

# Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακό Πρόγραμμα Σπουδών *Πληροφορικά και  
Επικοινωνιακά Συστήματα*

## Μεταπτυχιακή Διατριβή



Εφαρμογή Android για Παρουσίαση Υφιστάμενων  
Υποτροφιών

Βασίλειος Ντιγκιρλάκης

Επιβλέπων Καθηγητής  
Δρ. Δημήτριος Κυριακού

Μάιος 2016

# **Ανοικτό Πανεπιστήμιο Κύπρου**

**Σχολή Θετικών και Εφαρμοσμένων Επιστημών**

**Μεταπτυχιακό Πρόγραμμα Σπουδών *Πληροφοριακά και  
Επικοινωνιακά Συστήματα***

## **Μεταπτυχιακή Διατριβή**

**Εφαρμογή Android για Παρουσίαση Υφιστάμενων  
Υποτροφιών**

**Βασίλειος Ντιγκιρλάκης**

**Επιβλέπων Καθηγητής  
Δρ. Δημήτριος Κυριακού**

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων για απόκτηση μεταπτυχιακού τίτλου σπουδών στα Ευφυή Συστήματα από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών του Ανοικτού Πανεπιστημίου Κύπρου.

**Μάιος 2016**

ΛΕΥΚΗ ΣΕΛΙΔΑ

## Περίληψη

Με σκοπό να διευκολύνουμε μαθητές και φοιτητές από την Κύπρο και την Ελλάδα στην εύρεση προπτυχιακών, μεταπτυχιακών, διδακτορικών και μεταδιδακτορικών Υποτροφιών σε όλα τα Πανεπιστήμια και Κολλέγια θεωρήθηκε αναγκαία μια εφαρμογή για έξυπνες κινητές συσκευές (Smartphone – Tablets – Smart TV) με λειτουργικό Android.

Πιστεύουμε ότι μέσω της εφαρμογής αυτής και με δεδομένο ότι θα υπάρξει στο μέλλον μια ενεργή συμμετοχή των Πανεπιστημιακών Ιδρυμάτων – Κολλεγίων για την σωστή ενημέρωση του συστήματος που θα προκύψει θα υπάρχει μια πιο διαφανής σχέση υποψηφίων και Ιδρυμάτων.

Η εφαρμογή θα δώσει την δυνατότητα στους ενδιαφερόμενους υποτρόφους να πραγματοποιούν αναζητήσεις με κριτήρια αλλά και λέξεις κλειδιά. Επιπλέον το σύστημα θα διαθέτει ένα μηχανισμό συστάσεων βασισμένο στις αναζητήσεις άλλων χρηστών με παρόμοια ενδιαφέροντα. Μέσω αυτής της δυνατότητας κάθε παλιός αλλά και νέος χρήστης τους συστήματος θα μπορεί να λάβει προτάσεις βασισμένες στη γνώση που συσσωρεύει το σύστημα από τους χρήστες που πραγματοποιούν αναζητήσεις.

Για την διαδικασία «απλών αναζητήσεων» χρησιμοποιήθηκαν κριτήρια αναζήτησης που προέκυψαν από αναζήτηση συστημάτων στο διαδίκτυο αλλά και από τις ίδιες τις αναγγελίες υποτροφιών. Έγινε μια προσπάθεια να βρεθούν τα κοινά σημεία αυτών των αναγγελιών (από διαφορετικούς φορείς) ώστε να καταλήξουμε σε κάποια κοινά κριτήρια για όλους.

Για την διαδικασία συστάσεων «έξυπνες αναζητήσεις» χρησιμοποιήθηκε μια παραλλαγή της διαδικασίας παραγοντοποίησης πινάκων σε συνδυασμό με τον αλγόριθμο Levenshtein για την ομοιότητα των λέξεων κλειδιών της αναζήτησης ώστε το σύστημα να παράγει συστάσεις στους χρήστες που πραγματοποιούν την συγκεκριμένη διαδικασία.

Η συγκεκριμένη διατριβή έγινε με στόχο να απαντήσει τα ερευνητικά ερωτήματα, του κατά πόσο μια εφαρμογή έξυπνων κινητών συσκευών μπορεί να συνεισφέρει διευκολύνοντας την εύρεση διαθέσιμων υποτροφιών με τα κατάλληλα κριτήρια αναζήτησης, και του αν μπορεί να υπάρξει ένας μηχανισμός συστάσεων σε χρήστες της εφαρμογής με παρόμοια ενδιαφέροντα.

## **Summary**

In order to facilitate schoolchildren and students from Cyprus and Greece in finding the undergraduate, graduate, doctoral and postdoctoral scholarships in all universities and colleges was a necessary application for smart mobile devices (Smartphone - Tablets - Smart TV) running Android.

We believe that through this application and given that there will be in the future an active participation of Universities - Colleges for the proper information of the system established there will be a more transparent relationship candidates and Institutions.

The application will allow to interested scholars to carry out searches with criteria and keywords. Additionally the system has a mechanism of recommendations based on searches of other users with similar interests. Through this feature every old and new user of the system can receive recommendations based on knowledge that accumulates the system by people searching.

For the process “simple search” search criteria used obtained from internet search systems and the same scholarships announcements. An attempt was made to find common points of these announcements (from different institution) to come up with some common criteria for all.

For the recommendation process “smart searches” used a variation of matrix factorization (collaborative filtering) in conjunction with the Levenshtein algorithm to the similarity of the keywords of the search, so that the system generates recommendations to users who perform the specific procedure.

This thesis was aimed to answer the research questions, whether a smart mobile device application can contribute by facilitating finding available scholarships with appropriate search criteria, and whether there may be a mechanism of recommendations on the application users with similar interests.

## **Ευχαριστίες**

Θα ήθελα να ευχαριστήσω τον Επιβλέπων καθηγητή μου Δρ. Δημήτρη Κυριακού για την μεθοδική και περιεκτική του καθοδήγηση καθ' όλη τη διάρκεια, καθώς και για την κατανόησή του.

Επιπλέον θα ήθελα να ευχαριστήσω την οικογένεια μου, γιατί όλο αυτό έγινε, από χρόνο που στερήθηκα εγώ από αυτούς και αυτοί από εμένα.

# Περιεχόμενα

<b>Αναγκαιότητα και Σπουδαιότητα Έρευνας .....</b>	<b>1</b>
1.1 Υφιστάμενη Κατάσταση και Προβλήματα.....	1
1.1.1 Τι Είδους Υποτροφίες είναι Διαθέσιμες.....	1
1.1.2 Πως Μπορώ να βρω Υποτροφίες;.....	2
1.1.3 Ποια είναι τα απαραίτητα δικαιολογητικά;.....	2
1.1.4 Πώς μπορώ να κάνω αίτηση για υποτροφίες; .....	2
1.1.5 Πότε πρέπει να κάνω αίτηση για υποτροφίες; .....	2
1.1.6 Ποια θα είναι η οικονομική βοήθεια; .....	3
1.1.7 Υπάρχει αξιοκρατία και διαφάνεια στην επιλογή υποψηφίων; .....	3
1.2 Συστήματα Υποτροφιών .....	3
1.2.1 Ιστοσελίδες με Φίλτρα.....	3
1.2.2 Υποτροφίες χωρίς Φίλτρα .....	5
1.2.3 Εφαρμογές Εύρεσης Υποτροφιών Κινητών Συσκευών .....	5
<b>Επιθεώρηση Διαθέσιμων Τεχνολογιών.....</b>	<b>10</b>
2.1 Πλατφόρμες Ανάπτυξης .....	10
2.1.1 Eclipse .....	10
2.1.2 NetBeans .....	10
2.1.3 Android Studio.....	11
2.1.4 Apache Cordova .....	11
2.2 Βέλτιστες Πρακτικές σε εφαρμογές Android.....	12
2.3 Σχεδιαστικά Πρότυπα για Android εφαρμογές .....	14
2.4 Πρωτόκολλα Ανταλλαγής Μηνυμάτων .....	15
2.4.1 Rest Services.....	15
2.4.2 JSON .....	15
2.5 Συστήματα Συστάσεων .....	16
2.5.1 Γιατί σύστημα «φίλτρου συνεργασίας» .....	17
2.5.2 Διαθέσιμα συστήματα συστάσεων για Java .....	17
2.5.3 Συστήματα συστάσεων βάση «Φίλτρου» Συνεργασίας.....	18
2.5.4 Αξιολόγηση Συστημάτων Συστάσεων «Φίλτρου Συνεργασίας» .....	18
2.5.5 Παραγοντοποίηση Πινάκων – Matrix Factorization.....	19
2.5.6 Βασική Ιδέα.....	19
2.5.7 Μαθηματικά Παραγοντοποίησης Πινάκων.....	20
2.6 Απόσταση κατά Levenshtein (String Edit Distance).....	24
2.7 Διαθέσιμοι Database Servers.....	26
2.7.1 SQL Server .....	26

2.7.2	MySQL .....	27
2.7.3	PostgreSQL .....	27
2.7.4	Connection pooling (Ομαδοποίηση συνδέσεων Βάσης) .....	28
2.8	Middleware Εξυπηρέτησης και Σύνδεση με την Βάση Δεδομένων .....	28
2.8.1	Ορισμός Middleware .....	28
2.9	Απαιτήσεις Ασφαλείας (Security) .....	30
2.9.1	Android app Security Risks.....	30
2.9.2	Βέλτιστες Πρακτικές Ασφαλείας Android Apps .....	31
	<b>Μεθοδολογία.....</b>	<b>34</b>
3.1	Ερευνητικά Ερωτήματα .....	34
3.2	Επιλογή κριτηρίων αναζήτησης.....	34
3.2.1	Επίπεδο Υποτροφίας .....	34
3.2.2	Γλώσσες Υποτροφίας .....	35
3.2.3	Φορέας Υποτροφίας.....	35
3.2.4	Χώρα Υποτροφίας.....	35
3.2.5	Ενεργές Υποτροφίες.....	35
3.3	Άλλα στοιχεία υποτροφιών.....	35
3.3.1	Ποσό Παρεχόμενης Υποτροφίας .....	35
3.3.2	Ακαδημαϊκό Έτος .....	36
3.3.3	Σύντομη Περιγραφή Υποτροφίας.....	36
3.3.4	Ιδιότητες Υποτροφίας τύπου «Κειμένου» .....	36
3.4	UML – Unified Modeling Language Διαγράμματα .....	37
3.4.1	Περιπτώσεις Χρήσης – Use Cases.....	37
3.4.2	Διάγραμμα Κλάσεων – Class Diagram .....	43
3.4.3	Διάγραμμα Δραστηριοτήτων .....	45
3.4.4	Διαγράμματα Ευρωστίας .....	45
3.4.5	Διαγράμματα Ακολουθίας.....	52
3.5	Διάγραμμα Βάσης Δεδομένων .....	56
3.6	Περιγραφή Βάσης Δεδομένων .....	57
3.6.1	Πίνακες και Ανάλυση τους.....	57
	<b>Σχεδιασμός και Υλοποίηση εφαρμογής Android .....</b>	<b>59</b>
4.1	Middleware.....	59
4.1.1	Περιγραφή Λειτουργίας.....	59
4.2	Android App .....	68
4.2.1	Περιγραφή Λειτουργίας.....	68
4.2.2	Υποστήριξη διαφορετικών γλωσσών .....	69



4.2.3	To User Interface της εφαρμογής .....	70
4.2.4	Οθόνες εφαρμογής Android .....	71
<b>Έλεγχος και επαλήθευση της εφαρμογής .....</b>		<b>78</b>
5.1	Ανάλυση Ερωτηματολογίου Χρησιμότητας – Λειτουργικότητας από Μαθητές	78
5.1.1	1 <sup>ο</sup> Ερωτηματολόγιο Χρησιμότητας Υποψήφιων Υποτρόφων .....	78
5.1.2	2 <sup>ο</sup> Ερωτηματολόγιο Χρησιμότητας Υποψήφιων Υποτρόφων .....	84
5.2	Ανάλυση Ερωτηματολογίου Χρησιμότητας από Εκπαιδευτικά Ιδρύματα .....	88
5.3	Ανάλυση Ερωτηματολογίου Ευχρηστίας.....	95
5.4	Πειραματικές Δοκιμές Συστήματος Συστάσεων .....	96
5.4.1	Σενάριο 1 <sup>ο</sup> .....	96
5.4.2	Σενάριο 2 <sup>ο</sup> .....	96
<b>Συμπεράσματα και Επεκτάσεις.....</b>		<b>97</b>
6.1	Συμπεράσματα .....	97
6.2	Μελλοντικές Επεκτάσεις.....	98
 <b>Παραρτήματα</b>		
<b>SQL Full-Text Searches .....</b>		<b>99</b>
A.1	Η Χρησιμότητα των Full-text Searches .....	99
A.2	Απαιτήσεις για αναζήτηση πλήρους κειμένου.....	100
A.2.1	Ολοκληρωμένου κειμένου αναζητήσεις τύπου Boolean .....	101
A.3.1	Ολοκληρωμένου κειμένου αναζητήσεις φυσικής γλώσσας.....	105
A.3	Το φαινόμενο 50% Threshold.....	105
<b>UML – Unified Modeling Language .....</b>		<b>106</b>
B.1	Εισαγωγή.....	106
B.2	Διαγράμματα.....	107
<b>Βιβλιογραφία.....</b>		<b>108</b>



# Κεφάλαιο 1

## Αναγκαιότητα και Σπουδαιότητα Έρευνας

Οι ανάγκες της διατριβής επέρχονται από την απουσία ενός συστήματος που επιτρέπει σε ένα μαθητή να βρει συγκεντρωμένες τις διαθέσιμες υποτροφίες που προσφέρουν τα πανεπιστήμια/κολλέγια σε Κύπρο και Ελλάδα στον επιστημονικό τομέα που θέλει να ακολουθήσει. Στις ημέρες που ζούμε με την οικονομική κρίση να γίνεται ολοένα και πιο έντονη η ανάγκη μιας τέτοιας εφαρμογής γίνεται πιο επιτακτική δίνοντας τη δυνατότητα σε όσους δεν έχουν την οικονομική δυνατότητα να μειώσουν ή και να μηδενίσουν τις δαπάνες τους χωρίς να στερούνται το δικαίωμα στην μάθηση.

### 1.1 Υφιστάμενη Κατάσταση και Προβλήματα

Στη παρούσα κατάσταση η εύρεση υποτροφιών είναι μια αρκετά επίπονη διαδικασία η οποία απαιτεί αρκετό χρόνο αναζήτησης με διάφορα μέσα. Κάποια από τα μέσα που έχει στη διάθεσή του ο υποψήφιος σήμερα είναι :

- Αναζήτηση μέσω μηχανών αναζήτησης – search engines (Google, Yahoo, Bing)
- Οι ιστοσελίδες των Πανεπιστημίων και των Κολλεγίων
- Συγκεκριμένες ιστοσελίδες Κρατικών – Ιδιωτικών φορέων
- Κοινωφελή Ιδρύματα

Σκεφτόμενοι ως υποψήφιοι υπότροφοι θα έπρεπε να βρούμε απάντηση σε μια σειρά από ερωτήματα μέσα από τα οποία προκύπτουν και κάποια προβλήματα.

#### 1.1.1 Τι Είδους Υποτροφίες είναι Διαθέσιμες

Για την εύρεση των διαθέσιμων ειδών των υποτροφιών ο ενδιαφερόμενος θα πρέπει βρει τις ιστοσελίδες των Πανεπιστημίων / Κολλεγίων – Φορέων και σε καθένα από αυτά να αναζητήσει τα διαθέσιμα είδη.

Το πρόβλημα είναι ότι οι ιστοσελίδες δεν είναι πάντα ενημερωμένες (obsolete sites, υπάρχουν ιστοσελίδες οι οποίες έχουν να ενημερωθούν χρόνια).

### **1.1.2 Πως Μπορώ να βρω Υποτροφίες;**

Για την εύρεση των υποτροφιών ο ενδιαφερόμενος πρέπει να κάνει μια ευρεία αναζήτηση στο διαδίκτυο μέσω των διαθέσιμων μηχανών αναζήτησης Google, Yahoo, Bing.

Το πρόβλημα εδώ είναι ότι η διαδικασία αυτή είναι αρκετά χρονοβόρα και έχει αμφίβολα αποτελέσματα.

### **1.1.3 Ποια είναι τα απαραίτητα δικαιολογητικά;**

Ο μόνος τρόπος που υπάρχει για την εύρεση των απαραίτητων δικαιολογητικών για μια υποτροφία είναι

- Η επικοινωνία με το Ίδρυμα (email, τηλέφωνο, επίσκεψη κ.α.)
- Σε κάποιες περιπτώσεις αρκεί η επίσκεψη της ιστοσελίδας του ιδρύματος

Στην πλειοψηφία των περιπτώσεων δεν αρκεί η επίσκεψη στην ιστοσελίδα του ιδρύματος.

### **1.1.4 Πώς μπορώ να κάνω αίτηση για υποτροφίες;**

Ειδικά σε κρατικά Ιδρύματα ο συνηθής τρόπος για να κάνει ο υποψήφιος αίτηση είναι η επίσκεψη. Σε λίγες περιπτώσεις υπάρχει ηλεκτρονικό σύστημα αιτήσεων. Τα ανοικτά Πανεπιστήμια αποτελούν εξαίρεση αφού έχουν δώσει έμφαση στα ηλεκτρονικά συστήματα τόσο εκμάθησης όσο και αιτήσεων.

### **1.1.5 Πότε πρέπει να κάνω αίτηση για υποτροφίες;**

Το πιο δύσκολο ερώτημα είναι το «πότε». Καθώς δεν υπάρχει καθορισμένος χρόνος αναγγελίας ο ενδιαφερόμενος καλείται να ψάχνει σε τακτά χρονικά διαστήματα.

Το πρόβλημα είναι ότι συνήθως βλέπει κάποιος μέσω διαδικτύου μια υποτροφία αφού έχει παρέλθει η καταληκτική ημερομηνία αιτήσεων.

### **1.1.6 Ποια θα είναι η οικονομική βοήθεια;**

Και σε αυτό το ερώτημα δεν υπάρχει ένας μοναδικός τρόπος για να ικανοποιηθεί. Η πληροφορία αντλείται με διαφορετικούς τρόπους ανά περίπτωση. Πιο ξεκάθαρη εικόνα υπάρχει από τις ανακοινώσεις των κοινωφελών Ιδρυμάτων παρά από τα Πανεπιστημιακά ιδρύματα.

### **1.1.7 Υπάρχει αξιοκρατία και διαφάνεια στην επιλογή υποψηφίων;**

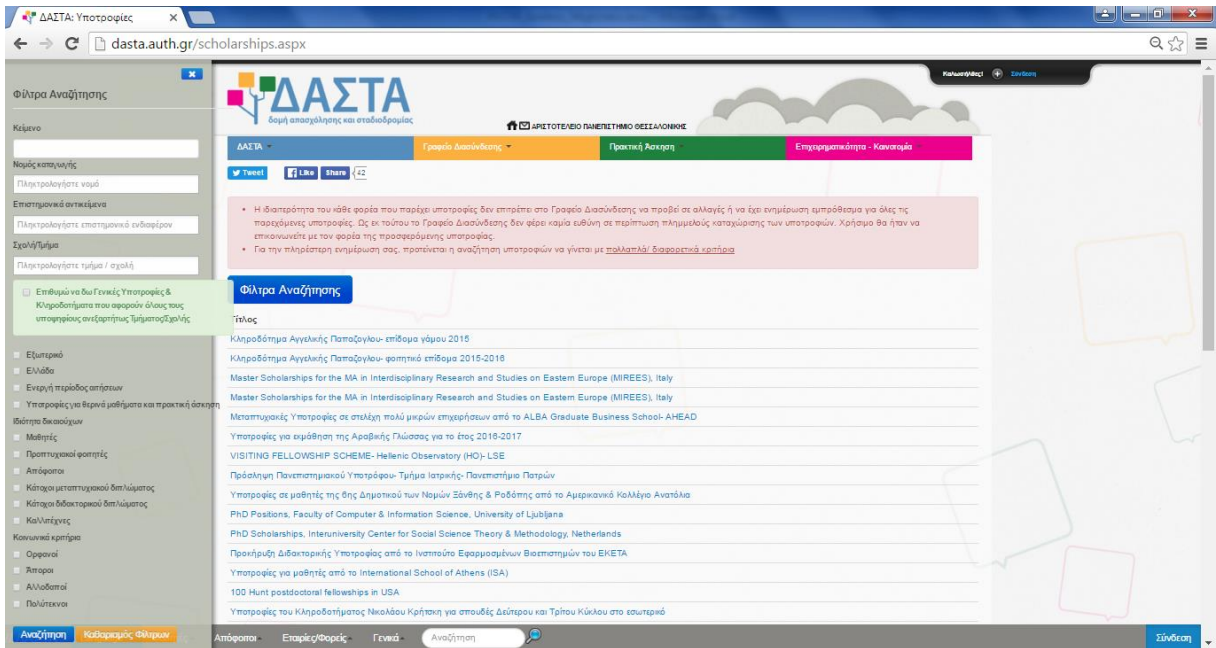
Προφανώς και δημιουργούνται υποψίες και ερωτηματικά περί διαφάνειας και αξιοκρατίας για τον τρόπο που γίνεται η επιλογή. Αν σκεφτούμε λίγο πιο βαθιά θα καταλάβουμε ότι όλα τα παραπάνω προβλήματα και οι ασάφειες σχετίζονται πιθανότατα με την διαφάνεια.

## **1.2 Συστήματα Υποτροφιών**

Κατά την διαδικασία αναζήτησης διαθέσιμων συστημάτων υποτροφιών δεν μπορούμε να πούμε ότι υπάρχει μια κοινή αντιμετώπιση. Κάθε ιστοσελίδα έχει υλοποιηθεί με ένα ξεχωριστό τρόπο, οι περισσότερες από αυτές απλά κοινοποιούν τις υποτροφίες σαν κείμενα που παρέχονται από τον φορέα της υποτροφίας, και λίγες από αυτές παρέχουν κάποια κριτήρια για αναζήτηση. Στην πράξη φαίνεται ότι δεν έχει γίνει κάποια σημαντική δουλειά στον τομέα αυτό.

### **1.2.1 Ιστοσελίδες με Φίλτρα**

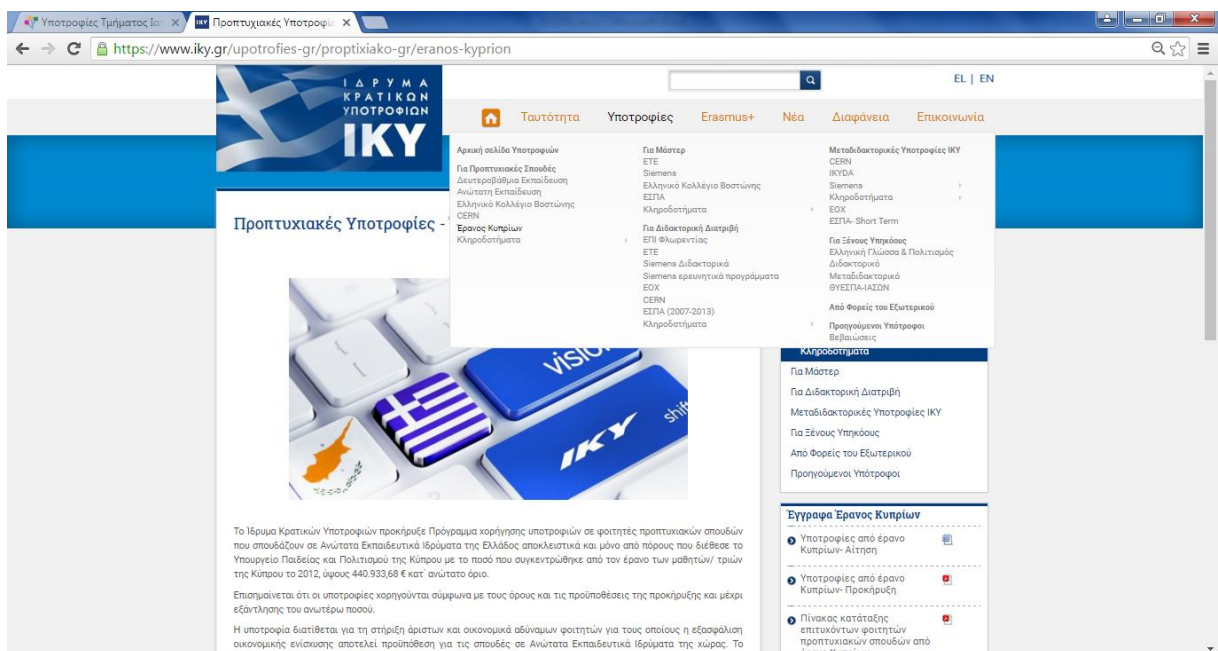
Στην παρακάτω εικόνα (Εικόνα 1) εμφανίζεται μια από τις ιστοσελίδες η οποία παρέχει κάποια κριτήρια – φίλτρα αναζήτησης στους χρήστες.



Εικόνα 1. Site ΔΑΣΤΑ

Το εν λόγω site παρέχει φίλτρα αναζήτησης όπως

- Κείμενο αναζήτησης
- Νομός Καταγωγής
- Σχολή Τμήμα
- Ενεργή περίοδος Αιτήσεων
- Επίπεδο Υποτροφίας
- Κοινωνικά κριτήρια



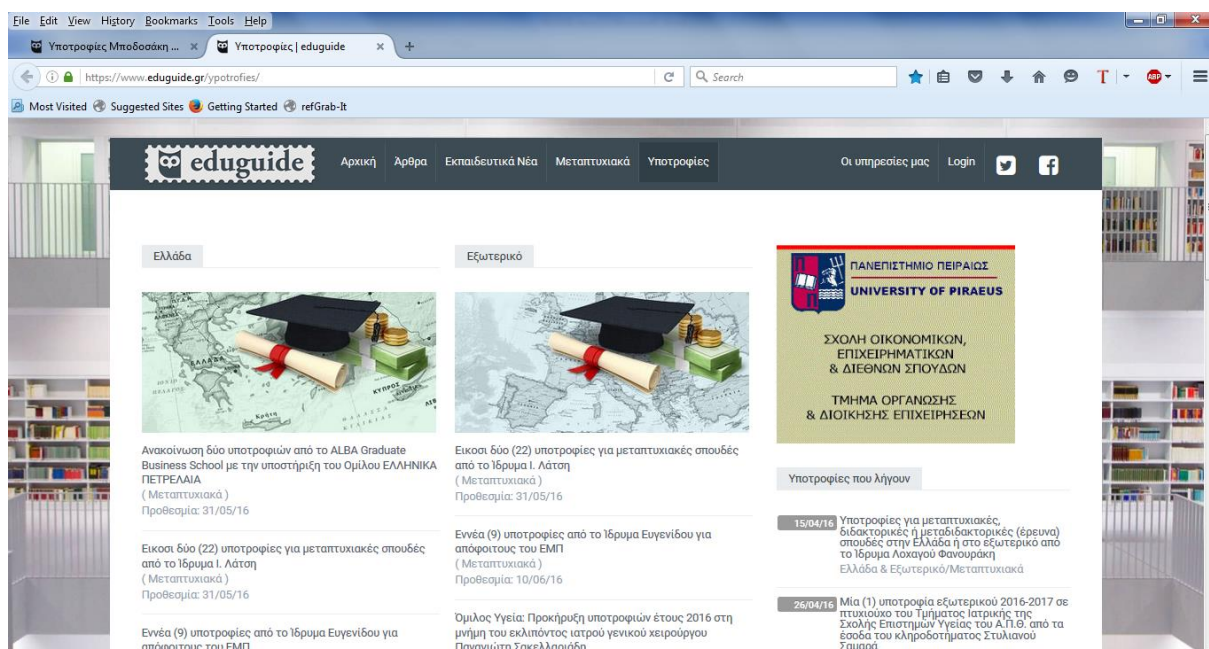
Εικόνα 2. Site Ιδρύματος Κρατικών Υποτροφιών

Στην παραπάνω ιστοσελίδα του Κρατικού Ιδρύματος Υποτροφιών (Εικόνα 2), τα φίλτρα αναζήτησης δεν υπάρχουν και πρακτικά βλέπουμε μόνο κατηγορίες υποτροφιών όπως φαίνεται στην εικόνα.

Αν κάνουμε μια απλή αναζήτηση στις διαφορές κατηγορίες θα δούμε γενικές επιδοτήσεις που μπορούν παρέχονται στην κάθε κατηγορία. Αλλά ερχόμαστε στο πρόβλημα που προαναφέρθηκε, ότι δηλαδή για τις περισσότερες υποτροφίες έχει παρέλθει η καταληκτική ημερομηνία αιτήσεων.

### 1.2.2 Υποτροφίες χωρίς Φίλτρα

Στην παρακάτω εικόνα (Εικόνα 3) φαίνονται υποτροφίες της ιστοσελίδας eduguide στην οποία φαίνεται ότι δεν υπάρχουν φίλτρα αναζήτησης παρά μόνο διαφοροποίηση υποτροφιών Ελλάδας και Εξωτερικού.

