡΟΜΗΝΙΑΣ

### Introduction

Ο χρήστης μπορεί να επιλέξει κάποιες ώρες και ημερομηνίες για να διευθετήσει την συνάντηση.

### Inputs

Ο χρήστης επιλέγει ώρες και ημερομηνίες.

### Processing

Η εφαρμογή θα πάρει τις ώρες και την ημερομηνίες και θα ενημερώσει το άτομο με το οποίο ο χρήστης θέλει να διευθέτηση συνάντηση.

### Outputs

Θα παρουσιάζεται μήνυμα επιτυχίας θα ενημερώνεται το άτομο με το οποίο ο χρήστης θέλει να διευθέτηση συνάντηση.

## 4. ΕΠΙΚΥΡΩΣΗ ΣΥΝΑΝΤΗΣΗΣ

### Introduction

Το άτομο με το οποίο ο χρήστης θέλει να διευθετήσει συνάντηση θα πρέπει να επιλέξει μία από τις ώρες και ημερομηνίες.

#### Inputs

Το άτομο με το οποίο ο χρήστης θέλει να διευθετήσει συνάντηση επιλέγει ώρα και ημερομηνία.

#### Processing

Η εφαρμογή θα πάρει την ώρα και την ημερομηνία και θα ενημερώσει το χρήστη για την διευθέτηση της συνάντησης.

#### Outputs

Θα ενημερώνεται ο χρήστης για την ώρα και την ημερομηνία που επέλεξε το άτομο με το οποίο θέλει να διευθετήσει συνάντηση

## **2. Επίβλεψη και μεταπτυχιακών διατριβών**

Όσον αφορά την επίβλεψη διπλωματικών η εφαρμογή θα καλύπτει προπτυχιακές, αλλά και μεταπτυχιακές διατριβές. Όλες οι λειτουργίες που έχουν αναφερθεί πιο πάνω και αφορούν διπλωματικές εργασίες μπορούν να χρησιμοποιηθούν και για μεταπτυχιακές διατριβές.

## **2.2 Ανάλυση απαιτήσεων και περιγραφή βάσης δεδομένων**

### **UML class diagram**

Στο σχήμα 2.2.1 υπάρχει ένα διάγραμμα κλάσεων (class diagram) που περιγράφει την σύνδεση μεταξύ των κλάσεων της android εφαρμογής . Στο class diagram περιέχονται οι κύριες κλάσεις της εφαρμογής και η σύνδεση μεταξύ τους. Τα βελάκια μεταξύ των κλάσεων υποδηλώνουν πως από την κλάση A μπορεί να γίνει μετάβαση στην κλάση B. Το διάγραμμα αναφέρεται συγκεκριμένα για το λειτουργικό σύστημα Android καθώς περιέχει συστατικά μέρη που χρησιμοποιεί αποκλειστικά το Android για την σχεδίαση εφαρμογών, όπως Activity και Service. Οι περισσότερες

κλάσεις είναι του τύπου Activity, το οποίο είναι μια σελίδα στην εφαρμογή. Activity είναι η κάθε κλάση η οποία διαθέτει δική της γραφική διεπαφή χρήστη. Ένα activity μπορεί να τρέχει μόνο στο προσκήνιο και κλείνει όταν ο χρήστης μετακινηθεί σε ένα άλλο activity. Επίσης υπάρχουν και service. Service είναι το συστατικό της εφαρμογής το οποίο τρέχει συνεχώς συνήθως για την εκτέλεση κάποιας χρονοβόρας διαδικασίας. Τα service δεν διαθέτουν γραφική διεπαφή χρήστη.

Το LoginActivity είναι η πρώτη σελίδα που εμφανίζεται στην εφαρμογή. Αφότου δώσει το όνομα χρήστη και τον κωδικό του ο χρήστης μεταφέρεται στο StartActivity (ή StartActivityStud αν είναι φοιτητής) από εκεί ο χρήστης μπορεί να μεταφερθεί στα υπόλοιπα Activity. Τα Activity ChooseContactActivity, ChooseADEContactActivity και ChooseADEContactActivityArticlesTools χρησιμοποιούνται για επιλογή επαφής από τον χρήστη. Τα υπόλοιπα Activity και Service χρησιμοποιούνται για την υλοποίηση των λειτουργιών και περισσότερες λεπτομέρειες υπάρχουν στο κεφάλαιο 4.3 που αναφέρεται στην υλοποίηση της εφαρμογής.



## **Data Store**

Στην βάση δεδομένων θα αποθηκεύονται τα στοιχεία των καθηγητών και φοιτητών, τα στάδια που θέτει ο καθηγητής, τα άρθρα ή τα εργαλεία που ανεβάζει ο καθηγητής, το ιστορικό των μηνυμάτων μεταξύ δύο ατόμων και οι εργασίες που δήλωσε πως ολοκλήρωσε ο φοιτητής. Η βάση θα βρίσκεται στον εξυπηρετητή (server) του πανεπιστημίου και η επικοινωνία μεταξύ της βάσης και της εφαρμογής θα γίνεται μέσω μιας υπηρεσίας διαδικτύου(web service).

## **Size Calculation**

- ΦΟΙΤΗΤΗΣ

Κλειδί του πίνακα αυτού: StudID

| ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ        | ΤΥΠΟΣ         | SIZE      | ΠΕΡΙΓΡΑΦΗ                  |
|-----------------------|---------------|-----------|----------------------------|
| StudID                | integer       | 4 Bytes   | Auto generated             |
| Username              | Nvarchar (30) | 62 Bytes  | Δεν εισάγεται από την αρχή |
| Email                 | varchar(60)   | 122 Bytes | Δεν εισάγεται από την αρχή |
| Password              | Nvarchar(50)  | 102 Bytes | Δεν εισάγεται από την αρχή |
| Όνομα                 | Nvarchar (25) | 52 Bytes  | Δεν εισάγεται από την αρχή |
| Επίθετο               | Nvarchar (30) | 62 Bytes  | Δεν εισάγεται από την αρχή |
| ID Επιβλέπον Καθηγητή | Integer       | 4 Bytes   | Δεν εισάγεται από την αρχή |

|                                |             |           |                               |
|--------------------------------|-------------|-----------|-------------------------------|
| ID<br>Ακαδημαϊκού<br>σύμβουλου | Integer     | 4 Bytes   | Δεν εισάγεται από την<br>αρχή |
| Μέσος όρος βαθμών              | Real        | 8 Bytes   | Δεν εισάγεται από την<br>αρχή |
| LastLogin                      | datetime    | 3 Bytes   | Δεν εισάγεται από την<br>αρχή |
| XMPPUsername                   | varchar(50) | 102 Bytes | Δεν εισάγεται από την<br>αρχή |
| XMPPPassword                   | varchar(50) | 102 Bytes | Δεν εισάγεται από την<br>αρχή |

- ΚΑΘΗΓΗΤΗΣ

Κλειδί του πίνακα αυτού: ProfID

| ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ | ΤΥΠΟΣ         | SIZE      | ΠΕΡΙΓΡΑΦΗ                     |
|----------------|---------------|-----------|-------------------------------|
| ProfID         | integer       | 4 Bytes   | Auto generated                |
| Username       | Nvarchar (30) | 62 Bytes  | Δεν εισάγεται από την<br>αρχή |
| Password       | Nvarchar (50) | 102 Bytes | Δεν εισάγεται από την<br>αρχή |
| Email          | Varchar (60)  | 102 Bytes | Δεν εισάγεται από την<br>αρχή |
| Όνομα          | Nvarchar (25) | 52 Bytes  | Δεν εισάγεται από την<br>αρχή |

|                 |               |           |                            |
|-----------------|---------------|-----------|----------------------------|
| Επίθετο         | Nvarchar (30) | 62 Bytes  | Δεν εισάγεται από την αρχή |
| InOffice        | Bit           | 1 Byte    | Δεν εισάγεται από την αρχή |
| OfficeLongitude | Float         | 4 Bytes   | Δεν εισάγεται από την αρχή |
| OfficeLatitude  | Float         | 4 Bytes   | Δεν εισάγεται από την αρχή |
| LastLogin       | Datetime      | 3 Bytes   | Δεν εισάγεται από την αρχή |
| XMPPUsername    | varchar(50)   | 102 Bytes | Δεν εισάγεται από την αρχή |
| XMPPPassword    | varchar(50)   | 102 Bytes | Δεν εισάγεται από την αρχή |

- ΣΤΑΔΙΑ

Κλειδί του πίνακα αυτού: StageID

| ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ    | ΤΥΠΟΣ          | SIZE      | ΠΕΡΙΓΡΑΦΗ                  |
|-------------------|----------------|-----------|----------------------------|
| StageID           | integer        | 4 Bytes   | Auto generated             |
| Ημερομηνία        | Date           | 3 Bytes   | Δεν εισάγεται από την αρχή |
| Σύντομη περιγραφή | Nvarchar (300) | 602 Bytes | Δεν εισάγεται από την αρχή |

|              |          |         |                            |
|--------------|----------|---------|----------------------------|
| ID Φοιτητή   | Integer  | 4 Bytes | Δεν εισάγεται από την αρχή |
| ID Καθηγητή  | Integer  | 4 Bytes | Δεν εισάγεται από την αρχή |
| TimeModified | Datetime | 3 Bytes | Δεν εισάγεται από την αρχή |
| Viewed       | Bit      | 1 Byte  | Δεν εισάγεται από την αρχή |

- ΑΡΘΡΑ/ΕΡΓΑΛΕΙΑ

Κλειδί του πίνακα αυτού: ArtID

| ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ    | ΤΥΠΟΣ          | SIZE      | ΠΕΡΙΓΡΑΦΗ                  |
|-------------------|----------------|-----------|----------------------------|
| ArtID             | integer        | 4 Bytes   | Auto generated             |
| Διαβασμένο        | bit            | 1 Byte    | Δεν εισάγεται από την αρχή |
| Σύντομη περιγραφή | Nvarchar (300) | 602 Bytes | Δεν εισάγεται από την αρχή |
| ID Φοιτητή        | Integer        | 4 Bytes   | Δεν εισάγεται από την αρχή |
| ID Καθηγητή       | Integer        | 4 Bytes   | Δεν εισάγεται από την αρχή |
| Title             | Varchar (50)   | 102 Bytes | Δεν εισάγεται από την αρχή |
| TimeModified      | Datetime       | 3 Bytes   | Δεν εισάγεται από την αρχή |

|        |     |        |                            |
|--------|-----|--------|----------------------------|
|        |     |        | αρχή                       |
| Viewed | Bit | 1 Byte | Δεν εισάγεται από την αρχή |

- ΕΡΓΑΣΙΕΣ ΦΟΙΤΗΤΗ

Κλειδί του πίνακα αυτού: WorkID

| ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ    | ΤΥΠΟΣ          | SIZE      | ΠΕΡΙΓΡΑΦΗ                  |
|-------------------|----------------|-----------|----------------------------|
| WorkID            | integer        | 4 Bytes   | Auto generated             |
| Ημερομηνία        | Date           | 3 Bytes   | Δεν εισάγεται από την αρχή |
| Σύντομη περιγραφή | Nvarchar (300) | 602 Bytes | Δεν εισάγεται από την αρχή |
| ID Φοιτητή        | Integer        | 4 Bytes   | Δεν εισάγεται από την αρχή |
| ID Καθηγητή       | Integer        | 4 Bytes   | Δεν εισάγεται από την αρχή |
| TimeModified      | Datetime       | 3 Bytes   | Δεν εισάγεται από την αρχή |
| Viewed            | Bit            | 1 Byte    | Δεν εισάγεται από την αρχή |

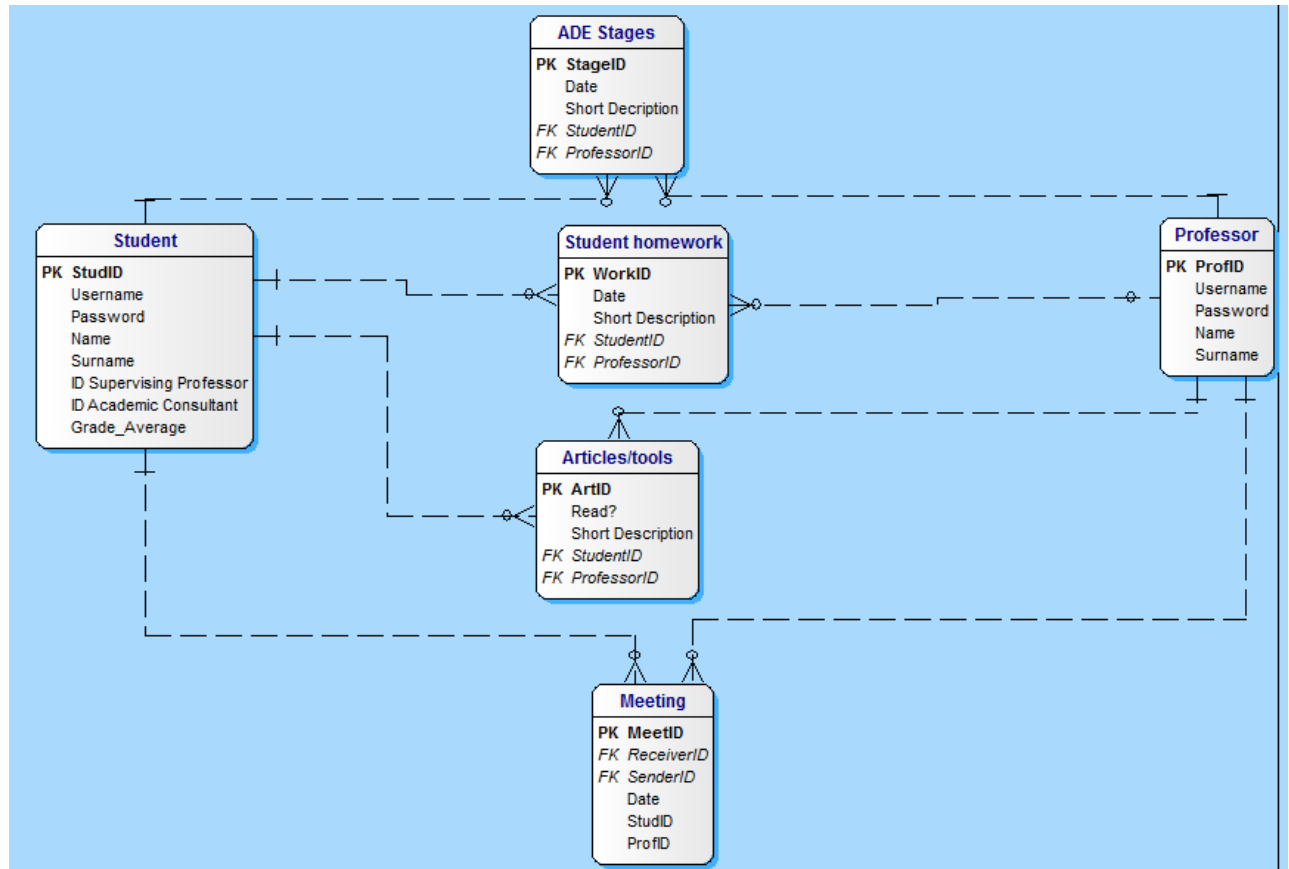
- ΣΥΝΑΝΤΗΣΕΙΣ

Κλειδί του πίνακα αυτού: MeetID

| ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ | ΤΥΠΟΣ         | SIZE      | ΠΕΡΙΓΡΑΦΗ                  |
|----------------|---------------|-----------|----------------------------|
| MeetID         | integer       | 4 Bytes   | Auto generated             |
| ID Αποστολέα   | Integer       | 4 Bytes   | Δεν εισάγεται από την αρχή |
| ID Παραλήπτη   | Integer       | 4 Bytes   | Δεν εισάγεται από την αρχή |
| Ημερομηνία     | Date          | 3 Bytes   | Δεν εισάγεται από την αρχή |
| Ώρα            | Datetime      | 3 Bytes   | Δεν εισάγεται από την αρχή |
| Θέμα           | Nvarchar (50) | 102 Bytes | Δεν εισάγεται από την αρχή |
| GroupID        | Varchar (50)  | 102 Bytes | Δεν εισάγεται από την αρχή |
| Confirmed      | Bit           | 1 Byte    | Δεν εισάγεται από την αρχή |
| TimeModified   | Datetime      | 3 Bytes   | Δεν εισάγεται από την αρχή |
| Viewed         | Bit           | 1 Byte    | Δεν εισάγεται από την αρχή |

## ER Diagram

Στο σχήμα 2.2.2 παρουσιάζεται το ER διάγραμμα που δείχνει όλους τους πίνακες και τις συσχετίσεις μεταξύ τους.