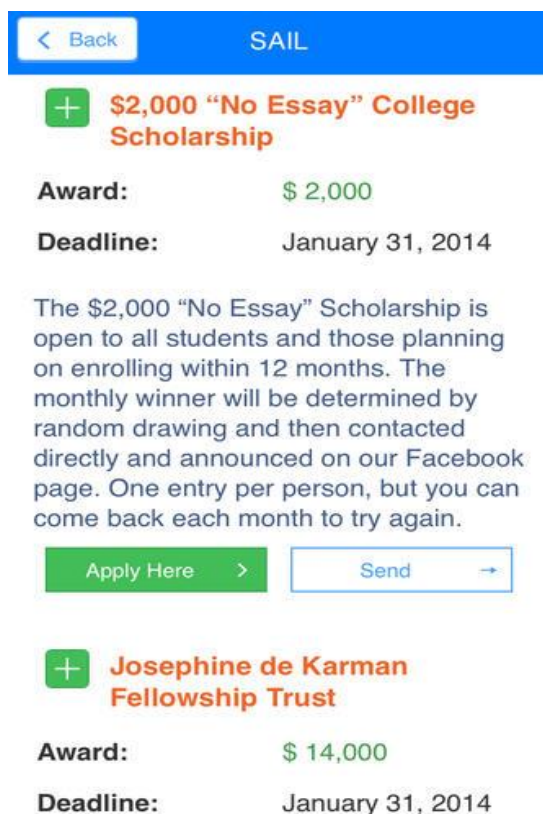


Εικόνα 3. Site eduguide

### 1.2.3 Εφαρμογές Εύρεσης Υποτροφιών Κινητών Συσκευών

Οι διαθέσιμες εφαρμογές κινητών συσκευών που υπάρχουν την παρούσα στιγμή δεν καλύπτουν την Ελλάδα και την Κύπρο. Συναντάμε εφαρμογές υποτροφιών για Android και iOS μόνο για τις Ηνωμένες Πολιτείες με τα παρακάτω χαρακτηριστικά.

### 1.2.3.1 Sail Scholarship Search



Εικόνα 4. Mobile App "Sail Scholarship Search"

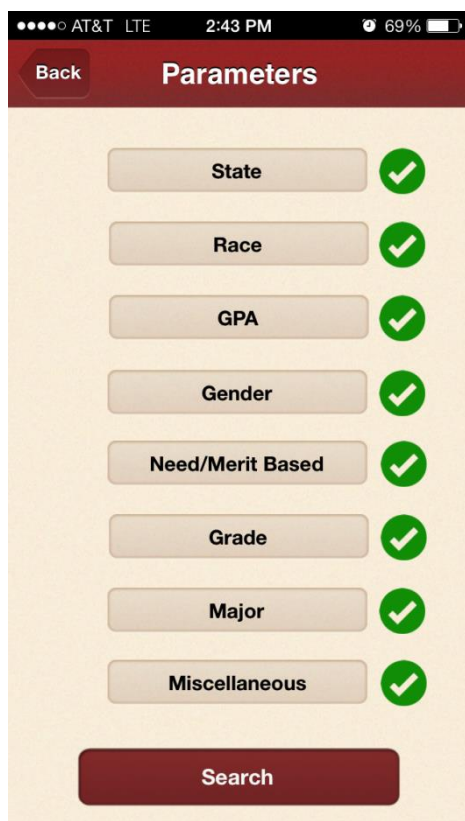
Η συγκεκριμένη εφαρμογή είναι απλή στην πλοήγηση, εύχρηστη, δίνει τη δυνατότητα εύρεσης υποτροφιών με βάση το προφίλ των Φοιτητών, περιλαμβάνει ειδικές υποτροφίες κολλεγίων ως μέρος των αποτελεσμάτων αναζήτησης, οι μαθητές μπορούν να επιλέξουν να στείλουν μεμονωμένες υποτροφίες στο email τους, τα οποία περιλαμβάνουν την πλήρη περιγραφή και τις συνδέσεις με τις υποτροφίες ή να επιλέξουν να δουν περισσότερες πληροφορίες σχετικά με τις υποτροφίες σε απευθείας σύνδεση, συμπεριλαμβανομένης της εφαρμογής και των πρόσθετων απαιτήσεων της κάθε υποτροφίας.

Οι μαθητές μπορούν να προσθέσουν τις υποτροφίες που τους ενδιαφέρουν σε μια λίστα (που ονομάζεται Scholarship Planner) και να πάρει το σύνολο των υποτροφιών που έχουν προσθέσει. Αυτή η λίστα των υποτροφιών χρησιμοποιούνται στη συνέχεια για να δώσει στον μαθητή ένα εκτιμώμενο κόστος συμμετοχής με βάση το κολέγιο που θα επιλέξει να παρακολουθήσει.



Ο Scholarship Planner δίνει στον Φοιτητή ένα κόστος συμμετοχής στο κολλέγιο που επιθυμεί, τα δίδακτρα, τα τέλη, τα βιβλία και τα έξοδα διαμονής και διατροφής για ένα πτυχίο τετραετούς φοίτησης.

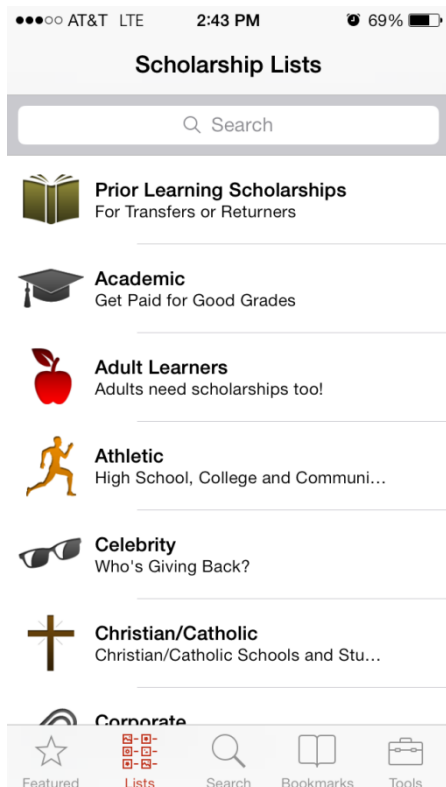
### 1.2.3.2 Scholly



Εικόνα 5. Mobile App “Scholly”

Η πλοήγηση είναι αρκετά έξυπνη και η εφαρμογή είναι Responsive (περιγράφηκε παραπάνω). Παρέχει υποτροφίες κολλεγίων βασισμένες στο προφίλ του χρήστη και επιτρέπει την αποθήκευση των υποτροφιών που μας ενδιαφέρουν για αργότερο έλεγχο. Οι φοιτητές μπορούν να προσθέσουν στο calendar της κινητής τους συσκευής τις καταληκτικές ημερομηνίες των υποτροφιών. Μια σύνοψη της υποτροφίας είναι εισηγμένη κατόπιν επιλογής με την επιλογή να δείτε την ιστοσελίδα για περισσότερες πληροφορίες.

### 1.2.3.3 Scholarship Advisor

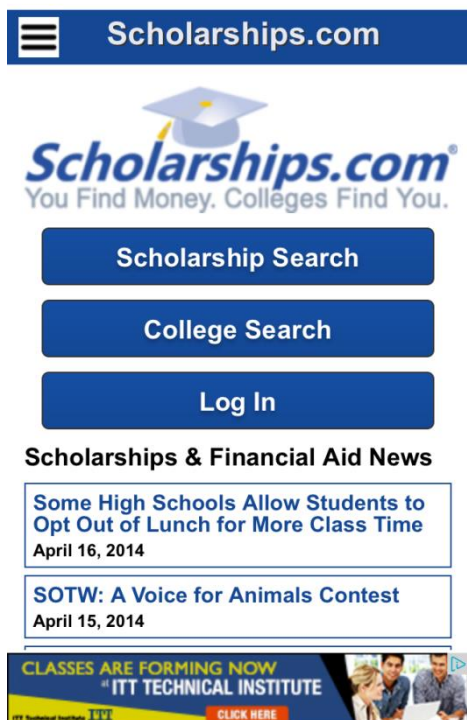


Εικόνα 6. Mobile App “Scholarship Advisor”

Το User Interface είναι πολύ καθαρό (μαύρο με λευκό με μια απόχρωση του κόκκινου). Οι υποτροφίες σε αυτή την εφαρμογή μπορούν να προβληθούν με πολλούς τρόπους. Οι φοιτητές μπορούν να επιλέξουν να δουν συγκεκριμένες κατηγορίες. Οι μαθητές μπορούν να δουν υποτροφίες με βάση το ποσό και να αποθηκεύσουν σελιδοδείκτη (bookmark) για τις υποτροφίες που βρίσκουν ενδιαφέρουσες. Πρόσθετα εργαλεία περιλαμβάνουν νέα για υποτροφίες, αναζήτηση υποτροφιών (η οποία απαιτεί από το μαθητή την εγγραφή του) και δημοφιλείς υποτροφίες. Οι μαθητές μπορούν επίσης να στείλουν email ή tweet υποτροφίες που τους ενδιαφέρει.

## 1.2.3.4 Scholarships.com

•••• AT&T LTE 2:42 PM 69%  
TCl College of Technology  
tcicollege.edu  
Learn from Industry Experts and Start Your Career! Apply today



Εικόνα 7. Mobile App “Scholarships.com”

Για να δούμε τις υποτροφίες πρέπει να συνδεθούμε με ένα email και ένα κωδικό πρόσβασης. Απαιτείται σχολικό έτος και η ημερομηνία γέννησης.

Επιτρέπει στους μαθητές να ψάξουν για το κολέγιο με την επιλογή αναζήτησης κολεγίου που παρέχει μια λίστα που χαρακτηρίζει τα δίδακτρα, τα ποσοστά αποδοχής, απαιτήσεις, τα ποσοστά αποφοίτησης και πτυχία που παρέχει.

Αρκετά κουραστική εφαρμογή λόγω των πολλών διαφημίσεων.

# Κεφάλαιο 2

## Επιθεώρηση Διαθέσιμων Τεχνολογιών

### 2.1 Πλατφόρμες Ανάπτυξης

Στο σημείο αυτό θα γίνει μια συνοπτική αναφορά στις διαθέσιμες πλατφόρμες που υπάρχουν αυτή την στιγμή για την ανάπτυξη εφαρμογών κινητών συσκευών.

#### 2.1.1 Eclipse

Το **Eclipse** [3] είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE - *integrated development environment*). Παρέχει ένα βασικό χώρο εργασίας και ένα επεκτάσιμο σύστημα από πρόσθετα (plugins) για την προσαρμογή του περιβάλλοντος. Το Eclipse είναι γραμμένο ως επί το πλείστον σε Java και κύρια χρήση του είναι για την ανάπτυξη εφαρμογών Java, αλλά μπορεί επίσης να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών σε άλλες γλώσσες προγραμματισμού με τη χρήση των plug ins, συμπεριλαμβανομένων: Ada, ABAP, C, C++, COBOL, Fortran, JavaScript, Julia, Lasso, Lua, NATURAL, Perl, PHP, Prolog, Python, R, Ruby (including Ruby on Rails framework), Rust, Scala, Clojure, Groovy, Scheme και Erlang.

Αξίζει να σημειωθεί ότι το συγκεκριμένο περιβάλλον ανάπτυξης έχει πια εγκαταλειφτεί από την Google, η οποία παροτρύνει τους χρήστες του Eclipse με χρήση του ADT plugin (Android Development Tools [6]) να μετακινήσουν τα project τους στο Android Studio.

#### 2.1.2 NetBeans

Το **NetBeans IDE** [2] είναι ένα δωρεάν και ανοικτού κώδικα ολοκληρωμένο περιβάλλον ανάπτυξης για την ανάπτυξη εφαρμογών για τα λειτουργικά συστήματα Windows, Mac, Linux και Solaris.

Το συγκεκριμένο IDE απλοποιεί την ανάπτυξη των web, enterprise, desktop, και εφαρμογών κινητών συσκευών που χρησιμοποιούν τα Java και HTML5 πλατφόρμες. Προσφέρει επίσης υποστήριξη για την ανάπτυξη σε PHP και C / C ++ εφαρμογών. Και για το NetBeans παρέχεται πρόσθετο plugin για ανάπτυξη εφαρμογών Android.

### 2.1.3 Android Studio

Το **Android Studio [26]** είναι ένα ολοκληρωμένο προγραμματιστικό περιβάλλον (IDE - *integrated development environment*) για ανάπτυξη εφαρμογών στην Android πλατφόρμα. Ανακοινώθηκε στις 16 Μαΐου 2013 στο συνέδριο Google I/O από την Google Product Manager, Katherine Chou. Το Android Studio είναι διαθέσιμο ελεύθερα με την άδεια Apache License 2.0.

Το Android Studio ήταν διαθέσιμο σε πρώιμο στάδιο για προεπισκόπηση ξεκινώντας από την έκδοση 0.1 τον Μάιο του 2013, έπειτα ξεκίνησε το δοκιμαστικό στάδιο από την έκδοση 0.8 που βγήκε τον Ιούνιο του 2014. Η πρώτη σταθερή έκδοση βγήκε το Δεκέμβριο του 2014, με την έκδοση 1.0.

Βασισμένο στο λογισμικό της Jet Brains IntelliJ IDEA, το Android Studio σχεδιάστηκε αποκλειστικά για Android προγραμματισμό. Είναι διαθέσιμο για Windows, Mac OS X και Linux, και αντικατέστησε το Eclipse Android Development Tools (ADT) ως το κύριο IDE της Google για ανάπτυξη Android εφαρμογών.

### 2.1.4 Apache Cordova

Το framework αυτό παρέχεται από την εταιρία Apache και επιτρέπει τη δημιουργία native εφαρμογών. Οι βασικότερες πλατφόρμες υποστηρίζονται (Android, iOS, Windows Phone κ.α.).

Η ανάπτυξη των εφαρμογών γίνεται μέσω τεχνολογιών διαδικτύου HTML5, CSS3 και JavaScript.

Το **Apache Cordova [25]** είναι το λογισμικό που αποτελεί τη βάση άλλων λογισμικών που υποστηρίζουν αυτού του είδους την μετατροπή (web σε Native). Τα άλλα frameworks παρέχουν συμπληρωματικά εργαλεία ανάπτυξης και το Cordova είναι υπεύθυνο για τη μετατροπή. Πρόκειται για λογισμικό ανοιχτού κώδικα το οποίο υποστηρίζει λειτουργίες για πάνω από 10 διαφορετικά λειτουργικά συστήματα.

Το σημαντικό είναι ότι με τον ίδιο κώδικα μπορούμε να παράγουμε εφαρμογή για οποιοδήποτε λειτουργικό σύστημα Android, iOS, Windows Phone. Επίσης το συγκεκριμένο framework διαθέτει plugins για τα περισσότερα σύγχρονα περιβάλλοντα ανάπτυξης (Microsoft Studio, Eclipse, NetBeans, κ.α.).

## 2.2 Βέλτιστες Πρακτικές σε εφαρμογές Android

Συνοπτικά θα δούμε τις βέλτιστες πρακτικές για την ανάπτυξη εφαρμογών Android

### ➤ User Inputs

Κάθε πεδίο κειμένου προορίζεται για μια διαφορετική δουλειά. Για παράδειγμα, ορισμένα πεδία κειμένου είναι για κείμενο (alphanumeric) και μερικά είναι για τους αριθμούς (numeric). Αν τα πεδία κειμένου είναι για τους αριθμούς, τότε είναι καλύτερα να εμφανίζουμε το αριθμητικό πληκτρολόγιο όταν εστιάζεται (focus on text field) το πεδίο κειμένου.

Αν το πεδίο αφορά κωδικό (password για περίπτωση που οι χρήστες πραγματοποιούν registration), τότε θα πρέπει να εμφανίζεται μια υπόδειξη κωδικού πρόσβασης (password hint), έτσι ώστε ο χρήστης μπορεί εύκολα να θυμάται τον κωδικό πρόσβασης.

### ➤ Background Jobs

Υπάρχουν ορισμένες εργασίες σε μια εφαρμογή που εκτελούνται στο παρασκήνιο της εφαρμογής (background tasks). Η δουλειά τους θα μπορούσε να είναι να φέρω κάποια πράγματα από το διαδίκτυο, διαδικασία αναζήτησης κ.α. Συνιστάται στις διεργασίες με εκτεταμένη αναμονή να μην πραγματοποιούνται στο UI thread αλλά στο παρασκήνιο από services ή ασύγχρονες διεργασίες (Asynchronous Tasks).

### ➤ Responsive UI

Έχουμε λοιπόν δύο προσεγγίσεις (σύγχρονες – ασύγχρονες διεργασίες), οι σύγχρονες δεν επηρεάζονται από τα γεγονότα (events) του κύκλου UI (User interface), έτσι ώστε να συνεχίζουν να λειτουργούν και με το κλείσιμο της εφαρμογής σε αντίθεση με τα Asynchronous Tasks.

Για την επίτευξη ενός Responsive UI προτείνεται η χρήση τέτοιων διεργασιών όπου αυτό χρειάζεται.

### ➤ Performance

Η απόδοση της εφαρμογής (performance) θα πρέπει να διαφοροποιείται ανάλογα με το αν η συσκευή είναι σε κατάσταση φόρτισης ή όχι. Η διαφοροποίηση δεν αφορά το front end αλλά το back end. Προτείνεται όταν η συσκευή φορτίζει να ενημερώνονται οι ρυθμίσεις της εφαρμογής, έτσι ώστε να μεγιστοποιείται ο ρυθμός ανανέωσης κάθε φορά που η συσκευή είναι συνδεδεμένη.

➤ Security & Privacy

Η εφαρμογή, τα δεδομένα του χρήστη και τα δεδομένα της εφαρμογής θα πρέπει να είναι ασφαλή. Η ασφάλεια μπορεί να αυξηθεί με τους παρακάτω παράγοντες.

- Χρησιμοποιήστε την εσωτερική αποθήκευση αντί της εξωτερικής για την αποθήκευση αρχείων της εφαρμογής
- Χρησιμοποιήστε τους παρόχους περιεχομένου (content providers), όπου είναι δυνατόν
- Χρήση SSL κατά τη σύνδεση με το διαδίκτυο ή τις κλήσεις σε web services.
- Χρησιμοποιήστε τα κατάλληλα δικαιώματα για την πρόσβαση σε διαφορετικές λειτουργίες του συσκευής

➤ Fragmentation

Ο κατακερματισμός είναι η δύναμη αλλά και η αδυναμία του Android, ένας πονοκέφαλος για τους προγραμματιστές που παρέχει επίσης τη βάση για την παγκόσμια εμβέλεια του Android. Οι Android συσκευές έρχονται σε όλα τα σχήματα και μεγέθη, με πολύ διαφορετικά επίπεδα απόδοσης και μεγέθους οθόνης. Επιπλέον, υπάρχουν πολλές διαφορετικές εκδόσεις του Android που είναι ταυτόχρονα ενεργές ανά πάσα στιγμή, προσθέτοντας ένα επιπλέον επίπεδο κατακερματισμού. Αυτό σημαίνει ότι η ανάπτυξη εφαρμογών που λειτουργούν σε όλο το φάσμα των Android συσκευών μπορεί να είναι εξαιρετικά δύσκολη και χρονοβόρα.

Απαραίτητες λοιπόν οι δοκιμές με πολλές αναλύσεις και εκδόσεις.

➤ Presentation

Οι καλύτερες πρακτικές υπαγορεύουν ότι το Presentation Layer θα πρέπει να διαχωρίζεται από το υπολογιστικό επίπεδο (computing Layer) όταν είναι εφικτό. Το Presentation Layer πρέπει να έχει δυναμικά controls ανάλογα με την ανάλυση και το orientation της συσκευής.

➤ Application Rate

Δυνατότητα αναφορών από τους χρήστες Bugs και βελτιώσεων.

➤ Google Analytics

- Μέτρηση δραστηριότητας του χρήστη με την εφαρμογή.
- Καθορίστε σε ποιά τοποθεσία βρίσκονται οι χρήστες της εφαρμογής.
- Μάθετε τι ψάχνουν οι άνθρωποι για την εφαρμογή σας.
- Απεικονίστε που κάνουν πιο πολύ click οι χρήστες.

## 2.3 Σχεδιαστικά Πρότυπα για Android εφαρμογές

➤ Responsive design

Το Responsive Web Design [9] παρέχει την καλύτερη δυνατή εμπειρία προβολής ενός δικτυακού τόπου, δεν έχει σημασία τι είδος συσκευής χρησιμοποιεί ο χρήστης για να τον δει. Η Wikipedia το περιγράφει ως «μια προσέγγιση που στοχεύει στην διαμόρφωση χώρων για να παρέχουν μια καλύτερη οπτική εμπειρία, άμεση ανάγνωση και πλοήγηση με ελάχιστη αλλαγή μεγέθους, μετακίνηση, και κύλιση, σε ένα ευρύ φάσμα συσκευών (από κινητά τηλέφωνα έως οθόνες επιτραπέζιων υπολογιστών).»

Αυτό γίνεται με τη χρήση fluid grid, τα οποία είναι ένας όρος για σχεδιασμό που λειτουργεί ανεξάρτητα από το ποιο είναι το μέγεθος της οθόνης. Έτσι, δεν έχει σημασία πόσο πολύ θα αλλάξετε το μέγεθος της οθόνης, η ίδια διάταξη θα ανταποκριθεί αυτόματα σε αυτό το μέγεθος, σαν μια μπάλα που μεγαλώνει ή συρρικνώνεται για να χωρέσει μέσα από πολλές διαφορετικές στεφάνες.

➤ Adaptive design

Το Adaptive Web Design [9] είναι διαφορετικό από το Responsive Design στο ότι δεν υπάρχει μια διάταξη που αλλάζει πάντα. Αντί αυτού, υπάρχουν πολλές διαφορετικές διατάξεις για πολλαπλά μεγέθη οθόνης, και η διάταξη που χρησιμοποιείται εξαρτάται από το μέγεθος της οθόνης που χρησιμοποιείται. Για παράδειγμα, θα μπορούσε να υπάρξει μια ειδική διάταξη για τα κινητά τηλέφωνα, ταμπλέτες, και επιτραπέζιους υπολογιστές - καθένα από τα οποία γίνονται εκ των προτέρων. Αυτά τα τρία σχέδια βρίσκονται σε αναμονή, έως ότου κάποιος επισκέπτεται την ιστοσελίδα, η ιστοσελίδα ανιχνεύει τον τύπο της συσκευής που χρησιμοποιείται, και παραδίδει την προκαθορισμένη διάταξη για



τη συγκεκριμένη συσκευή. Έτσι, αντί μιας ενιαίας μπάλας που προσαρμόζεται σε διαφορετικού μεγέθους στεφάνες, τότε θα έχετε πολλές διαφορετικές μπάλες να χρησιμοποιούνται ανάλογα με το μέγεθος της στεφάνης.

➤ **Material design**

Το Material Design [5] (κωδικό όνομα: **quantum paper**) είναι ένα σύνολο σχεδιαστικών κανόνων (design language) που δημιουργήθηκε από τη Google και ανακοινώθηκε για πρώτη φορά στο συνέδριο Google I/O στις 25 Ιουνίου, 2014. Επεκτείνει το μοντέλο των «cards» (πρωτοεμφανίστηκε στο Google Now) που είναι σχεδιασμός, βασισμένος σε διάταξη πίνακα (grid layout), με κινούμενα σχέδια και μεταβάσεις που ανταποκρίνονται στις κινήσεις των χρηστών, χρήση padding και εφέ βάθους με κατάλληλο φωτισμό και σκίαση των αντικειμένων. Ο σχεδιαστής Matias Duarte εξήγησε ότι, σε αντίθεση με το πραγματικό χαρτί, το ψηφιακό υλικό μπορεί να επεκταθεί και να μορφοποιηθεί έξυπνα. Το material έχει φυσικές επιφάνειες και άκρα. Οι σκιές δίνουν νόημα σε αυτό που αγγίζεις. Η Google υποστηρίζει ότι η καινούρια σχεδιαστική τους γλώσσα βασίζεται στο χαρτί και το μολύβι.

## 2.4 Πρωτόκολλα Ανταλλαγής Μηνυμάτων

### 2.4.1 Rest Services

Τα Rest Services (Representational State Transfer) είναι ένα αρχιτεκτονικό στυλ, και μια προσέγγιση στην επικοινωνία που χρησιμοποιείται συχνά στην ανάπτυξη των υπηρεσιών Web. Η χρήση των Rest συχνά προτιμάται έναντι του στυλ των περισσότερο βαρέων SOAP Services (Simple Object Access Protocol), επειδή τα Rest αξιοποιούν καλύτερα το εύρος ζώνης, το οποίο τα καθιστά ως καλύτερη επιλογή για χρήση μέσω του Διαδικτύου. Η προσέγγιση SOAP απαιτεί την συγγραφή ή τη χρήση ενός προγράμματος διακομιστή – server program (για να εξυπηρετήσει τα δεδομένα) και ενός προγράμματος πελάτη – client program (για να αιτηθεί τα δεδομένα).

### 2.4.2 JSON

Το **JSON** [19] (JavaScript Object Notation) είναι ένα ελαφρύ πρότυπο ανταλλαγής δεδομένων. Είναι εύκολο για τους ανθρώπους να το διαβάσουν και γράψουν. Είναι εύκολο για τις μηχανές να το αναλύσουν (parse) και να το παράγουν (generate). Είναι

βασισμένο πάνω σε ένα υποσύνολο της γλώσσας προγραμματισμού JavaScript, Standard ECMA-262 Έκδοση 3η - Δεκέμβριος 1999. Το JSON είναι ένα πρότυπο κειμένου το οποίο είναι τελείως ανεξάρτητο από γλώσσες προγραμματισμού αλλά χρησιμοποιεί πρακτικές (conventions) οι οποίες είναι γνωστές στους προγραμματιστές της οικογένειας προγραμματισμού C, συμπεριλαμβανομένων των C, C++, C#, Java, JavaScript, Perl, Python, και πολλών άλλων. Αυτές οι ιδιότητες κάνουν το JSON μια ιδανική γλώσσα προγραμματισμού ανταλλαγής δεδομένων.

Το JSON είναι χτισμένο σε δύο δομές:

- Μια συλλογή από ζευγάρια ονομάτων/τιμών. Σε διάφορες γλώσσες προγραμματισμού, αυτό αντιλαμβάνεται ως ένα object, καταχώριση, δομή, λεξικό, πίνακα hash (hash table), λίστα κλειδιών, ή associative πίνακα.
- Μία ταξινομημένη λίστα τιμών. Στις περισσότερες γλώσσες προγραμματισμού, αυτό αντιλαμβάνεται ως ένας πίνακας (array), διάνυσμα, λίστα, ή ακολουθία.

Αυτά είναι τα universal data structures. Ουσιαστικά όλες οι μοντέρνες γλώσσες προγραμματισμού τα υποστηρίζουν με τον έναν ή τον άλλον τρόπο. Λογικό είναι πως ένα πρότυπο δεδομένων το οποίο είναι εύκολα μεταβαλλόμενο με γλώσσες προγραμματισμού οι οποίες επίσης είναι βασισμένες σε αυτές τις δομές.

## 2.5 Συστήματα Συστάσεων

Αν προσπαθούσαμε να δώσουμε ένα ορισμό για το τι είναι ένα σύστημα συστάσεων [20], θα μπορούσαμε να πούμε ότι είναι ένα σύστημα το οποίο έχει την δυνατότητα να συλλέγει και να παρουσιάζει σε κάθε χρήστη (παλιό και νέο) κάποιες επιλογές αντικειμένων τα οποία ανήκουν στο πεδίο των ενδιαφερόντων του. Αυτή ακριβώς η συλλογή και στη συνέχεια παρουσίαση των αντικειμένων, σχετικά με τα ενδιαφέροντα κάποιου χρήστη, αποτελεί μια πρόταση/σύσταση του συστήματος προς τον χρήστη.

Τα συστήματα συστάσεων έκαναν την εμφάνισή τους τα τελευταία χρόνια. Αποτελούν πεδίο που συγκεντρώνει μεγάλο ενδιαφέρον τόσο σε ακαδημαϊκό όσο και σε εμπορικό/οικονομικό επίπεδο. Ήδη σε πολλές περιπτώσεις ηλεκτρονικού εμπορίου έχουν υιοθετηθεί τεχνικές συστάσεων προς τους πελάτες τους, βάση της αγοραστικής συμπεριφοράς τους.

Λόγω της ραγδαίας εξάπλωσης του παγκόσμιου ιστού (World-Wide-Web) και των ηλεκτρονικών εφαρμογών που αναπτύσσονται σε αυτόν (ηλεκτρονικό εμπόριο),

προέκυψε έντονα η ανάγκη της εξατομίκευσης της πληροφορίας και της προσαρμογής της στα ενδιαφέροντα του χρήστη. Παρουσιάστηκε λοιπόν η ανάγκη ύπαρξης μηχανισμών φιλτραρίσματος της πληροφορίας και των δεδομένων, ώστε να προσαρμόζονται κατάλληλα στις ανάγκες και τα ενδιαφέροντα των χρηστών.

Αυτό ακριβώς το φιλτράρισμα και η προσαρμογή της πληροφορίας στα ενδιαφέροντα του χρήστη, αποτελούν τους λόγους ανάπτυξης και τους στόχους των συστημάτων συστάσεων.

Τα σύγχρονα συστήματα συστάσεων βασίζονται σε δύο κυρίως τρόπους λειτουργίας, ώστε να πραγματοποιήσουν μια πρόταση/σύσταση σε κάποιο χρήστη.

- Συστάσεις βάση περιεχομένου (content-based)
- Συστάσεις βάση «φίλτρου» συνεργασίας (collaborative-filtering) [18][20]

Στη δική μας περίπτωση θα ασχοληθούμε με τις συστάσεις βάση «φίλτρου» συνεργασίας (collaborative filtering).

### **2.5.1 Γιατί σύστημα «φίλτρου συνεργασίας»**

Τα συστήματα που λειτουργούν βάσει περιεχομένου χρησιμοποιούν τα περιεχόμενα ή τα χαρακτηριστικά εκείνων των αντικειμένων (υποτροφιών) τα οποία έχει βαθμολογήσει και έχει κρίνει ο χρήστης. Ως εκ τούτου δεν χρησιμοποιούν καθόλου την αξιολόγηση αντικειμένων όπως αυτή προκύπτει από παλιότερους χρήστες. Συνεπώς χρησιμοποιούν μόνο την γραμμή του πίνακα που αντιστοιχεί στον χρήστη τον οποίο θέλουμε να δώσουμε συστάσεις και τα διανύσματα χαρακτηριστικών για τα διάφορα αντικείμενα, αγνοώντας τα δεδομένα από τις γραμμές των που προκύπτουν από τις προτιμήσεις άλλων χρηστών.

Αντιθέτως, τα συστήματα που λειτουργούν βάση φίλτρου συνεργασίας θέτουν τις συστάσεις τους βασιζόμενα μόνο στις αξιολογήσεις των άλλων χρηστών, που είναι όμως παρόμοιοι και σχετικοί με το εκάστοτε συγκεκριμένο χρήστη.

### **2.5.2 Διαθέσιμα συστήματα συστάσεων για Java**

Το Apache Mahout είναι ένα Java framework στην περιοχή της εξόρυξης δεδομένων. Έχει ενσωματωμένο σύστημα Συστάσεων, μια Collaborative engine για εξατομικευμένες συστάσεις.

Το Vogoo είναι ένα PHP framework που υλοποιεί ένα σύστημα συστάσεων βάση Φίλτρου Συνεργασίας.

Μια έκδοση Java της μεθόδου συνεργατικού φιλτραρίσματος υλοποιείται στη βιβλιοθήκη Cofi. Αυτό αναπτύχθηκε από τον Daniel Lemire, ο δημιουργός των αλγορίθμων Slope-One. Υπάρχει επίσης μια έκδοση της PHP διαθέσιμη στην ιστοσελίδα του Lemire.

Έγινε προσπάθεια για την χρήση του Apache Mahout, αλλά δεν κρίθηκε ικανοποιητικός ο τρόπος που χρησιμοποιείτε η βιβλιοθήκη. Πρακτικά είναι φτιαγμένη για επεξεργασία batch και όχι πραγματικού χρόνου που είναι και το ζητούμενο για την δική μας περίπτωση.

### **2.5.3 Συστήματα συστάσεων βάση «Φίλτρου» Συνεργασίας**

Ο κύριος στόχος των συστημάτων (collaborative filtering) αυτών είναι να αναγνωρίσουν εκείνους τους χρήστες που έχουν παρόμοιες προτιμήσεις με ένα χρήστη. Στη συνέχεια προτείνονται σε αυτόν τον χρήστη οι προτιμήσεις των χρηστών με παρόμοιες προτιμήσεις. Αυτή η προσέγγιση στηρίζεται στην αντίληψη πως όταν αναζητούμαι μια πληροφορία, συνήθως εξετάζουμε τις απόψεις ή τις συμβουλές ατόμων με παρόμοιες προτιμήσεις και ενδιαφέροντα (των οποίων βέβαια την άποψη εμπιστευόμαστε). Τα συστήματα αυτά εκμεταλλεύονται τις επιλογές που κάποιοι άλλοι χρήστες έχουν «αναζητήσει» και «αξιολογήσει».

Για να είναι σε θέση ένα τέτοιο σύστημα να αποφασίσει για την σχετικότητα ή όχι των χρηστών πρέπει να υπάρχει ένας μηχανισμός σύγκρισης. Πρακτικά πρέπει να υπάρχει ένας μηχανισμός διατήρησης προφίλ των χρηστών αλλά και των αξιολογήσεων του.

### **2.5.4 Αξιολόγηση Συστημάτων Συστάσεων «Φίλτρου Συνεργασίας»**

Σύμφωνα με την Διδακτορική διατριβή του Γ. Αλεξανδρίδη για Συνεργατικά Συστήματα Συστάσεων με χρήση άμεσης & έμμεσης κοινωνικής πληροφορίας, η Παραγοντοποίηση Πινάκων παρουσίασε το μικρότερο μέσο απόλυτο σφάλμα από άλλες προσεγγίσεις στο dataset της MovieLens 100k, και δεδομένου ότι η πολυπλοκότητα της συγκεκριμένης κρίνεται κατάλληλη για εφαρμογή της σε περιβάλλον που θα πρέπει να παράγει συστάσεις σε πραγματικό χρόνο, θεωρείται η πιο κατάλληλη για την περίπτωση συστάσεων του συστήματος που θα υλοποιήσουμε.

### 2.5.5 Παραγοντοποίηση Πινάκων – Matrix Factorization

Οι συστάσεις (recommendations) μπορούν να δημιουργηθούν από ένα ευρύ φάσμα των αλγορίθμων. Ενώ οι μέθοδοι συστάσεων με βάση το χρήστη (user-based) ή με βάση τα αντικείμενα (item-based) είναι απλοί και διαισθητικοί, οι τεχνικές παραγοντοποίησης πινάκων [4][13][18][23] (matrix factorization) είναι συνήθως πιο αποτελεσματικές επειδή μας επιτρέπουν να αναγνωρίσουμε τα λανθάνοντα γνωρίσματα που διέπουν τις αλληλεπιδράσεις μεταξύ των χρηστών και αντικειμένων.

Φυσικά η παραγοντοποίηση πινάκων είναι απλώς ένα μαθηματικό εργαλείο, και ως εκ τούτου εφαρμόζεται σε πολλά σενάρια όπου κάποιος θα ήθελε να μάθει κάτι κρυμμένο κάτω από τα δεδομένα.

### 2.5.6 Βασική Ιδέα

Ακριβώς όπως υποδηλώνεται από το όνομα «Παραγοντοποίηση Πίνακα» είναι προφανώς η παραγοντοποίηση, δηλαδή η εύρεση δύο ή περισσότερων πινάκων τέτοιων ώστε ο πολλαπλασιασμός τους να δίνει τον αρχικό πίνακα.

Όπως αναφέρθηκε παραπάνω, η παραγοντοποίηση πινάκων μπορεί να χρησιμοποιηθεί για να ανακαλύψει λανθάνοντα γνωρίσματα που διέπουν τις αλληλεπιδράσεις ανάμεσα σε δύο διαφορετικά είδη οντοτήτων. Μια προφανής εφαρμογή της είναι η πρόβλεψη βαθμολογιών βάση «Φίλτρου» συνεργασίας – Collaborative Filtering.

Σε ένα σύστημα σύστασης όπως το Netflix ή το MovieLens, υπάρχει μια ομάδα χρηστών και ένα σύνολο από στοιχεία (ταινίες για τα δύο παραπάνω συστήματα). Δεδομένου ότι κάθε χρήστες έχουν αξιολογήσει κάποια στοιχεία στο σύστημα, θα θέλαμε να προβλέψουμε πώς οι χρήστες θα βαθμολογήσουν τα στοιχεία που δεν έχουν ακόμη αξιολογηθεί, έτσι ώστε να μπορούμε να κάνουμε συστάσεις προς τους χρήστες. Σε αυτή την περίπτωση, όλες οι πληροφορίες που έχουμε για τις υπάρχουσες αξιολογήσεις μπορούν να αναπαρασταθούν με ένα πίνακα. Ας υποθέσουμε τώρα έχουμε 5 χρήστες και 10 αντικείμενα, και οι βαθμολογίες είναι ακέραιοι αριθμοί που κυμαίνονται από 1 έως 5, ο πίνακας μπορεί να εμφανίζεται όπως παρακάτω (η παύλα σημαίνει ότι ο χρήστης δεν έχει βαθμολογήσει την αντικείμενο) :

	D1	D2	D3	D4
U1	5	3	-	1
U2	4	-	-	1

U3	1	1	-	5
U4	1	-	-	4
U5	-	1	5	4

**Πίνακας 1**

Ως εκ τούτου, το έργο της πρόβλεψης των κελιών χωρίς αξιολόγηση μπορεί να θεωρηθεί ως η συμπλήρωση των κενών (κελιά με παύλες), έτσι ώστε οι τιμές θα είναι συνεπής με τα υφιστάμενα αξιολογημένα κελιά στον πίνακα.

Η αίσθηση από τη χρήση παραγοντοποίησης πίνακα για να λύσει αυτό το πρόβλημα είναι ότι πρέπει να υπάρχουν κάποια λανθάνοντα γνωρίσματα που καθορίζουν τον τρόπο που ο χρήστης βαθμολογεί ένα στοιχείο. Για παράδειγμα, δύο χρήστες θα δίνουν υψηλές βαθμολογίες σε μια ορισμένη ταινία, αν και στους δύο αρέσουν οι ηθοποιοί της ταινίας, ή αν η ταινία είναι μια ταινία δράσης, το οποίο είναι ένα είδος που προτιμάται από τους χρήστες. Ως εκ τούτου, εάν μπορούμε να ανακαλύψουμε αυτά τα λανθάνοντα γνωρίσματα, θα πρέπει να είμαστε σε θέση να προβλέψουμε βαθμολογία σε σχέση με ένα συγκεκριμένο χρήστη και ένα συγκεκριμένο στοιχείο, επειδή τα χαρακτηριστικά που σχετίζονται με το χρήστη θα πρέπει να ταιριάζουν με τα χαρακτηριστικά που σχετίζονται με το στοιχείο.

Στην προσπάθεια να ανακαλύψουμε τα διαφορετικά χαρακτηριστικά, μπορούμε επίσης να κάνουμε την παραδοχή ότι ο αριθμός των χαρακτηριστικών θα είναι μικρότερος από τον αριθμό των χρηστών και τον αριθμό των στοιχείων. Δεν πρέπει να είναι δύσκολο να κατανοήσουμε αυτή την υπόθεση, διότι προφανώς δεν θα ήταν λογικό να υποθέσουμε ότι κάθε χρήστης συνδέεται με ένα μοναδικό χαρακτηριστικό (αν και αυτό δεν είναι αδύνατο). Και ούτως ή άλλως, αν αυτή είναι η περίπτωση, δεν θα υπήρχε λόγος να προβούμε σε συστάσεις, γιατί κάθε ένας από αυτούς τους χρήστες δεν θα ενδιαφερόταν για τα βαθμολογημένα αντικείμενα από τους άλλους χρήστες. Ομοίως, το ίδιο επιχείρημα ισχύει και για τα στοιχεία.

## **2.5.7 Μαθηματικά Παραγοντοποίησης Πινάκων**

### **2.5.7.1 Μικρή Εισαγωγή Πράξεων Πινάκων**

$$\begin{bmatrix} 1 & 4 \\ 0 & 3 \end{bmatrix} \times \begin{bmatrix} 0 & 3 & 0 \\ 2 & 1 & 4 \end{bmatrix} = \begin{bmatrix} 1*0 + 4*2 & 1*3 + 4*1 & 1*0 + 4*4 \\ 0*0 + 3*2 & 0*3 + 3*1 & 0*0 + 3*4 \end{bmatrix} = \begin{bmatrix} 8 & 7 & 16 \\ 6 & 3 & 12 \end{bmatrix}$$

**Σχήμα 1.** Πολλαπλασιασμός πινάκων

Στο παραπάνω πίνακα φαίνεται ο πολλαπλασιασμός δύο πινάκων. Όταν πολλαπλασιάζουμε πίνακες, ο πρώτος πίνακας (A στο σχήμα) πρέπει να έχει τον ίδιο αριθμό στηλών με τον αριθμό γραμμών του δεύτερου πίνακα (B στο σχήμα). Το αποτέλεσμα του πολλαπλασιασμού παράγει ένα πίνακα Γ ο οποίος έχει ίδιο αριθμό γραμμών με τον A και ίδιο αριθμό στηλών με τον B.

$$\begin{pmatrix} a & d \\ b & e \\ c & f \end{pmatrix}^T = \begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix}$$

**Σχήμα 2.** Ανάστροφος Πίνακας

Ο ανάστροφος ενός πίνακα A προκύπτει εάν διατάξουμε τις γραμμές του A ως στήλες με την ίδια σειρά (οπότε αυτόματα οι στήλες γίνονται γραμμές). Συμβολίζεται  $A^T$ . Έτσι το στοιχείο που βρίσκεται στη θέση ij, στον ανάστροφο θα πάρει τη θέση ji. Πιο αυστηρά, Εάν  $A = (a_{ij})$  είναι ένας  $m \times n$  πίνακας, τότε ο ανάστροφος  $A^T = (a_{ij}')$  είναι ο  $n \times m$  πίνακας όπου  $a_{ij}' = a_{ji}$ , για όλα τα i, j.

### 2.5.7.2 Παραγοντοποίηση Πινάκων

Έχοντας μια αίσθηση τώρα πίσω από την παραγοντοποίηση πίνακα καθώς και τις βασικές πράξεις πινάκων που θα χρειαστούμε, μπορούμε να προχωρήσουμε στα μαθηματικά της Παραγοντοποίησης Πινάκων. Πρώτον, έχουμε ένα σύνολο U των χρηστών, καθώς και ένα σύνολο D των στοιχείων (υποτροφιών).

Έστω ότι έχουμε έναν Πίνακα R διάστασης  $|U| \times |D|$  (όπου U οι χρήστες και D τα στοιχεία - υποτροφίες). Στο πίνακα αυτό έχει καταχωρηθεί σε κάθε κελί το πόσες φορές ο χρήστης  $U_i$  έχει αξιολογήσει την συγκεκριμένη υποτροφή  $D_j$ . Επίσης υποθέτουμε ότι θα θέλαμε να ανακαλύψουμε K λανθάνοντα γνωρίσματα. Θα πρέπει

λοιπόν να βρεθούν δύο πίνακες  $\mathbf{P}$  (a |U|xK πίνακα) και  $\mathbf{Q}$  (a|D|xK πίνακα) των οποίων το γινόμενο τους να προσεγγίζει τον  $\mathbf{R}$ :

$$\mathbf{R} \approx \mathbf{P} \mathbf{x} \mathbf{Q}^T = \hat{\mathbf{R}}$$

Με τον τρόπο αυτό, κάθε γραμμή του  $\mathbf{P}$  θα αντιπροσωπεύει τη δύναμη των ενώσεων μεταξύ ενός χρήστη και των χαρακτηριστικών. Ομοίως, κάθε σειρά του  $\mathbf{Q}$  θα αντιπροσωπεύει τη δύναμη των ενώσεων μεταξύ ενός στοιχείου και των χαρακτηριστικών.

Για να πάρουμε την πρόβλεψη μιας αξιολόγησης ενός στοιχείου  $d_j$  από  $u_i$ , μπορούμε να υπολογίσουμε το γινόμενο των δύο διανυσμάτων που αντιστοιχούν σε  $u_i$  και  $d_j$ :

$$\hat{r}_{ij} = p_i^T q_j = \sum_{k=1}^k p_{ik} q_{kj}$$

Τώρα, πρέπει να βρούμε έναν τρόπο ώστε να βρούμε τον  $\mathbf{P}$  και τον  $\mathbf{Q}$ . Ένας τρόπος για να προσεγγίσουμε αυτό το πρόβλημα είναι η αρχικοποίηση των δύο πινάκων με κάποιες τιμές, να υπολογίσουμε πόσο «διαφορετικό» είναι το γινόμενο από το  $\mathbf{M}$ , και στη συνέχεια να προσπαθήσουμε να ελαχιστοποιήσουμε τη διαφορά αυτή επαναληπτικά. Μία τέτοια μέθοδος ονομάζεται σταδιακή μείωση – gradient descent, με στόχο την εύρεση ενός τοπικού ελάχιστου της διαφοράς.

Η διαφορά εδώ, που συνήθως ονομάζεται το σφάλμα μεταξύ της εκτιμώμενης βαθμολογίας και της πραγματικής βαθμολογίας, μπορεί να υπολογιστεί από την ακόλουθη εξίσωση για κάθε ζεύγος χρήστη-στοιχείου (υποτροφίας) :

$$e_{ij}^2 = (r_{ij} - \hat{r}_{ij})^2 = (r_{ij} - \sum_{k=1}^K p_{ik} q_{kj})^2$$

Εδώ θεωρούμε το τετραγωνικό σφάλμα, επειδή η εκτιμώμενη βαθμολογία μπορεί να είναι είτε υψηλότερη ή χαμηλότερη από την πραγματική βαθμολογία.

Για να ελαχιστοποιηθεί το σφάλμα, πρέπει να ξέρουμε σε ποια κατεύθυνση θα πρέπει να τροποποιηθούν οι τιμές των  $p_{ik}$  και  $q_{kj}$ . Με άλλα λόγια, πρέπει να γνωρίζουμε την διαβάθμιση από τις τρέχουσες τιμές, και ως εκ τούτου διαφοροποιούμε την παραπάνω εξίσωση σε σχέση με αυτές τις δύο μεταβλητές ξεχωριστά :

$$\frac{\partial}{\partial p_{ik}} e_{ij}^2 = -2 (r_{ij} - \hat{r}_{ij})(q_{kj}) = -2e_{ij}q_{kj}$$

$$\frac{\partial}{\partial q_{ik}} e_{ij}^2 = -2 (r_{ij} - \hat{r}_{ij})(p_{ik}) = -2e_{ij}p_{ik}$$

Έχοντας υπολογίσει τη διαβάθμιση, μπορούμε τώρα να διατυπώσουμε τους κανόνες ενημέρωσης για τα  $p_{ik}$  και  $q_{kj}$  :



$$p'_{ik} = p_{ik} + a \frac{\partial}{\partial p_{ik}} e_{ij}^2 = p_{ik} + 2ae_{ij}q_{kj}$$

$$q'_{ik} = q_{ik} + a \frac{\partial}{\partial p_{kj}} e_{ij}^2 = q_{kj} + 2ae_{ij}p_{ik}$$

Το  $\alpha$  είναι μια σταθερά της οποίας η τιμή καθορίζει τον ρυθμό προσέγγισης στο ελάχιστο. Συνήθως επιλέγουμε μια μικρή τιμή του  $\alpha$ , δηλαδή 0,0002. Αυτό οφείλεται στο γεγονός ότι, αν κάνουμε πολύ μεγάλο βήμα προς το ελάχιστο, διατρέχουμε τον κίνδυνο να χάσουμε το ελάχιστο και να ταλαντευόμαστε γύρω από αυτό.

Μια ερώτηση που θα μπορούσε να έρθει στο μυαλό μας τώρα είναι : αν βρούμε δύο Πίνακες  $\mathbf{P}$  και  $\mathbf{Q}$  τέτοιους ώστε  $\mathbf{P} \times \mathbf{Q}$  να προσεγγίζει τον  $\mathbf{R}$ , δεν είναι ότι οι προβλέψεις μας για όλες τις μη βαθμολογημένες αξιολογήσεις θα είναι μηδενικές; Στην πραγματικότητα, δεν προσπαθούμε πραγματικά να καταλήξουμε από τον  $\mathbf{P}$  και  $\mathbf{Q}$  στην ακριβή αναπαραγωγή του  $\mathbf{R}$ . Αντί αυτού, θα προσπαθήσουμε μόνο να ελαχιστοποιήσουμε τα λάθη των παρατηρούμενων ζευγών χρήστη-στοιχείου. Με άλλα λόγια, αν αφήσουμε το  $T$  να είναι ένα σύνολο από πλειάδες, καθένα από τα οποία έχει τη μορφή της  $(u_i, d_j, r_{ij})$ , έτσι ώστε  $T$  να περιέχει όλα τα παρατηρούμενα ζεύγη χρήστη-στοιχείου μαζί με τις σχετικές αξιολογήσεις - rating, προσπαθούμε μόνο να ελαχιστοποιήσουμε το  $e_{ij}$  για  $(u_i, d_j, r_{ij}) \in T$ . (Με άλλα λόγια, το  $T$  είναι το σύνολο μας με τα δεδομένα εκπαίδευσης.) Όσο για τους υπόλοιπους αγνώστους, θα είμαστε σε θέση να καθορίσουμε τις τιμές τους αφού οι σχέσεις μεταξύ χρηστών, στοιχείων και χαρακτηριστικών έχουν αντληθεί.

Χρησιμοποιώντας τους παραπάνω κανόνες ενημέρωσης, μπορούμε στη συνέχεια να εκτελέσουμε επαναληπτικά τη διαδικασία έως ότου το σφάλμα συγκλίνει στο ελάχιστο. Μπορούμε να ελέγξουμε το συνολικό σφάλμα, όπως αυτό υπολογίζεται με βάση την ακόλουθη εξίσωση και να καθορίσουμε πότε θα πρέπει να σταματήσει τη διαδικασία.

$$E = \sum_{(u_i, d_j, r_{ij}) \in T} e_{ij}^2 = \sum_{(u_i, d_j, r_{ij}) \in T} \left( r_{ij} - \sum_{k=1}^K p_{ik} q_{kj} \right)^2$$

### 2.5.7.3 Regularization - Κανονικοποίηση

Ο παραπάνω αλγόριθμος είναι ένα πολύ βασικός αλγόριθμος για την παραγοντοποίηση ενός πίνακα. Υπάρχουν πολλές μέθοδοι που μπορούν να κάνουν τα πράγματα να φαίνονται πιο περίπλοκα. Μια κοινή επέκταση σε αυτό το βασικό αλγόριθμο είναι να

εισαγάγει την κανονικοποίηση - regularization για να αποφευχθεί η υπερπροσαρμογή - overfitting. Αυτό γίνεται με την προσθήκη μιας παραμέτρου  $\beta$  και τροποποίησης του τετραγωνικού σφάλματος ως εξής :

$$e_{ij}^2 = (r_{ij} - \sum_{k=1}^K p_{ik} q_{kj})^2 + \frac{\beta}{2} \sum_{k=1}^K (\|P\|^2 - \|Q\|^2)$$

Με άλλα λόγια, η νέα παράμετρος  $\beta$  χρησιμοποιείται για τον έλεγχο των μεγεθών των γνωρισμάτων χρηστών - χαρακτηριστικών και των στοιχείων (υποτροφιών) - χαρακτηριστικών διανυσμάτων έτσι ώστε P και Q να δώσουν μια καλή προσέγγιση του R, χωρίς να χρειάζεται να περιέχουν μεγάλους αριθμούς. Στην πράξη, το  $\beta$  έχει οριστεί στην περιοχή του 0,02. Οι νέοι κανόνες ενημέρωσης για αυτό το τετραγωνικό σφάλμα μπορούν να ληφθούν με μία διαδικασία παρόμοια με αυτή που περιγράφεται παραπάνω. Οι νέοι κανόνες ενημερωμένης έκδοσης είναι οι εξής.

$$p'_{ik} = p_{ik} + \alpha \frac{\partial}{\partial p_{ik}} e_{ij}^2 = p_{ik} + \alpha(2e_{ij}q_{kj} - \beta p_{ik})$$

$$q'_{ik} = q_{ik} + \alpha \frac{\partial}{\partial p_{kj}} e_{ij}^2 = q_{kj} + \alpha(2e_{ij}p_{ik} - \beta q_{kj})$$

## 2.6 Απόσταση κατά Levenshtein (String Edit Distance)

Η απόσταση Levenshtein [12][14] ή SED (String Edit Distance) (Levenshtein (1966)) ορίζεται μεταξύ δύο συμβολοσειρών (strings) τυχαίου μήκους, δεν απαιτείται δηλαδή οι συμβολοσειρές αυτές να έχουν το ίδιο μήκος. Η απόσταση Levenshtein μετρά τις διαφορές μεταξύ των δύο συμβολοσειρών, όχι μόνο εκεί που υπάρχουν διαφορετικά γράμματα αλλά ακόμη και εκεί που το ένα string έχει κάποιο χαρακτήρα ενώ το άλλο όχι.

Μαθηματικά, η απόσταση Levenshtein μεταξύ δύο συμβολοσειρών a, b (μήκους |a| και |b| αντίστοιχα) δίνεται από  $lev_{a,b}(|a|, |b|)$  όπου

$$lev_{a,b}(|i|, |j|) = \begin{cases} \max(i, j) & \text{αν } \min(i, j) = 0, \\ \min \begin{cases} lev_{a,b}(i-1, j) + 1 \\ lev_{a,b}(i, j-1) + 1 \\ lev_{a,b}(i-1, j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{διαφορετικά} \end{cases}$$

Για παράδειγμα, οι λέξεις «καλάμι» και «καλάθι» έχουν μεγάλη ομοιότητα καθώς με μία και μόνο αλλαγή γράμματος, μ με θ, η πρώτη λέξη μπορεί να μετατραπεί στη δεύτερη. Επιπλέον, η λέξη «άβουλος» μπορεί να μετατραπεί στη λέξη «άβολος» με τη διαγραφή

ενός γράμματος, του «υ», ενώ ανάλογα η αντίστροφη μετατροπή γίνεται με την προσθήκη του γράμματος «υ».

Πιο αναλυτικά, η απόσταση Levenshtein μεταξύ δύο συμβολοσειρών,  $s_1$  και  $s_2$ , ορίζεται ως το μικρότερο πλήθος των αλλαγών που πρέπει να λάβουν χώρα, προκειμένου η συμβολοσειρά  $s_1$  να μετατραπεί στην  $s_2$ . Οι αλλαγές που ενδέχεται να γίνουν είναι:

- αλλαγή ενός γράμματος σε κάποιο άλλο
- προσθήκη ενός γράμματος
- διαγραφή ενός γράμματος

Κάθε μία από τις αλλαγές αυτές μπορεί να έχει διαφορετικό βάρος στον υπολογισμό της απόστασης, ωστόσο στην απλούστερη περίπτωση οι παραπάνω τρεις διαδικασίες είναι ισότιμες και αυξάνουν κατά μία μονάδα την απόσταση ενός ζεύγους συμβολοσειρών. Για τον υπολογισμό της απόστασης κατασκευάζεται ένας πίνακας όπως αυτός που εικονίζεται στο σχήμα 3. Κάθε ένα από τα στοιχεία του πίνακα αυτού συμπληρώνεται με βάση τον ακόλουθο ψευδοκώδικα, όπου το  $m[i,j]$  αναφέρεται στο στοιχείο του πίνακα που βρίσκεται στη θέση  $i, j$ , ενώ  $|s_1|$  και  $|s_2|$  είναι τα πλήθη των γραμμάτων των δύο συμβολοσειρών αντίστοιχα.

$$m[i,j] = d(s_1[1..i], s_2[1..j])$$

$$m[0,0] = 0$$

$$m[i,0] = i, i=1..|s_1| \quad m[0,j] = j, j=1..|s_2|$$

$$m[i,j] = \min(m[i-1,j-1]$$

$$+ \text{if } s_1[i]=s_2[j] \text{ then } 0 \text{ else } 1 \text{ fi}, m[i-1, j] + 1,$$

$$m[i, j-1] + 1), i=1..|s_1|, j=1..|s_2|$$

		A	M	A	P	T	Ω	Λ	O	Σ
A	0	1	2	3	4	5	6	7	8	9
Γ	1	0	1	2	3	4	5	6	7	8
A	2	1	1	2	3	4	5	6	7	8
Π	3	2	2	1	2	3	4	5	6	7
Η	4	3	3	2	2	3	4	5	6	7
T	5	4	4	3	3	3	4	5	6	7
O	6	5	5	4	4	3	4	5	6	7
Σ	7	6	6	5	5	4	4	5	5	6
Σ	8	7	7	6	6	5	5	5	6	5

**Σχήμα 3.** Υπολογισμός πίνακα με αποστάσεις Levenshtein για τις λέξεις «αγαπητός» και «αμαρτωλός». Στον πίνακα φαίνονται και οι συντομότεροι τρόποι μετασχηματισμού της μιας λέξης στην άλλη καθώς και το κόστος κάθε μετασχηματισμού.

Ένα παράδειγμα υπολογισμού της απόστασης για τις λέξεις «αγαπητός» και «αμαρτωλός») παρουσιάζεται στον παραπάνω πίνακα. Ο πίνακας αυτός μπορεί να υπολογιστεί γραμμή γραμμή, καθώς το κάθε στοιχείο μίας γραμμής εξαρτάται αποκλειστικά από τα στοιχεία της προηγούμενης, αλλά και εκείνης στην οποία βρίσκεται. Συνεπώς, η πολυπλοκότητα αυτής της μετρικής είναι  $O(|s_1|*|s_2|)$  και, σε περίπτωση που τα  $s_1, s_2$  είναι ίσα με  $n$ , η πολυπλοκότητα είναι  $O(n^2)$ .

Είναι φανερό ότι η απόσταση δύο συμβολοσειρών δεν εξαρτάται από τη μεταξύ τους διάταξη, όπως επίσης και ότι υπάρχουν περισσότερες της μίας εναλλακτικές διαδρομές για τη μετατροπή της μίας συμβολοσειράς στην άλλη με το ελάχιστο δυνατό κόστος. Σε κάθε περίπτωση πάντως, η απόσταση Levenshtein δίνει το μικρότερο δυνατό κόστος μετατροπής.

Κάποιες από τις παραλλαγές του αλγόριθμου αποδίδουν διαφορετικά βάρη στην κάθε αλλαγή, πέραν του 0 και 1. Ένα τυπικό παράδειγμα είναι τα *linear gap-costs* (Gotoh (1990)), όπου μία συνεχόμενη σειρά από εισαγωγές (ή διαγραφές) μήκους  $x$  έχει κόστος  $ax + b$ , όπου τα  $a, b$  είναι κάποιες σταθερές παράμετροι. Μάλιστα, σε περίπτωση που το  $b > 0$  η συγκεκριμένη παραλλαγή «τιμωρεί» ακόμη και μικρές συνεχόμενες αλλαγές, αποδίδοντάς τους μεγαλύτερο βάρος.

## 2.7 Διαθέσιμοι Database Servers

Για την μοντελοποίηση των δεδομένων των υποτροφιών μπορούμε να χρησιμοποιήσουμε κάποια σχεσιακή βάση δεδομένων οι οποίες παρέχονται για δωρεάν χρήση. Οποιαδήποτε βάση δεδομένων και αν χρησιμοποιήσουμε είναι σε θέση να εξυπηρετήσει τις ανάγκες της διατριβής.

Παρακάτω γίνεται μια συνοπτική παρουσίαση των πιο γνωστών εφαρμογών Βάσεων Δεδομένων για ελεύθερη χρήση.

### 2.7.1 SQL Server

Ο **SQL Server [15]** (προφέρεται «σίκουελ σέρβερ») είναι μια σχεσιακή βάση δεδομένων, η οποία αναπτύσσεται από τη Microsoft. Οι κύριες γλώσσες που χρησιμοποιούνται είναι η T-SQL και η ANSI SQL. Ο SQL Server βγήκε για πρώτη φορά στην αγορά το 1989 σε συνεργασία με την Sybase. Η τελευταία έκδοση του είναι ο SQL Server 2014 και το 2016 θα ανακοινωθεί και η επίσημη νέα έκδοση του.