2.2.2 ER Διάγραμμα

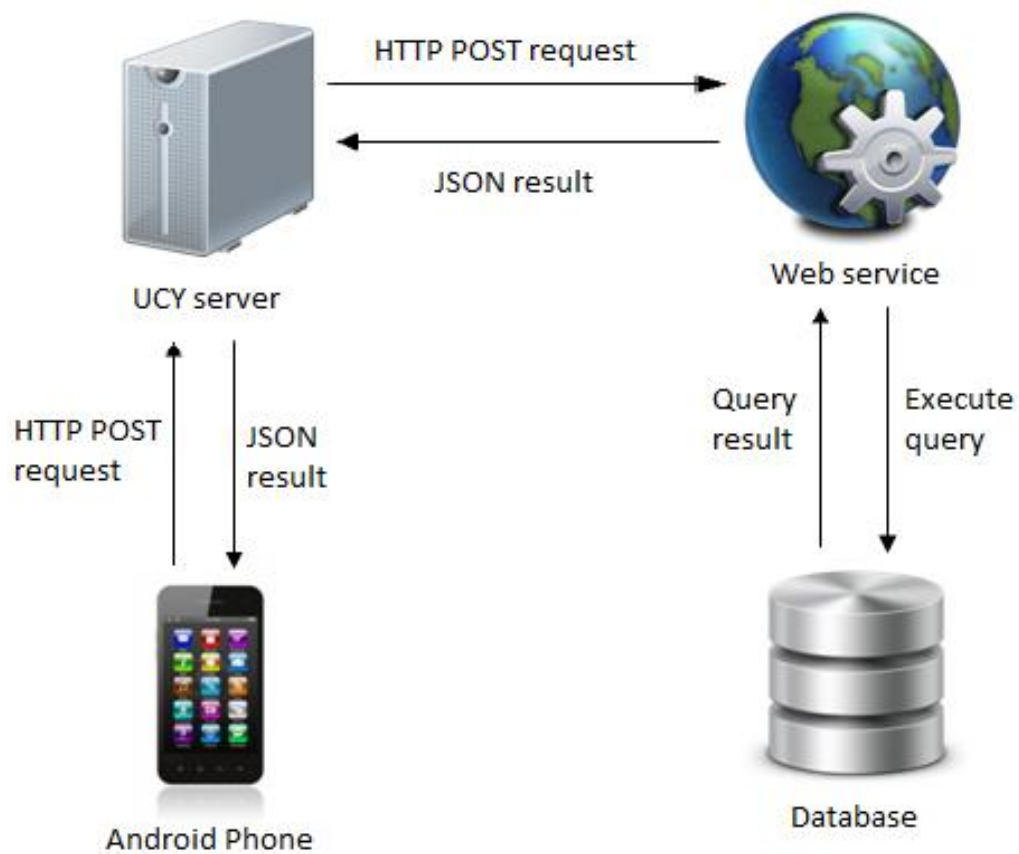
## 2.3 Επικοινωνία μεταξύ εφαρμογής, web service και βάσης δεδομένων

Η επικοινωνία μεταξύ της android εφαρμογής και της βάσης δεδομένων είναι ουσιώδους σημασίας για την εκπόνηση των περισσότερων λειτουργιών. Σχεδόν όλες οι λειτουργίες της εφαρμογής απαιτούν την αποθήκευση και ανάκτηση δεδομένων από την βάση δεδομένων.

Για την πρόσβαση στην βάση δεδομένων από την εφαρμογή υλοποιήθηκε μια Representational State Transfer (RESTful)[13] υπηρεσία διαδικτύου (web service) που ουσιαστικά παίζει ρόλο διαμεσολαβητή μεταξύ της εφαρμογής και της βάσης δεδομένων.

Η διαδικασία που ακολουθείτε για την επικοινωνία μεταξύ εφαρμογής, υπηρεσίας διαδικτύου (web service) και βάσης δεδομένων είναι η εξής. Αρχικά η εφαρμογή στέλνει κωδικοποιημένα SQL queries μέσω Hypertext Transfer Protocol (HTTP)[22] POST αιτήσεων προς την υπηρεσία διαδικτύου (web service). Ακολούθως, η υπηρεσία διαδικτύου (web service) διαβάζει το SQL query, το αποκωδικοποιεί και το εκτελεί στην βάση δεδομένων. Το τελευταίο βήμα που εκτελείται από την υπηρεσία διαδικτύου (web service) είναι να διαβάσει το αποτέλεσμα από την βάση δεδομένων, να το κωδικοποιήσει στην μορφή JSON και να το επιστρέψει σαν απάντηση στην HTTP αίτηση της εφαρμογής. Τέλος, η εφαρμογή παίρνει το αποτέλεσμα που επέστρεψε η υπηρεσία διαδικτύου (web service), το αποκωδικοποιεί από την JSON μορφή του και το χρησιμοποιεί στην λειτουργία που θέλει να ολοκληρώσει.

Παρακάτω υπάρχει ένα διάγραμμα (σχήμα 2.3.1) που παρουσιάζει την επικοινωνία μεταξύ βάσης δεδομένων και της εφαρμογής, μέσω της υπηρεσίας διαδικτύου (web service).



### 2.3.1 Επικοινωνία μεταξύ των συστατικών μερών του συστήματος

## Κεφάλαιο 3. Τεχνολογίες/Πρωτόκολλα Ανάπτυξης Εφαρμογής

---

|                                                             |    |
|-------------------------------------------------------------|----|
| 3.1 C#/ASP .NET .....                                       | 31 |
| 3.2 WCF RESTful web service .....                           | 33 |
| 3.3 SQL server 2008 .....                                   | 34 |
| 3.4 Really Simple Context Middleware (RSCM).....            | 35 |
| 3.5 Extensible Messaging and Presence Protocol (XMPP) ..... | 36 |

---

Στο κεφάλαιο αυτό γίνεται λεπτομερής ανάλυση των τεχνολογιών και των πρωτοκόλλων που χρησιμοποιήθηκαν κατά την διάρκεια της υλοποίησης της εφαρμογής και γενικά ολόκληρου του συστήματος.

### 3.1 C#/ASP.NET

Σε αυτήν την ενότητα περιγράφεται η γλώσσα προγραμματισμού C# και το framework Active Server Pages (ASP).NET. Αρχικά να αναφέρουμε πως και οι δύο αυτές τεχνολογίες ανήκουν στο .NET framework που σχεδίασε η Microsoft.

Η γλώσσα C# είναι μια καινούργια, πρωτοεμφανίστηκε το 2000, αντικειμενοστραφής γλώσσα σχεδιασμένη από την Microsoft. Αρχικός σκοπός της Microsoft ήταν η δημιουργία μιας αντικειμενοστραφής γλώσσας προγραμματισμού που θα έμοιαζε στην C, για αυτό άλλωστε και το αρχικό της όνομα ήταν Cool (C-like Object Oriented Language). Στόχος της γλώσσας αυτής ήταν να είναι μια απλή, μοντέρνα, γενικών χρήσεων αντικειμενοστραφής γλώσσα.

Η γλώσσα C# έχει όλα τα χαρακτηριστικά των αντικειμενοστραφή γλωσσών όπως την δημιουργία κλάσεων/αντικειμένων, την κληρονομικότητα, boxing και unboxing, exception handling, τον πολυμορφισμό κτλπ. Η C# μοιάζει πολύ στην σύνταξη και στα χαρακτηριστικά στην γλώσσα προγραμματισμού Java και έχει χαρακτηριστεί μέχρι και απομίμηση της. Επίσης η γλώσσα αυτή διαθέτει μια πληθώρα από βιβλιοθήκες οι οποίες μπορούν να χρησιμοποιηθούν από τους

προγραμματιστές καθώς επίσης και διάφορες βοηθητικές λειτουργίες όπως, έλεγχος διαστάσεων του πίνακα, έλεγχος για προσπάθεια χρήσης μη αρχικοποιημένης μεταβλητής, αυτόματο garbage collection κτλ. Τέλος η C# σαν γλώσσα που ανήκει στο .NET framework διαθέτει ένα ακόμα βασικό χαρακτηριστικό, παράγει ενδιάμεσο κώδικα (CIL) ο οποίος μπορεί να χρησιμοποιηθεί σε συνδυασμό με οποιαδήποτε άλλη γλώσσα ανήκει στο .NET framework.

Το ASP .NET είναι ένα framework σχεδιασμένο από την Microsoft με σκοπό την δημιουργία δυναμικών ιστοσελίδων, διαδικτυακών εφαρμογών και webservises που τρέχουν στον εξυπηρετητή (server-side). Και αυτό το framework είναι σχετικά καινούργιο αφού πρωτοεμφανίστηκε το 2002.

Ιστοσελίδες που χρησιμοποιούν το ASP .NET αποτελούνται από Web form αρχεία με επέκταση (extension) .aspx. Σε αυτά τα Web form αρχεία περιέχεται HTML κώδικας που μπορεί να διαβαστεί από όλους τους φυλλομετρητές (browsers) καθώς επίσης και τα scripts που εκτελούνται στον εξυπηρετητή (server). Τα scripts συνήθως μπορούν να αποτελούνται από αντικειμενοστραφή κώδικα που χρησιμοποιείται για ανταπόκριση στις ενέργειες του χρήστη, πρόσβαση στην βάση δεδομένων κτλ. Τα scripts μπορούν να είναι γραμμένα χρησιμοποιώντας είτε την γλώσσα C#, είτε την γλώσσα VisualBasic, είτε την γλώσσα F#, δηλαδή γλώσσες που ανήκουν στο .NET framework. Επίσης από την δεύτερη έκδοση του framework και μετά δίνεται η δυνατότητα στον προγραμματιστή αν το επιθυμεί να τοποθετήσει τα scripts σε ένα διαφορετικό αρχείο με extension aspx.cs ή aspx.vb ή aspx.fs ανάλογα με την γλώσσα προγραμματισμού. Επιπλέον πλεονέκτημα του ASP .NET είναι, όπως και της C#, πως μπορεί να χρησιμοποιηθεί σε συνδυασμό και με κώδικα που είναι γραμμένος σε κάποια άλλη γλώσσα του .NET framework.

Στην παρούσα εργασία, οι τεχνολογίες αυτές χρησιμοποιήθηκαν για την υλοποίηση της διαδικτυακής εφαρμογής και της υπηρεσίας διαδικτύου.

### 3.2 WCF RESTful web service

Το Windows Communication Foundation (WCF) [12] αποτελεί ένα εργαλείο που αποτελείται από ένα σύνολο από Application programming interface (APIs) και χρησιμοποιείται για την δημιουργία service-oriented εφαρμογών, όπως για παράδειγμα μια RESTful υπηρεσία διαδικτύου (web service). Το WCF είναι σχεδιασμένο από την Microsoft και ανήκει στο .NET framework.

Κάθε υπηρεσία που σχεδιάζεται με το WCF διαθέτει μια διεπαφή (interface) το οποίο μπορεί να χρησιμοποιήσει ο πελάτης (client) για να κάνει χρήση της υπηρεσίας. Επίσης κάθε υπηρεσία διαθέτει ένα ή περισσότερα σημεία σύνδεσης (endpoints) τα οποία χρησιμοποιεί ο πελάτης (client) για να συνδεθεί με το service. Τα σημεία σύνδεσης (endpoints) αποτελούνται από το URL και κάποιες παραμέτρους για το πώς θα μεταφερθούν τα δεδομένα προς τον πελάτη (client). Πέρα από τα σημεία σύνδεσης (endpoints), υπηρεσίες που σχεδιάζονται με το WCF διαθέτουν και συμπεριφορές (behaviors). Οι συμπεριφορές (behaviors) επιτρέπουν στους προγραμματιστές να δημιουργούν δικές τους τροποποιήσεις στον τρόπο που επεξεργάζονται ή μεταφέρονται τα μηνύματα μεταξύ πελάτη (client) και υπηρεσίας. Για παράδειγμα, μπορεί ο προγραμματιστής να προσθέσει χαρακτηριστικά για ασφάλεια, όπως εξουσιοδότηση και ταυτοποίηση.

Ο όρος Representational State Transfer (REST) αναφέρεται σε μια αρχιτεκτονική λογισμικού η οποία χρησιμοποιείται κυρίως για τον σχεδιασμό υπηρεσιών διαδικτύου (web services). Οι υπηρεσίες διαδικτύου (web services) που χρησιμοποιούν αυτή την αρχιτεκτονική εστιάζουν στην μεταφορά δεδομένων μέσω Hypertext Transfer Protocol (HTTP) αιτήσεων από των εξυπηρετητή (server) προς οποιοδήποτε πελάτη (client) που είναι σχεδιασμένος σε οποιαδήποτε προγραμματιστική γλώσσα.

Εφαρμογές που χρησιμοποιούν την REST αρχιτεκτονική αποτελούνται από τους πελάτες (clients) και τους εξυπηρετητή (servers). Όπως αναφέρθηκε και πιο πάνω ο πελάτης (client) στέλνει HTTP αιτήσεις προς τον εξυπηρετητή (server) (οι οποίες μπορούν να είναι οποιουδήποτε τύπου, GET, POST, PUT, DELETE), ο εξυπηρετητής (server) επεξεργάζεται τις αιτήσεις αυτές και επιστρέφει μια απάντηση. Συνήθως οι πληροφορίες που ανταλλάσσονται μεταξύ πελάτη (client) και εξυπηρετητή

(server) είναι της μορφής JavaScript Object Notation (JSON) ή Extensible Markup Language(XML).

Η αρχιτεκτονική REST τα τελευταία χρόνια χρησιμοποιείται όλο και περισσότερο κερδίζοντας έδαφος από την «αντίπαλο» αρχιτεκτονική SOAP. Αυτό συμβαίνει κυρίως λόγω της απλότητας και της ευκολίας χρήσης της REST αρχιτεκτονικής.

Οι τεχνολογίες αυτές έχουν χρησιμοποιηθεί για την υλοποίηση της υπηρεσίας διαδικτύου που λειτουργεί σαν μεσολαβητής μεταξύ της Android εφαρμογής και της βάσης δεδομένων.

### **3.3 SQL Server 2008**

Για την αποθήκευση των δεδομένων της εφαρμογής χρειαζόμαστε μια βάση δεδομένων. Για την δημιουργία της βάσης δεδομένων χρησιμοποιήσαμε τον SQL server 2008.

Ο SQL server 2008 χρησιμοποιείται για την σχεδίαση και διαχείριση βάσεων δεδομένων. Ο SQL server είναι σχεδιασμένος από την Microsoft στην γλώσσα προγραμματισμού C++.

Σαν σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων, τα δεδομένα αποθηκεύονται στην βάση δεδομένων μέσα σε συσχετιζόμενους πίνακες. Ακολούθως τα δεδομένα μπορούν να διαβαστούν, τροποποιηθούν ή να προστεθούν καινούργια μέσω αυτών των πινάκων. Η δημιουργία μιας βάσης δεδομένων στον SQL server μπορεί να γίνει με δύο τρόπους, είτε μέσω του γραφικού περιβάλλοντος που προσφέρει το εργαλείο SQL server management studio είτε μέσω της Data Definiton Language (DDL) T-SQL. Ακολούθως για την επεξεργασία των δεδομένων της βάσης δεδομένων απαιτείται η δημιουργία query statements τα οποία είναι γραμμένα είτε στην T-SQL είτε στην ANSI SQL.

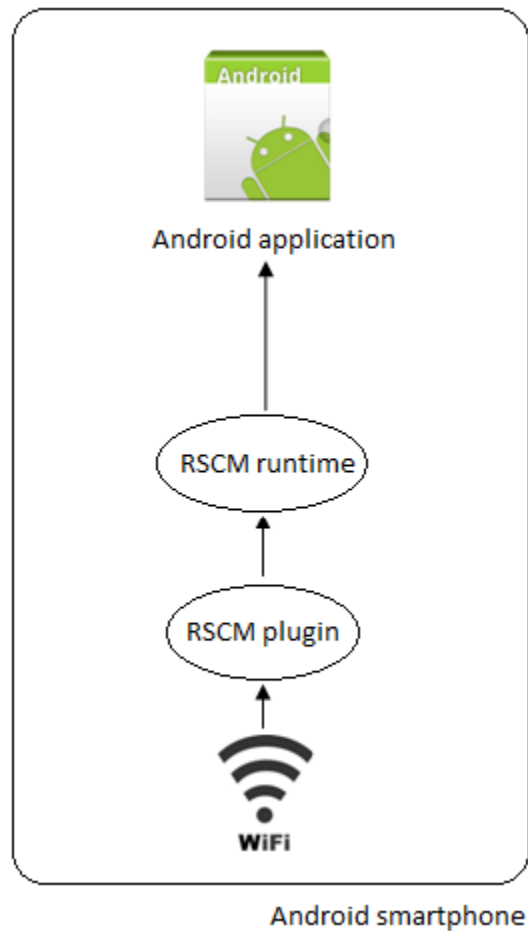
Μια βάση δεδομένων σχεδιασμένη με τον SQL server, μπορεί να είναι εγκατεστημένη σε έναν εξυπηρετητή (server) και να έχουν πρόσβαση σε αυτή απομακρυσμένοι χρήστες.

### **3.4 Really Simple Context Middleware (RSCM)**

Το Really Simple Context Middleware (RSCM) είναι ένα μεσισμικό που προσφέρει εύκολη πρόσβαση σε πληροφορίες συγκεκριμένου, σχεδιασμένο από τον Νέαρχο Πασπαλλή. Το RSCM αποτελείται από δύο κύρια συστατικά, το RSCM library που περιέχει τον κύριο όγκο του κώδικα που απαιτείται για την δημιουργία εφαρμογών που κάνουν χρήση του μεσισμικού και το RSCM runtime το οποίο περιέχει μια android υπηρεσία (service) που τρέχει στο παρασκήνιο και διευκολύνει την επικοινωνία μεταξύ των plugins που παράγουν την πληροφορία συγκεκριμένου και των εφαρμογών που την καταναλώνουν.

Τα plugins αποτελούν επίσης σημαντικό στοιχείο του μεσισμικού. Τα plugins είναι ανεξάρτητα συστατικά μέρη τα οποία μπορούν εύκολα να επαναχρησιμοποιηθούν και σκοπός τους είναι να παράγουν πληροφορία συγκεκριμένου από τους σένσορες της συσκευής. Τα plugins χωρίζονται σε δύο κατηγορίες, context sensors και context reasoners. Στην κατηγορία context sensors ανήκουν τα απλά plugins που μόνο εξάγουν πληροφορία συγκεκριμένου από τους σένσορες της συσκευής. Στην κατηγορία context reasoners ανήκουν τα plugins που όχι μόνο εξάγουν πληροφορία συγκεκριμένου από τους σένσορες αλλά επεξεργάζονται την πληροφορία αυτή και εξάγουν μια σύνοψη ή κάποια συμπεράσματα βασισμένα στην πληροφορία συγκεκριμένου. Για αυτό το λόγο plugins που ανήκουν στην κατηγορία context reasoners, χαρακτηρίζονται σαν παραγωγοί αλλά και καταναλωτές πληροφορίας συγκεκριμένου, ενώ τα plugins στην κατηγορία context sensors μόνο παράγουν πληροφορία συγκεκριμένου. Καταναλωτές πληροφορίας συγκεκριμένου είναι επίσης και οι εφαρμογές που χρησιμοποιούν το RSCM. Στο σχήμα 3.4.1 υπάρχει ένα παράδειγμα χρήσης του wifi σένσορα για εξαγωγή πληροφοριών συγκεκριμένου με την χρήση του RSCM.

Στην android εφαρμογή το RSCM χρησιμοποιείται για την εξαγωγή της θέσης του καθηγητή. Η οποία ακολούθως συγκρίνεται με την θέση του γραφείου του για να εξακριβωθεί αν βρίσκεται σε αυτό.



3.4.1 Παράδειγμα χρήσης RSCM

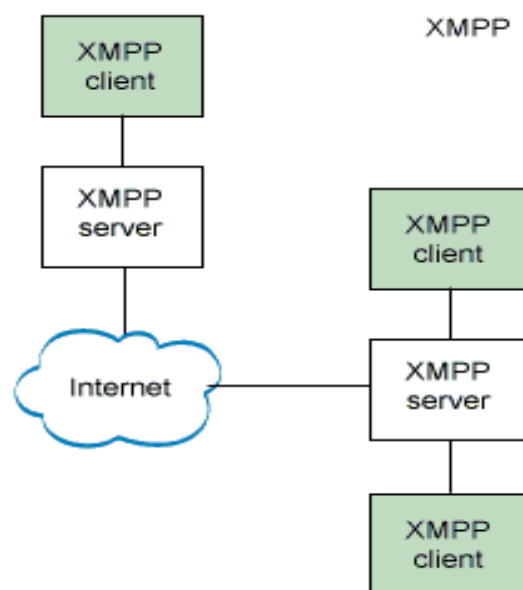
### 3.5 Extensible Messaging and Presence Protocol (XMPP)

Το Extensible Messaging and Presence Protocol (XMPP) ή αλλιώς Jabber είναι ένα πρωτόκολλο που προσφέρει επικοινωνία σε πραγματικό χρόνο. Βασικές λειτουργίες που προσφέρει το XMPP είναι η ανταλλαγή άμεσων μηνυμάτων, η ανταλλαγή πληροφορίας παρουσίας του χρήστη και η ανταλλαγή δεδομένων. Το XMPP είναι ανοικτό πρωτόκολλο και μπορεί ο κάθε προγραμματιστής να προσθέσει επεκτάσεις σε αυτό. Επίσης το XMPP χρησιμοποιείται ευρέως από διάφορες μεγάλες

εταιρίες λογισμικού όπως η Google, η οποία χρησιμοποιεί το πρωτόκολλο στην εφαρμογή της Google Talk.

Η επικοινωνία μεταξύ δύο χρηστών πραγματοποιείται με την μεσολάβηση ενός εξυπηρετητή (server). Υπάρχουν διάφοροι δημόσιοι εξυπηρετητή (server) π.χ. jabber.org τους οποίους μπορεί να χρησιμοποιήσει ο καθένας δωρεάν. Όλοι οι εξυπηρετητές (server) υλοποιούν τις βασικές λειτουργίες του XMPP πρωτοκόλλου και ο κάθε εξυπηρετητής (server) διαθέτει επίσης υλοποιημένες και ορισμένες επεκτάσεις του πρωτοκόλλου όπως για παράδειγμα την λειτουργία για ανάκτηση offline μηνυμάτων. Στο σχήμα 3.5.1 φαίνεται ο τρόπος επικοινωνίας μεταξύ συσκευών (XMPP clients) και εξυπηρετητών (XMPP server) για την ανταλλαγή μηνυμάτων.

Το XMPP πρωτόκολλο χρησιμοποιείται στην android εφαρμογή για ανταλλαγή άμεσων μηνυμάτων μεταξύ καθηγητών και φοιτητών.



3.5.1 Δομή XMPP

## Κεφάλαιο 4. Υλοποίηση

---

|                                             |    |
|---------------------------------------------|----|
| 4.1 Υλοποίηση υπηρεσίας διαδικτύου .....    | 38 |
| 4.2 Υλοποίηση βάσης δεδομένων.....          | 40 |
| 4.3 Υλοποίηση εφαρμογής .....               | 42 |
| 4.3.1 Γραφική διεπαφή χρήστη .....          | 43 |
| 4.3.2 Υλοποίηση λειτουργιών .....           | 48 |
| 4.4 Υλοποίηση διαδικτυακής εφαρμογής .....  | 57 |
| 4.4.1 Αρχική σελίδα .....                   | 57 |
| 4.4.2 Εγγραφή καθηγητή .....                | 58 |
| 4.4.3 Εγγραφή φοιτητή .....                 | 60 |
| 4.4.4 Επεξεργασία λογαριασμού .....         | 61 |
| 4.4.5 Επεξεργασία από τον διαχειριστή ..... | 64 |

---

Σε αυτό το κεφάλαιο γίνεται περιγραφή του τρόπου υλοποίησης ολόκληρου του συστήματος. Αρχικά αναλύεται ο τρόπος υλοποίησης της εφαρμογής και ακολούθως της διαδικτυακής εφαρμογής, της υπηρεσίας διαδικτύου (web service) και τέλος της βάσης δεδομένων.

### 4.1 Υλοποίηση υπηρεσίας διαδικτύου

Η υπηρεσία διαδικτύου (web service) έχει σχεδιαστεί χρησιμοποιώντας το εργαλείο WCF και ακολουθεί την αρχιτεκτονική REST. Η υπηρεσία διαδικτύου (web service) είναι εγκατεστημένο στον IIS server του πανεπιστημίου και λειτουργεί σαν μεσολαβητής στην επικοινωνία μεταξύ της android εφαρμογής και της βάσης δεδομένων. Δουλειά της υπηρεσίας διαδικτύου (web service) είναι να παίρνει τις HTTP POST αιτήσεις που γίνονται από την εφαρμογή, να της επεξεργάζεται, να τρέχει το query στην βάση και ακολούθως να επιστρέφει το αποτέλεσμα στην

εφαρμογή κωδικοποιημένο σε JSON μορφή. Η υπηρεσία διαδικτύου (web service) υλοποιείται κυρίως σε δύο αρχεία το `IRestServiceImplPost.cs` και το `RestServiceImplPost.svc.cs`.

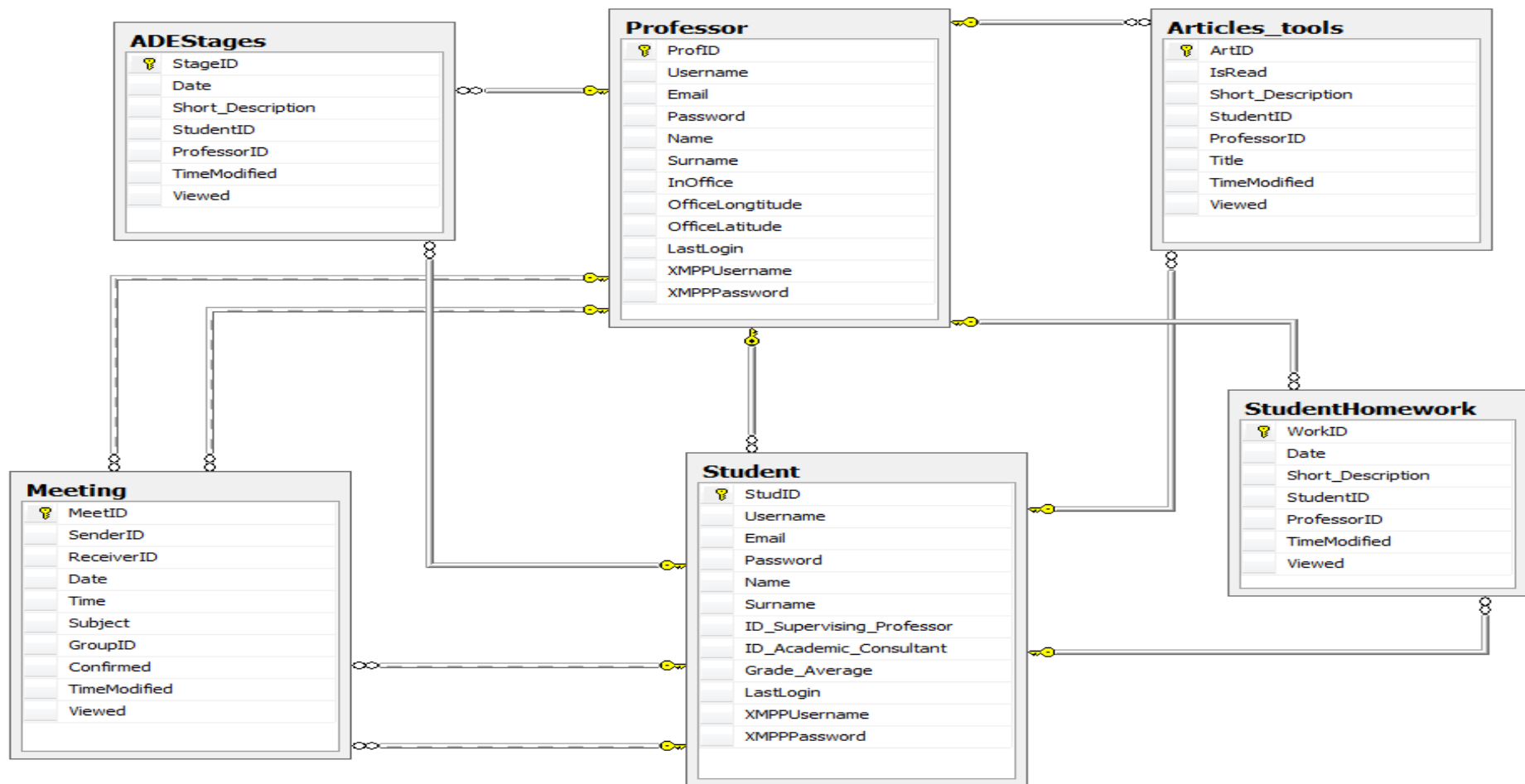
Στο αρχείο `IRestServiceImplPost.cs` περιέχεται το διεπαφή ή αλλιώς το συμβόλαιο της υπηρεσίας διαδικτύου (web service). Μέσω αυτού του διεπαφή καθορίζεται ο τύπος των HTTP αιτήσεων που δέχεται η υπηρεσία διαδικτύου (web service), στην περίπτωση μας POST αιτήσεις. Επίσης καθορίζεται ο η μορφή των δεδομένων που παίρνει η υπηρεσία διαδικτύου (web service) και την μορφή των δεδομένων που στέλνει πίσω η υπηρεσία διαδικτύου (web service). Σε αυτή υλοποίηση, όπως αναφέρθηκε και πριν, και τα δύο πεδία παίρνουν την τιμή JSON. Επιπλέον στο διεπαφή καθορίζεται το UriTemplate το οποίο αφορά το τελευταίο σκέλος του URL που χρησιμοποιείται για πρόσβαση στην υπηρεσία διαδικτύου (web service). Για παράδειγμα, σε αυτή την υλοποίηση έχουμε UriTemplate ίσο με `JSONData`, έτσι για να έχει πρόσβαση κάποιος στην υπηρεσία διαδικτύου (web service) πρέπει να στείλει μια HTTP αίτηση στο `www3.cs.ucy.ac.cy/professor2studentWS/JSONData`. Τέλος, στο διεπαφή περιέχεται και μια συνάρτηση η οποία θα υλοποιηθεί στο άλλο αρχείο και ουσιαστικά περιέχει όλη την λειτουργία της υπηρεσίας διαδικτύου (web service).

Στο αρχείο `RestServiceImplPost.svc.cs` περιέχεται η υλοποίηση της κλάσης `RestServiceImplPost` η οποία υλοποιεί την διεπαφή που αναλύθηκε πιο πάνω. Στο αρχείο αυτό υπάρχει η μέθοδος `JSONData` η οποία καλείται κάθε φορά που γίνεται μια αίτηση στην υπηρεσία διαδικτύου (web service). Στην μέθοδο αυτή περνά σαν παράμετρο το query. Αρχικά το query επεξεργάζεται για να έρθει στην αρχική του μορφή δηλαδή πριν από την επεξεργασία που έγινε στην android εφαρμογή. Ακολούθως ελέγχεται ο τύπος του query, πιθανοί τύποι είναι SELECT, INSERT, UPDATE, DELETE και FORGET. Ο τελευταίος τύπος αναφέρεται στην λειτουργία για επαναφορά του κωδικού του χρήστη. Οι υπόλοιποι τύποι αφορούν τα βασικά SQL query. Για τους τύπους INSERT, UPDATE, DELETE εκτελείται το query και επιστρέφεται Done ή Not Done αναλόγως αν έχει ολοκληρωθεί σωστά το query. Στην περίπτωση που ο τύπος του query είναι SELECT, εκτελείται το query, το αποτέλεσμα το οποίο αποθηκεύεται σε `dataTable` κωδικοποιείται σε JSON μορφή και επιστρέφεται στην android εφαρμογή.

Επίσης στο αρχείο web.config καθορίζονται κάποιες επιπλέον σημαντικές παράμετροι, όπως για παράδειγμα ο καθορισμός της διεπαφής στο αρχείο IRestServiceImplPost.cs ως το συμβόλαιο της υπηρεσίας διαδικτύου (web service).