Τα εργαλεία διαχείρισης της συγκεκριμένης βάσης δεδομένων είναι ένα από τα κύρια προτερήματα του SQL Server.

Η έκδοση του SQL Express παρέχεται για δωρεάν χρήση μέχρι 10GB και πρόσφατα δόθηκε και η έκδοση του SQL Developer.

### 2.7.2 MySQL

Ο **MySQL [1]** (προφέρεται «μυι εσκιουελ») είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) ανοιχτού κώδικα. Τον Ιούλιο του 2013, ήταν το δεύτερο παγκόσμιος πιο χρησιμοποιούμενο RDBMS, και πρώτο ανοιχτού κώδικα client-server μοντέλο RDBMS.

Το σχέδιο ανάπτυξης του MySQL έχει κάνει τον πηγαίο κώδικα του διαθέσιμο υπό τους όρους της GNU General Public License<sup>[21]</sup>, καθώς και κάτω από μια ποικιλία από αποκλειστικές συμφωνίες. Ο MySQL ανήκει και χρηματοδοτείται από μια και μόνο κερδοσκοπική εταιρεία, τη σουηδική εταιρεία MySQL AB, τώρα ανήκει στην Oracle Corporation.

Για εμπορική χρήση, είναι διαθέσιμες αρκετές εκδόσεις επί πληρωμή, οι οποίες προσφέρουν επιπλέον λειτουργικότητα. Ο MySQL είναι μια δημοφιλής επιλογή βάσεων δεδομένων για χρήση σε εφαρμογές web, και υποστηρίζει μια ευρεία γκάμα λειτουργικών συστημάτων στα οποία είναι διαθέσιμος (Linux, Windows).

Εφαρμογές που χρησιμοποιούν τη βάση δεδομένων MySQL περιλαμβάνουν: TYPO3, MODx, Joomla, WordPress, phpBB, myBB, Drupal και άλλα λογισμικά, αυτός κατά τη γνώμη μου και το γεγονός του ανοιχτού κώδικα είναι ο κύριος λόγος εξάπλωσής του. Ο MySQL χρησιμοποιείται επίσης σε πολλές υψηλού προφίλ, ιστοσελίδες μεγάλης κλίμακας, συμπεριλαμβανομένης της Google (αν και όχι για τις αναζητήσεις), και του Facebook, Twitter, Flickr, και του YouTube.

### 2.7.3 PostgreSQL

Η **PostgreSQL [10]** είναι και αυτό ένα σύστημα διαχείρισης σχεσιακών βάσεων ανοιχτού κώδικα το οποίο συμπεριφέρεται καλά σε UNIX based συστήματα όπως το Linux. Είναι πολύ πιο ισχυρή, επεκτάσιμη, και συμβατή με τα πρότυπα από ότι ο MySQL, και μπορεί να χειριστεί πολλαπλές συναλλαγές εύκολα. Η ανάπτυξη της διαρκεί ήδη πάνω από δύο δεκαετίες και βασίζεται σε μια αποδεδειγμένα καλή αρχιτεκτονική η οποία έχει δημιουργήσει μια ισχυρή αντίληψη των χρηστών της γύρω από την

αξιοπιστία, την ακεραιότητα δεδομένων και την ορθή λειτουργία. Η Postgres Plus υποστηρίζει τα πιο δημοφιλή περιβάλλοντα, συμπεριλαμβανομένων των Linux 32, Linux 64, Windows και Mac OS X.

#### **2.7.4 Connection pooling (Ομαδοποίηση συνδέσεων Βάσης)**

Στην τεχνολογία λογισμικού, το **connection pooling** [24] είναι μια κρυφή μνήμη των συνδέσεων βάσης δεδομένων που διατηρείται έτσι ώστε οι συνδέσεις μπορούν να ξαναχρησιμοποιηθούν όταν μελλοντικές αιτήσεις απαιτηθούν προς την βάση δεδομένων. Το connection pooling χρησιμοποιείται για τη βελτίωση της απόδοσης της εκτέλεσης εντολών σε μια βάση δεδομένων. Το άνοιγμα και η διατήρηση μιας σύνδεσης βάσης δεδομένων για κάθε χρήστη, ειδικά τα request που γίνονται σε μια δυναμική βάση δεδομένων από εφαρμογές web, είναι δαπανηρή και αφήνει σκουπίδια (στη μνήμη). Με το connection pooling, μετά από την δημιουργία μιας νέας σύνδεσης, αυτή τοποθετείται στο pool και χρησιμοποιείται και πάλι (πρακτικά με το κλείσιμο της σύνδεσης από την κάθε εφαρμογή δεν φεύγει από το pool), έτσι μια νέα σύνδεση δεν χρειάζεται να την δημιουργήσει ξανά (αυτό είναι και το χρονοβόρο). Αν όλες οι συνδέσεις του pool χρησιμοποιούνται, μία νέα σύνδεση γίνεται και προστίθεται και αυτή στο pool. Η ομαδοποίηση αυτή των συνδέσεων μειώνει επίσης δραστικά τον χρόνο που ένας χρήστης πρέπει να περιμένει για να δημιουργηθεί μια σύνδεση με τη βάση δεδομένων.

## **2.8 Middleware Εξυπηρέτησης και Σύνδεση με την Βάση Δεδομένων**

### **2.8.1 Ορισμός Middleware**

Η επικοινωνία μεταξύ πελάτη (client) και του διακομιστή (server) δεν είναι άμεση. Συχνά, όταν ένας πελάτης απαιτεί μία υπηρεσία από έναν διακομιστή το αίτημα περνά μέσω πολλών ενδιάμεσων στρωμάτων λογισμικού που παρεμβάλλονται μεταξύ των πελατών και των διακομιστών.

Αυτό το ενδιάμεσο λογισμικό ονομάζεται **Middleware**. Αποτελεί ζωτικό μέρος κάθε συστήματος πελάτη-διακομιστή. Ευτυχώς, τις περισσότερες φορές, ο προγραμματιστής

και ο σχεδιαστής δεν χρειάζεται ν' ανησυχούν για το πώς λειτουργεί. Χρειάζεται μόνο να το χρησιμοποιούν μέσω επικλήσεων μεθόδων. Η έννοια του Middleware είναι κάπως ομιχλώδης. Ο καλύτερος ορισμός είναι των Orfali, Harkey και Edwards (1998) [17]:

*Το Middleware είναι ένας ευρύς όρος που καλύπτει όλα τα καταναμημένα λογισμικά που χρειάζονται για την υποστήριξη της συνεργασίας πελατών και διακομιστών. Πού αρχίζει το middleware και πού τελειώνει; Αρχίζει με την ομάδα API στην πλευρά του πελάτη που χρησιμοποιείται για την επίκληση μιας υπηρεσίας, και καλύπτει την μεταβίβαση του αιτήματος στο δίκτυο και την αντίστοιχη απάντηση.*

Το καλύτερο παράδειγμα middleware είναι το λογισμικό διεπαφής ενός browser και του Παγκόσμιου Ιστού. Όταν γίνεται αίτηση για μια ιστοσελίδα, ας πούμε όταν ένας χρήστης κάνει κλικ σ' έναν υπερσύνδεσμο (hyperlink) μιας σελίδας, ένα κείμενο, που αποτελεί μέρος ενός πρωτοκόλλου, αποστέλλεται στο middleware ρωτώντας για την σελίδα. Στην συνέχεια, το middleware θα εντοπίσει την σελίδα, θα την ανακτήσει και θα την στείλει πίσω στον browser. Τμήμα του Middleware είναι ένα πρωτόκολλο που το αφορά η ανεύρεση σελίδων, η παρακολούθηση λαθών, η αναφορά λαθών, η μεταβίβαση δεδομένων και η επικοινωνία με τα κατώτερα στρώματα λογισμικού, που αποτελεί μέρος του TCP/IP.

Υπάρχουν δύο κατηγορίες middleware (R. Pressman and D.C, 2000) το γενικό middleware και η υπηρεσία middleware. Το γενικό middleware είναι λογισμικό που συσχετίζεται με υπηρεσίες που ζητούνται από όλους τους πελάτες και τους διακομιστές. Κανονικά το περισσότερο από αυτό το middleware αφορά τα λειτουργικά συστήματα. Το τυπικό λογισμικό που υπάγεται σ' αυτό το πλαίσιο περιλαμβάνει:

- Λογισμικό για την διεξαγωγή επεξεργασιών όπως η μεταφορά ακατέργαστων δεδομένων χαρακτήρων στο Internet.
- Λογισμικό που συγχρονίζει αντιγραμμένα αρχεία.
- Λογισμικό που διαχειρίζεται μία καταναμημένη συλλογή αρχείων.

Η υπηρεσία middleware συσχετίζεται με μία συγκεκριμένη υπηρεσία όπως η εκτύπωση αρχείων σ' έναν απομακρυσμένο υπολογιστή. Τα τυπικά παραδείγματα περιλαμβάνουν:

- Λογισμικό που επιτρέπει σ' έναν πελάτη να θέσει ερώτημα σε μία βάση δεδομένων, π.χ. λογισμικό που ερμηνεύει και εκτελεί ένα ερώτημα που τίθεται από τον χρήστη ενός πελάτη υπολογιστή και επεξεργάζεται τα δεδομένα στέλνοντας τα αποτελέσματα στον πελάτη.

- Λογισμικό που αφορά συνδυασμούς κατανεμημένων αντικειμένων όπως το RMI. Το λογισμικό αυτό επιτρέπει στον χρήστη να δημιουργήσει ένα αντικείμενο και στους πελάτες ν' αποστείλουν μηνύματα σ' αυτό το αντικείμενο. Εφαρμόζει λειτουργίες που αφορούν την ασφάλεια, τον εντοπισμό ενός αντικειμένου, την διαβίβαση δεδομένων, που σχετίζονται με την μέθοδο ορισμών, σ' ένα κατανεμημένο αντικείμενο και την μεταφορά των αποτελεσμάτων ενός μηνύματος που έχει αποσταλεί σ' ένα κατανεμημένο αντικείμενο.
- Το middleware που αφορά τις ομάδες συζητήσεων που δίνει την δυνατότητα στον χρήστη να διαβάζει και να στέλνει μηνύματα στην ομάδα.

## 2.9 Απαιτήσεις Ασφαλείας (Security)

Ένα από τα μεγαλύτερα χτυπήματα κατά του Android είναι ότι δεν είναι αρκετά ασφαλές για επιχειρησιακή χρήση, αλλά τα πράγματα από έκδοση σε έκδοση γίνονται όλο και καλύτερα.

Αρκετά θέματα ασφαλείας υπάρχουν για το Android, ειδικά όταν πρόκειται για το Android security app, την κεντρική διαχείριση και την ικανότητα να προστατεύονται τα δεδομένα σε περίπτωση απώλειας ή κλοπής συσκευών. Αυτά όλα είναι λογικοί λόγοι για τους οργανισμούς να σκεφτούν δύο φορές πριν επιτρέψουν τις Android συσκευές στο εταιρικό περιβάλλον τους.

### 2.9.1 Android app Security Risks

Η έλλειψη ελέγχου των εφαρμογών είναι ίσως το πιο σοβαρό από τα ζητήματα ασφαλείας που αντιμετωπίζουν οι επιχειρήσεις. Λαμβάνοντας υπόψη ότι η Apple έχει μια αυστηρή διαδικασία εφαρμογής της έγκρισης και δεν επιτρέπει μη εγκεκριμένες εφαρμογές να εγκατασταθούν στις συσκευές της, ενώ οι χρήστες του Android είναι ελεύθεροι να εγκαταστήσουν οποιαδήποτε εφαρμογή αναπτύσσει ο καθένας. Το Android Market προσπαθεί να προσφέρει κάποιο επίπεδο ελέγχου επί των διαθέσιμων εφαρμογών στους χρήστες, αλλά και άλλες εφαρμογές - μερικά δυνητικά επιβλαβείς - είναι διαθέσιμα σε εναλλακτικές καταστήματα app ή ακόμα και στις ιστοσελίδες των προγραμματιστών.



Για να κάνει τους χρήστες να γνωρίζουν τα πιθανά ζητήματα ασφάλειας του Android γύρω από τις εφαρμογές, οι περισσότερες συσκευές εξηγούν σε ποια δεδομένα και ποια χαρακτηριστικά απαιτείται να έχει πρόσβαση η εφαρμογή, και στη συνέχεια ρωτά τον χρήστη να πατήσει OK πριν από την εγκατάσταση. Για παράδειγμα, ο χρήστης θα ξέρει αν μια εφαρμογή θέλει να χρησιμοποιήσει την λίστα των επαφών του, ή αν θέλει να έχει πρόσβαση στην κάρτα SIM για την αποστολή μηνυμάτων κειμένου.

Στις περισσότερες περιπτώσεις, όμως, αυτά τα προειδοποιητικά μηνύματα δεν βοηθάνε και πολύ. Αν ένας χρήστης εγκαθιστά μια εφαρμογή που υπόσχεται να κάνει ευκολότερη την επικοινωνία με τους άλλους, είναι προφανές ότι αυτός ή αυτή θα πατήσει OK μετά την ανάγνωση αυτών των προειδοποιήσεων. Θα μπορούσε να βοηθήσει αν το Android θα μπορούσε να προειδοποιήσει τον χρήστη κατά τη στιγμή που η εφαρμογή ξεκινά την αποστολή μηνυμάτων κειμένου, αλλά δεν υπάρχει τέτοια λειτουργία διαθέσιμη αυτή τη στιγμή. Προς το παρόν, ένα anti-malware λογισμικό μπορεί να προσφέρει κάποια προστασία έναντι αυτών των κινδύνων για την ασφάλεια των εφαρμογών.

## **2.9.2 Βέλτιστες Πρακτικές Ασφαλείας Android Apps**

### **2.9.2.1 Αποθήκευση Δεδομένων**

Η πιο κοινή ανησυχία ασφάλειας για μια εφαρμογή για το Android είναι το κατά πόσον τα δεδομένα που αποθηκεύετε στη συσκευή είναι προσβάσιμη σε άλλες εφαρμογές. Υπάρχουν τρεις θεμελιώδεις τρόποι για να αποθηκεύσετε τα δεδομένα στη συσκευή:

#### **Χρήση εσωτερικού χώρου αποθήκευσης**

Από προεπιλογή, τα αρχεία που έχετε δημιουργήσει για τον εσωτερικό χώρο αποθήκευσης είναι προσβάσιμα μόνο με την εφαρμογή σας. Αυτή η προστασία υλοποιείται από το Android και είναι επαρκής για τις περισσότερες εφαρμογές.

Πρέπει γενικά να αποφεύγεται η χρήση του `MODE_WORLD_WRITEABLE` ή `MODE_WORLD_READABLE` τρόπων για IPC αρχεία, επειδή δεν παρέχουν τη δυνατότητα να περιορίσει η πρόσβαση δεδομένων σε

συγκεκριμένες εφαρμογές, ούτε παρέχει κάποιο έλεγχο σχετικά με τη μορφή των δεδομένων. Αν θέλετε να μοιραστείτε τα δεδομένα σας με άλλες διαδικασίες εφαρμογής, ίσως, να εξετάσει η χρήση ενός παρόχου περιεχομένου, η οποία προσφέρει δικαιώματα read και write σε άλλες εφαρμογές και μπορεί να κάνει δυναμική την χορηγηση της άδειας για κάθε περίπτωση χωριστά.

Να παρέχει επιπλέον προστασία για τα ευαίσθητα δεδομένα, μπορείτε να επιλέξετε να κρυπτογραφήσετε τα τοπικά αρχεία χρησιμοποιώντας ένα κλειδί που δεν είναι άμεσα προσβάσιμα από την εφαρμογή. Για παράδειγμα, ένα κλειδί μπορεί να τοποθετηθεί σε ένα **keystore** και προστατεύεται με κωδικό πρόσβασης χρήστη που δεν είναι αποθηκευμένα στη συσκευή. Αν και αυτό δεν προστατεύει τα δεδομένα από ένα συμβιβασμό ρίζα που μπορούν να παρακολουθούν το χρήστη εισάγοντας τον κωδικό πρόσβασης, μπορεί να παρέχει προστασία για ένα χαμένο συσκευή χωρίς κρυπτογράφηση συστήματος αρχείων .

### **Χρησιμοποιώντας Εξωτερικά μέσα αποθήκευσης**

Τα αρχεία που δημιουργούνται στο εξωτερικό μέσο αποθήκευσης , όπως SD κάρτες, είναι παγκοσμίως αναγνώσιμο και εγγράψιμο. Επειδή το μέσο εξωτερικής αποθήκευσης μπορεί να αφαιρεθεί από το χρήστη, αλλά και τροποποιούνται από οποιαδήποτε εφαρμογή, δεν θα πρέπει να αποθηκεύονται ευαίσθητες πληροφορίες χρησιμοποιώντας εξωτερικά μέσα αποθήκευσης.

Όπως και με τα δεδομένα από οποιοδήποτε μη αξιόπιστη πηγή, θα πρέπει να εκτελέσει την επικύρωση των εισροών κατά το χειρισμό των δεδομένων από εξωτερικά μέσα αποθήκευσης.

### **Χρήση Δικτύου**

Οι συναλλαγές του δικτύου είναι εγγενώς επικίνδυνες για την ασφάλεια, διότι συνεπάγεται τη διαβίβαση των δεδομένων που είναι δυνητικά ιδιωτικό στον χρήστη. Είναι πολύ σημαντικό η συσκευή να εκτελεί συναλλαγές δικτύου χρησιμοποιώντας βέλτιστες πρακτικές για την διατήρηση της ασφάλειας των δεδομένων του χρήστη ανά πάσα στιγμή.

### **Αυθεντικοποίηση**

Κατά την διαδικασία Εισαγωγής και Εγγραφής στο σύστημα πρέπει να εξασφαλίζεται το validation πεδίων όπως το email, password με κανόνες συντακτικούς για λόγους ορθότητας αλλά και για λόγους πολυπλοκότητας ειδικά σε πεδία κωδικών.

Η απαίτηση αυτή καλύπτεται με την χρήση regular expressions τόσο για το email στο οποίο πρέπει να γίνεται επαλήθευση ότι δεν περιέχει μη επιτρεπούς χαρακτήρες όσο και στο password στο οποίο μπορούμε να έχουμε ισχυρά passwords τόσο κατά την εισαγωγή όσο και κατά την εγγραφή στο σύστημα.

### **Ακεραιότητα**

Μηχανισμός hash μπορεί να εξασφαλίσει την ακεραιότητα των μηνύματων από τον αποστολέα στον παραλήπτη. Η εφαρμογή υποτροφιών δεν διαχειρίζεται sensitive data οπότε δεν κρίνεται απαραίτητη η χρήση μηχανισμού ακεραιότητας των δεδομένων που μεταφέρονται μέσω διαδικτύου και αρκεί και μόνο η κρυπτογράφηση αυτών.

### **Εμπιστευτικότητα**

Πρέπει να προτιμάται η χρήση https αντί του http η οποία μας επιτρέπει την μεταφορά κρυπτογραφημένων δεδομένων από την client εφαρμογή στο middleware. Καθώς επίσης και το πιστοποιητικό SSL που θα χρησιμοποιηθεί στο κεντρικό σύστημα πρέπει να είναι signed ώστε το site να είναι trusted.

Στην δική μας περίπτωση η εφαρμογή του Android μπορεί να υποστηρίξει κλήσεις http αλλά και https. Επίσης η κεντρική εφαρμογή με απλές ρυθμίσεις στον Tomcat μπορεί να παρέχει τα services και σε https. Το μόνο πρόβλημα που θα είχαμε είναι ότι το πιστοποιητικό μας δεν θα ήταν αξιόπιστο (trusted) καθώς αυτή η διαδικασία απαιτεί την πιστοποίηση από κάποιο φορέα όπως η Verisign επί πληρωμή.

### **Stateless Connections**

Οι κλήσεις που πραγματοποιούνται μέσω http ή https είναι state less κατά συνέπεια κρίνεται σκόπιμη η δημιουργία ενός μοναδικού κωδικού (session id) ο οποίος θα δημιουργείται κατά την είσοδο στο σύστημα από το middleware και θα χρησιμοποιείται από την client εφαρμογή σε κάθε επόμενο request προς το middleware.

# Κεφάλαιο 3

## Μεθοδολογία

### 3.1 Ερευνητικά Ερωτήματα

Ο σκοπός της παρούσας διατριβής είναι να απαντήσει τα ερευνητικά ερωτήματα «του κατά πόσο μια εφαρμογή έξυπνων κινητών συσκευών με τα κατάλληλα κριτήρια μπορεί να συνεισφέρει διευκολύνοντας την εύρεση διαθέσιμων υποτροφιών σε Ελλάδα και Κύπρο»

Και

«αν μπορεί να υπάρξει ένα μηχανισμός συστάσεων υποτροφιών σε χρήστες της εφαρμογής με παρόμοια ενδιαφέροντα»

Άρα σε πρώτη φάση θα περιγράψουμε τα κριτήρια που επιλέχθηκαν για τις αναζητήσεις καθώς και τα υπόλοιπα στοιχεία μιας υποτροφίας στο σύστημα.

### 3.2 Επιλογή κριτηρίων αναζήτησης

Όπως είδαμε στο Κεφάλαιο 1, δεν υπάρχουν κοινά και προφανή κριτήρια που μπορούν να χρησιμοποιηθούν για την αναζήτηση υποτροφιών. Σκεφτόμενοι ως υποψήφιοι φοιτητές και προσπαθώντας να βρούμε κοινά στοιχεία κατά την αναζήτηση, έγινε μια ομαδοποίηση των διαθέσιμων υποτροφιών στις παρακάτω κατηγορίες.

#### 3.2.1 Επίπεδο Υποτροφίας

Στην κατηγορία αυτή διακρίνονται τα επίπεδα σπουδών στα οποία παρέχεται η κάθε υποτροφία, και είναι τα παρακάτω

- Δευτεροβάθμιας Εκπαίδευσης
- Προπτυχιακή
- Μεταπτυχιακή
- Διδακτορική
- Μετά Διδακτορική

### **3.2.2 Γλώσσες Υποτροφίας**

Με δεδομένο ότι η εφαρμογή στοχεύει να εξυπηρετήσει υποψήφιους από την Ελλάδα και Κύπρο, οι υποστηριζόμενες γλώσσες είναι τα Ελληνικά και τα Αγγλικά.

### **3.2.3 Φορέας Υποτροφίας**

Αφορά τον φορέα που παρέχει την υποτροφία, είδαμε ότι οι υποτροφίες σε Κύπρο και Ελλάδα παρέχονται είτε από Πανεπιστήμια και Κολλέγια άμεσα, είτε από Ιδιωτικούς φορείς (Κληροδοτήματα, Κοινωφελή Ιδρύματα κ.α.) στα οποία τις περισσότερες φορές δεν καθορίζεται το Εκπαιδευτικό Ίδρυμα στο οποίο θα παρέχεται η υποτροφία αλλά οι κλάδοι σπουδών για τους οποίους παρέχονται φυσικά συναντάμε και το αντίστροφο.

### **3.2.4 Χώρα Υποτροφίας**

Η χώρα για την οποία παρέχεται η υποτροφία, η οποία διακρίνεται σε Ελλάδα, Κύπρο και Εξωτερικό. Βρέθηκαν υποτροφίες οι οποίες παρέχονται για χώρες του εξωτερικού όπως Ηνωμένες Πολιτείες, Ηνωμένο Βασίλειο κ.α.

### **3.2.5 Ενεργές Υποτροφίες**

Με την επιλογή ενός κριτηρίου ενεργής υποτροφίας θα μπορούσαν να εξαιρεθούν οι υποτροφίες για τις οποίες έχει παρέλθει η ημερομηνία των αιτήσεων από τον φορέα που τις παρέχει.

## **3.3 Άλλα στοιχεία υποτροφιών**

Στην κατηγορία αυτή έπειτα από αναζήτηση σε διάφορες ιστοσελίδες του διαδικτύου που προσφέρουν υποτροφίες, καταλήξαμε σε μια σειρά από ιδιότητες που πρέπει να έχει η οντότητα «Υποτροφίες» ώστε να μπορεί ο κάθε υποψήφιος να βρει αυτό που τον ενδιαφέρει, στη συνέχεια γίνεται μια περιγραφή των ιδιοτήτων αυτών.

### **3.3.1 Ποσό Παρεχόμενης Υποτροφίας**

Βρέθηκαν ιστοσελίδες οι οποίες είχαν σαφή αναφορά στο ποσό της υποτροφίας και στη συχνότητα καταβολής αυτού του ποσού, βρέθηκαν όμως και ιστοσελίδες για τις οποίες δεν ήταν διακριτό το ποσό καθώς και η συχνότητα. Κρίθηκε σκόπιμο ότι η οντότητα «Υποτροφία» πρέπει να έχει το ποσό καθώς και τη συχνότητα καταβολής του στον δικαιούχο γιατί αυτό το στοιχείο είναι αρκετά σημαντικό για κάποιον υποψήφιο στην προσπάθεια αναζήτησης χρηματοδότησης του. Για το λόγο αυτό στην κλάση και κατά

συνέπεια στον Πίνακα της Βάσης Δεδομένων υπάρχουν τέσσερις ιδιότητες – κολώνες αντίστοιχα για την αποθήκευση του στοιχείου αυτού.

- FinanceRateFreq – αφορά την συχνότητα καταβολής της χρηματοδότησης ενός ποσού και μπορεί να έχει τιμές όπως Μηνιαία, Εξάμηνη, Ετήσια και Εφάπαξ.
- Amount – το ποσό αυτό καθ' αυτό
- CurrencyISO – το νόμισμα κατά ISO του ποσού (EUR, USD, κ.α.)
- DurationInMonths – δείχνει την διάρκεια της χρηματοδότησης σε μήνες

### **3.3.2 Ακαδημαϊκό Έτος**

Στην ιδιότητα αυτή καταχωρείται σε μορφή String το ακαδημαϊκό έτος έναρξης της κάθε υποτροφίας, παράδειγμα για το τρέχον έτος «2015-2016».

### **3.3.3 Σύντομη Περιγραφή Υποτροφίας**

Στην ιδιότητα αυτή «ShortDescription» καταχωρείται μια συνοπτική περιγραφή της κάθε υποτροφίας. Συνήθως είναι και η επικεφαλίδα που εντοπίζεται στις διάφορες ιστοσελίδες υποτροφιών.

### **3.3.4 Ιδιότητες Υποτροφίας τύπου «Κειμένου»**

Στην κατηγορία αυτή και στις ιδιότητες που φαίνονται παρακάτω επιλέχθηκε σε ιδιότητες τύπου κειμένου «Text» να καταχωρούνται τα πλήρη στοιχεία μιας υποτροφίας όπως ακριβώς την συναντάμε.

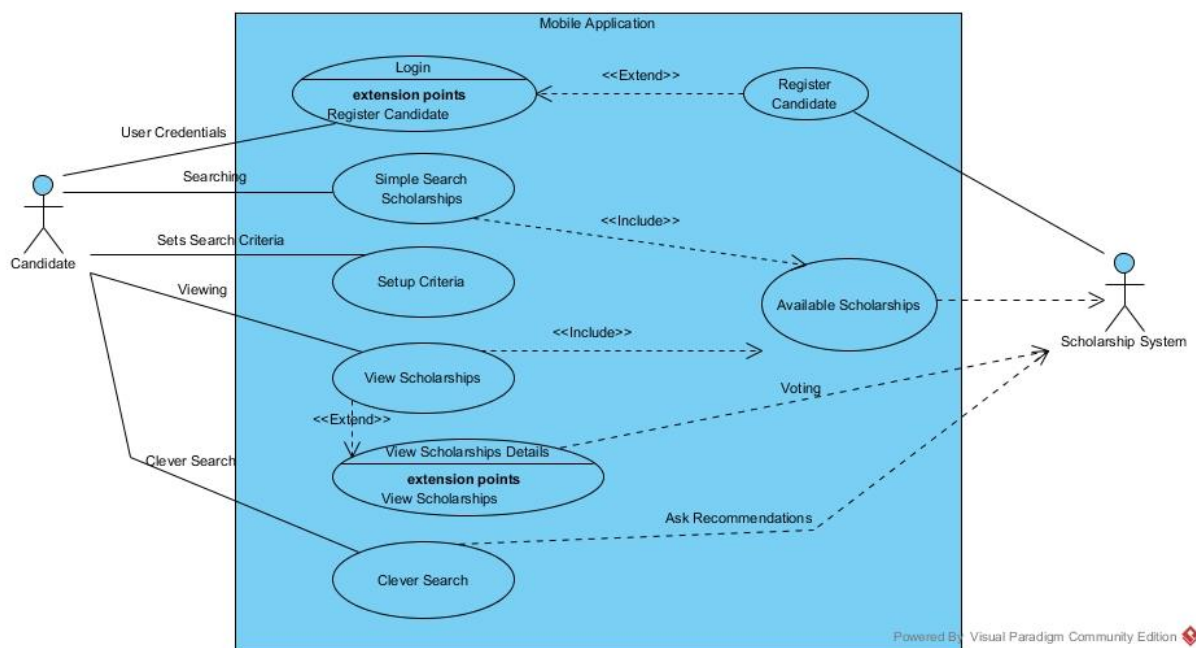
- Η πρώτη ιδιότητα μακροσκελής περιγραφή - «LongDescription», αφορά το πλήρες κείμενο από την ανακοίνωση μιας υποτροφίας και είναι υποχρεωτικό πεδίο.
- Η δεύτερη ιδιότητα προαπαιτούμενα - «Prerequisites», αφορά τα προαπαιτούμενα μιας υποτροφίας. Υπάρχουν για παράδειγμα περιπτώσεις υποτροφιών που απαιτούν συγκεκριμένο τόπο καταγωγής ή και συγκεκριμένη βαθμολογία για τον κάθε υποψήφιο. Λόγω της μεγάλης ανομοιομορφίας που παρουσιάζεται στις διάφορες αναγγελίες υποτροφιών θεωρήθηκε ο πιο ασφαλής τρόπος ώστε να μην υπάρχει απώλεια στοιχείων να αποθηκεύονται σε πλήρες κείμενο τα προαπαιτούμενα.
- Η Τρίτη ιδιότητα Αποδεικτικά έγγραφα - «Documentation», αφορά στα διάφορα έγγραφα που πρέπει να προσκομιστούν από τον υποψήφιο προκειμένου να

υποβάλει την αίτησή του. Και εδώ ο μόνος τρόπος αποθήκευσης ώστε να μην απωλέσθαι κάποια σημαντική πληροφορία είναι το κείμενο.