## 4.2 Υλοποίηση βάσης δεδομένων

Η βάση δεδομένων του συστήματος έχει σχεδιαστεί χρησιμοποιώντας το SQL server 2008 και είναι εγκατεστημένη στον IIS server του πανεπιστημίου. Η βάση δεδομένων διαθέτει έξι πίνακες οι οποίοι είναι οι εξής, Student, Professor, ADEStages, Meeting, Articles\_tools και StudentHomework. Περισσότερες λεπτομέρειες σχετικά με τα χαρακτηριστικά του κάθε πίνακα υπάρχουν στην ενότητα 2.2. Κάποια χαρακτηριστικά διαθέτουν προκαθορισμένη αρχική (default) τιμή. Το χαρακτηριστικό LastLogin στους πίνακες Professor και Student παίρνει αρχική τιμή ίση με την ημερομηνία και ώρα την συγκεκριμένη στιγμή. Με τον ίδιο τρόπο αρχικοποιείται και το χαρακτηριστικό TimeModified στους υπόλοιπους πίνακες. Επίσης στους άλλους πίνακες αρχικοποιείται με false το χαρακτηριστικό Viewed. Επιπλέον στον πίνακα Professor αρχικοποιείται με false το χαρακτηριστικό InOffice. Στο σχήμα 4.2.1 παρουσιάζονται οι συσχετίσεις μεταξύ των πινάκων.



4.2.1 Σχήμα συσχετίσεων

Όπως φαίνεται από το σχήμα οι δύο κυριότεροι πίνακες στην βάση δεδομένων είναι ο πίνακας Student και ο πίνακας Professor. Τα πρωτεύοντα κλειδιά των πινάκων αυτών τοποθετούνται σαν foreign key σε όλους τους υπόλοιπους πίνακες. Στους πίνακες ADEStages, Articles\_tools και StudentHomework τα πεδία StudentID και ProfessorID είναι ξένα κλειδιά (foreign keys) και σχετίζονται με το StudID του πίνακα Student και το ProfID του πίνακα Professor αντίστοιχα. Στον πίνακα Meeting, τα πεδία SenderID και ReceiverID είναι ξένα κλειδιά (foreign keys) και σχετίζονται και τα δύο με το StudID του πίνακα Student και το ProfID του πίνακα Professor. Επίσης οι πίνακες Professor και Student συνδέονται μεταξύ τους καθώς υπάρχουν δύο ξένα κλειδιά (foreign keys) στον πίνακα Student, τα πεδία ID\_Supervising\_Professor και το ID\_Academic\_Consultant, τα οποία σχετίζονται με το ProfID του πίνακα Professor.

Πέρα από αυτούς τους έξι πίνακες η βάση δεδομένων διαθέτει και τρεις όψεις (Views). Οι τρεις όψεις (views) αυτές είναι hwview, meetingview, stagesview. Οι όψεις hwview και stagesview έχουν πανομοιότυπα χαρακτηριστικά τα οποία είναι: Date, Short\_Description, StudentID, ProfessorID, TimeModified, Viewed και ένα επιπλέον χαρακτηριστικό για κάθε όψη (view), WorkID και StageID αντίστοιχα. Η όψη meetingview διαθέτει τα χαρακτηριστικά MeetID, SenderID, ReceiverID, Date, Time, Subject, GroupID, Confirmed, TimeModified και Viewed. Τα χαρακτηριστικά των όψεων (views) έχουν τον ίδιο τύπο με τα χαρακτηριστικά των πινάκων που σχετίζονται. Η όψη hwview σχετίζεται με τον πίνακα StudentHomework, η όψη stagesview με τον πίνακα ADEStages και η όψη meetingview με τον πίνακα Meetings. Οι όψεις ουσιαστικά περιέχουν ακριβώς τα ίδια δεδομένα με τους πίνακες που σχετίζονται αλλά ταξινομημένα με βάση την ημερομηνία. Η ταξινόμηση αυτή χρησιμεύει στην φόρτωση αυτών των γεγονότων (event) στο ημερολόγιο της android εφαρμογής.

#### **4.3 Υλοποίηση εφαρμογής**

Σε αυτή την ενότητα θα αναλυθεί λεπτομερώς ο τρόπος και οι τεχνικές που χρησιμοποιήθηκαν για την υλοποίηση των κυριότερων λειτουργιών της android εφαρμογής.

Πρώτα να αναφέρουμε κάποια βασικά χαρακτηριστικά των android εφαρμογών. Μια εφαρμογή για το λειτουργικό σύστημα android αποτελείται κυρίως από πολλά activities, ενώ μπορεί να διαθέτει επίσης services και broadcast receivers.

Ένα Activity είναι μια σελίδα στην εφαρμογή. Ουσιαστικά activity είναι η κάθε κλάση η οποία διαθέτει δική της γραφική διεπαφή χρήστη. Ένα activity μπορεί να τρέχει μόνο στο προσκήνιο και κλείνει όταν ο χρήστης μετακινηθεί σε ένα άλλο activity.

Ένα Service είναι το συστατικό της εφαρμογής το οποίο τρέχει συνεχώς συνήθως για την εκτέλεση κάποιας χρονοβόρας διαδικασίας (π.χ. κατέβασμα αρχείου) ή για την εκτέλεση κάποιας λειτουργίας που δεν απαιτείται αλληλεπίδραση με τον χρήστη (π.χ. εξακρίβωση θέσης). Τα service δεν διαθέτουν γραφική διεπαφή χρήστη.

Τέλος ένα broadcast receiver είναι το συστατικό της εφαρμογής το οποίο «ακούει» και περιμένει για ένα συγκεκριμένο broadcast που προκαλεί κάποιο event. Για παράδειγμα ο broadcast receiver μπορεί να καλείται στο ξεκίνημα της συσκευής ή στο ξεκίνημα της εφαρμογής. Μόλις συμβεί το συγκεκριμένο event το broadcast receiver θα εκτελέσει τον κώδικά του και θα τερματίσει.

Επίσης μια εφαρμογή σε android διαθέτει και άλλα σημαντικά στοιχεία όπως το Manifest αρχείο το οποίο περιέχει σημαντικές λεπτομέρειες για την εφαρμογή π.χ. ποιες εκδόσεις του android υποστηρίζει η εφαρμογή κτλπ, το strings αρχείο το οποίο περιέχει το κείμενο που χρησιμοποιείται στην εφαρμογή και το styles αρχείο που περιέχει τα θέματα και άλλα δεδομένα που έχουν να κάνουν με την εμφάνιση της εφαρμογής.

#### **4.3.1 Γραφική διεπαφή χρήστη**

Μια διεπαφή χρήστη για android εφαρμογή είναι γραμμένη σε XML και διαθέτει διάφορα Views και Viewgroups. Views είναι τα στοιχεία τα οποία αποτελούν το γραφικό περιβάλλον και προσφέρουν δυνατότητα αλληλεπίδρασης με την εφαρμογή στον χρήστη. Παραδείγματα view είναι το button, το textview, το checkbox κτλπ. Viewgroups είναι σύνολα από views τα οποία επίσης υποδεικνύουν

το πώς είναι δομημένα τα Views μέσα στο συγκεκριμένο group. Παραδείγματα viewgroup είναι το RelativeLayout που επιτρέπει την τοποθέτηση των views οπουδήποτε στο viewgroup (στο πάνω/κάτω μέρος του group, δεξιά/αριστερά από ένα άλλο view, πάνω/κάτω από ένα άλλο view κτλπ.) και το LinearLayout που επιτρέπει την τοποθέτηση views στην σειρά (το ένα δίπλα στο άλλο ή το ένα πάνω από το άλλο) κτλπ. Επίσης αξίζει να αναφερθεί πως μέσα σε ένα Viewgroup μπορούν να υπάρχουν πέρα από Views και Viewgroups. Για παράδειγμα μέσα σε ένα LinearLayout μπορεί να υπάρχει ένα button και ένα RelativeLayout. Ακολουθεί ένα παράδειγμα αρχείου (σχήμα 4.3.1.1) που περιέχει μια γραφική διεπαφή. Στο παράδειγμα αυτό υπάρχει ένα LinearLayout μέσα στο οποίο υπάρχουν δύο RelativeLayouts. Μέσα σε κάθε RelativeLayout υπάρχει ένα textview και ένα imageView.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <RelativeLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" >

        <TextView
            android:id="@+id/supervisor"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentLeft="true"
            android:text="@string/Supervisor"
            android:textSize="22dp" >
        </TextView>

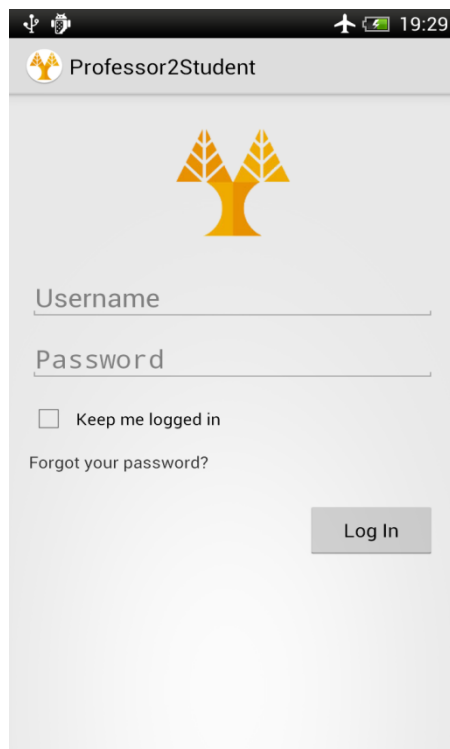
        <ImageView
            android:id="@+id/OnlineOffline_supervisor"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentRight="true"
            android:layout_centerVertical="true"
            android:contentDescription="@drawable/online"
            android:src="@drawable/offline" >
        </ImageView>
    </RelativeLayout>

    <RelativeLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" >

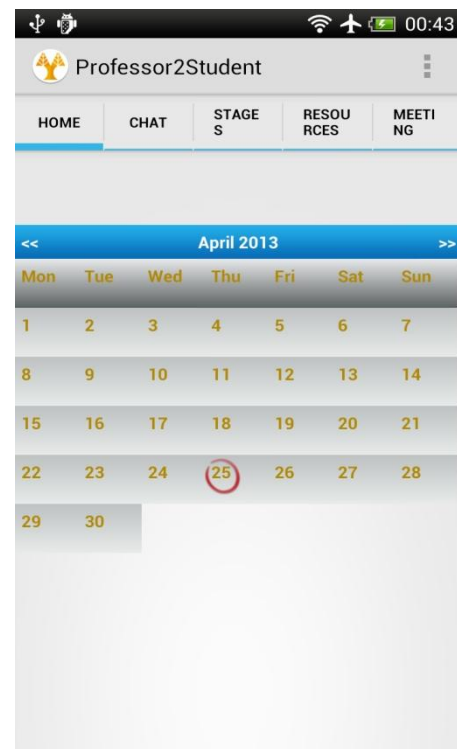
        <TextView
            android:id="@+id/acaconsultant"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
```

#### 4.3.1.1 CheckPositionActivity.xml

Αρχικά θα αναλύσουμε πώς είναι σχεδιασμένη η γραφική διεπαφή χρήστη. Το πρώτο παράθυρο που θα εμφανιστεί στον χρήστη μόλις ξεκινήσει την εφαρμογή θα αφορά το log in, παρόμοιο με το σχήμα 4.3.1.2. Μετά από το log in θα εμφανιστεί το παράθυρο home, που αποτελεί την κεντρική σελίδα της εφαρμογής. Στο 4.3.1.3 φαίνεται ένα παράδειγμα του home. Σε αυτό το παράθυρο όπως και σε όλα τα κύρια παράθυρα υπάρχει ένα tab host με 5 tabs. Επίσης ένα άλλο σημαντικό στοιχείο στο home είναι το ημερολόγιο που βρίσκεται στο κέντρο της οθόνης, το ημερολόγιο αυτό δημιουργήθηκε εξ ολοκλήρου από την αρχή με την χρήση ενός grid view. Στο ημερολόγιο αυτό μαρκάρονται οι μέρες στις οποίες υπάρχει κάποιο καθορισμένο event καθώς και η σημερινή μέρα. Οι μέρες με κάποιο event διαθέτουν ένα μπλε σημάδι στην δεξιά πάνω γωνία ενώ η σημερινή μέρα κυκλώνεται με κόκκινο κύκλο.



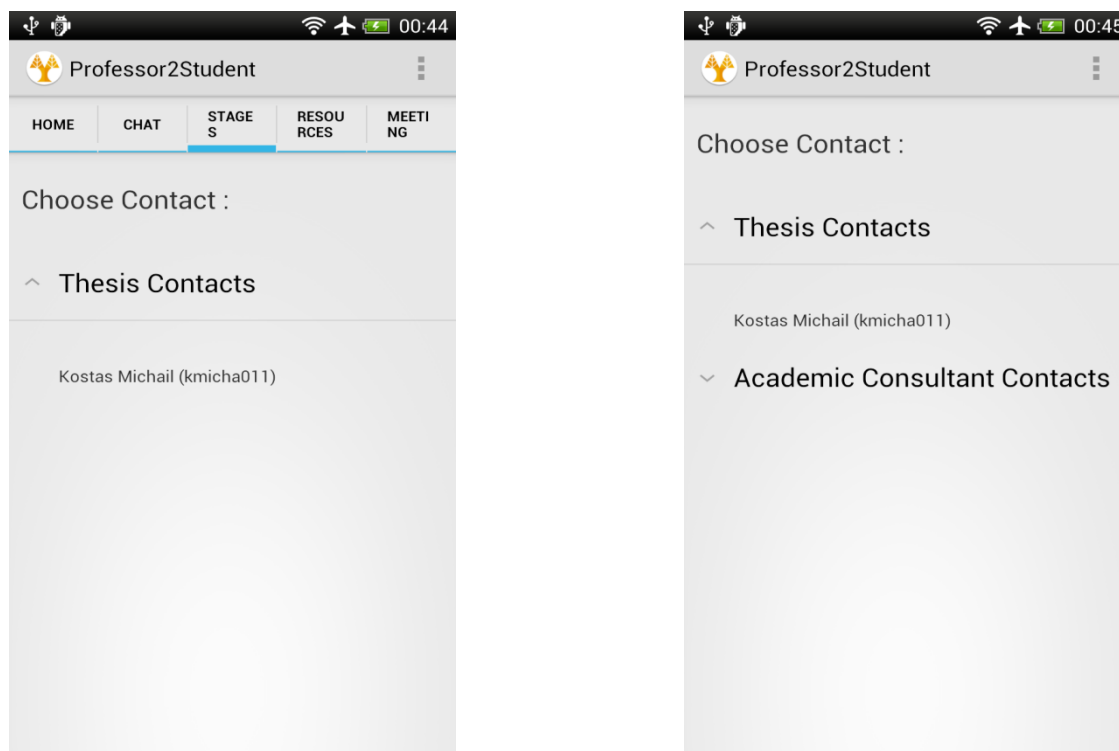
4.3.1.2 Log in activity



4.3.1.3 Home activity

Επίσης σημαντικό στοιχείο της διεπαφής χρήστη είναι το παράθυρο για την επιλογή επαφής. Το παράθυρο αυτό εμφανίζεται όταν ο χρήστης επιλέξει να θέσει ένα στάδιο, να κάνει αναφορά σε ένα άρθρο/εργαλείο ή να προγραμματίσει μια συνάντηση. Αυτό το παράθυρο έχει δύο εκδοχές, στην μια υπάρχουν δύο expandable lists, η μια λίστα αφορά τις επαφές που σχετίζονται με τον ακαδημαϊκό σύμβουλο και

η άλλη επαφές που σχετίζονται με την διπλωματική εργασία. Στην άλλη εκδοχή υπάρχει μόνο η λίστα με επαφές για την διπλωματική εργασία. Παράδειγμα των δύο εκδοχών υπάρχει στο σχήμα 4.3.1.4.

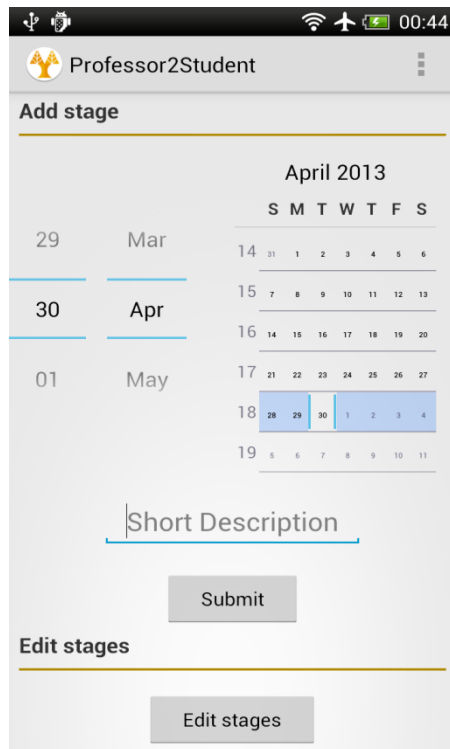


#### 4.3.1.4 Activity για επιλογή επαφής

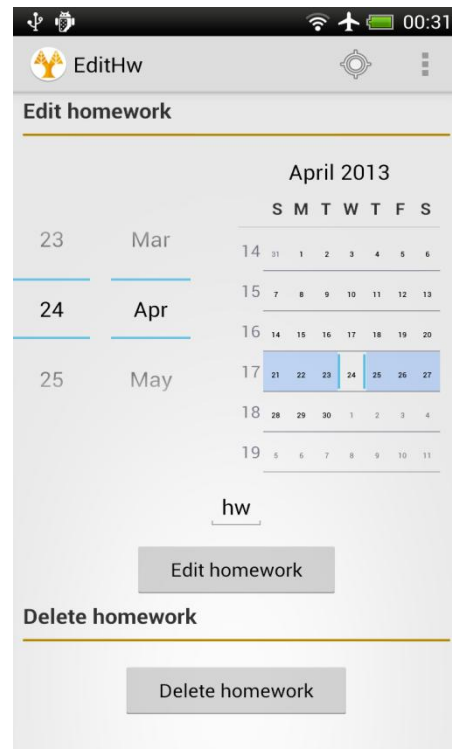
Μετά από την επιλογή επαφής, θα εμφανιστεί στον χρήστη ένα παράθυρο για να καταχωρήσει τα απαραίτητα στοιχεία π.χ. για στάδιο στην διπλωματική πρέπει να δοθεί ημερομηνία και μια σύντομη περιγραφή. Παράδειγμα φαίνεται στο σχήμα 4.3.1.5. Για κάθε event που καταχωρείται υπάρχει η επιλογή της επεξεργασίας και της διαγραφής του. Το event αρχικά επιλέγεται από μια λίστα και ακολούθως μπορεί να επιλεγεί αν θα διαγραφεί ή επεξεργαστεί. Παράδειγμα υπάρχει στο σχήμα 4.3.1.6.

Υπάρχουν επίσης και άλλα ξεχωριστά παράθυρα όπως για παράδειγμα το παράθυρο για ανταλλαγή μηνυμάτων και ο έλεγχος για την θέση του καθηγητή, παραδείγματα για τα δύο αυτά παράθυρα υπάρχουν στο 4.3.1.7 και στο 4.3.1.8. Στο παράθυρο για ανταλλαγή μηνυμάτων υπάρχει το όνομα του παραλήπτη, το αν είναι online ή όχι και ακολουθούν τα μηνύματα που ανταλλάσσονται μεταξύ των χρηστών. Στο παράθυρο για έλεγχο της θέσης του καθηγητή υπάρχει το όνομα του επιβλέποντα

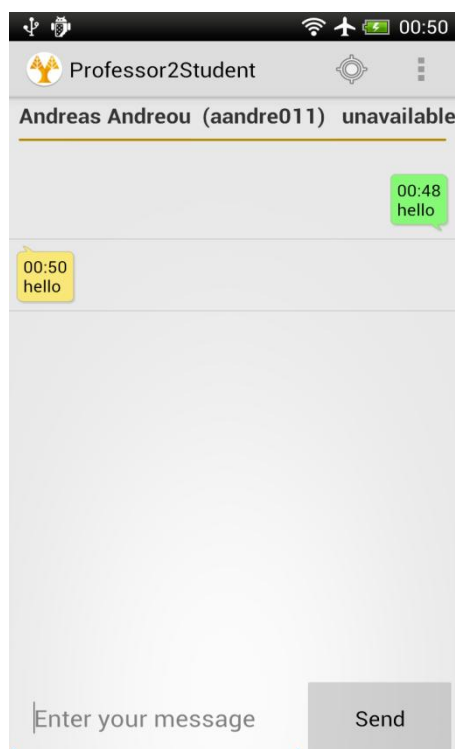
καθηγητή και του ακαδημαϊκού συμβούλου και δίπλα μια πράσινη ή γκριζα κουκκίδα που υποδηλώνει αν βρίσκεται ή όχι ο καθηγητής στο γραφείο του.



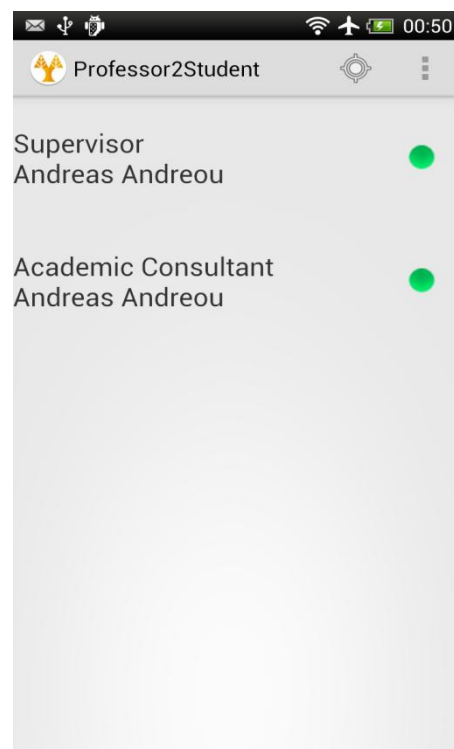
4.3.1.5 Activity για δημιουργία σταδίου



4.3.1.6 Activity για επεξεργασία σταδίου



4.3.1.7 Activity για ανταλλαγή μηνυμάτων



4.3.1.8 Activity για έλεγχο θέσης καθηγητή

### 4.3.2 Υλοποίηση λειτουργιών

Η δημιουργία Activity, Service και Broadcast receiver για μια android εφαρμογή απαιτεί μια συγκεκριμένη διαδικασία.

Ξεκινώντας από το Activity, για την δημιουργία μιας κλάσης-Activity απαιτείται πρωτίστως η κλάση αυτή να είναι κληρονόμος της κλάσης android.app.activity. Η κλάση αυτή περιέχει την abstract μέθοδο onCreate() η οποία πρέπει υποχρεωτικά να υλοποιηθεί στην καινούργια κλάση-Activity. Η μέθοδος αυτή καλείται από το σύστημα κάθε φορά που ξεκινάει το συγκεκριμένο Activity.

Για την δημιουργία μιας κλάσης-Service απαιτείται πρωτίστως η κλάση αυτή να είναι κληρονόμος της κλάσης android.app.service ή της κλάσης android.app.intentservice. Η κλάση android.app.service περιέχει την abstract μέθοδο onCreate(), την μέθοδο onStartCommand() και την μέθοδο onBind() οι οποίες πρέπει υποχρεωτικά να υλοποιηθούν στην καινούργια κλάση-Service. Η μέθοδος onCreate() καλείται μόνο όταν το service πρώτη φορά καλείται και δεν είναι δημιουργημένο, η μέθοδος onStartCommand() καλείται κάθε φορά που μια εφαρμογή ξεκινάει το ήδη δημιουργημένο service και η μέθοδος onBind() καλείται όταν μια εφαρμογή ζητήσει να δεσμεύσει το service.

Για την δημιουργία μιας κλάσης-Broadcast receiver απαιτείται πρωτίστως η κλάση αυτή να είναι κληρονόμος της κλάσης android.content.BroadcastReceiver. Η κλάση αυτή περιέχει την abstract μέθοδο onReceive() η οποία πρέπει υποχρεωτικά να υλοποιηθεί στην καινούργια κλάση-BroadcastReceiver. Η μέθοδος αυτή καλείται όταν σταλθεί η μετάδοση για την οποία περιμένει ο BroadcastReceiver.

Πέρα από τις βασικές λειτουργίες, η υλοποίηση σχεδόν όλων των λειτουργιών της εφαρμογής απαιτεί την επικοινωνία με την βάση δεδομένων που βρίσκεται εγκατεστημένη στον εξυπηρετητή του πανεπιστημίου. Ο τρόπος με τον οποίο επικοινωνεί η εφαρμογή με την βάση δεδομένων έχει αναλυθεί στην ενότητα 2.3. Σε αυτή την ενότητα θα περιγραφεί με περισσότερη λεπτομέρεια ο τρόπος αποστολής αιτήσεων από την εφαρμογή προς τον εξυπηρετητή.

Για την επικοινωνία με τον εξυπηρετητή έχει δημιουργηθεί μια κλάση με το όνομα DBconnect η οποία κληρονομεί από την κλάση android.os.AsyncTask. Η κλάση αυτή επιτρέπει την εκτέλεση χρονοβόρων διεργασιών ασύγχρονα.

Κληρονομώντας από την κλάση αυτή απαιτείται η υλοποίηση της μεθόδου `doInBackground()`. Στην υλοποίηση αυτής της μεθόδου περιέχεται ουσιαστικά όλη η διαδικασία που εκτελείται από την πλευρά της εφαρμογής για την σύνδεση με την βάση δεδομένων. Στην μέθοδο αυτή περνά σαν παράμετρος το `query` που θέλουμε να εκτελεστεί. Ακολουθώς το `query` αυτό κωδικοποιείται με την AES κωδικοποίηση και αλλάζουν κάποιοι χαρακτήρες που μπορεί να δημιουργήσουν πρόβλημα στην μεταφορά. Μετά δημιουργείται η HTTP POST αίτηση και στέλνεται προς τον εξυπηρετητή. Τέλος, διαβάζεται η απάντηση του εξυπηρετητή και επιστρέφεται στο Activity ή Service για αποκωδικοποίηση της από την JSON μορφή στην οποία βρίσκεται. Λειτουργίες που χρησιμοποιούν αυτή τη διαδικασία είναι οι εξής: Προσθήκη, επεξεργασία και διαγραφή σταδίων διπλωματικής, δηώσεων εργασιών, άρθρων/εργαλείων συναντήσεων, το `log in`, η επιλογή επαφής και οποιεσδήποτε άλλες λειτουργίες οι οποίες απαιτούν ανάγνωση, επεξεργασία ή προσθήκη δεδομένων από και προς την βάση δεδομένων. Πιο κάτω υπάρχει κομμάτι του κώδικα που υπάρχει στην μέθοδο `doInBackground()` (σχήμα 4.3.2.1) και ο κώδικας για την κωδικοποίηση του `query` (σχήμα 4.3.2.2).

```

HttpPost httpPost = new HttpPost(
    "http://www3.cs.ucy.ac.cy/professor2studentWS/RestServiceImplPost.svc/JSONData");
// topo8etisi tou query se JSON object
JSONObject query = new JSONObject();
try {
    query.put("query", type);
} catch (Exception ex) {
}
String text = "{\"JSONDataResult\":{\"json\":\"[]\"}}";
String message;
// an i siskeui einai sindedemeni sto diadiktio
if (isNetworkAvailable()) {
    try {
        // dimiourgise to minima xrisimopointas to query
        message = query.toString();
        message = "{\"query\":\"" + message + "\"}";
        // pros8esi header sto http post request
        httpPost.setEntity(new StringEntity(message, "UTF8"));
        httpPost.setHeader("Content-type", "application/json");
        // apostoli request kai paralavi apantisis
        HttpResponse response = httpClient.execute(httpPost,
            localContext);

        // epe3ergasia apantisis
        HttpEntity entity = response.getEntity();
        text = getASCIISContentFromEntity(entity);
    }
}

```

#### 4.3.2.1 DBconnect

```

public String encryptText(String cipherText) throws Exception {

    // dimiourgia key specification me ton algorithmo AES
    SecretKeySpec ks = new SecretKeySpec(
        "*****".getBytes("UTF-8"), "AES");

    // dimiourgia parameter specification
    IvParameterSpec iv = new IvParameterSpec(
        "*****".getBytes("ASCII"));

    // dimiourgia kai initialization tou cipher
    Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
    cipher.init(Cipher.ENCRYPT_MODE, ks, iv);

    // encode tou string
    byte[] bytes = cipher.doFinal(cipherText.getBytes());
    byte[] encoded = Base64.encodeBase64(bytes);

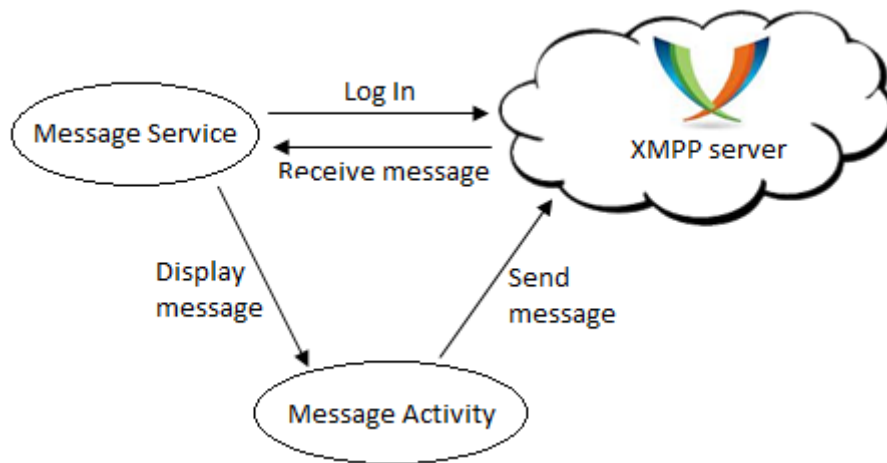
    return new String(encoded, "ASCII");
}

```

#### 4.3.2.2 Κωδικοποίηση query

Υπάρχουν επίσης λειτουργίες που χρησιμοποιούν και άλλες τεχνικές όπως η λειτουργία για την ανταλλαγή μηνυμάτων και η λειτουργία για την εξακρίβωση της θέσης του καθηγητή.

Για την υλοποίηση της λειτουργίας ανταλλαγής μηνυμάτων χρησιμοποιήθηκε το XMPP πρωτόκολλο και η βιβλιοθήκη asmack[16] η οποία βοηθά στην υλοποίηση εφαρμογών που κάνουν χρήση του συγκεκριμένου πρωτόκολλου. Επίσης για την υλοποίηση της συγκεκριμένης λειτουργίας απαιτείται και η χρήση ενός δημόσιου XMPP εξυπηρετητή ο οποίος λειτουργεί σαν μεσολαβητής στην επικοινωνία δύο χρηστών. Σαν XMPP εξυπηρετητής επιλέχθηκε ο jabber.rootbash.com. Η επιλογή του συγκεκριμένου εξυπηρετητή έγινε βάση συγκεκριμένων κριτηρίων. Σημαντικά κριτήρια αποτελούσαν η υποστήριξη offline μηνυμάτων και η υποστήριξη ελληνικών χαρακτήρων. Η υλοποίηση της λειτουργίας ανταλλαγής μηνυμάτων έγινε μέσω ενός service, MessageService και ενός activity, MessageActivity. Η αλληλεπίδραση μεταξύ του activity, του service και του XMPP server περιγράφεται στο σχήμα 4.3.2.7.



#### 4.3.2.7 Υλοποίηση XMPP

Το `MessageService` ξεκινά μόλις ο χρήστης κάνει `log in` στην εφαρμογή. Μόλις ξεκινήσει το service καλείται η μέθοδος `onStartCommand()`. Μέσα σε αυτή τη μέθοδο αρχικά γίνεται το `log in` του χρήστη στον XMPP εξυπηρετητή. Ακολούθως προσθέτονται οι επαφές του (σχετικοί καθηγητές και φοιτητές) στο σύνολο των XMPP επαφών του και δημιουργείται ένας listener ο οποίος «ακούει» για νέα μηνύματα. Τέλος το service κάνει έλεγχο για offline μηνύματα και αν υπάρχουν τα περνά στον listener ο οποίος επεξεργάζεται τα μηνύματα αυτά και τα εμφανίζει στο `listview` του `MessageActivity`.

Το `messageActivity` περιέχει ένα `listview` το οποίο περιέχει τα μηνύματα που ανταλλάχθηκαν καθώς επίσης και ένα `edittext` για να γράψει ο χρήστης το μήνυμα και ένα `button` για να στείλει το μήνυμα. Κάθε φορά που στέλνεται ή παραλαμβάνεται ένα μήνυμα προστίθεται στην λίστα μηνυμάτων και καλείται η μέθοδος `setListAdapter()`, η οποία ανανεώνει το `listview` έτσι ώστε να περιέχει και τα καινούργια μηνύματα. Για το λειτουργία αυτή δημιουργήθηκε ένας listener ο οποίος εντοπίζει αλλαγές που γίνονται στην λίστα μηνυμάτων. Επίσης, αν παραληφθεί μήνυμα την στιγμή που ο χρήστης δεν βρίσκεται στο `MessageActivity` ή βρίσκεται στο `MessageActivity` αλλά συνομιλεί με κάποια άλλη επαφή, θα σταλθεί στον χρήστη ένα `android notification` μαζί με δόνηση της συσκευής και ήχο ειδοποίησης, χρησιμοποιώντας το `NotificationManager`. Παράδειγμα του κώδικα για το `MessageService` φαίνεται στο σχήμα 4.3.2.3 και για το `MessageActivity` στο σχήμα 4.3.2.4.