### 3.4 UML – Unified Modeling Language Διαγράμματα

Όλα τα διαγράμματα UML που ακολουθούν, έχουν παραχθεί από το **Visual Paradigm Community Edition** το οποίο σε αυτή την έκδοση παρέχεται χωρίς να απαιτείται κάποια άδεια χρήσης.

#### 3.4.1 Περιπτώσεις Χρήσης – Use Cases



Σχήμα 4. Διάγραμμα περιπτώσεων χρήσης

##### 3.4.1.1 Περίπτωση χρήσης «Σύνδεση»

**Όνομα ΠΧ :** Σύνδεση στο σύστημα Υποτροφιών

**Κύριος χειριστής :** Φοιτητής

**Σύντομη περιγραφή :** Ο χρήστης πραγματοποιεί Σύνδεση στο σύστημα.

**Εμπλεκόμενοι & ενδιαφέροντα :** Εφαρμογή κινητής συσκευής (Mobile App), Σύστημα υποτροφιών

**Προϋποθέσεις (preconditions) :** Ο χρήστης δεν έχει ήδη κάνει σύνδεση

**Μετασυνθήκες (postconditions) :** Ο χρήστης συνδέεται στο σύστημα

**Βασική ροή :**

1. Ο «Χρήστης» επιλέγει από το αρχικό μενού της εφαρμογής την επιλογή «Σύνδεση»
2. Η «Εφαρμογή» εμφανίζει την δραστηριότητα (activity) στην οποία ο χρήστης πρέπει να καταχωρήσει το email και τον κωδικό ασφαλείας.
3. Ο «Χρήστης» πληκτρολογεί το email και τον κωδικό και πατά το πλήκτρο Σύνδεση.
4. Η «Εφαρμογή» κρυπτογραφεί τον κωδικό ασφαλείας και καλεί το Rest Service “doLogin” του κεντρικού συστήματος Υποτροφιών.
5. Το «Κεντρικό σύστημα» λαμβάνει τα στοιχεία και ελέγχει την ύπαρξη στην βάση δεδομένων του χρήστη εφόσον τον βρει και υπάρχει ταύτιση και του κωδικού ασφαλείας επιστρέφει στην Εφαρμογή της κινητής συσκευής ένα μοναδικό UUID και ότι η διαδικασία σύνδεσης έγινε με επιτυχία.
6. Η «Εφαρμογή» λαμβάνει το μήνυμα του κεντρικού συστήματος και εμφανίζει στον χρήστη κατάλληλο μήνυμα.

#### **Εναλλακτικές ροές :**

- 5α. Το κεντρικό σύστημα δεν βρίσκει τον χρήστη στην Βάση Δεδομένων ή ο κωδικός ασφαλείας δεν ταυτίζεται.
- 5α1. Το «Κεντρικό Σύστημα» στέλνει το κατάλληλο μήνυμα λάθους στην εφαρμογή του κινητού.

#### **3.4.1.2 Περίπτωση χρήσης «Εγγραφή»**

**Όνομα ΠΧ :** Εγγραφή νέου χρήστη στο σύστημα Υποτροφιών

**Κύριος χειριστής :** Φοιτητής

**Σύντομη περιγραφή :** Ο χρήστης πραγματοποιεί Εγγραφή στο σύστημα.

**Εμπλεκόμενοι & ενδιαφέροντα :** Εφαρμογή κινητής συσκευής (Mobile App), Σύστημα υποτροφιών

**Προϋποθέσεις (preconditions) :** Ο χρήστης δεν έχει ήδη κάνει σύνδεση

**Μετασυνθήκες (postconditions) :** Ο χρήστης συνδέεται στο σύστημα

#### **Βασική ροή :**

1. Ο «Χρήστης» επιλέγει από το αρχικό μενού της εφαρμογής την επιλογή «Εγγραφή»



2. Η «Εφαρμογή» εμφανίζει την δραστηριότητα (activity) στην οποία ο χρήστης πρέπει να καταχωρήσει το email, τον κωδικό ασφαλείας, επώνυμο και όνομα.
3. Ο «Χρήστης» πληκτρολογεί το email τον κωδικό, επώνυμο και όνομα και πατά το πλήκτρο Εγγραφή.
4. Η «Εφαρμογή» κρυπτογραφεί τον κωδικό ασφαλείας και καλεί το Rest Service «doRegister» του κεντρικού συστήματος Υποτροφιών.
5. Το «Κεντρικό σύστημα» λαμβάνει τα στοιχεία και καταχωρεί τον νέο χρήστη στην βάση δεδομένων, και επιστρέφει στην Εφαρμογή της κινητής συσκευής ένα μοναδικό UUID και ότι η διαδικασία σύνδεσης έγινε με επιτυχία.
6. Η «Εφαρμογή» λαμβάνει το μήνυμα του κεντρικού συστήματος και εμφανίζει στον χρήστη κατάλληλο μήνυμα.

#### **Εναλλακτικές ροές :**

5α. Το κεντρικό σύστημα βρίσκει τον χρήστη στην Βάση Δεδομένων ήδη καταχωρημένο.

5α1. Το «Κεντρικό Σύστημα» στέλνει το κατάλληλο μήνυμα λάθους στην εφαρμογή του κινητού.

### **3.4.1.3 Περίπτωση χρήσης «Κριτήρια Αναζήτησης»**

**Όνομα ΠΧ :** Εφαρμογή κριτηρίων αναζήτησης

**Κύριος χειριστής :** Φοιτητής - «Χρήστης»

**Σύντομη περιγραφή :** Ο χρήστης επιλέγει από τα βασικά κριτήρια αναζήτησης.

**Εμπλεκόμενοι & ενδιαφέροντα :** Εφαρμογή κινητής συσκευής (Mobile App)

**Προϋποθέσεις (preconditions) :** Ο χρήστης έχει ήδη κάνει σύνδεση

**Μετασυνθήκες (postconditions) :** Καμία

#### **Βασική ροή :**

1. Ο «Χρήστης» επιλέγει από το αρχικό μενού της εφαρμογής την επιλογή «Κριτήρια Αναζήτησης»
2. Η «Εφαρμογή» εμφανίζει την δραστηριότητα (activity) στην οποία ο χρήστης πρέπει να επιλέξει τα κριτήρια αναζήτησης, τύπο εκπαιδευτικού ιδρύματος, Γλώσσα προγράμματος σπουδών, Επίπεδο σπουδών, Χώρα ιδρύματος υποτροφίας, που θέλει να εφαρμοστούν.
3. Ο «Χρήστης» επιλέξει από τις εμφανιζόμενες κατηγορίες τα φίλτρα που επιθυμεί.

4. Η «Εφαρμογή» δημιουργεί μια κλάση ανάλογα με τα κριτήρια που επέλεξε ο χρήστης και την αποθηκεύει στην μνήμη της συσκευής.

**Εναλλακτικές ροές :**

Καμία

**3.4.1.4 Περίπτωση χρήσης «Απλής Αναζήτησης»**

**Όνομα ΠΧ :** Απλή Αναζήτηση Υποτροφιών

**Κύριος χειριστής :** Φοιτητής -«Χρήστης»

**Σύντομη περιγραφή :** Ο χρήστης εισάγει λέξεις κλειδιά και στη συνέχεια πατάει το πλήκτρο αναζήτησης.

**Εμπλεκόμενοι & ενδιαφέροντα :** Εφαρμογή κινητής συσκευής (Mobile App), Σύστημα υποτροφιών

**Προϋποθέσεις (preconditions) :** Εφαρμογή κριτηρίων αναζήτησης (προαιρετικά), Ο χρήστης έχει ήδη κάνει σύνδεση.

**Μετασυνθήκες (postconditions) :** Καμία

**Βασική ροή :**

1. Ο «Χρήστης» επιλέγει από το «Μενού Επιλογών», την επιλογή «Απλή Αναζήτηση».
2. Η «Εφαρμογή» εμφανίζει την δραστηριότητα (activity) στην οποία ο χρήστης μπορεί αν επιθυμεί να καταχωρήσει λέξεις κλειδιά για αναζήτηση.
3. Ο «Χρήστης» εισάγει λέξεις κλειδιά στο πεδίο «Αναζήτηση»
4. Ο «Χρήστης» πατάει το κουμπί «Αναζήτηση» (φακός εύρεσης)
5. Η «Mobile App» επικοινωνεί με το «Σύστημα υποτροφιών» το οποίο πραγματοποιεί αναζήτηση στην Βάση Δεδομένων εφαρμόζοντας όλα τα κριτήρια και τις λέξεις κλειδιά που έχει εισάγει ο χειριστής.
6. Το «Σύστημα υποτροφιών» δημιουργεί τη λίστα με τις διαθέσιμες υποτροφίες βάση των κριτηρίων και των λέξεων κλειδιών
7. Το «Σύστημα υποτροφιών» στέλνει την λίστα στην «Mobile App»
8. Η «Mobile App» εμφανίζει στην οθόνη την λίστα με τα αποτελέσματα της αναζήτησης.

**Εναλλακτικές ροές :**

- 6α. Δεν βρέθηκαν διαθέσιμες υποτροφίες με τα κριτήρια και τις λέξεις κλειδιά

6α1. Η «Mobile App» εμφανίζει στην οθόνη μήνυμα ότι δεν βρέθηκαν αποτελέσματα με τα επιλεγμένα κριτήρια και τις λέξεις κλειδιά.

### **3.4.1.5 Περίπτωση χρήσης «Έξυπνης Αναζήτησης»**

**Όνομα ΠΧ :** Έξυπνη Αναζήτηση Υποτροφιών

**Κύριος χειριστής :** Φοιτητής «Χειριστής»

**Σύντομη περιγραφή :** Ο χρήστης εισάγει λέξεις κλειδιά και στη συνέχεια πατάει το πλήκτρο αναζήτησης.

**Εμπλεκόμενοι & ενδιαφέροντα :** Εφαρμογή κινητής συσκευής (Mobile App), Σύστημα υποτροφιών

**Προϋποθέσεις (preconditions) :** Εφαρμογή κριτηρίων αναζήτησης (προαιρετικά), Ο χρήστης έχει ήδη κάνει σύνδεση.

**Μετασυνθήκες (postconditions) :** Καμία

**Βασική ροή :**

1. Ο «Χρήστης» επιλέγει από το «Μενού Επιλογών», την επιλογή «Έξυπνη Αναζήτηση».
2. Η «Εφαρμογή» εμφανίζει την δραστηριότητα (activity) στην οποία ο χρήστης μπορεί αν επιθυμεί να καταχωρήσει λέξεις κλειδιά για αναζήτηση.
3. Ο «Χρήστης» εισάγει λέξεις κλειδιά στο πεδίο «Αναζήτηση»
4. Ο «Χρήστης» πατάει το κουμπί «Αναζήτηση» (φακός εύρεσης)
5. Η «Mobile App» επικοινωνεί με το «Σύστημα υποτροφιών» το οποίο πραγματοποιεί αναζητήσεις με ίδια κριτήρια και παρόμοιες λέξεις κλειδιά. Για όσα βρει εφαρμόζει τον recommendation algorithm (παραγοντοποίησης πινάκων).
6. Το «Σύστημα υποτροφιών» δημιουργεί τη λίστα με τις διαθέσιμες υποτροφίες από τα αποτελέσματα του προηγούμενο βήματος.
7. Το «Σύστημα υποτροφιών» στέλνει την λίστα στην «Mobile App»
8. Η «Mobile App» εμφανίζει στην οθόνη την λίστα με τα αποτελέσματα της αναζήτησης.

**Εναλλακτικές ροές :**

6α. Δεν βρέθηκαν διαθέσιμες υποτροφίες με τα κριτήρια και τις λέξεις κλειδιά

6α1. Η «Mobile App» εμφανίζει στην οθόνη μήνυμα ότι δεν βρέθηκαν αποτελέσματα με τα επιλεγμένα κριτήρια και τις λέξεις κλειδιά.

### **3.4.1.6 Περίπτωση χρήσης «Λεπτομέρειες Υποτροφίας»**

**Όνομα ΠΧ :** Εμφάνιση Λεπτομερειών Υποτροφίας

**Κύριος χειριστής :** Φοιτητής

**Σύντομη περιγραφή :** Ο χρήστης βλέπει τις λεπτομέρειες μιας υποτροφίας αφού επιλέξει από μια λίστα επιλογών.

**Εμπλεκόμενοι & ενδιαφέροντα :** Εφαρμογή κινητής συσκευής (Mobile App), Σύστημα υποτροφιών

**Προϋποθέσεις (preconditions) :** Ο χρήστης έχει προηγουμένως πραγματοποιήσει «Αναζήτηση Υποτροφιών» με «Απλή Αναζήτηση» ή με «Έξυπνη Αναζήτηση»

**Μετασυνθήκες (postconditions) :** Καμία

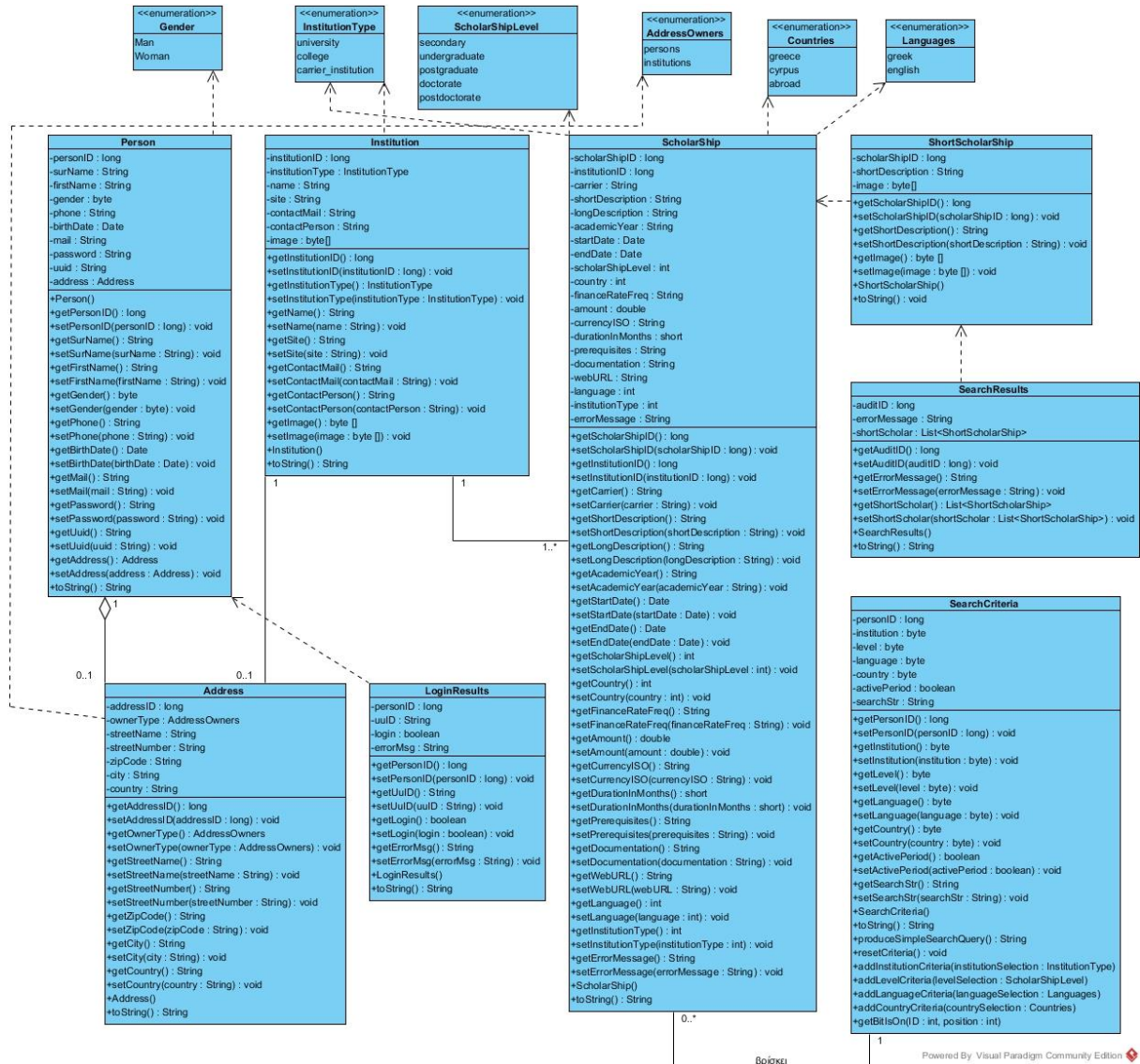
**Βασική ροή :**

1. Ο «Χρήστης» βλέπει στην οθόνη την λίστα με τις υποτροφίες, μετά από τις περιπτώσεις χρήσης «Απλής ή Έξυπνης Αναζήτησης».
2. Ο «Χρήστης» πατάει παρατεταμένα πάνω στο κείμενο της υποτροφίας
3. Η «Mobile App» ζητά από το «Σύστημα υποτροφιών» τις λεπτομέρειες της επιλεγμένης υποτροφίας
4. Το «Σύστημα υποτροφιών» φέρνει από την βάση όλα τα στοιχεία της επιλεγμένης υποτροφίας και τα στέλνει στην «Mobile App»
5. Η «Mobile App» εμφανίζει σε νέο activity τις λεπτομέρειες της υποτροφίας.

**Εναλλακτικές ροές :**

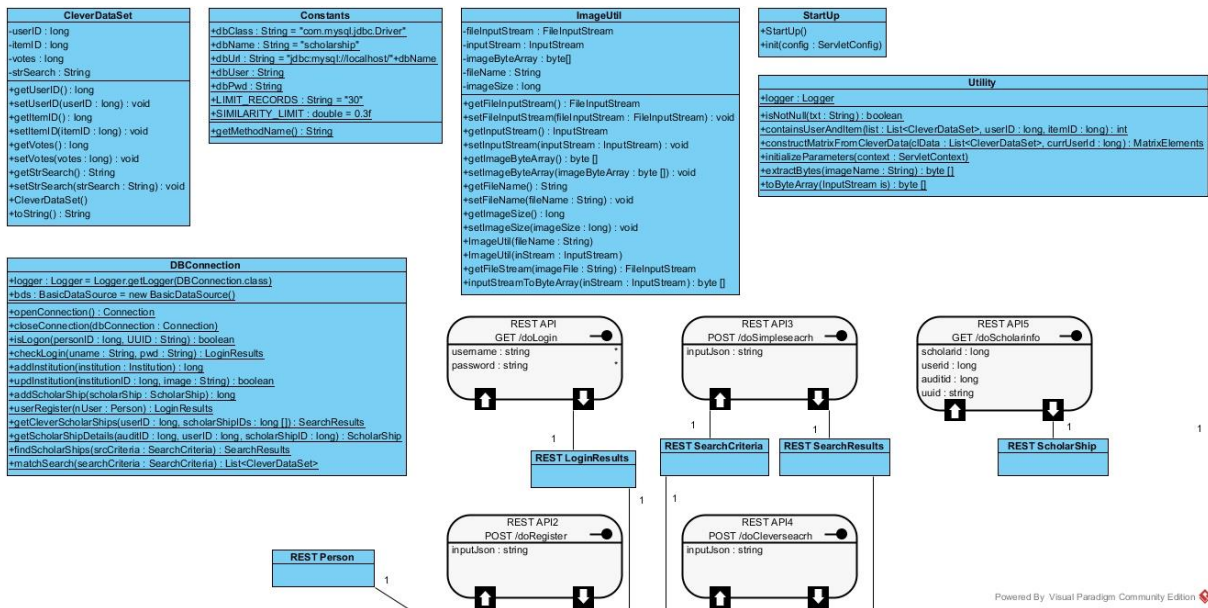
Καμία

### 3.4.2 Διάγραμμα Κλάσεων – Class Diagram

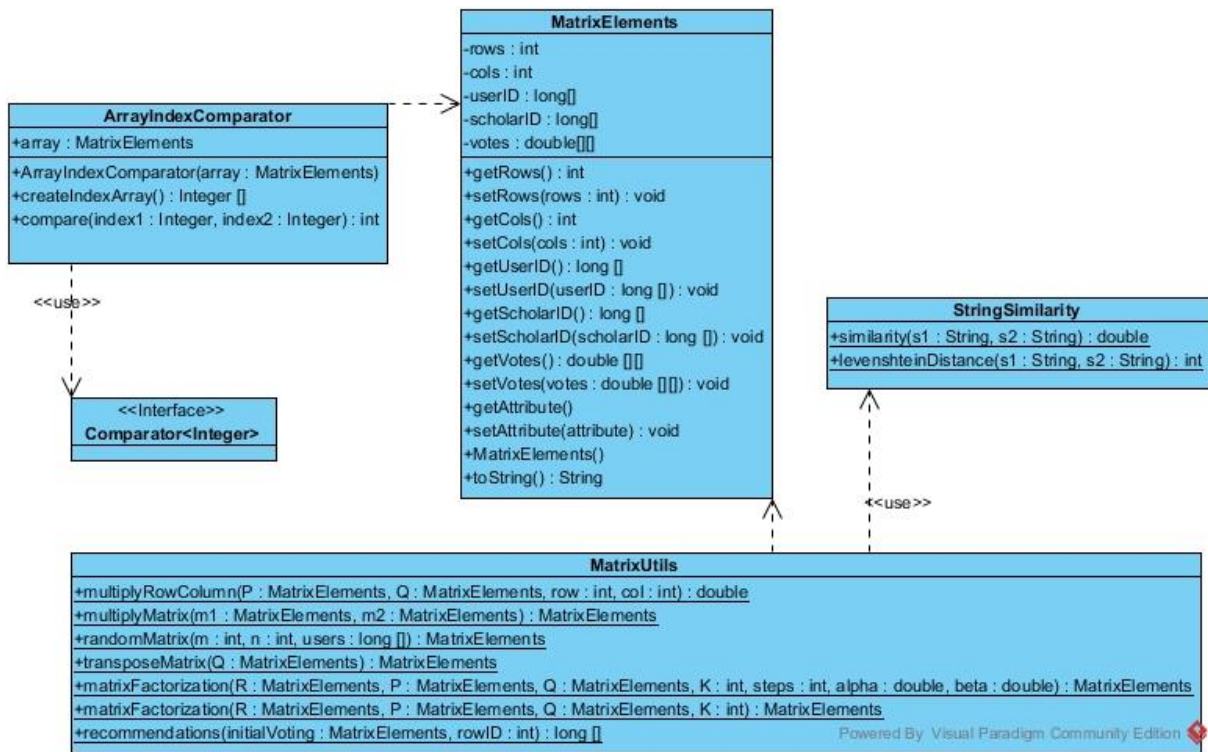


Σχήμα 5. Κύριο Διάγραμμα κλάσεων

Στο κύριο διάγραμμα κλάσεων φαίνονται όλες οι κλάσεις οι οποίες είναι κοινές για την εφαρμογή client (Mobile App) και Middleware. Για τον σκοπό αυτό όλες αυτές οι κλάσεις πρέπει να ανήκουν σε ένα κοινό java package ώστε μια αλλαγή να επηρεάζει και τις δύο εφαρμογές και η συντήρηση του κώδικα να γίνεται πιο εύκολα.

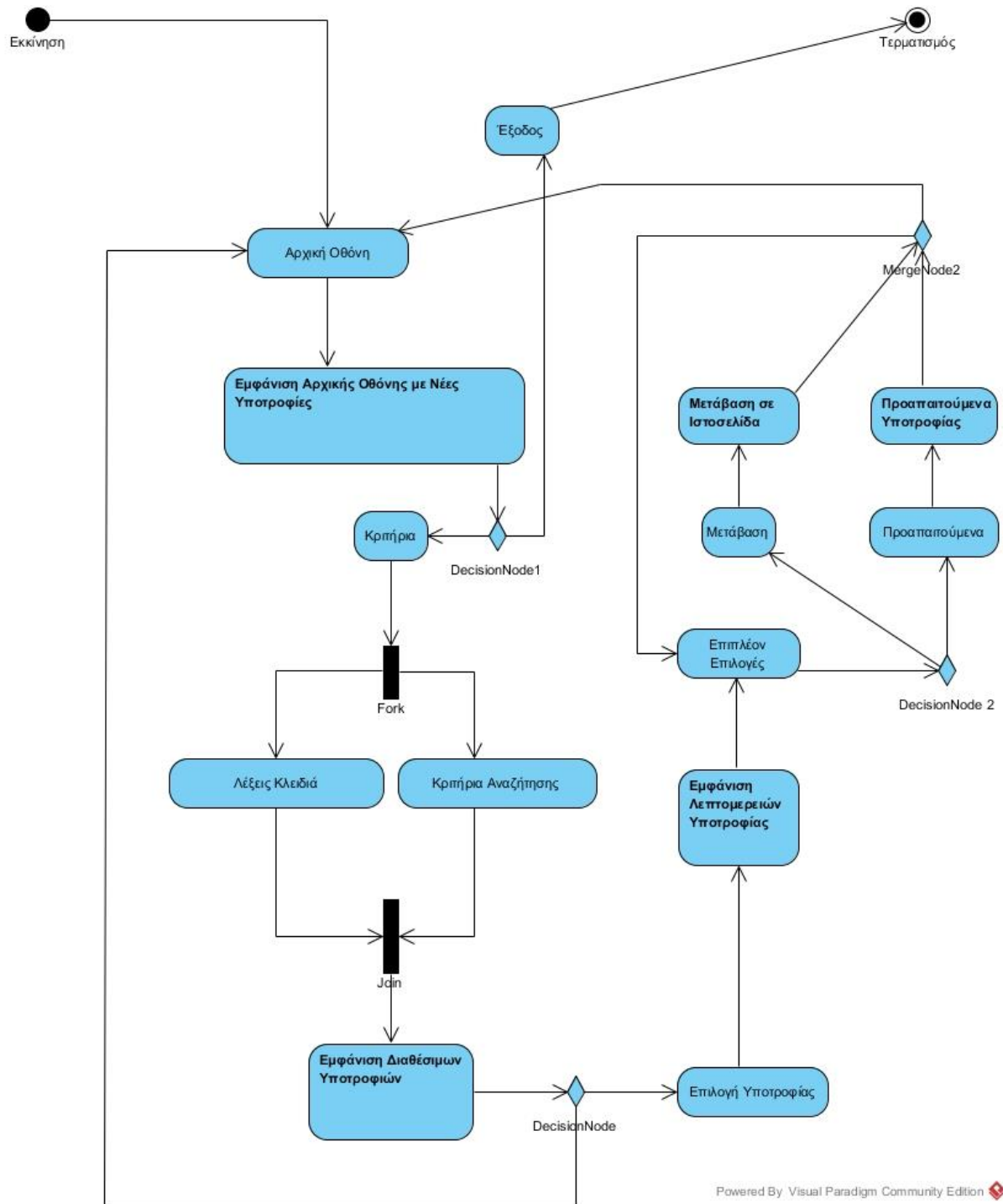


Σχήμα 6. Διάγραμμα κλάσεων Διεργασιών Middleware



Σχήμα 7. Διάγραμμα κλάσεων μηχανισμού συστάσεων

### 3.4.3 Διάγραμμα Δραστηριοτήτων

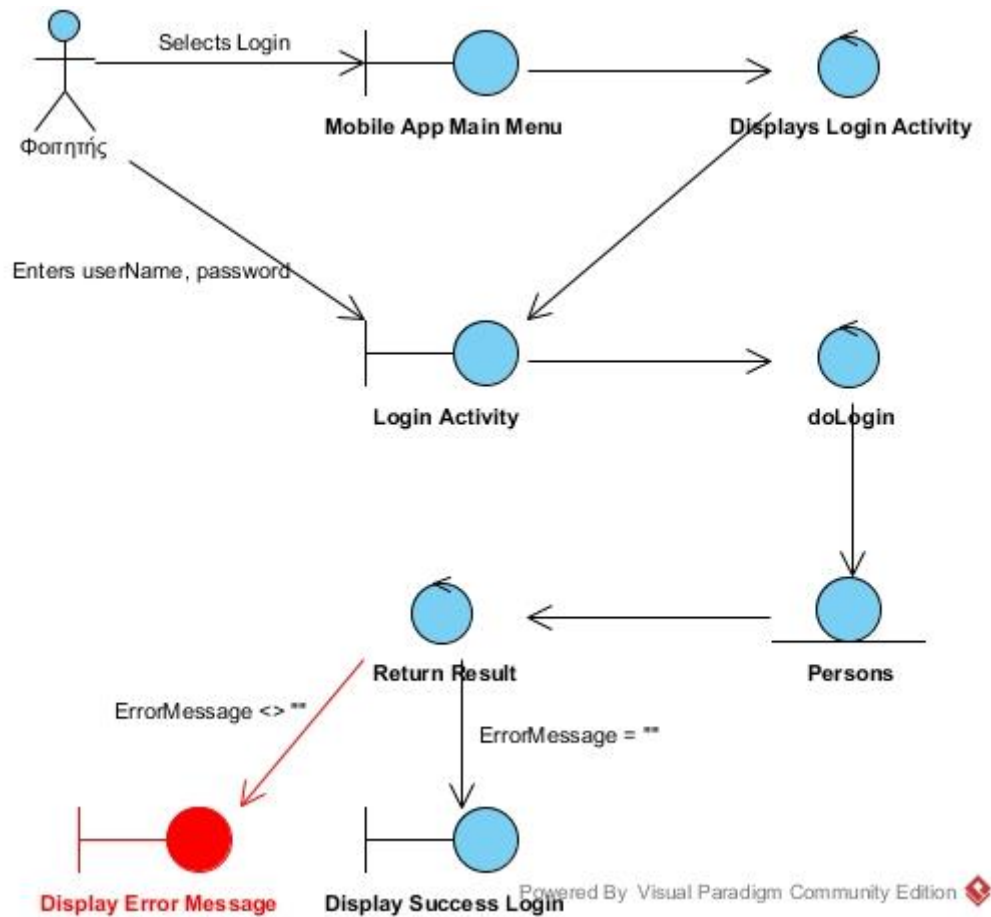


Σχήμα 8. Διάγραμμα δραστηριοτήτων

### 3.4.4 Διαγράμματα Ευρωστίας

Παρακάτω εμφανίζονται τα διαγράμματα Ευρωστίας (Robustness) για κάθε περίπτωση χρήσης όπως αυτές περιγράφηκαν στην ενότητα 3.4.1