```

connection.connect();
SASLAuthentication.supportSASLMechanism("PLAIN", 0);
connection.login(settings.getString("XMPPUsername", ""),
decryptText(settings.getString("XMPPPassword", "")));
        decryptText(settings.getString("XMPPPassword", "")));
System.out.println(connection.isAuthenticated());
Presence presence = new Presence(Presence.Type.unavailable);
connection.sendPacket(presence);

roster.createEntry(dbresult.getJSONObject(i).getString("XMPPUsername"),
        dbresult.getJSONObject(i).getString("Name"))

connection.addPacketListener(new PacketListener() {
    public void processPacket(Packet packet) {
        Message message = (Message) packet;
        if (message.getBody() != null) {
            final String fromName = StringUtils
                .parseBareAddress(message.getFrom());
            DateFormat dateFormat = new SimpleDateFormat(
                "HH:mm");

            Calendar cal = Calendar.getInstance();
            final Msg m = new Msg(message.getBody(), 0,
                dateFormat.format(cal.getTime()));
            messages.add(m);
            mHandler.post(new Runnable() {
                public void run() {

                    MessageActivity.mMessageValueModel.get(fromName).addMsg(m);
                }
            });
        }
    }
});

```

#### 4.3.2.3 MessageService

```

// custom listener gia allages stin lista
mMessageValueModel.get(getReceiver()).setListener(
    new ValueChangeListener() {
        public void onValueChanged(ArrayList<Msg> messages) {
            // TODO Auto-generated method stub
            setListAdapter();
        }
    });

public void setListAdapter() {
    // anavathmisi tou listview me ta minimata
    ArrayAdapter<Msg> adapter = new MessageArrayAdapter(mContext,
        R.layout.listitem discuss, mMessageValueModel
            .get(getReceiver()).getValue());
    listView.setAdapter(adapter);
    listView.setSelection(listView.getAdapter().getCount() - 1);
}

```

#### 4.3.2.4 MessageActivity

Για την υλοποίηση της λειτουργίας εξακρίβωσης θέσης του καθηγητή χρησιμοποιήθηκαν δύο τεχνικές. Ο ένας τρόπος για εξαγωγή της θέσης του καθηγητή είναι με την χρήση του LocationManager του android και ο άλλος τρόπος με την χρήση του RSCM. Και για τις δύο μεθόδους χρησιμοποιείται ένας BroadcastReceiver και ένα Service. Η μέθοδος που χρησιμοποιεί τον LocationManager απαιτεί την χρήση ενός επιπλέον BroadcastReceiver. Ο χρήστης μπορεί να επιλέξει ποια μέθοδο εξαγωγής θέσης επιθυμεί να χρησιμοποιήσει από το μενού settings. Αν ο χρήστης επιλέξει την επιλογή “continuously” για update interval, θα χρησιμοποιηθεί το RSCM ενώ αν επιλέξει κάποιο χρονικό διάστημα θα χρησιμοποιηθεί το LocationManager του android. Επίσης ο καθηγητής για να ενεργοποιήσει την λειτουργία για εξακρίβωση θέσης πρέπει να χρησιμοποιήσει το SetOfficeActivity για να δώσει την τοποθεσία του γραφείου του.

Πρώτα θα αναλυθεί η τεχνική εξαγωγής θέσης με τη χρήση του LocationManager. Με την εκκίνηση της συσκευής ή της εφαρμογής στέλνεται μια μετάδοση για την οποία περιμένει ο BroadcastReceiver MyScheduleReceiver. Αμέσως εκτελείται η μέθοδος onReceive(). Η μέθοδος αυτή χρησιμοποιείται και για τις δύο τεχνικές εξαγωγής θέσης. Για την τεχνική με τον LocationManager δημιουργείται ένα AlarmManager στο οποίο δίνετε το χρονικό περιθώριο στο οποίο πρέπει να στέλνεται ένα alarm. Το χρονικό περιθώριο αυτό θα είναι ίσο με την επιλογή που έδωσε ο χρήστης σαν update interval στο μενού settings (προκαθορισμένη τιμή είναι 30). Δουλειά του AlarmManager είναι να στέλνει μια μετάδοση (alarm) κάθε ένα προκαθορισμένο διάστημα που καθορίζεται από τον χρήστη. Ακολούθως η μετάδοση αυτή λαμβάνεται από τον δεύτερο BroadcastReceiver MyStartServiceReceiver. Αμέσως καλείται η μέθοδος onReceive() του MyScheduleReceiver. Η μέθοδος αυτή ξεκινάει το service LocationReasoner αφού πρώτα ελέγξει αν ο χρήστης επιτρέπει τον έλεγχο της θέσης του. Στο συγκεκριμένο service, αρχικά στην μέθοδο onCreate() δημιουργείται το αντικείμενο για τον LocationManager. Ακολούθως στην μέθοδο onStartCommand() ζητείται αναβάθμιση θέσης, θέτεται ο listener για τις αναβαθμίσεις και υλοποιούνται οι απαραίτητες μέθοδοι. Η μέθοδος η οποία χρειάζεται απαραίτητα να υλοποιηθεί είναι η μέθοδος onLocationChanged(). Η μέθοδος αυτή καλείται μόλις υπάρξει αλλαγή στην θέση. Μέσα σε αυτή τη μέθοδο, αρχικά διαβάζεται οι συντεταγμένες της νέας θέσης και οι συντεταγμένες της θέσης γραφείου. Ακολούθως, υπολογίζεται η απόσταση μεταξύ

των δύο αυτών συντεταγμένων και αν η απόσταση αυτή είναι μικρότερη από 50 μέτρα αποθηκεύεται στην βάση πως ο καθηγητής βρίσκεται στο γραφείο του. Παράδειγμα κώδικα από το MyScheduleReceiver υπάρχει στο σχήμα 4.3.2.5 και του LocationReasoner στο 4.3.2.6.

```
AlarmManager service = (AlarmManager) context
    .getSystemService(Context.ALARM_SERVICE);
Intent i = new Intent(context, MyStartServiceReceiver.class);
PendingIntent pending = PendingIntent.getBroadcast(context, 0, i,
    PendingIntent.FLAG_CANCEL_CURRENT);
Calendar cal = Calendar.getInstance();
// Start 30 seconds after boot completed
cal.add(Calendar.SECOND, 30);
// Fetch every 30 seconds
// InexactRepeating allows Android to optimize the energy
consumption
    if (settings.getBoolean("location", false)) {
        if (settings.getString("update_interval", "30").equals("1"))
        {
            Intent intent2 = new Intent(context,
RSCMlocation.class);
            context.startService(intent2);
        } else {
            Intent intent3 = new Intent(context,
RSCMlocation.class);
            context.stopService(intent3);
            service.setInexactRepeating(AlarmManager.RTC_WAKEUP,
cal.getTimeInMillis(), Long.parseLong(settings.getString("update_interval",
",
        "30"))*60*1000,
            pending);
```

#### 4.3.2.5 MyScheduleReceiver

```

locationManager.requestSingleUpdate(LocationManager.NETWORK_PROVIDER, this,
null)
public void onLocationChanged(final Location location) {
    final double longitude = location.getLongitude();
    final double latitude = location.getLatitude();
    double d2r = Math.PI / 180;
    double dlong = (OfficeLongitude - longitude) * d2r;
    double dlat = (OfficeLatitude - latitude) * d2r;
    double a = Math.pow(Math.sin(dlat / 2.0), 2) + Math.cos(latitude *
d2r)
        * Math.cos(OfficeLatitude * d2r)
        * Math.pow(Math.sin(dlong / 2.0), 2);
    double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
    double d = 6367000 * c;
    // an i apostasi einai mikroteri apo 50 metra ipologizete pws o
    // kathigitis vriskete sto grafeio tou
    if (d <= 50) {
        String query2 = "UPDATE Professor SET InOffice = 'True' WHERE
ProfID = '" + ProfID + "'";
        new DBconnect().execute(query2);
    } else {
        String query2 = "UPDATE Professor SET InOffice = 'False'
WHERE ProfID = '" + ProfID + "'";
        new DBconnect().execute(query2);
    }
}

```

#### 4.3.2.6 LocationReasoner

Για την τεχνική εξαγωγής θέσης με την χρήση του μεσισμικού RSCM πρέπει να αναφέρουμε πως απαιτείται η χρήση του RSCM runtime και του plugin για coarse location (βλ. ενότητα 3.4). Το plugin έχει υλοποιηθεί με την χρήση του LocationManager και την αίτηση για περιοδικά updates. Το plugin έχει εξαχθεί σε ένα apk και πρέπει να είναι εγκατεστημένο στην συσκευή. Αρχικά χρησιμοποιεί και αυτή η τεχνική τον BroadcastReceiver MyScheduleReceiver. Σε αυτή την περίπτωση δεν δημιουργείται AlarmManager, απλά ξεκινάει το service RSCMlocation. Το service αυτό υλοποιεί την μέθοδο onContextValueChanged() η οποία καλείται μόλις έχουμε νέα τιμή για την θέση του χρήστη. Ακολούθως η διαδικασία είναι παρόμοια με αυτή που εκτελείται και στην προηγούμενη τεχνική, όπου συγκρίνονται οι συντεταγμένες μεταξύ της νέας θέσης και της θέσης γραφείου και αν η απόσταση είναι μικρότερη από 50 μέτρα αποθηκεύεται στην βάση πως ο καθηγητής βρίσκεται στο γραφείο του. Ο κώδικας που χρησιμοποιείται για σύγκριση της θέσης είναι παρόμοιος με τον κώδικα που χρησιμοποιείται στο LocationReasoner (σχήμα 4.3.2.6)

μόνο που αντί για την μέθοδο `onLocationChanged()` υπάρχει η μέθοδος `onContextValueChanged()`.

#### **4.4 Υλοποίηση διαδικτυακής εφαρμογής**

Η διαδικτυακή εφαρμογή έχει υλοποιηθεί με την χρήση της γλώσσας C# μέσω του ASP.NET framework. Σκοπός της διαδικτυακής εφαρμογής είναι να παρέχει κάποιες βασικές λειτουργίες για επεξεργασία και δημιουργία λογαριασμών από φοιτητές, καθηγητές και τον διαχειριστή του συστήματος. Η διαδικτυακή εφαρμογή χωρίζεται βάση των τεσσάρων βασικών λειτουργιών της. Πρώτη βασική λειτουργία είναι η δημιουργία λογαριασμού από τους καθηγητές και η δεύτερη βασική λειτουργία αφορά την δημιουργία λογαριασμού από τους φοιτητές. Τρίτη βασική λειτουργία αποτελεί η επεξεργασία του λογαριασμού κάποιου χρήστη. Η επεξεργασία αυτή περιέχει, αλλαγή των στοιχείων του χρήστη, αλλαγή του κωδικού του και στην περίπτωση που ο χρήστης είναι φοιτητής, την δήλωση του μέσου όρου της βαθμολογίας του. Η τελευταία βασική λειτουργία αφορά την επεξεργασία των χρηστών και των δεδομένων από ένα διαχειριστή ή τους καθηγητές. Ο διαχειριστής μπορεί να προσθέσει, διαγράψει ή να επεξεργαστεί οποιοδήποτε φοιτητή ή καθηγητή, ενώ ο καθηγητής διαθέτει τις ίδιες δυνατότητες αλλά μόνο για φοιτητές. Οι λειτουργίες που αφορούν τους καθηγητές και τον διαχειριστή είναι κλειδωμένες μέσω των στοιχείων authentication και authorization στο web.config αρχείο.

##### **4.4.1 Αρχική σελίδα**

Ανοίγοντας την διαδικτυακή εφαρμογή εμφανίζεται η ακόλουθη σελίδα,

## Interaction Between Professors and Students

### Registration

Professor Registration

Student Registration

### Edit Account

Edit Account

Administrator Edit

Σε αυτή τη σελίδα εμφανίζονται τέσσερα κουμπιά για τις τέσσερις βασικές επιλογές που διαθέτει ο χρήστης (Professor Registration, Student Registration, Edit Account και Administrator Edit). Οι δύο πρώτες επιλογές αφορούν εγγραφή καθηγητή ή φοιτητή στο σύστημα και οι δύο τελευταίες αφορούν επεξεργασία λογαριασμών που βρίσκονται ήδη στο σύστημα.

#### 4.4.2 Εγγραφή καθηγητή

Αν ο χρήστης είναι καθηγητής και επιλέξει να εγγραφεί στο σύστημα επιλέγοντας το Professor Registration θα του εμφανιστεί η ακόλουθη σελίδα,

## Interaction Between Professors and Students

### Registration

Registration Code:

Continue Registration

Σε αυτή την σελίδα ζητείται από τον χρήστη να δώσει τον κωδικό, τον οποίο πρέπει να γνωρίζουν μόνο καθηγητές, για να προχωρήσει στην εγγραφή. Μόλις δώσει τον κωδικό θα του εμφανιστεί η ακόλουθη σελίδα,

## Register

\*Indicates required fields

Username: \*

Email: \*

Password/Κωδικός: \*

Confirm Password/Επιβεβαίωση κωδικού: \*

Name/Όνομα: \*

Surname/Επιθετο: \*

Create chat account [here](#)

Chat Username: \*

@jabber.rootbash.com

Chat Password/Κωδικός Chat: \*

Confirm Chat Password /  
Επιβεβαίωση κωδικού Chat: \*

Register

Σε αυτή τη σελίδα ο καθηγητής πρέπει να συμπληρώσει όλα τα στοιχεία (όλα είναι υποχρεωτικά). Τα πρώτα στοιχεία (Username, Email, Password, Name και

Surname) αφορούν προσωπικά στοιχεία του καθηγητή. Ακολούθως ο καθηγητής θα πρέπει να δημιουργήσει έναν jabber.rootbash.com λογαριασμό, χρησιμοποιώντας τον σύνδεσμο (link) που υπάρχει στην σελίδα, που θα του επιτρέπει να χρησιμοποιήσει τις λειτουργίες chat της εφαρμογής. Τέλος, χρησιμοποιώντας τα στοιχεία που έδωσε στην δημιουργία του jabber.rootbash.com λογαριασμού του, θα πρέπει να συμπληρώσει τα τελευταία στοιχεία (Chat Username, Chat password) και να ολοκληρώσει την εγγραφή του. Με την ολοκλήρωση της εγγραφής καταχωρείται ο καθηγητής στην βάση δεδομένων με την χρήση ενός INSERT query.

#### 4.4.3 Εγγραφή φοιτητή

Αν ο χρήστης είναι φοιτητής και επιλέξει να εγγραφεί στο σύστημα επιλέγοντας το Student Registration θα του εμφανιστεί η ακόλουθη σελίδα,

**Register** \*Indicates required fields

Username: \*

Email: \*

Password/Κωδικός: \*

Confirm Password/Επιβεβαίωση κωδικού: \*

Name/Όνομα: \*

Surname/Επιθετο: \*

Supervising Professor:

Academic Consultant: \*

Create chat account [here](#)

Chat Username: \*  @jabber.rootbash.com

Chat Password/Κωδικός Chat: \*

Confirm Chat Password / Επιβεβαίωση κωδικού Chat: \*

**Register**

Σε αυτή τη σελίδα ο φοιτητής πρέπει να συμπληρώσει τα στοιχεία του (όλα είναι υποχρεωτικά εκτός από το Supervising Professor). Τα πρώτα στοιχεία (Username, Email, Password, Name και Surname) αφορούν προσωπικά στοιχεία του

φοιτητή. Ακολουθούν δύο drop down lists όπου ο φοιτητής θα πρέπει να επιλέξει τον επιβλέποντα καθηγητή του (αν υπάρχει) και τον ακαδημαϊκό του σύμβουλο. Στις δύο λίστες υπάρχουν τα ονόματα όλων των καθηγητών, τα στοιχεία αυτά διαβάζονται από την βάση δεδομένων. Ακολούθως ο φοιτητής θα πρέπει να δημιουργήσει έναν jabber.rootbash.com λογαριασμό, χρησιμοποιώντας τον σύνδεσμο (link) που υπάρχει στην σελίδα, που θα του επιτρέψει να χρησιμοποιήσει τις λειτουργίες chat της εφαρμογής. Τέλος, χρησιμοποιώντας τα στοιχεία που έδωσε στην δημιουργία του jabber.rootbash.com λογαριασμού του, θα πρέπει να συμπληρώσει τα τελευταία στοιχεία (Chat Username, Chat password) και να ολοκληρώσει την εγγραφή του. Με την ολοκλήρωση της εγγραφής καταχωρείται ο φοιτητής στην βάση δεδομένων με την χρήση ενός INSERT query.

#### **4.4.4 Επεξεργασία λογαριασμού**

Αν ο χρήστης θέλει να τροποποιήσει κάποια από τα στοιχεία που έχει δώσει κατά την εγγραφή του πρέπει να επιλέξει το Edit Account και θα του παρουσιαστεί η ακόλουθη σελίδα,

## Interaction Between Professors and Students

### Edit Account

Username:

Password:

Submit

Εδώ ο χρήστης, νοουμένου ότι είναι ήδη εγγεγραμμένος στο σύστημα, θα πρέπει να δώσει το Username και το Password που είχε δώσει κατά την εγγραφή του. Πατώντας το κουμπί submit, θα ελεγχθεί αν υπάρχει ο χρήστης στην βάση και αν υπάρχει θα διαβαστούν τα στοιχεία του και θα παρουσιαστεί η επόμενη σελίδα.

### Edit Details

|                        |                                                  |                        |                                                        |
|------------------------|--------------------------------------------------|------------------------|--------------------------------------------------------|
| Name:                  | <input type="text" value="Fidias"/>              | Name:                  | <input type="text"/>                                   |
| Surname:               | <input type="text" value="Ioannides"/>           | Surname:               | <input type="text"/>                                   |
| Username:              | <input type="text" value="fioann011"/>           | Username:              | <input type="text"/>                                   |
| Email:                 | <input type="text" value="fidias@outlook.com"/>  | Email:                 | <input type="text"/>                                   |
| Supervising Professor: | <input type="text" value="Georgia Kapitsaki"/>   | Supervising Professor: | <input type="text" value="None"/>                      |
| Academic Consultant:   | <input type="text" value="Georgia Kapitsaki"/>   | Academic Consultant:   | <input type="text" value="Georgia Kapitsaki (gkapi)"/> |
| Chat Username:         | <input type="text" value="fioann011@jabber.de"/> | Chat Username:         | <input type="text"/>                                   |

Edit

Change Password

Set average grade

Σε αυτή τη σελίδα υπάρχουν δύο στήλες. Στην πρώτη στήλη παρουσιάζονται τα ήδη αποθηκευμένα στοιχεία του χρήστη και στην δεύτερη μπορεί ο χρήστης να δώσει τα νέα στοιχεία που θέλει να καταχωρηθούν. Δεν απαιτείται ο χρήστης να δώσει νέα δεδομένα για όλα τα πεδία αν επιθυμεί να αλλάξει μόνο μερικά. Για παράδειγμα, μπορεί να συμπληρώσει στην δεύτερη στήλη μόνο το πεδίο Name και να πατήσει Edit, αν θέλει να αλλάξει μόνο αυτό το πεδίο και να κρατήσει τα υπόλοιπα όπως ήταν πριν. Οι αλλαγές αποθηκεύονται στην βάση με την χρήση ενός UPDATE query για κάθε πεδίο. Αν ο χρήστης επιθυμεί να αλλάξει τον κωδικό του, μπορεί να πατήσει το κουμπί Change Password και θα του παρουσιαστεί η ακόλουθη σελίδα,



#### Change Password

|                                       |                          |
|---------------------------------------|--------------------------|
| Old password:                         | <input type="password"/> |
| New password:                         | <input type="password"/> |
| Confirm new password:                 | <input type="password"/> |
| <input type="button" value="Submit"/> |                          |

Σε αυτή την σελίδα ο χρήστης μπορεί δίνοντας πρώτα τον παλιό του κωδικό και ακολούθως τον νέο που επιθυμεί να αλλάξει τον κωδικό του. Η αλλαγή του κωδικού πραγματοποιείται με την χρήση ενός UPDATE query στο οποίο ελέγχεται αν έχει δώσει σωστό παλιό κωδικό ο χρήστης.

Αν ο φοιτητής θέλει να θέσει τον γενικό του βαθμό, μπορεί να πατήσει το κουμπί set average grade και θα του παρουσιαστεί η ακόλουθη σελίδα,



#### Set Average Password

|                                       |                          |
|---------------------------------------|--------------------------|
| Password:                             | <input type="password"/> |
| Set average grade:                    | <input type="text"/>     |
| <input type="button" value="Submit"/> |                          |

Σε αυτή την σελίδα ο φοιτητής δίνει αρχικά τον κωδικό του για επιβεβαίωση και ακολούθως δίνει τον γενικό του βαθμό. Η δήλωση του βαθμού πραγματοποιείται με την χρήση ενός UPDATE query στο οποίο ελέγχεται αν έχει δώσει σωστό κωδικό ο χρήστης.

#### 4.4.5 Επεξεργασία από τον διαχειριστή

Αν ο χρήστης είναι ο διαχειριστής του συστήματος μπορεί να επεξεργαστεί τα στοιχεία των καθηγητών και των φοιτητών που είναι εγγεγραμμένοι στο σύστημα πατώντας στο Administrator Edit. Αρχικά θα του εμφανιστεί η ακόλουθη σελίδα,



### Interaction Between Professors and Students

#### Administrator Edit

Administrator Code:

Continue with edit

Σε αυτή την σελίδα ζητείται από τον χρήστη να δώσει κωδικό, ο οποίος μπορεί να είναι ο κωδικός που γνωρίζει μόνο ο διαχειριστής ή ο κωδικός που γνωρίζουν οι καθηγητές, για να προχωρήσει στην επεξεργασία. Αν ο χρήστης δώσει τον κωδικό του διαχειριστή θα του εμφανιστεί η ακόλουθη σελίδα,

## Interaction Between Professors and Students

**Administrator Edit**

Professors Edit

Students Edit

Σε αυτή τη σελίδα ο διαχειριστής μπορεί να επιλέξει αν θέλει να επεξεργαστεί τα στοιχεία των καθηγητών ή των φοιτητών. Για παράδειγμα αν ο διαχειριστής επιλέξει να επεξεργαστεί τους φοιτητές, θα διαβαστούν από την βάση δεδομένων όλοι οι φοιτητές και θα του εμφανιστεί η πιο κάτω σελίδα,

| Edit Students <span>+ Add student</span> |                      |        |           |                       |                     |        |                                                                                       |
|------------------------------------------|----------------------|--------|-----------|-----------------------|---------------------|--------|---------------------------------------------------------------------------------------|
| Username                                 | Email                | Name   | Surname   | Supervising professor | Academic consultant | Delete | Edit                                                                                  |
| fioann055                                | fidias@outlook.com   | Fidias | Ioannides | Zoe Erotokritou       | Zoe Erotokritou     | ✗      |  |
| fioann011                                | fidias@outlook.com   | Fidias | Ioannides | Georgia Kapitsaki     | Georgia Kapitsaki   | ✗      |  |
| mdimitriou                               | mdimitriou@ucy.ac.cy | Μάριος | Δημητρίου | Georgia Kapitsaki     | Georgia Kapitsaki   | ✗      |  |

Εδώ ο διαχειριστής μπορεί να επιλέξει αν θέλει να διαγράψει, να επεξεργαστεί ή να προσθέσει κάποιο φοιτητή (τα ίδια ισχύουν και για τους καθηγητές). Αν επιλέξει να προσθέσει ένα φοιτητή θα του εμφανιστεί η ακόλουθη σελίδα,

## Add Student

|                                          |                                                            |
|------------------------------------------|------------------------------------------------------------|
| Name:                                    | <input type="text"/>                                       |
| Surname:                                 | <input type="text"/>                                       |
| Username:                                | <input type="text"/>                                       |
| Password:                                | <input type="password"/>                                   |
| Confirm Password:                        | <input type="password"/>                                   |
| Email:                                   | <input type="text"/>                                       |
| Supervising Professor:                   | --Select-- <input type="button" value="v"/>                |
| Academic Consultant:                     | Georgia Kapitsaki (gkapi) <input type="button" value="v"/> |
| Create chat account <a href="#">here</a> |                                                            |
| Chat Username:*                          | <input type="text"/> @jabber.rootbash.com                  |
| Chat Password:*                          | <input type="password"/>                                   |
| Confirm Chat Password: *                 | <input type="password"/>                                   |

Add

Η σελίδα αυτή περιέχει τα ίδια πεδία όπως και κατά την εγγραφή ενός φοιτητή. Ενώ αν επιλέξει να επεξεργαστεί έναν φοιτητή θα εμφανιστεί η ακόλουθη σελίδα,