### 3.4.4.1 Σύνδεσης χρήστη



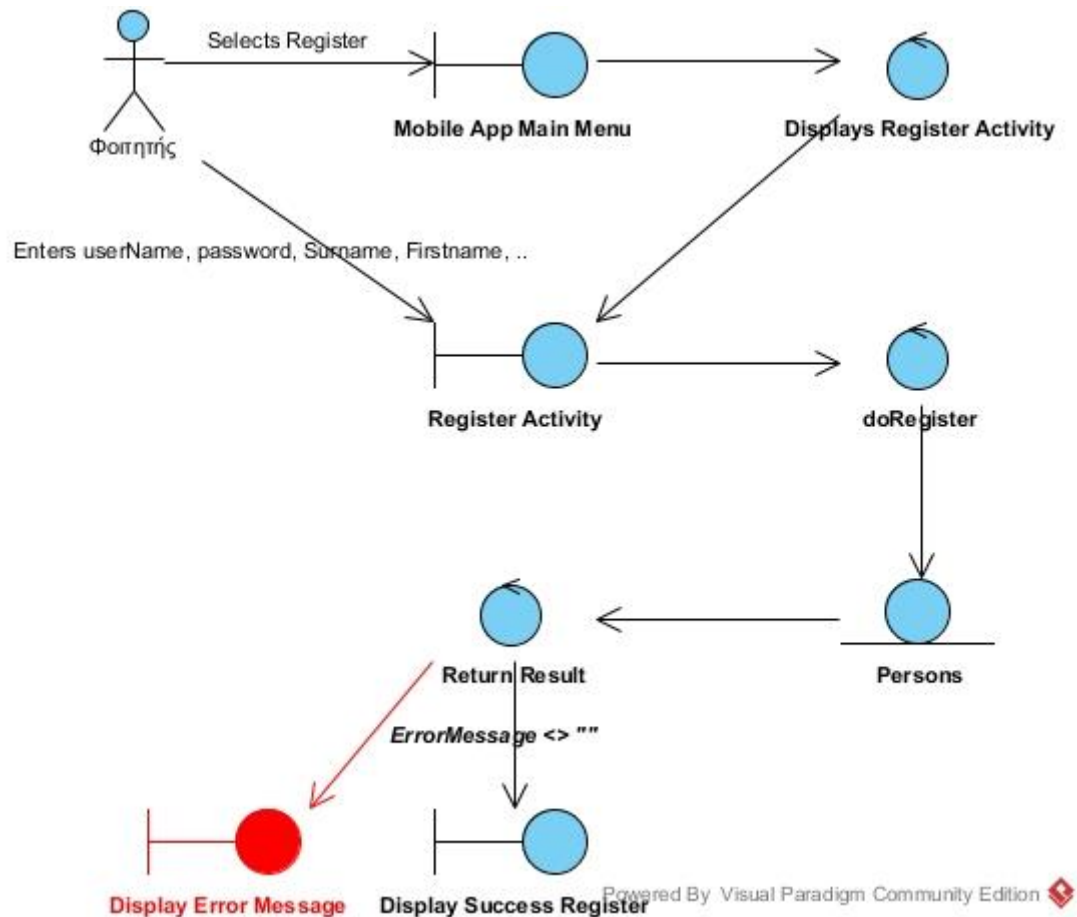
Σχήμα 9. Διάγραμμα Ευρωστίας περίπτωσης χρήσης «Σύνδεση»

#### Βήματα Εκτέλεσης

1. Ο χρήστης επιλέγει «Login» από το αρχικό μενού
2. Η εφαρμογή εμφανίζει το Login activity
3. Ο χρήστης εισάγει username και password
4. Το Login Activity καλεί την doLogin
5. Η μέθοδος doLogin ελέγχει την ύπαρξη ή όχι του χρήστη, επιστρέφοντας κατάλληλο error message
6. Αν το error message είναι κενό μήνυμα επιτυχημένου Login διαφορετικά μήνυμα αποτυχίας.



### 3.4.4.2 Εγγραφής Χρήστη

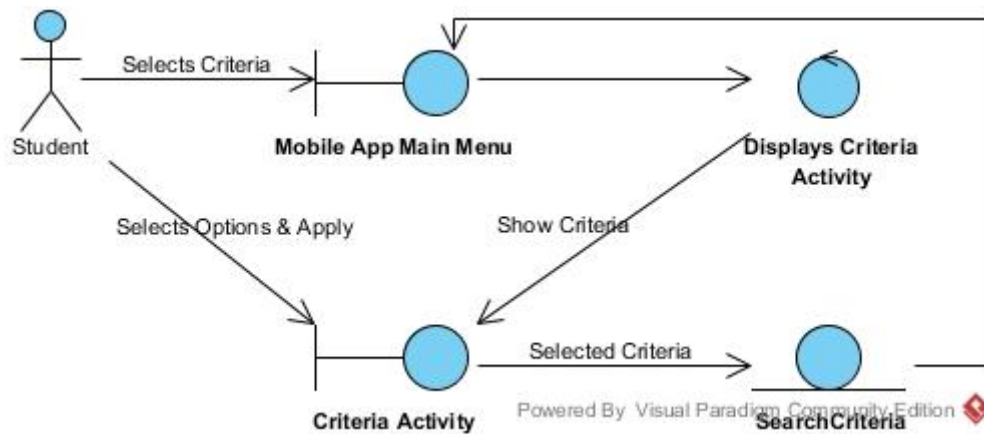


Σχήμα 10. Διάγραμμα Ευρωστίας περίπτωσης χρήσης «Εγγραφή»

#### Βήματα Εκτέλεσης

1. Ο χρήστης επιλέγει «Register» από το αρχικό μενού
2. Η εφαρμογή εμφανίζει το Register activity
3. Ο χρήστης εισάγει username, password, όνομα και επώνυμο
4. Το Register Activity καλεί την doRegister
5. Η μέθοδος doRegister ελέγχει την ύπαρξη ή όχι του χρήστη και αν δεν υπάρχει εισάγει το χρήστη επιστρέφοντας κατάλληλο error message
6. Αν το error message είναι κενό μήνυμα επιτυχημένου Register διαφορετικά μήνυμα αποτυχίας.

### 3.4.4.3 Εφαρμογής - Αλλαγής κριτηρίων

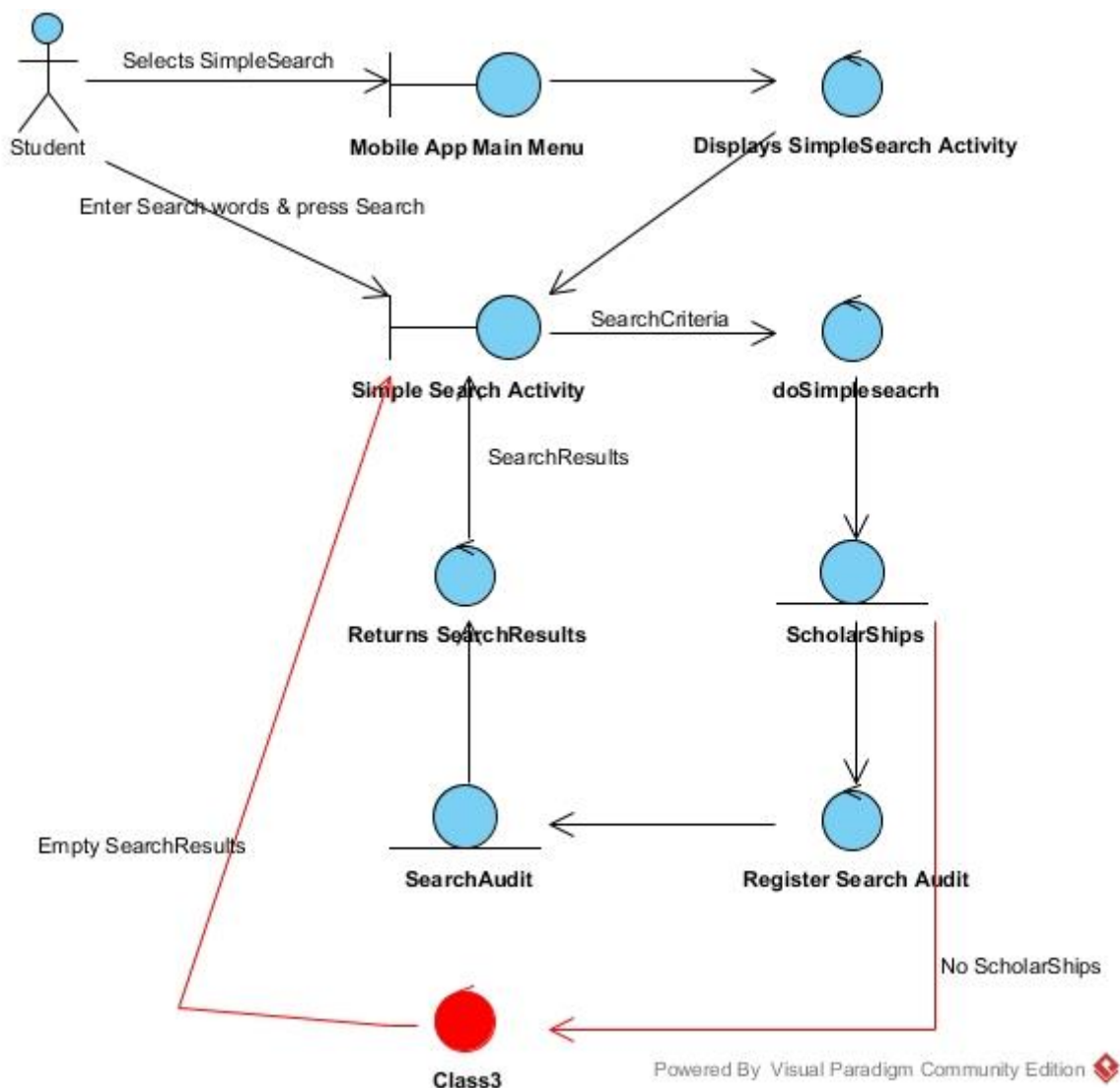


Σχήμα 11. Διάγραμμα Ευρωστίας περίπτωσης χρήσης «Κριτήρια Αναζήτησης»

#### **Βήματα Εκτέλεσης**

1. Ο χρήστης επιλέγει «Search Filters» από το αρχικό μενού
2. Η εφαρμογή εμφανίζει το Search Criteria activity
3. Ο χρήστης επιλέγει τα κριτήρια που επιθυμεί
4. Το Search Criteria activity ενημερώνει τα κριτήρια
5. Επιστρέφει στην αρχική οθόνη

### 3.4.4.4 Απλής Αναζήτησης



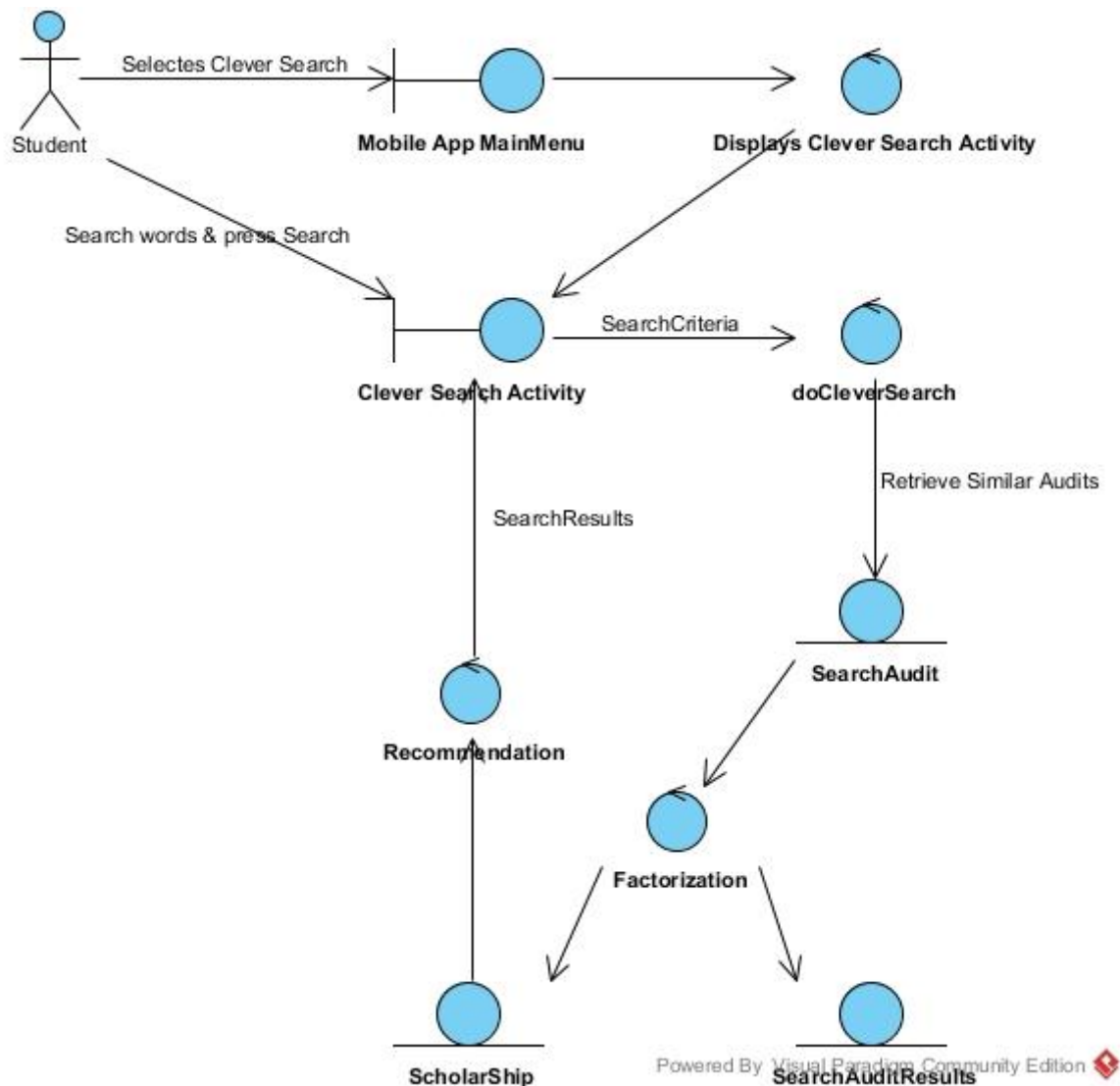
Σχήμα 12. Διάγραμμα Ευρωστίας περίπτωσης χρήσης «Απλής Αναζήτησης»

#### Βήματα Εκτέλεσης

1. Ο χρήστης επιλέγει «Simple Search» από το αρχικό μενού
2. Η εφαρμογή εμφανίζει το Simple Search activity
3. Ο χρήστης εισάγει λέξη/εις προς αναζήτηση και πατά αναζήτηση
4. Το Simple Search activity καλεί την μέθοδο doSimpleSearch με τα τρέχοντα κριτήρια αναζήτησης και την λέξη/εις αναζήτησης
5. Η doSimpleSearch ζητάει από τις υποτροφίες με αυτά τα κριτήρια και την λέξη/εις αναζήτησης
6. Αν υπάρχουν υποτροφίες καταχωρεί στον SearchAudit τα κριτήρια και την λέξη/εις αναζήτησης και τον χρήστη

7. Επιστρέφει στο Simple Search activity τα αποτελέσματα αναζήτησης

### 3.4.4.5 Έξυπνης Αναζήτησης



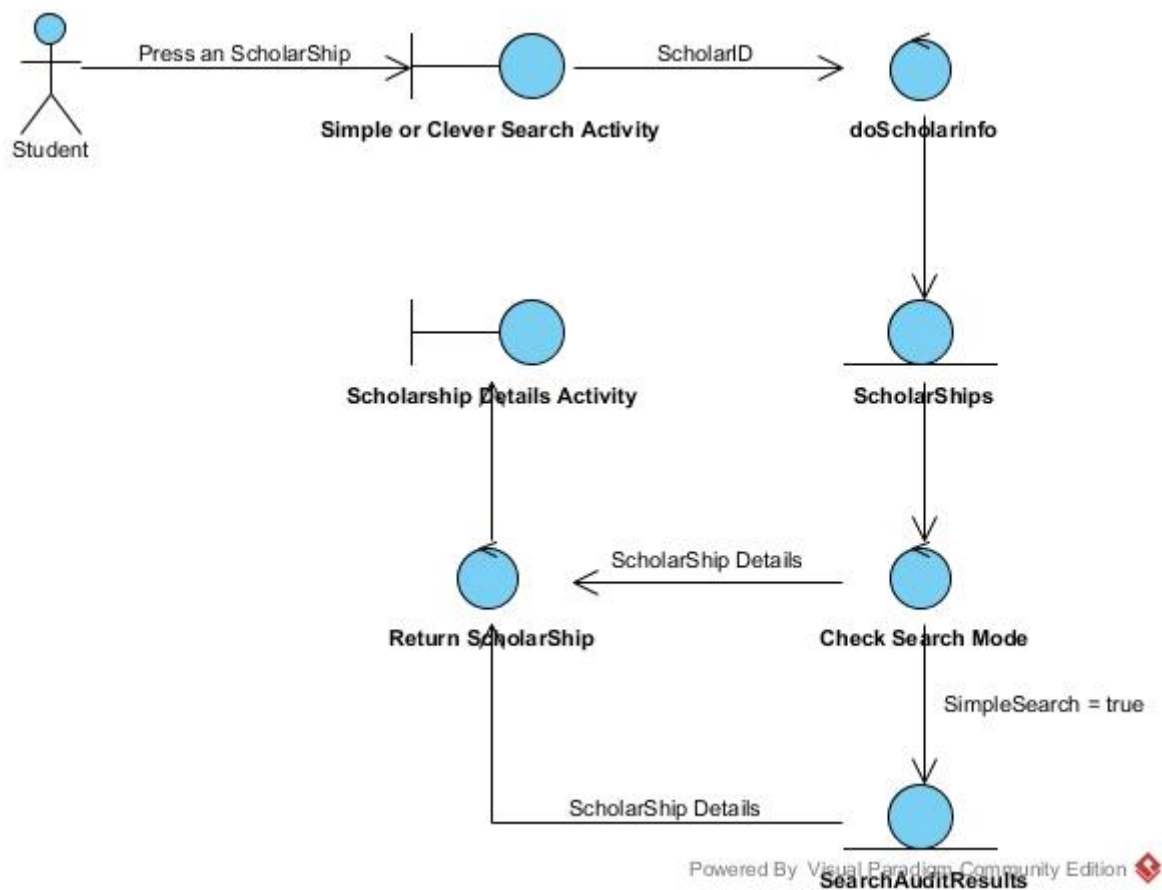
Σχήμα 13. Διάγραμμα Ευρωστίας περίπτωσης χρήσης «Έξυπνης Αναζήτησης»

#### Βήματα Εκτέλεσης

1. Ο χρήστης επιλέγει «Clever Search» από το αρχικό μενού
2. Η εφαρμογή εμφανίζει το Clever Search activity
3. Ο χρήστης εισάγει λέξη/εις προς αναζήτηση και πατά αναζήτηση
4. Το Clever Search activity καλεί την μέθοδο doCleverSearch με τα τρέχοντα κριτήρια αναζήτησης και την λέξη/εις αναζήτησης
5. Η doCleverSearch ζητά τις όμοιες αναζητήσεις από τον SearchAudit

6. Πραγματοποιεί παραγοντοποίηση πινάκων στα αποτελέσματα (εφόσον υπάρχουν)
7. Παράγει συστάσεις για τον χρήστη και αντλεί τις υποτροφίες για αυτές
8. Επιστρέφει στο Clever Search activity τα αποτελέσματα αναζήτησης

#### 3.4.4.6 Λεπτομέρειες Υποτροφίας



**Σχήμα 14.** Διάγραμμα Ευρωστίας περίπτωσης χρήσης «Λεπτομέρειες Υποτροφίας»

#### **Βήματα Εκτέλεσης**

1. Ο χρήστης επιλέγει μια υποτροφία από αυτές που εμφανίζονται στο Simple Search Activity ή το Clever Search Activity
2. Το Simple or Clever Search Activity καλεί την doScholarInfo με το ID της υποτροφίας
3. Η doScholarInfo ζητά τις λεπτομέρειες της υποτροφίας από την Βάση με αυτό το ID
4. Αν το Search Mode είναι SimpleSearch τότε ενημέρωση με μια ψήφο τον SearchAuditResults

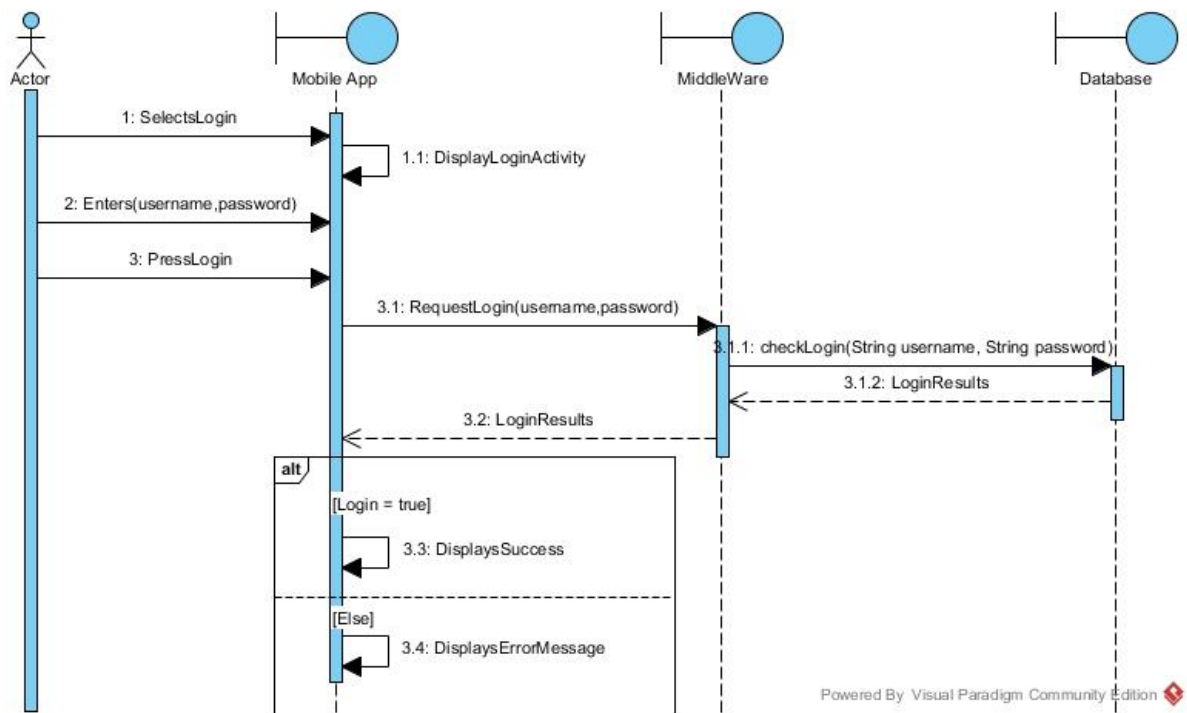
5. Επέστρεψε τις λεπτομέρειες της υποτροφίας στο call Activity

### 3.4.5 Διαγράμματα Ακολουθίας

Παρακάτω εμφανίζονται τα διαγράμματα Ακολουθίας (Sequence) για κάθε περίπτωση χρήσης όπως αυτές περιγράφηκαν στην ενότητα 3.4.1 και φαίνονται και από τα αντίστοιχα διαγράμματα ευρωστίας της ενότητας 3.4.4.

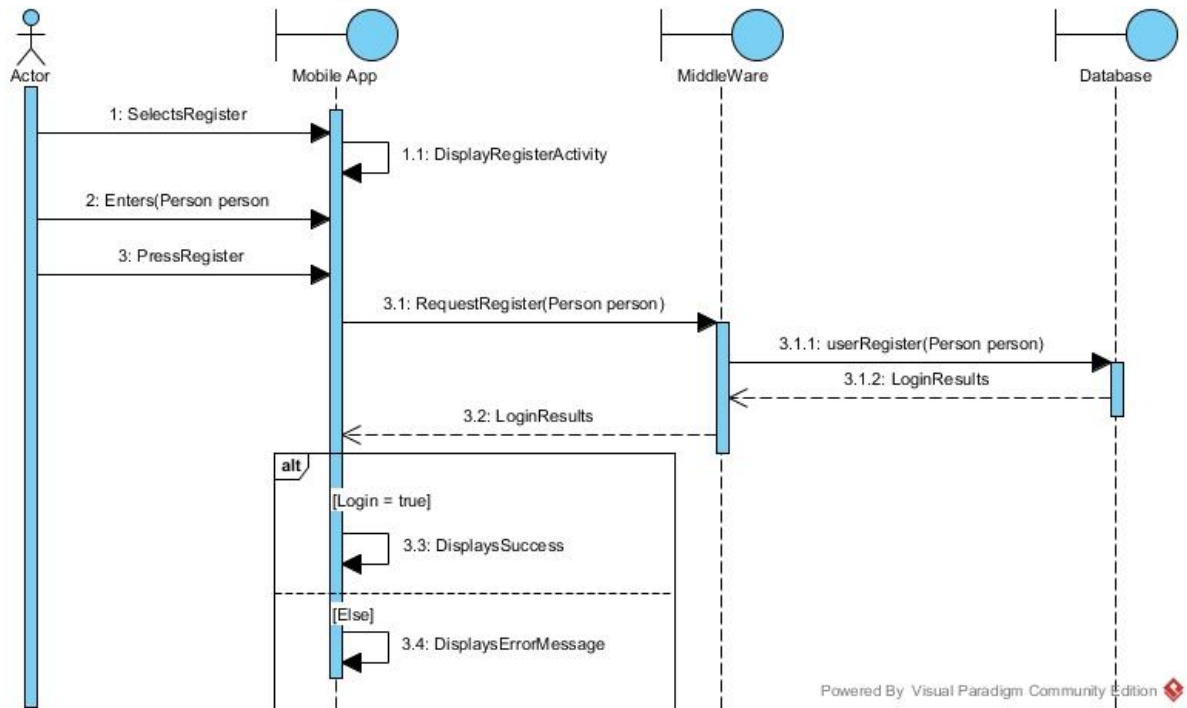
Τα διαγράμματα ακολουθίας αντικατοπτρίζουν ακριβώς τον τρόπο λειτουργίας κάθε περίπτωσης χρήσης και είναι τα κύρια διαγράμματα που θα χρησιμοποιηθούν μαζί με τα Διαγράμματα κλάσεων κατά την ανάπτυξη του συστήματος.