## Edit Student

|                        |                                                    |                        |                                                             |
|------------------------|----------------------------------------------------|------------------------|-------------------------------------------------------------|
| Name:                  | <input type="text" value="Fidias"/>                | Name:                  | <input type="text"/>                                        |
| Surname:               | <input type="text" value="Ioannides"/>             | Surname:               | <input type="text"/>                                        |
| Username:              | <input type="text" value="foann055"/>              | Username:              | <input type="text"/>                                        |
| Email:                 | <input type="text" value="fidias@outlook.com"/>    | Email:                 | <input type="text"/>                                        |
| Supervising Professor: | <input type="text" value="Zoe Erotokritou"/>       | Supervising Professor: | Zoe Erotokritou (zeroto11) <input type="button" value="v"/> |
| Academic Consultant:   | <input type="text" value="Zoe Erotokritou"/>       | Academic Consultant:   | Zoe Erotokritou (zeroto11) <input type="button" value="v"/> |
| Chat Username:         | <input type="text" value="foann055ucy@jabber.de"/> | Chat Username:         | <input type="text"/>                                        |

Edit

Η σελίδα αυτή περιέχει τα ίδια πεδία όπως και κατά την επεξεργασία του λογαριασμού κάποιου φοιτητή από τον ίδιο.

Αν στο πρώτο βήμα ο χρήστης δώσει τον κωδικό των καθηγητών τότε θα του εμφανιστεί η ακόλουθη σελίδα,

## Interaction Between Professors and Students

**Add Student**

Add Student

Πατώντας στο κουμπί Add student, θα του εμφανιστεί μια φόρμα για πρόσθεση ενός φοιτητή παρόμοια με αυτή που φαίνεται πιο πάνω για τον διαχειριστή.

Επιπλέον αξίζει να αναφερθεί πως όλες οι φόρμες διαθέτουν Validators οι οποίοι εμφανίζουν τα κατάλληλα μηνύματα αν η είσοδος σε κάποιο πεδίο δεν ήταν η αναμενόμενη. Επίσης όλοι οι κωδικοί κωδικοποιούνται με την κρυπτογράφηση AES προτού αποθηκευτούν στην βάση δεδομένων.

## Κεφάλαιο 5. Δοκιμή και αξιολόγηση

---

|                                            |    |
|--------------------------------------------|----|
| 5.1 Παρουσίαση αποτελεσμάτων δοκιμών ..... | 68 |
| 5.2 Ανάλυση αποτελεσμάτων.....             | 70 |

---

Στο κεφάλαιο 5 γίνεται αναφορά στην διαδικασία που τηρήθηκε για δοκιμή της εφαρμογής και γίνεται η ανάλυση των αποτελεσμάτων που εξάχθηκαν από αυτή.

### 5.1 Παρουσίαση αποτελεσμάτων δοκιμών

Μετά την ολοκλήρωση της υλοποίησης, ακολούθησαν εκτενείς δοκιμές για την συμπεριφορά της ολοκληρωμένης εφαρμογής με όλα τα συστατικά μέρη συνενωμένα. Μετά από τις δοκιμές, η εφαρμογή δόθηκε σε καθηγητές και φοιτητές του πανεπιστημίου για αξιολόγηση. Αφότου δοκίμασαν την εφαρμογή τους ζητήθηκε να συμπληρώσουν το ακόλουθο ερωτηματολόγιο που περιέχει ερωτήσεις σχετικά με την εμπειρία χρήσης της εφαρμογής.

#### Ερωτηματολόγιο για την εφαρμογή Professor2Student

Παρακαλώ όπως υποδείξετε σε ποιο βαθμό συμφωνείτε ή διαφωνείτε με τις επόμενες δηλώσεις

( 1 : Διαφωνώ πλήρως, 2 : Διαφωνώ, 3 : Ούτε συμφωνώ ούτε διαφωνώ, 4 : Συμφωνώ, 5 : Συμφωνώ πλήρως )

**\* Required**

**1. Η εφαρμογή είναι εύκολη στην χρήση \***

1    2    3    4    5

---

Διαφωνώ πλήρως ☐ ☐ ☐ ☐ ☐ Συμφωνώ πλήρως

---

**2. Μπορώ να μάθω να χρησιμοποιώ όλες τις λειτουργίες της εφαρμογής εύκολα \***

1    2    3    4    5

---

Διαφωνώ πλήρως ☐ ☐ ☐ ☐ ☐ Συμφωνώ πλήρως

---

**3. Η περιήγηση μέσω των μενού της εφαρμογής είναι εύκολη \***

1 2 3 4 5

---

Διαφωνώ πλήρως ☐ ☐ ☐ ☐ ☐ Συμφωνώ πλήρως

---

**4. Τα μενού και υπομενού της εφαρμογής είναι σωστά δομημένα \***

1 2 3 4 5

---

Διαφωνώ πλήρως ☐ ☐ ☐ ☐ ☐ Συμφωνώ πλήρως

---

**5. Όλα τα αντικείμενα στην εφαρμογή (κουμπιά, textboxes κτλ.) είναι σωστά τοποθετημένα και η λειτουργία τους είναι εύκολα κατανοητή \***

1 2 3 4 5

---

Διαφωνώ πλήρως ☐ ☐ ☐ ☐ ☐ Συμφωνώ πλήρως

---

**6. Η εφαρμογή επιστρέφει κατάλληλα μηνύματα σύμφωνα με τις ενέργειες του χρήστη \***

1 2 3 4 5

---

Διαφωνώ πλήρως ☐ ☐ ☐ ☐ ☐ Συμφωνώ πλήρως

---

**7. Η εφαρμογή είναι αποδοτική και ο χρόνος απόκρισης μετά από τις ενέργειες του χρήστη είναι μικρός \***

1 2 3 4 5

---

Διαφωνώ πλήρως ☐ ☐ ☐ ☐ ☐ Συμφωνώ πλήρως

---

**8. Η εφαρμογή διαθέτει όλες τις λειτουργίες που περίμενα να έχει \***

1 2 3 4 5

---

Διαφωνώ πλήρως ☐ ☐ ☐ ☐ ☐ Συμφωνώ πλήρως

---

**9. Υπάρχει κάποια λειτουργία που δεν διαθέτει η εφαρμογή και πιστεύετε πως θα ήταν χρήσιμη;**

**10. Υπάρχει κάποια λειτουργία που διαθέτει η εφαρμογή και πιστεύετε πως δεν είναι χρήσιμη;**

**11. Επιπλέον σχόλια σχετικά με την εφαρμογή**

**12. Είστε καθηγητής ή φοιτητής \***

☒

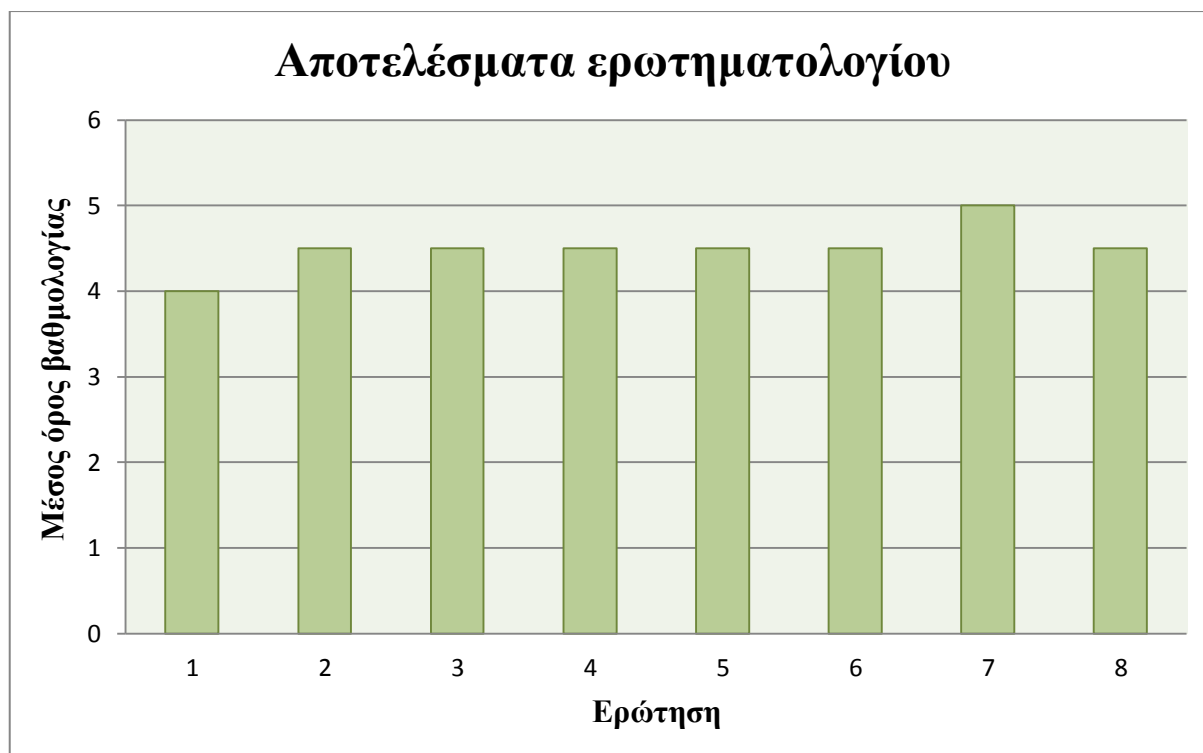
Καθηγητής

☐

Φοιτητής

## **5.2 Ανάλυση αποτελεσμάτων**

Στο σχήμα 5.2.1 παρουσιάζονται τα αποτελέσματα του ερωτηματολογίου. Στην γραφική παράσταση ο Χ άξονας περιέχει τον αριθμό της κάθε ερώτησης και ο Υ άξονας τον μέσο όρο βαθμολογίας που εκτείνεται μεταξύ 1 και 5. Για παράδειγμα η ερώτηση 4, “Τα μενού και τα υπομενού της εφαρμογής είναι σωστά δομημένα” έχει μέσο όρο βαθμολογίας 4,5.



#### 5.2.1 Αποτελέσματα ερωτηματολογίου

Οι ερωτήσεις που περιέχονται στο ερωτηματολόγιο καλύπτουν πολλές πτυχές της εφαρμογής, όπως ευχρηστία, αποδοτικότητα, δομή και πληρότητα. Έτσι από τις απαντήσεις στις ερωτήσεις αυτές μπορούν να εξαχθούν χρήσιμα αποτελέσματα για ολόκληρη την εφαρμογή. Τα αποτελέσματα από τις οκτώ ερωτήσεις είναι πολύ ενθαρρυντικά μιας και ο μέσος όρος βαθμολογίας όλων των ερωτήσεων κυμαίνεται μεταξύ 4 και 5. Ψηλότερη βαθμολογία έχει η ερώτηση 7 που αφορά την αποδοτικότητα της εφαρμογής, ενώ χαμηλότερη βαθμολογία έχει η ερώτηση 1 που αφορά την ευχρηστία της εφαρμογής, δείχνοντας πως ίσως τα μεγαλύτερα περιθώρια βελτίωσης αφορούν εκείνο τον τομέα. Την ευχρηστία αφορούν επίσης οι ερωτήσεις 2 και 6 που έχουν και οι δύο μέσο όρο βαθμολογίας 4.5. Οι ερωτήσεις 3, 4 και 5 αφορούν κυρίως την δομή της εφαρμογής. Και οι 3 αυτές ερωτήσεις έχουν μέσο όρο βαθμολογίας 4.5, οπότε μπορούμε να συμπεράνουμε πως οι χρήστες βρήκαν την δομή της εφαρμογής αρκετά καλή. Η ερώτηση 8, καθώς επίσης και οι ερωτήσεις με σχόλια, αφορά την πληρότητα της εφαρμογής. Ο μέσος όρος βαθμολογίας είναι επίσης 4.5 οπότε και πάλι μπορούμε να συμπεράνουμε πως οι χρήστες βρήκαν αρκετά πλήρης την εφαρμογή. Σύμφωνα με τις απαντήσεις των χρηστών στο ερωτηματολόγιο, φαίνεται πως βρήκαν αρκετά εύχρηστη, καλά δομημένη και πλήρης την εφαρμογή ενώ έμειναν πολύ ικανοποιημένη από την απόδοση της. Σίγουρα υπάρχουν περιθώρια

βελτίωσης σε όλους τους τομείς και σύμφωνα με τους χρήστες περισσότερο στον τομέα της ευχρηστίας.

Επίσης πέρα από τις βαθμολογίες λήφθηκαν χρήσιμες πληροφορίες και από τις τρεις τελευταίες ερωτήσεις. Έγιναν σχόλια που αφορούσαν τη διευκόλυνση του συσχετισμού μεταξύ καθηγητή και φοιτητή και να προστεθεί η δυνατότητα ο καθηγητής να μπορεί να δει την κατάσταση στην οποία βρίσκεται κάθε στάδιο της διπλωματικής συγκεντρωτικά. Επίσης έγινε σχόλιο να γίνει πιο γενική η εφαρμογή για να είναι ευκολότερη η προσθήκη νέων λειτουργιών. Τα σχόλια αυτά αποτελούν χρήσιμα συμπεράσματα που μπορούν να χρησιμοποιηθούν σε μετέπειτα βελτιώσεις της εφαρμογής.

## Κεφάλαιο 6. Συμπεράσματα – Μελλοντική Εργασία

---

|                              |    |
|------------------------------|----|
| 6.1 Συμπεράσματα .....       | 73 |
| 6.2 Μελλοντική Εργασία ..... | 75 |

---

Στο τελευταίο κεφάλαιο περιέχονται τα συμπεράσματα που εξάχθηκαν από την ολοκλήρωση της εργασίας αυτής και ακολούθως αναφέρεται μελλοντική εργασία που θα ήταν χρήσιμο να γίνει για επέκταση της συγκεκριμένης εργασίας.

### 6.1 Συμπεράσματα

Η επικοινωνία μεταξύ καθηγητών και φοιτητών αποτελεί διαχρονικά ένα καίριο πρόβλημα, καθώς επίσης και η επίβλεψη της πορείας του φοιτητή από τους καθηγητές του. Ιδιαίτερα σημαντική ανάγκη είναι η βελτίωση στην επικοινωνία, στην οργάνωση και στην αλληλεπίδραση μεταξύ φοιτητών που εργάζονται σε μια διπλωματική εργασία και των καθηγητών που τους επιβλέπουν, έτσι ώστε να διευκολύνεται η ολοκλήρωση της εργασίας και να περιορίζεται η ταλαιπωρία τόσο για τον καθηγητή όσο και για τον φοιτητή. Επίσης σημαντικό είναι να παρέχεται η δυνατότητα συνεχής επίβλεψης της πορείας του φοιτητή από τον ακαδημαϊκό του σύμβουλο, με στόχο την έγκαιρη βοήθεια του όταν χρειαστεί.

Πλέον με την ραγδαία ανάπτυξης των έξυπνων κινητών συσκευών δίνεται η δυνατότητα για άμεση επικοινωνία από οπουδήποτε και αν βρίσκεται ο χρήστης σε οποιαδήποτε χρονική στιγμή. Οι κινητές συσκευές πλέον έχουν αυξημένη υπολογιστική ισχύ και διαθέτουν την δυνατότητα συνεχούς σύνδεσης με το διαδίκτυο, ενώ η δημοτικότητα τους ανεβαίνει μέρα με την μέρα. Οι δυνατότητες αυτές μπορούν να δώσουν λύσεις στο πρόβλημα επικοινωνίας και αλληλεπίδρασης μεταξύ φοιτητών και καθηγητών που έχει αναφερθεί πιο πάνω.

Το σύστημα που σχεδιάστηκε έχει σαν στόχο να δώσει λύσεις στα προβλήματα αυτά. Η εφαρμογή στο λειτουργικό σύστημα android διαθέτει όλες τις δυνατότητες των κινητών συσκευών που αναφέρθηκαν πιο πάνω και προσφέρει την

δυνατότητα για εύκολη και γρήγορη επικοινωνία οπουδήποτε και οποιαδήποτε στιγμή. Μέσω της εφαρμογής αυτής καθηγητές και φοιτητές μπορούν να επικοινωνούν και να αλληλεπιδρούν με περισσότερη ευκολία. Η εφαρμογή προσφέρει λειτουργίες όπως ανταλλαγή μηνυμάτων, καθορισμό συναντήσεων, καθορισμό σταδίων διπλωματικής κλπ. που εξυπηρετούν αυτό το στόχο. Η εφαρμογή έχει ήδη δοκιμαστεί από καθηγητές και φοιτητές του τμήματος, με τα πρώτα μηνύματα να είναι ενθαρρυντικά.

Προς το παρόν για την χρήση της εφαρμογής υπάρχει ένας σημαντικός περιορισμός. Απαιτείται και ο φοιτητής και ο καθηγητής να διαθέτουν android συσκευή, παρότι τα περισσότερα μέρη του συστήματος και πολλά μέρη της εφαρμογής μπορούν να χρησιμοποιηθούν και σε άλλα λειτουργικά συστήματα. Η βάση δεδομένων, η διαδικτυακή εφαρμογή και η υπηρεσία διαδικτύου μπορούν να χρησιμοποιηθούν από οποιοδήποτε λειτουργικό σύστημα και το μόνο που χρειάζεται είναι η δημιουργία της εφαρμογής σε ένα άλλο λειτουργικό που θα τα χρησιμοποιεί. Επίσης τα μηνύματα XMPP μπορούν να ανταλλαχθούν μεταξύ χρηστών που χρησιμοποιούν διαφορετικό λειτουργικό σύστημα.

Για την ανάπτυξη του συστήματος αυτού χρησιμοποιήθηκαν κατά κύριο λόγο τα εργαλεία Microsoft Visual Studio 2010 Ultimate (για την διαδικτυακή εφαρμογή και την υπηρεσία διαδικτύου), Eclipse IDE σε συνδυασμό με το Android ADT plugin (για την δημιουργία της Android εφαρμογής) και Microsoft SQL Server 2008 (για την δημιουργία της βάσης δεδομένων). Επίσης για σχεδιασμό ορισμένων εικόνων χρησιμοποιήθηκε τα εργαλεία Gimp και Paint.NET. Όλα τα εργαλεία που χρησιμοποιήθηκαν ήταν εύκολα στην χρήση και διευκόλυναν πολύ την ανάπτυξη ολόκληρου του συστήματος.

Κλείνοντας πιστεύουμε πως η εφαρμογή αυτή μπορεί να αποτελέσει ένα σημαντικό εργαλείο για ικανοποίηση της ανάγκης άμεσης και εύκολης επικοινωνίας μεταξύ των καθηγητών και των φοιτητών του τμήματος.

## 6.2 Μελλοντική Εργασία

Η εργασία αυτή μπορεί να αποτελέσει σημείο αφετηρίας για μετέπειτα διπλωματικές εργασίες με στόχο την πρόσθεση περισσότερων λειτουργιών και βελτίωσης της εφαρμογής.

Κυριότερο πρόβλημα που μπορεί να λυθεί σε μετέπειτα διπλωματικές εργασίες είναι η ανάγκη και ο καθηγητής και ο φοιτητής να διαθέτει κινητή συσκευή λειτουργικού συστήματος Android. Παρότι το Android είναι με διαφορά το δημοφιλέστερο λειτουργικό για κινητές συσκευές, θα ήταν χρήσιμη η υλοποίηση της εφαρμογής και για άλλα λειτουργικά συστήματα όπως iOS, Windows Phone και BlackberryOS. Η υλοποίηση της εφαρμογής για κάποιο από τα προαναφερθέντα λειτουργικά συστήματα διευκολύνεται από το γεγονός πως όλα τα στοιχεία του συστήματος, όπως η διαδικτυακή εφαρμογή, η υπηρεσία διαδικτύου (web service) και η βάση δεδομένων μπορούν να χρησιμοποιηθούν από οποιοδήποτε άλλο λειτουργικό σύστημα και όχι μόνο το Android. Επίσης η ανταλλαγή μηνυμάτων μπορεί να γίνει μεταξύ δύο οποιονδήποτε συσκευών αφού το πρωτόκολλο XMPP μπορεί να υλοποιηθεί στα πλείστα λειτουργικά.

Επιπλέον υπάρχουν βελτιώσεις που λόγω χρόνου δεν έγιναν και μπορούν να γίνουν σε μετέπειτα εργασίες στην ήδη υπάρχουσα android εφαρμογή. Χρήσιμες λειτουργίες θα ήταν η πρόσθεση δυνατότητας ανάγνωσης του ιστορικού μηνυμάτων, δυνατότητας ομαδικής συνομιλίας (group chat) και δυνατότητα φωνητικής ή βίντεο κλήσης. Επίσης χρήσιμο θα ήταν ο φοιτητής να μπορεί να θέτει τους βαθμούς του για κάθε μάθημα ξεχωριστά και από τους βαθμούς αυτούς να υπολογίζεται ο μέσος όρος. Οι βαθμοί αυτοί θα μπορούσαν να εμφανίζονται και στον ακαδημαϊκό σύμβουλο μέσω της κινητής εφαρμογής. Ακόμη, μπορεί να προστεθεί ένας μηχανισμός για έλεγχο συγκρούσεων συναντήσεων (δύο συναντήσεις να έχουν ίδια ημερομηνία και ώρα) και να ενημερώνεται ο χρήστης όταν αυτό συμβαίνει.

Επίσης, από το ερωτηματολόγιο, λήφθηκαν χρήσιμες πληροφορίες για επιπλέον λειτουργίες και βελτιώσεις που μπορούν να προστεθούν μελλοντικά στην εφαρμογή. Μια εισήγηση αφορούσε τη διευκόλυνση του συσχετισμού μεταξύ καθηγητή και φοιτητή και να γίνει πιο γενική η εφαρμογή για να είναι ευκολότερη η προσθήκη νέων λειτουργιών. Επιπλέον προτάθηκε να προστεθεί η δυνατότητα ο καθηγητής να μπορεί να δει την κατάσταση στην οποία βρίσκεται κάθε στάδιο της

διπλωματικής συγκεντρωτικά. Οι εισηγήσεις αυτές αποτελούν χρήσιμες ιδέες που μπορούν να προστεθούν μελλοντικά στην εφαρμογή.

# **Βιβλιογραφία**

- [1] Tutorial on Android components:  
<http://developer.android.com/guide/components/index.html>
- [2] Android Development Tutorial:  
<http://www.vogella.com/articles/Android/article.html>
- [3] Location and Sensors in Android,  
<http://developer.android.com/guide/topics/sensors/index.html>
- [4] Really Simple Context Middleware: <http://code.google.com/p/rscm/>
- [5] Nearchos Paspallis, Middleware-based development of context-aware applications with reusable components, Ph.D. Thesis, University of Cyprus, September 2009.
- [6] Youngki Lee, S. S. Iyengar, Chulhong Min, Younghyun Ju, Seungwoo Kang, Taiwoo Park, Jinwon Lee, Yunseok Rhee, Junehwa Song, MobiCon: A Mobile Context-Monitoring Platform, Communications of the ACM, Vol. 55 No. 3, Pages 54-65, <http://cacm.acm.org/magazines/2012/3/146244-mobicon-a-mobile-context-monitoring-platform/fulltext>
- [7] Darren Carlson, Andreas Schrader, A wide-area context-awareness approach for Android, Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services, pp. 383-386
- [8] Alan Johnston, SIP: Understanding the Session Initiation Protocol, Second Edition, 2003.
- [9] Radvision, SIP Protocol overview,  
<http://www.radvision.com/NR/rdonlyres/51855E82-BD7C-4D9D-AA8A-E822E3F4A81F/0/RADVISIONSIPProtocolOverview.pdf/>
- [10] SIP RFC, <http://www.ietf.org/rfc/rfc3261.txt>
- [11] Session Initiation Protocol for Android,  
<http://developer.android.com/guide/topics/connectivity/sip.html>
- [12] WCF <http://msdn.microsoft.com/en-us/library/dd456779.aspx>
- [13] REST using WCF <http://msdn.microsoft.com/en-us/magazine/dd315413.aspx>
- [14] XMPP protocol <http://xmpp.org/>
- [15] XMPP public server <http://jabber.rootbash.com/>
- [16] Android XMPP libraries (asmack) <http://beem-project.com/>
- [17] XMPP RFC <http://www.ietf.org/rfc/rfc3920.txt>

- [18] XMPP XEP0013 extension (Flexible offline message retrieval)  
<http://xmpp.org/extensions/xep-0013.html>
- [19] XMPP XEP0077 extension (In-band registration)  
<http://xmpp.org/extensions/xep-0077.html>
- [20] Android <http://www.android.com/about/>
- [21] Operating system market share survey  
<http://www.idc.com/getdoc.jsp?containerId=prUS23946013>
- [22] Hypertext Transfer Protocol (HTTP)  
<http://www.w3.org/Protocols/rfc2616/rfc2616.html>