#### 3.4.5.1 Σύνδεσης Χρήστη



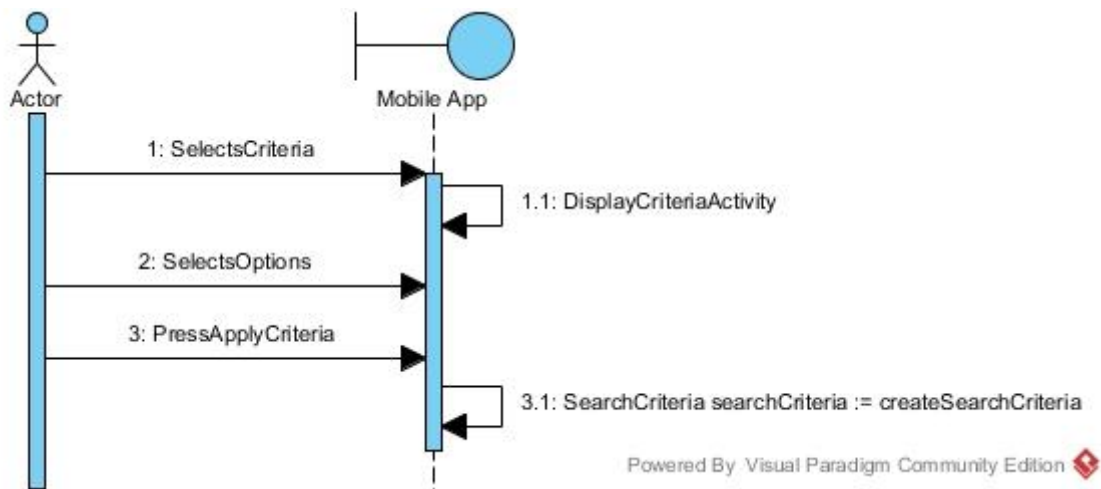
Σχήμα 15. Διάγραμμα Ακολουθίας περίπτωσης χρήσης «Σύνδεση»

### 3.4.5.2 Εγγραφής Χρήστη



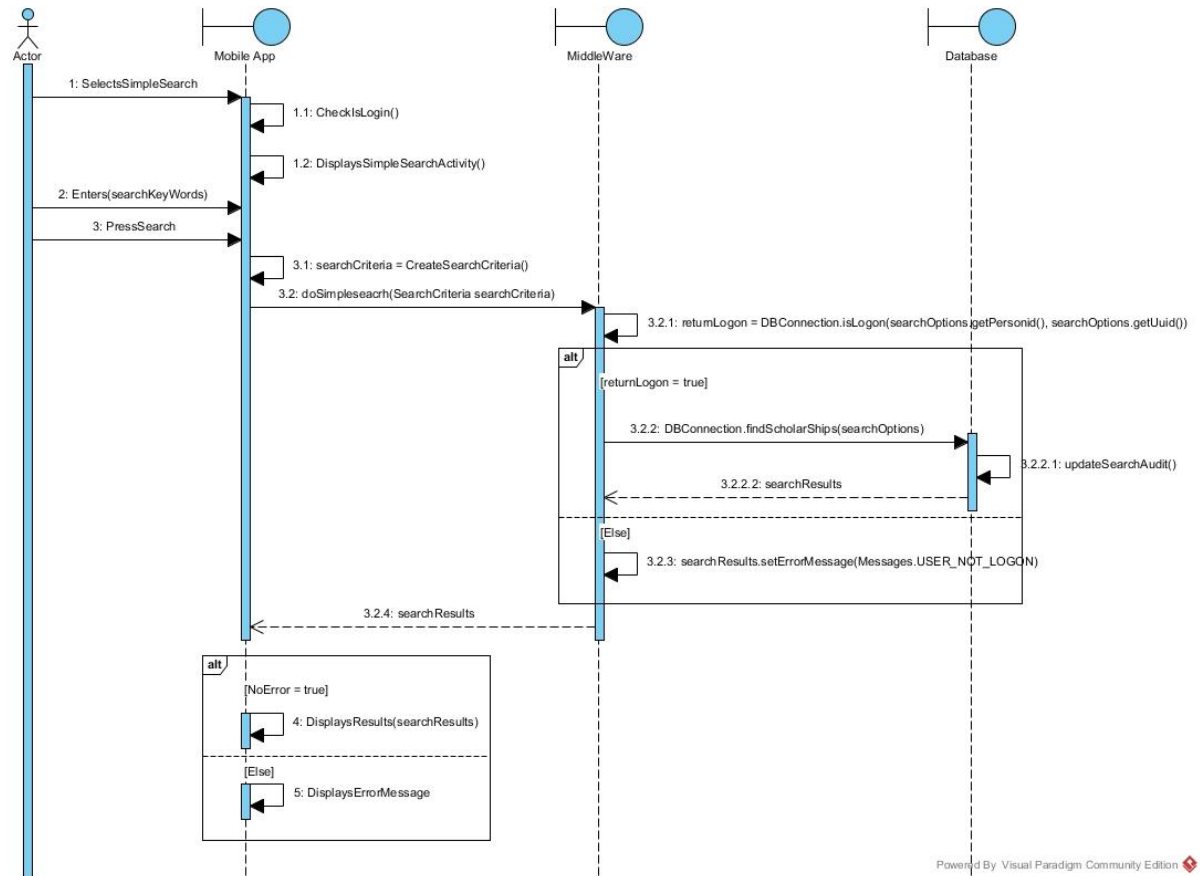
Σχήμα 16. Διάγραμμα Ακολουθίας περίπτωσης χρήσης «Εγγραφής»

### 3.4.5.3 Εφαρμογής - Αλλαγής Κριτηρίων



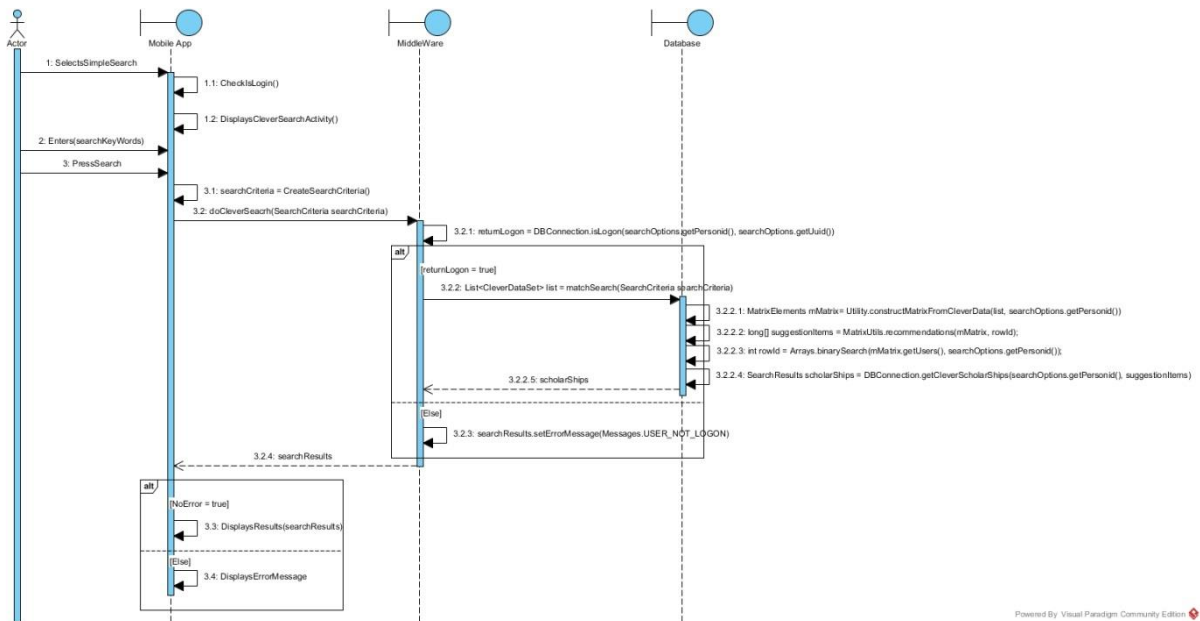
Σχήμα 17. Διάγραμμα Ακολουθίας περίπτωσης χρήσης «Κριτήρια Αναζήτησης»

### 3.4.5.4 Απλής Αναζήτησης



Σχήμα 18. Διάγραμμα Ακολουθίας περίπτωσης χρήσης «Απλής Αναζήτησης»

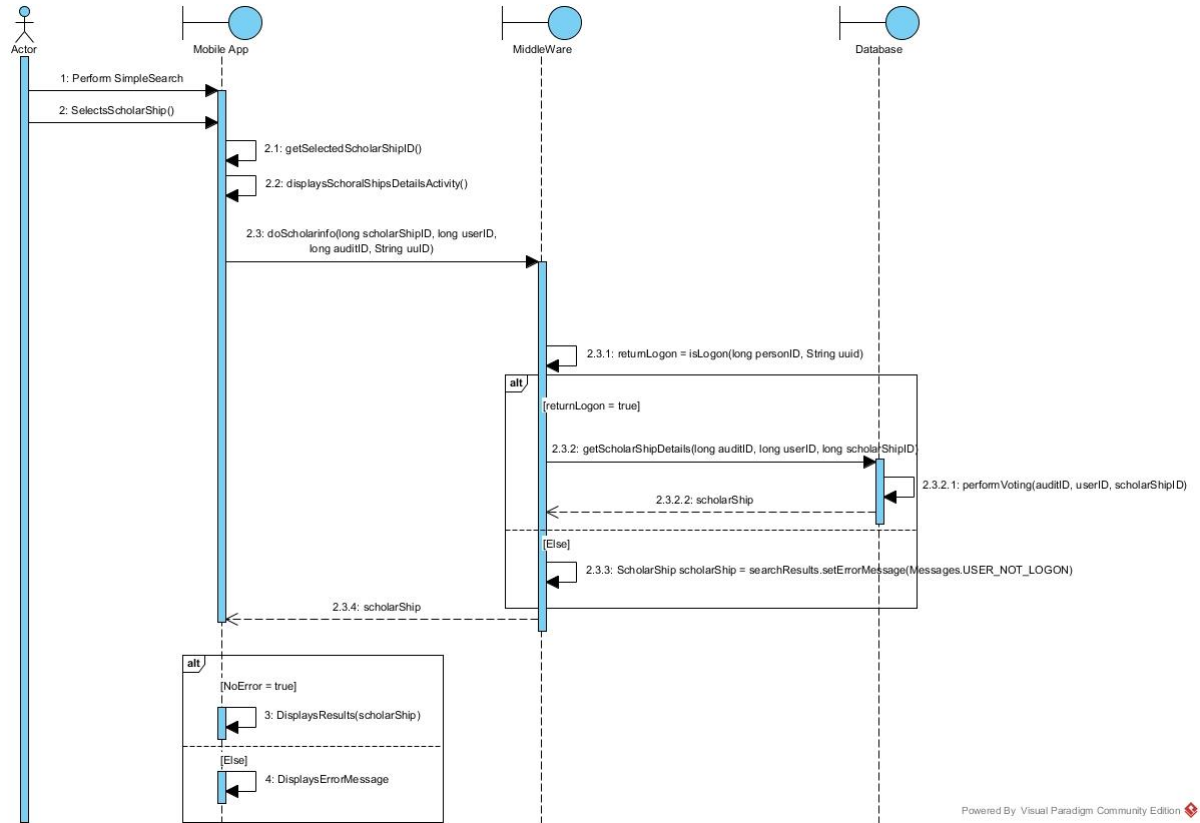
### 3.4.5.5 Έξυπνης Αναζήτησης



Σχήμα 19. Διάγραμμα Ακολουθίας περίπτωσης χρήσης «Έξυπνης Αναζήτησης»

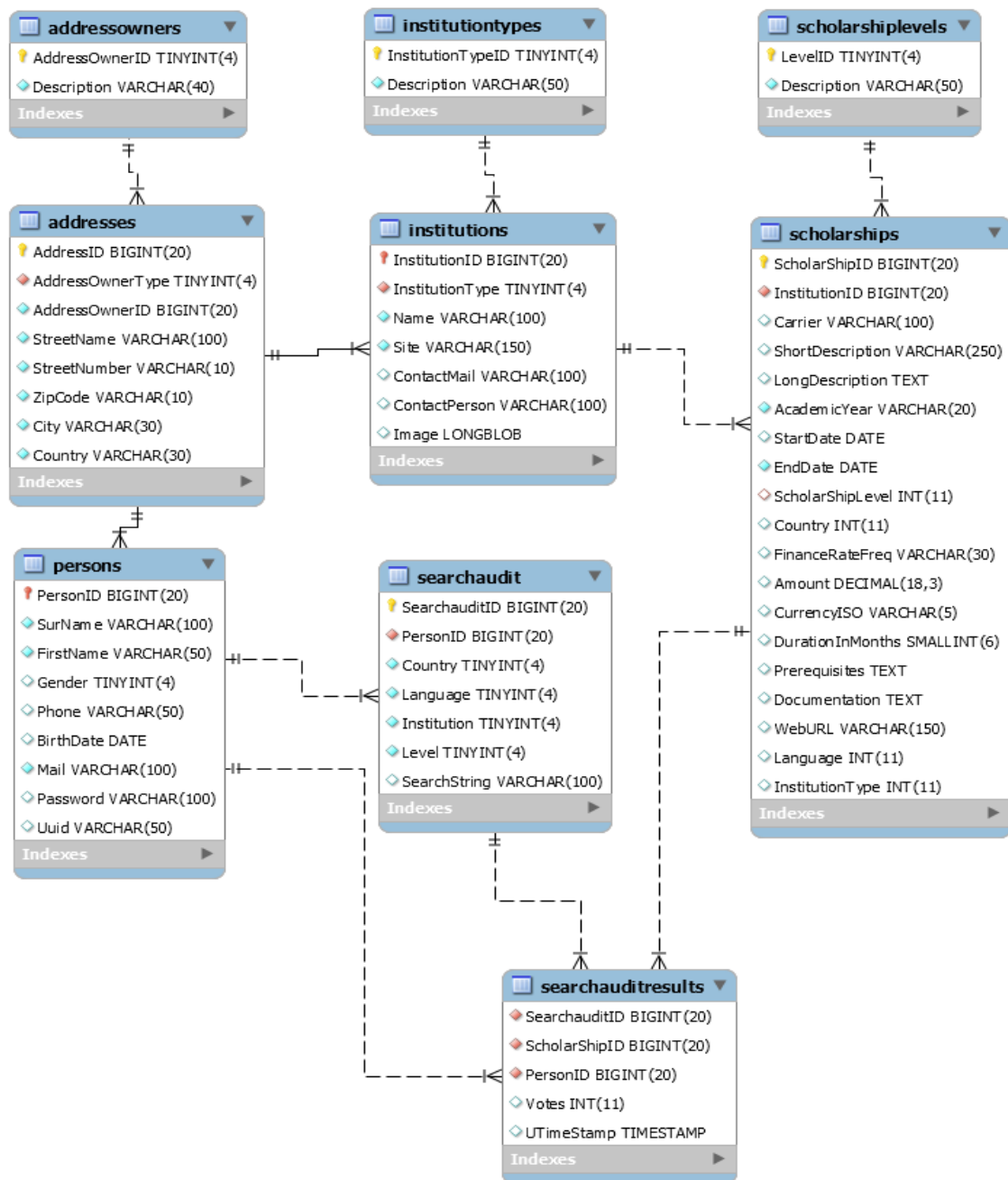


### 3.4.5.6 Λεπτομέρειες Υποτροφίας



Σχήμα 20. Διάγραμμα Ακολουθίας περίπτωσης χρήσης «Λεπτομέρειες Υποτροφίας»

### 3.5 Διάγραμμα Βάσης Δεδομένων



Σχήμα 21. Διάγραμμα Βάσης Δεδομένων SholarShip

Βλέποντας το διάγραμμα της Βάσης δεδομένων και κάνοντας μια σύγκριση με το διάγραμμα κλάσεων της ενότητας 3.4.2 σχήμα 5, μπορούμε να παρατηρήσουμε την πλήρη ταύτιση των δύο διαγραμμάτων (με εξαίρεση της μεθόδους του διαγράμματος κλάσεων που αφορούν το implementation).

## 3.6 Περιγραφή Βάσης Δεδομένων

Για την διατριβή αυτή επιλέχθηκε η MySQL για υλοποίηση της Βάσης Δεδομένων (θα μπορούσε να υλοποιηθεί και σε οποιαδήποτε άλλη σχεσιακή βάση από αυτές που αναφέρθηκαν). Ο κυριότερος λόγος χρήσης της MySQL είναι ότι χρησιμοποιείται ευρέως σε Web εφαρμογές και για προϊόν χωρίς license είναι μια αξιόπιστη βάση δεδομένων.

### 3.6.1 Πίνακες και Ανάλυση τους

Θα περιγραφούν οι κύριοι πίνακες με εξαίρεση τους πίνακες που αφορούν enumeration ή types όπως ο πίνακας AddressOwners.

#### 3.6.1.1 Addresses

Στον πίνακα αυτό αποθηκεύονται στοιχεία διευθύνσεων είτε αυτά αφορούν χρήστες της εφαρμογής (Persons), είτε αφορούν φορείς (Institutions) η διαφοροποίηση γίνεται από την κολόνα AddressOwnerType η οποία παίρνει την τιμή 1 για διεύθυνση χρήστη και 2 για την περίπτωση φορέα. Οι υπόλοιπες κολόνες δεν χρειάζονται περαιτέρω ανάλυση αφού μόνο το όνομα τους είναι αρκετό για την κατανόηση τους.

#### 3.6.1.2 Institutions

Πίνακας στον οποίο αποθηκεύονται οι φορείς που παρέχουν υποτροφίες όπως Πανεπιστήμια, κολλέγια και Ιδρύματα κοινωφελή. Για κάθε φορέα αποθηκεύεται ο τύπος του, το όνομα, η ιστοσελίδα του, mail και πρόσωπο επικοινωνίας καθώς και η εικόνα (logo). Ο πίνακας σχετίζεται με σχέση 1 προς πολλά με τον πίνακα ScholarShips, πρακτικά αυτός ο πίνακας θα πρέπει να γεμίζει – ενημερώνεται από τους ίδιους τους φορείς.

#### 3.6.1.3 Persons

Ο πίνακας αποθηκεύει τα στοιχεία των χρηστών με στοιχεία Επώνυμο, Όνομα, Φύλλο, Τηλέφωνο, Ημερομηνία γέννησης, email και κωδικό πρόσβασης. Περιέχει και μια κολόνα UUID στην οποία αποθηκεύεται ένα μοναδικό τυχαίο randomUUID μέσω του

java.util.UUID κάθε φορά που πραγματοποιείται επιτυχημένο login από κάποιον χρήστη.

#### **3.6.1.4 ScholarShips**

Ο πίνακας αυτό αποτελεί και την καρδιά του συστήματος μιας και σε αυτόν αποθηκεύονται οι υποτροφίες. Έχει μια κολόνα σύνδεσης με τον πίνακα Institutions (ποιος φορέας παρέχει την υποτροφία) τις κολόνες των κριτηρίων που περιγράφηκαν καθώς και τα κείμενα προκήρυξης υποτροφίας, προαπαιτούμενων και δικαιολογητικών. Στην ενότητα 3.1 και 3.2 περιγράφονται με αρκετή λεπτομέρεια.

#### **3.6.1.5 SearchAudit & SearchAuditResults**

Οι δύο πίνακες χρησιμοποιούνται αποκλειστικά από το σύστημα συστάσεων με την καταγραφή των κριτηρίων των απλών αναζητήσεων και των λέξεων κλειδιών και τον μηχανισμό voting. Η περιγραφή λειτουργίας τους περιγράφεται αναλυτικά στην ενότητα 2.5.

# Κεφάλαιο 4

## Σχεδιασμός και Υλοποίηση εφαρμογής Android

### 4.1 Middleware

#### 4.1.1 Περιγραφή Λειτουργίας

Η κεντρική εφαρμογή υλοποιήθηκε μέσω Eclipse ως Web project που γίνεται hosting σε Apache Tomcat 8. Είναι εξολοκλήρου γραμμένη σε Java και χρησιμοποιεί την βιβλιοθήκη Google Gson [21] για μετατροπή κλάσεων σε Json και αντίστροφα. Υποστηρίζει compression μέσω παραμετροποίησης του Web Server Tomcat (πράγμα που θεωρήθηκε αναγκαίο αφού κατά την αποστολή των υποτροφιών στέλνουμε και τις εικόνες των φορέων που τις παρέχουν).

##### 4.1.1.1 Database Connectivity (Τρόπος σύνδεσης με ΒΔ)

Για τις ανάγκες διασύνδεσης με την Βάση Δεδομένων χρησιμοποιήθηκε ο μηχανισμός Connection Pooling (όπως αυτός περιγράφηκε στην ενότητα 2.7.4). Όλες οι μέθοδοι που έχουν πρόσβαση στην Βάση Δεδομένων έχουν υλοποιηθεί σε μια μόνο κλάση (DBConnection.java) και όλες οι public μέθοδοι είναι shared ώστε να μπορούν να χρησιμοποιηθούν από οποιαδήποτε άλλη κλάση του Middleware χωρίς Instantiation.

##### **Συνοπτική Περιγραφή Μεθόδων Κλάσης DBConnection**

- `openConnection() : Connection`  
Δημιουργεί μια καινούργια σύνδεση στην ΒΔ και επιστρέφει το αντικείμενο Connection.
- `closeConnection(Connection dbConnection) : void`  
Κλείνει μια υπάρχουσα σύνδεση με την ΒΔ.
- `isLogon(long personID, String UUID) : Boolean`  
Ελέγχει αν ο χρήστης είναι ήδη συνδεδεμένος
- `checkLogin(String uname, String pwd) : LoginResults`

Ελέγχει κατά την λειτουργία της σύνδεσης το username και το password του χρήστη και επιστρέφει ανάλογα στο αντικείμενο LoginResults

- addInstitution(Institution institution) : long  
Προσθήκη ενός Institution στον πίνακα της ΒΔ.
- updInstitution(long institutionID, String image) : Boolean  
Ενημερώνει την εικόνα ενός Institution στην ΒΔ.
- addScholarShip(ScholarShip scholarship) : long  
Προσθέτει μια υποτροφία στον αντίστοιχο πίνακα της ΒΔ.
- userRegister(Person person) : LoginResults  
Πραγματοποιεί εγγραφή ενός νέου χρήστη
- getCleverScholarShips(long userID, long[] scholarShipIDs) : SearchResults
- getScholarShipDetails(long auditID, long userID, long scholarShipID) : ScholarShip
- findScholarShips(SearchCriteria srcCriteria) : SearchResults
- matchSearch(SearchCriteria srcCriteria) : List<CleverDataSet>

#### **4.1.1.2 Διαδικασία Σύνδεσης**

Πρόκειται για μια απλή διαδικασία κατά την οποία το κεντρικό σύστημα λαμβάνει μέσω ενός http request της μεθόδου doLogin τις παραμέτρους username και password (κρυπτογραφημένο) και ελέγχει την ύπαρξη του χρήστη στην Βάση Δεδομένων και συγκεκριμένα στον πίνακα Person με το σωστό password. Σε περίπτωση που βρεθεί ο χρήστης με το σωστό κωδικό δημιουργεί μια κλάση τύπου LoginResults στην οποία αποθηκεύει και ένα UUID (string) το οποίο ισχύει για όσο είναι ενεργή η client εφαρμογή. Πρακτικά σε κάθε request μετά την επιτυχημένη σύνδεση στέλνεται και το UUID. Σε περίπτωση μη εύρεσης του χρήστη στην Βάση ή λανθασμένου κωδικού στη κλάση αυτή η ιδιότητα errorMessage γεμίζει με ένα κατάλληλο μήνυμα λάθους και επιπλέον η ιδιότητα login γίνεται false.

#### **4.1.1.3 Διαδικασία Εγγραφής Χρήστη**

Κατά την πρώτη εκκίνηση της εφαρμογής ο χρήστης πρέπει να κάνει εγγραφή στο σύστημα. Η κεντρική διαδικασία που παρέχει την υπηρεσία εγγραφής του χρήστη ονομάζεται doRegister και πρακτικά λαμβάνει από το http request μια κλάση τύπου

Person (για λόγους απλότητας κατά την εγγραφή δεν καταχωρούνται από την εφαρμογή κινητής συσκευής όλα τα στοιχεία παρά μόνο τα απαραίτητα). Το κεντρικό σύστημα αφού ελέγξει την μη ύπαρξη χρήστη με το ίδιο username καταχωρεί στον πίνακα της Βάσης Person την νέα εγγραφή και επιστρέφει στον client μια κλάση LoginResults με τον ίδιο τρόπο που γίνεται και κατά την διαδικασία σύνδεσης χρήστη.

#### 4.1.1.4 Απλή Αναζήτηση

Με την επιλογή των κριτηρίων που πραγματοποιείται από τον χρήστη στην εφαρμογή της κινητής συσκευής δημιουργείται η κλάση SearchCriteria (που περιγράψαμε παραπάνω). Ανάλογα τώρα με την επιλογή κριτηρίων του χρήστη σε κάθε κατηγορία τίθεται τιμή στα αντίστοιχη ιδιότητα (property) της κλάσης σύμφωνα με την παρακάτω διαδικασία.

Έστω ότι ο χρήστης επιλέγει από την κατηγορία «Επίπεδο Σπουδών» (ScholarShipLevel) η οποία παρέχει τις κάτωθι επιλογές

ScholarShipLevel	
SECONDARY	<input type="checkbox"/>
UNDERGRADUATE	<input checked="" type="checkbox"/>
POSTGRADUATE	<input checked="" type="checkbox"/>
DOCTORATE	<input checked="" type="checkbox"/>
POSTDOCTORATE	<input type="checkbox"/>

Πίνακας 2. Επίπεδα Σπουδών Υποτροφιών

Κατά τη διάρκεια της αποθήκευσης των κριτηρίων η εφαρμογή (client) θα αναθέσει τιμή στην ιδιότητα «level»

σύμφωνα με τον τύπο

$$level = \sum_{i=1}^v check_i 2^{i-1}, \text{ όπου } check_i = 1 \text{ if checked else } 0$$

Έτσι για το παραπάνω παράδειγμα έχουμε **level = 2 + 4 + 8 = 14**

Με το ίδιο τρόπο υπολογίζονται και οι υπόλοιπες ιδιότητες των κριτηρίων της κλάσης. Οι λέξεις κλειδιά για αναζήτηση αποθηκεύονται στην ιδιότητα SearchStr. Στη συνέχεια η κλάση αυτή μετατρέπεται σε json format και αποστέλλεται στο κεντρικό σύστημα.

Το κεντρικό σύστημα λαμβάνοντας ένα http request (POST) ένα String (Json format) το μετατρέπει σε κλάση SearchCriteria. Από αυτή την κλάση αναζήτησης συνθέτει ένα query με where clause, που συντίθεται από τα κριτήρια αναζήτησης και τις λέξεις κλειδιά για αναζήτηση στα αποθηκευμένα κείμενα των υποτροφιών στην βάση δεδομένων από την μέθοδο produceSimpleSearchQuery.

Για κάθε ιδιότητα της κλάσης που αφορά κριτήρια αναζήτησης δημιουργείται μια συνθήκη της μορφής

*ScholarShipLevel & 6 = ScholarShipLevel*

Και τελικά όλες μαζί αυτές οι συνθήκες με AND operator συνθέτουν τη συνολική WHERE Clause του SQL query που θα εκτελεστεί στην βάση δεδομένων.

Όσον αφορά τώρα τις λέξεις κλειδιά που θα αναζητηθούν στα πεδία κειμένου των υποτροφιών (LongDescription, Prerequisites, Documentation) πραγματοποιείται αναζήτηση σε BOOLEAN MODE όπως αυτή περιγράφεται στο Παράρτημα Α.

#### **4.1.1.4.1 Μηχανισμός Voting**

Για κάθε απλή αναζήτηση που πραγματοποιείτε από κάποιον χρήστη, τα κριτήρια αναζήτησης καθώς και οι λέξεις κλειδιά για αναζήτηση δημιουργούν μια νέα εγγραφή στον πίνακα της Βάσης Δεδομένων «SearchAudit» στην οποία καταχωρούνται οι τιμές των κριτηρίων αναζήτησης από την κλάση «SearchCriteria», ο κωδικός του χρήστη και οι λέξεις κλειδιά προς αναζήτηση. Για την περίπτωση που υπάρχει στο πίνακα εγγραφή με τα ίδια δεδομένα δεν γίνεται καμία καταχώρηση. Σε κάθε περίπτωση στα αποτελέσματα αναζήτησης «SearchResults» επιστρέφεται και ο κωδικός auditID της εγγραφής στον πίνακα.

Στην συνέχεια εφόσον ο χρήστης της εφαρμογής κινητής συσκευής επιλέξει κάποια υποτροφία από την λίστα των αποτελεσμάτων στέλνονται στο κεντρικό σύστημα τα στοιχεία auditID, userID, και scholarShipID (η εγγραφή του πίνακα SearchAudit, ο κωδικός χρήστη που πραγματοποίησε την επιλογή και ο κωδικός της επιλεγμένης υποτροφίας αντίστοιχα). Το κεντρικό σύστημα με την παραλαβή του μηνύματος για λεπτομέρειες της επιλεγμένης υποτροφίας με τα στοιχεία που είπαμε δημιουργεί μια νέα εγγραφή στον πίνακα «SearchAuditResults», θέτοντας στην κολόνα “Votes” την τιμή ένα (μια ψήφο), σε περίπτωση που υπάρχει εγγραφή στον πίνακα με το ίδιο auditID, scholarShipID και personID τότε απλά αυξάνει η τιμή της κολόνας Votes κατά ένα και ενημερώνεται το TimeStamp της συγκεκριμένης γραμμής του πίνακα.



Με αυτό τον τρόπο πραγματοποιείται μια ψηφοφορία εν αγνοία του χρήστη για τις υποτροφίες που ψάχνει λεπτομέρειες κάποιας υποτροφίας που τον ενδιαφέρει.

#### 4.1.1.5 Έξυπνη Αναζήτηση

Κατά την Έξυπνη Αναζήτηση ο χρήστης πραγματοποιεί ακριβώς ίδια διαδικασία με την απλή, η διαφορά είναι ότι σε αυτή την περίπτωση τα αποτελέσματα δεν προέρχονται από αναζήτηση στον πίνακα Υποτροφιών, αλλά από τον μηχανισμό συστάσεων που υποστηρίζει το κεντρικό σύστημα.

Κατά την παραλαβή του http request από την κεντρική εφαρμογή (ακριβώς όπως και στην απλή αναζήτηση) δημιουργείται από το Json η κλάση SearchCriteria. Στη συνέχεια το σύστημα εκτελεί το ένα query της παρακάτω μορφής :

```
SELECT searchauditresults.*, searchaudit.SearchString
FROM searchaudit INNER JOIN searchauditresults ON
    searchaudit.SearchauditID = searchauditresults.SearchauditID
WHERE Country = 7 AND Language = 3 AND Institution = 7 AND Level = 31
ORDER BY 5 DESC,3,2 LIMIT 40
```

Για την συλλογή από τους πίνακες SearchAudit και SearchAuditResults με ίδια κριτήρια ανεξαρτήτως χρήστη που πραγματοποίησε την αναζήτηση. Αξίζει να σημειωθεί ότι στο query αυτό υπάρχει μια πρόβλεψη για το μέγιστο όριο εγγραφών που μπορεί να επιστρέψει (LIMIT 40), η συγκεκριμένη μεταβλητή είναι παράμετρος του συστήματος (LIMIT\_RECORDS) και μπορεί να αλλάξει όπως θα δούμε παρακάτω στην ενότητα 4.1.1.8.

Από τις εγγραφές που επιστρέφονται το σύστημα εφαρμόζει τον αλγόριθμο string similarity (Levenshtein distance) ανάμεσα στο String αναζήτησης και αυτών που έχουν επιστραφεί από την βάση δεδομένων με την εκτέλεση του προηγούμενου query. Κάθε εγγραφή της οποίας η ομοιότητα είναι μικρότερη από την παράμετρο του συστήματος SIMILARITY\_LIMIT αποκόπτεται από την λίστα εγγραφών.

Στην συνέχεια από την λίστα των εγγραφών καθώς και από τον χρήστη για τον οποίο πρέπει να δώσουμε συστάσεις δημιουργείται ένα πίνακας (όπως Πίνακας 1). Για τον πίνακα αυτόν καλείται η μέθοδος recommendations, η οποία αφού πραγματοποιήσει παραγοντοποίηση πινάκων όπως αυτή περιγράφηκε στην ενότητα 2.5.7, παράγεται ένας πίνακας παραγοντοποίησης με ίδια διάσταση με τον αρχικό. Από αυτόν τον πίνακα

επιλέγουμε κατά σειρά φθίνουσα για την γραμμή του χρήστη που πρέπει να δώσουμε συστάσεις οι κολόνες (στην πράξη οι υποτροφίες). Με την λίστα των κωδικών υποτροφιών ζητάμε από την Βάση Δεδομένων τα στοιχεία των υποτροφιών με την μέθοδο `getCleverScholarShips` και δημιουργούμε μια κλάση αποτελεσμάτων τύπου `SearchResults` την οποία και αποστέλλει το κεντρικό σύστημα στην εφαρμογή κινητής συσκευής.

#### 4.1.1.6 Παράδειγμα διαδικασίας «Απλής Αναζήτησης - Έξυπνης Αναζήτησης»

Έστω ότι η εφαρμογή κινητής συσκευής καλεί την μέθοδο `doSimpleSearch` με παράμετρο εισόδου το παρακάτω `Json String` :

```
{"uuid": "cf6e9c5bc0094df2aa6cb52db73444e4", "searchStr": "Πληροφορικής", "personid": 5, "institution": 7, "language": 3, "level": 4, "gps": 1, "country": 7, "activePeriod": true}
```

Το κεντρικό σύστημα αφού ελέγξει ότι ο χρήστης έχει κάνει login στο σύστημα θα εκτελέσει τα κάτωθι queries :

- ```
SELECT scholarships.ScholarShipID, scholarships.ShortDescription, institutions.Image
FROM scholarships INNER JOIN institutions ON scholarships.InstitutionID =
institutions.InstitutionID
WHERE (scholarships.EndDate >= CURDATE())
AND (scholarships.InstitutionType & 7 = scholarships.InstitutionType)
AND (scholarships.ScholarShipLevel & 4 = scholarships.ScholarShipLevel)
AND (scholarships.Country & 7 = scholarships.Country)
AND (scholarships.Language & 3 = scholarships.Language)
AND MATCH (LongDescription, Prerequisites, Documentation) AGAINST(' Πληροφορικής
' IN BOOLEAN MODE)
ORDER BY MATCH (LongDescription, Prerequisites, Documentation) AGAINST(' Πληροφορικής'
IN BOOLEAN MODE) DESC
```

Το 1<sup>ο</sup> query θα μας φέρει τις υποτροφίες για τις οποίες οι λέξεις κλειδιά αλλά και τα επιλεγμένα κριτήρια βρεθούν στην Βάση Δεδομένων. (έστω ότι επιστρέφει τις υποτροφίες 2,3,7,8)

- ```
INSERT INTO searchaudit (PersonID, Country, Language, Institution, Level, SearchString) VALUES
(5,7,3,7,4,'Πληροφορικής')
```

Το 2<sup>ο</sup> εφόσον στο 1<sup>ο</sup> υπάρχουν εγγραφές (δεν επιστραφεί κενό recordset) θα κάνει εισάγει στον πίνακα `searchaudit` (με τα κριτήρια, χρήστη και λέξεις κλειδιά)

Για κάθε τώρα Απλή Αναζήτηση των χρηστών γίνεται η ίδια διαδικασία και έστω ότι ο πίνακας searchaudit έχει την μορφή

SearchAuditID	PersonID	Country	Language	Institution	Level	SearchString
1	2	7	3	7	4	Πληροφορική
2	3	7	3	7	4	Πληροφορική
3	4	7	3	7	4	Πληροφορική
4	5	7	3	7	4	Πληροφορική

**Πίνακας 2.** Αναζήτηση Επιπέδου Μεταπτυχιακού με την λέξη «Πληροφορική»

Αν κάθε χρήστης από αυτούς που φαίνονται στον Πίνακα 2 (2,3,4,5) από τα αποτελέσματα των αναζητήσεων επέλεγε και κάποια/ες υποτροφίες από αυτές που επέστρεφε το σύστημα θα είχαμε ως αποτέλεσμα στον πίνακα searchauditresults τις παρακάτω εγγραφές (όπου ο χρήστης 2 έχει επιλέξει δύο φορές την υποτροφία 2, μία την 3, 3 φορές την 8 και καμία την 7, κ.τ.λ.)

SearchAuditID	ScholarShipID	PersonID	Votes	UTimeStamp
1	2	2	2	2016-05-02 12:29
1	3	2	1	2016-05-02 12:59
1	8	2	3	2016-05-02 13:06
2	3	3	1	2016-05-02 13:07
2	7	3	2	2016-05-02 13:08
2	8	3	1	2016-05-02 13:07
3	2	4	2	2016-05-02 13:09
3	7	4	3	2016-05-02 13:10
3	8	4	1	2016-05-02 13:11
4	8	5	4	2016-05-02 13:06

**Πίνακας 3.** Ψηφοφορία αναζητήσεων του πίνακα 1 από τους χρήστες στα αποτελέσματα των υποτροφιών.

Έστω ένας νέος χρήστης (ο χρήστης με id 6) πραγματοποιούσε με ίδια κριτήρια και λέξη κλειδί «Πληροφορική» Έξυπνη Αναζήτηση. Αυτό για το σύστημα θα είχε ως αποτέλεσμα να δημιουργηθεί ένα πίνακας της παρακάτω μορφής

## ΥΠΟΤΡΟΦΙΕΣ

		2	3	7	8
ΧΡΗΣΤΕΣ	Χρήστης/Υποτροφία				
	2	2	1	0	3
	3	0	1	2	1
	4	2	0	3	1
	5	0	0	0	4
	6	0	0	0	0

**Πίνακας 4.** Πίνακας στον οποίο θα γίνει παραγοντοποίηση

Πριν την δημιουργία του πίνακα έχουν απορριφθεί οι εγγραφές των οποίων η ομοιότητα των λέξεων δεν είναι στα επιθυμητά επίπεδα.

Ο Πίνακας 4 δίνεται ως input στον αλγόριθμο συστάσεων ο οποίος παράγει ένα πίνακα της μορφής

## ΥΠΟΤΡΟΦΙΕΣ

		2	3	7	8
ΧΡΗΣΤΕΣ	Χρήστης/ Υποτροφία				
	2	2.032	1.202	3.830	2.854
	3	1.161	0.569	1.998	1.176
	4	1.932	0.722	2.961	1.093
	5	2.958	1.703	5.501	3.977
	6	0.610	0.282	1.023	0.555

**Πίνακας 5.** Παραγοντοποιημένος πίνακας

Ο αλγόριθμος συστάσεων λοιπόν για τον χρήστη με personID 6 θα επιστρέψει τις υποτροφίες με αύξουσα σειρά 7,2,8,3 στην εφαρμογή.

Αντίστοιχα η εκτέλεση αν δεν ήταν νέος χρήστης (και έπρεπε να δοθούν συστάσεις για τον χρήστη 5) ο αρχικός πίνακας 4 προς παραγοντοποίηση δεν θα είχε την τελευταία γραμμή του χρήστη 6, θα φτιαχνόταν ο παραγοντοποιημένος πίνακας με μια γραμμή λιγότερη και το σύστημα θα πρότεινε πάλι με αύξουσα σειρά για τον χρήστη 5 τις υποτροφίες του παραγοντοποιημένου πίνακα, και οι προτάσεις για τον χρήστη 5 θα ήταν όπως φαίνεται στον πίνακα 6, πάλι με αύξουσα σειρά οι υποτροφίες 7,8,2,3.

		ΥΠΟΤΡΟΦΙΕΣ			
		2	3	7	8
ΧΡΗΣΤΕΣ	Χρήστης/ Υποτροφία				
	2	2.305	1.197	4.237	2.573
	3	1.047	0.594	2.014	1.171
	4	1.537	0.793	2.815	1.715
	5	3.500	1.762	6.336	3.907

**Πίνακας 6.** Παραγοντοποιημένος πίνακας για σύσταση στον χρήστη 5

#### 4.1.1.7 Middleware Logging

Για όλες τις διεργασίες που πραγματοποιούνται στον κεντρικό σύστημα χρησιμοποιήθηκε μηχανισμός logging και συγκεκριμένα η βιβλιοθήκη log4j 1.2.17 της Apache (Apache log4j 1.2.17 is distributed under the Apache License version 2.0).

Η παραμετροποίηση της συγκεκριμένης βιβλιοθήκης γίνεται μέσω του αρχείου log4j.properties που βρίσκεται στην εφαρμογή του κεντρικού συστήματος. Δεν θα επεκταθούμε περισσότερο για την συγκεκριμένη βιβλιοθήκη μιας και είναι ευρέως γνωστή και χρησιμοποιούμενη για Web εφαρμογές.

#### 4.1.1.8 Παράμετροι Κεντρικού Συστήματος - Middleware

Στην κλάση του κεντρικού συστήματος Constants βρίσκονται μια σειρά από public static παραμέτρους που επηρεάζουν την λειτουργία του συστήματος.

- **public static** String *dbClass*;  
ο driver σύνδεσης με την βάση δεδομένων.
- **public static** String *dbName*;  
το όνομα της βάσης δεδομένων.
- **public static** String *dbUrl*;  
το URL της βάσης δεδομένων.
- **public static** String *dbUser*;  
το username για σύνδεση με την βάση δεδομένων.
- **public static** String *dbPwd*;  
το password για σύνδεση με την βάση δεδομένων.
- **public static** String *LIMIT\_RECORDS*;  
ο μέγιστος αριθμός εγγραφών στο αρχικό βήμα της έξυπνης αναζήτησης που θα ανασυρθεί.
- **public static double** *SIMILARITY\_LIMIT*;  
ο βαθμός ομοιότητας κάτω από τον οποίο θα εξαιρεθούν εγγραφές από την βάση.

Όλες οι παραπάνω μεταβλητές είναι δηλωμένες στο configuration αρχείο της εφαρμογής (web.xml) και φορτώνονται κατά την διάρκεια εκκίνησης της εφαρμογής από τον Web Server (στην δική μας περίπτωση χρησιμοποιείται ο Apache Tomcat v 8.0)

## 4.2 Android App

### 4.2.1 Περιγραφή Λειτουργίας

Η εφαρμογή φτιάχτηκε εξ' ολοκλήρου με το Android Studio version 1.5.1. Με την έναρξη της εφαρμογής ο χρήστης βλέπει το αρχικό Activity (MainActivity) στο οποίο υπάρχει μενού με τις προσφερόμενες ενέργειες που μπορεί να πραγματοποιήσει. Όλες οι ενέργειες που παρέχονται από το αρχικό μενού δημιουργούν και αντίστοιχα Activities όπως παρακάτω :

- LoginActivity
- RegisterActivity
- SearchFilters
- SimpleSearch

- CleverSearch
- ScholarShipDetails

Κάθε activity από τα παραπάνω καλεί και μια κλήση στο κεντρικό σύστημα μέσω της AsyncHttpClient. Αυτό συμβαίνει γιατί κατά κανόνα θέλουμε η εφαρμογή να ανταποκρίνεται και να μην δείχνει ότι έχει κολλήσει κατά την διάρκεια των κλήσεων του κεντρικού συστήματος. Κάθε κλήση στα κεντρικά Rest Services πραγματοποιείται πάντα ασύγχρονα λοιπόν, και τερματίζει είτε με Success ή Failure events. Η βιβλιοθήκη που χρησιμοποιήθηκε για αυτές τις κλήσεις είναι η android-async-http-1.4.9.jar [22]. Η μετατροπή των κλάσεων σε Json και αντίστροφα έγινε με χρήση της βιβλιοθήκης της Google Gson library (gson-2.5.jar) [21]. Η ίδια βιβλιοθήκη χρησιμοποιήθηκε και στο κεντρικό σύστημα για μετατροπή από Json σε κλάση και αντίστροφα.

#### 4.2.2 Υποστήριξη διαφορετικών γλωσσών

Είναι πάντα μια καλή πρακτική να εξάγουμε UI strings από τον κώδικα της εφαρμογής μας και να τα κρατάμε σε ένα εξωτερικό αρχείο. Το Android το κάνει αυτό εύκολα με έναν κατάλογο πόρων (resources directory) για κάθε project Android.

Με την δημιουργία ενός project σε Android Studio δημιουργείτε κάτω από τον κατάλογο *res/values* το αρχείο strings.xml, πρακτικά στο αρχείο αυτό αποθηκεύουμε constants strings σε μορφή xml όπως παρακάτω :

```
<resources>
  <string name="app_name">Scholar Ships</string>
  <string name="navigation_drawer_open">Open navigation drawer</string>
  <string name="navigation_drawer_close">Close navigation drawer</string>
  <string name="action_settings">Settings</string>
</resources>
```

Όλες αυτές οι σταθερές είναι προσβάσιμες τόσο από αλλά resources αρχεία που αφορούν την εμφάνιση των οθονών (layout), των μενού (menu) αλλά και το ίδιο το AndroidManifest.xml με μια δήλωση της μορφής **android:label="@string/app\_name"** όσο και μέσα από τις κλάσεις των activities με την δήλωση **R.string.app\_name**.

Δημιουργώντας τώρα ένα κατάλογο της παρακάτω μορφής κάτω από τον res/

```
MyProject/
  res/
    values/
      strings.xml
    values-el/
      strings.xml
    values-fr/
      strings.xml
```

τότε έχουμε την δυνατότητα να έχουμε αντίγραφο του αρχικού strings.xml στα οποία αλλάζουμε το περιεχόμενο των σταθερών στην γλώσσα που επιθυμούμε, παρακάτω βλέπουμε πως διαμορφώνεται το αρχείο strings.xml για γλώσσα el (ελληνικά).

```
<resources>
  <string name="app_name">Υποτροφίες</string>
  <string name="navigation_drawer_open">Άνοιγμα πλοήγησης</string>
  <string name="navigation_drawer_close">Κλείσιμο πλοήγησης</string>
  <string name="action_settings">Ρυθμίσεις</string>
</resources>
```

Με αυτή την προσέγγιση κάτω από τον κατάλογο values βρίσκεται η προκαθορισμένη (default) γλώσσα της εφαρμογής μας, και στους υπόλοιπους καταλόγους οι γλώσσες που θέλουμε να υποστηρίζει η εφαρμογή μας.

Με αυτόν τον τρόπο όταν ο χρήστης αλλάζει γλώσσα για τη συσκευή του και η εφαρμογή θα αλλάξει αυτόματα όλα τα πεδία τύπου κειμένου στην επιλεγμένη γλώσσα εφόσον έχει οριστεί για αυτή αρχείο strings.xml.

Η εφαρμογή των υποτροφιών υποστηρίζει πλήρως δύο γλώσσες Αγγλικά και Ελληνικά με την τεχνική που περιγράψαμε τώρα.

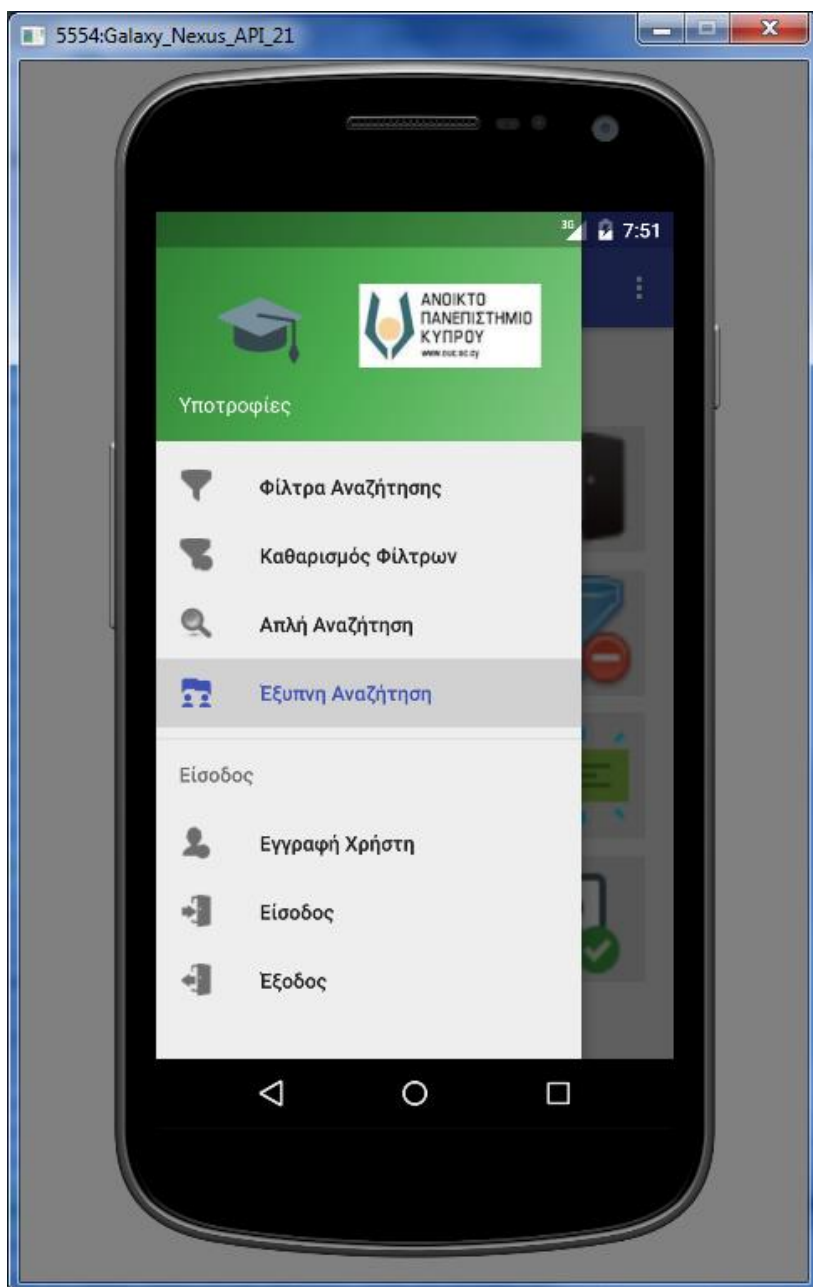
### 4.2.3 Το User Interface της εφαρμογής

Η εφαρμογή ακολουθεί το σχεδιαστικό πρότυπο που προτάσσει η Google (Material Design) όπως αυτό περιγράφηκε στην ενότητα 2.3. Αξίζει να σημειωθεί ότι όλες οι οθόνες μπορούν να αλλάξουν εμφάνιση χωρίς να αλλάξει ο κώδικας, αρκεί να κρατήσουμε αναλλοίωτα τα control names. Εικόνες, περιγραφές, μηνύματα κ.α. μπορούν να αλλάξουν με την αλλαγή των αντίστοιχων Layout xml files.

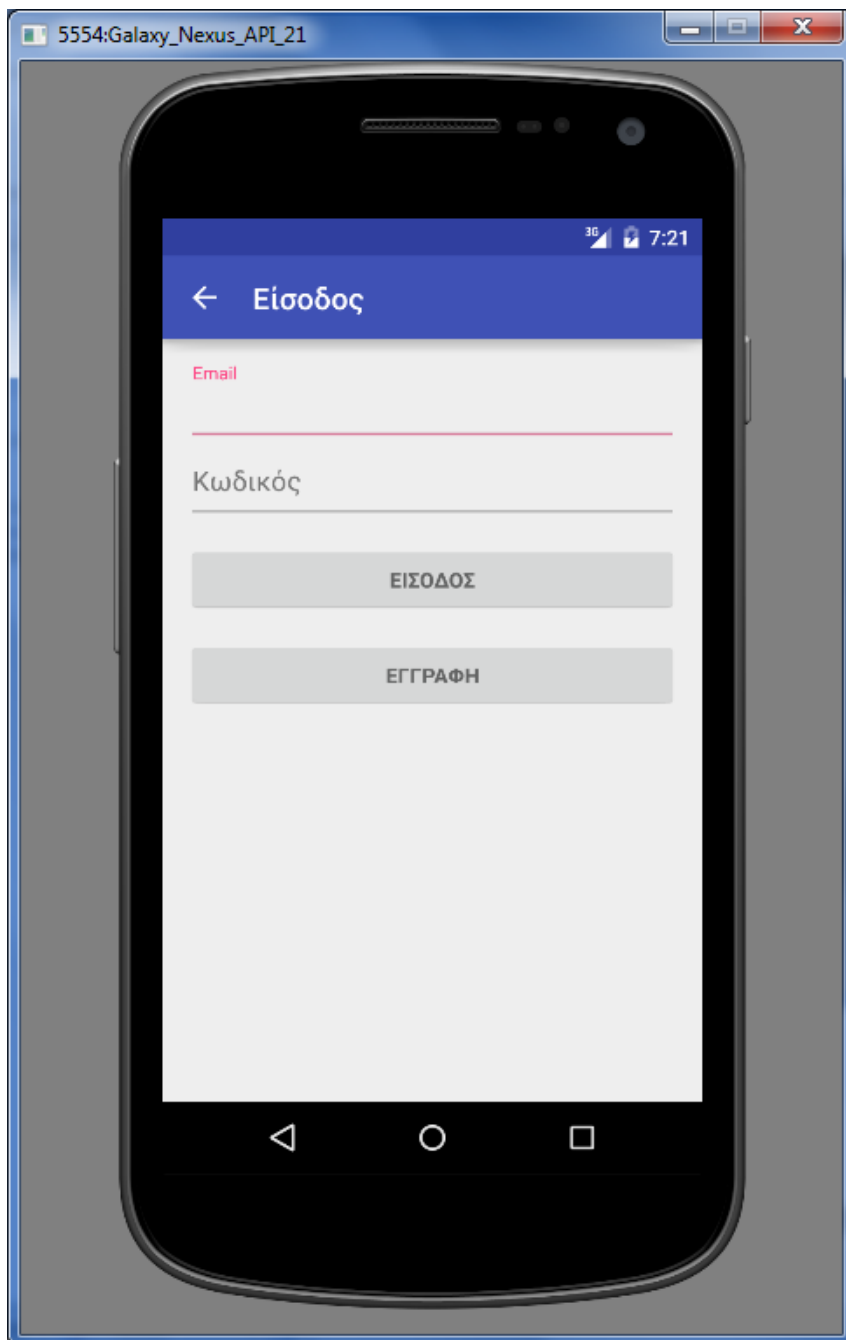


## 4.2.4 Οθόνες εφαρμογής Android

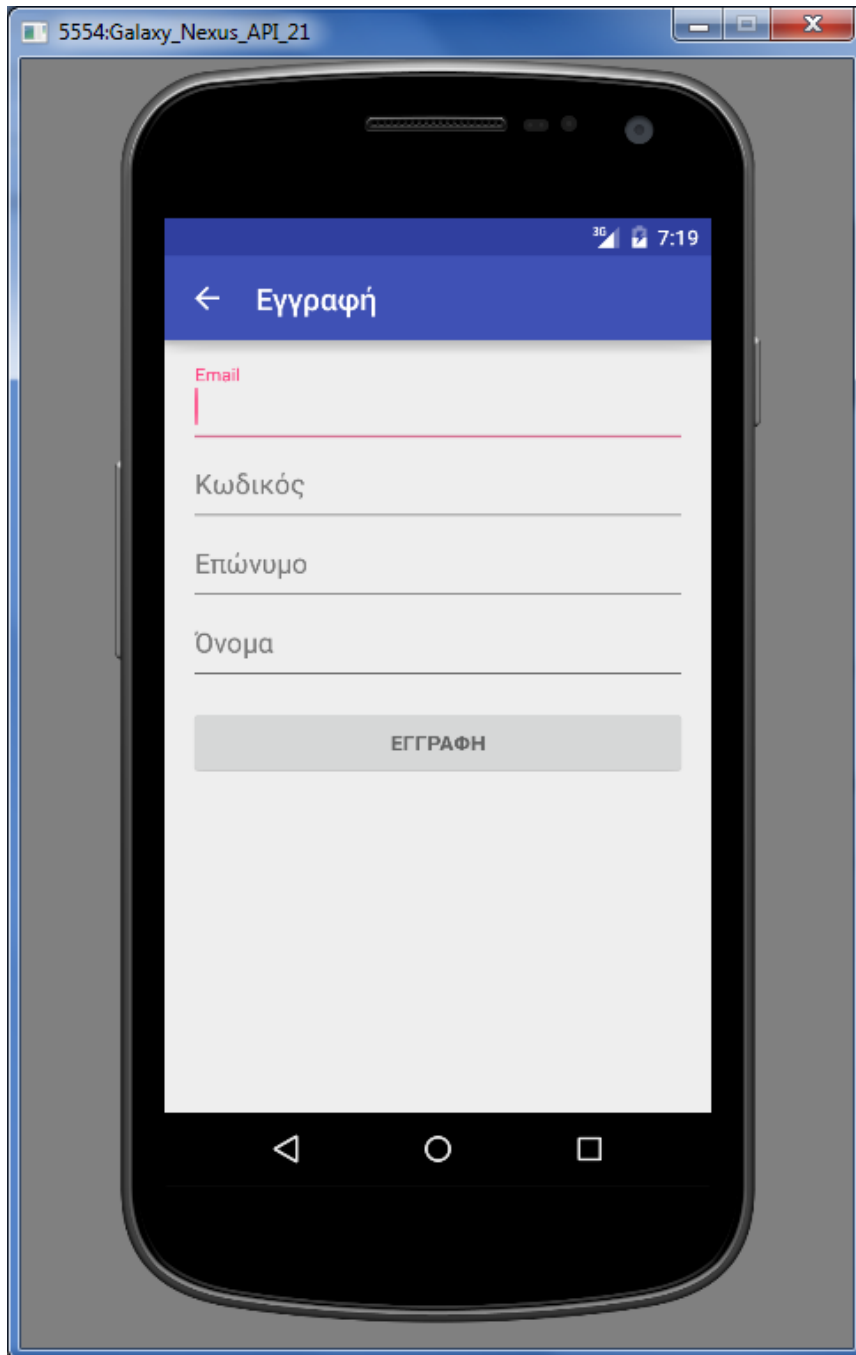
### 4.2.4.1 Αρχική Οθόνη και μενού Επιλογών



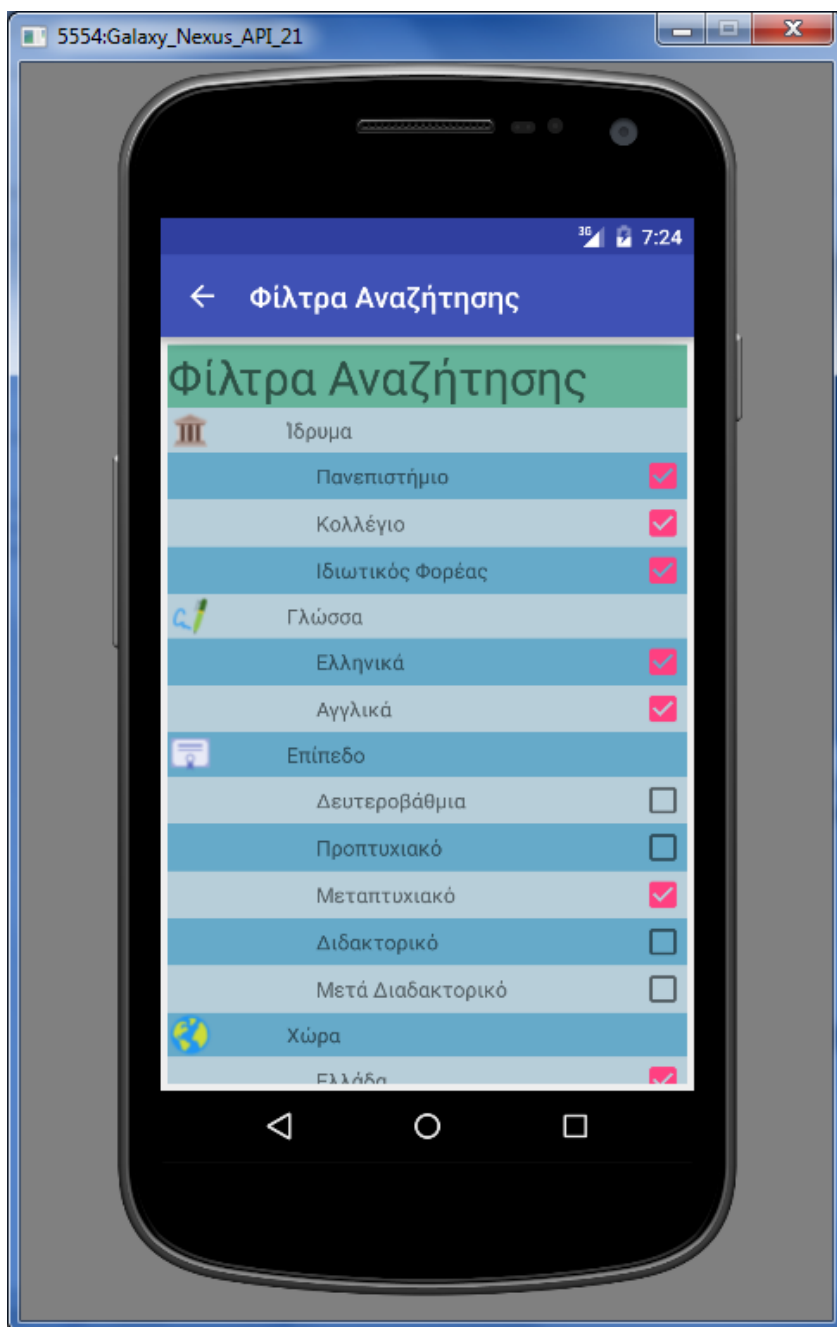
#### 4.2.4.2 Οθόνη Σύνδεσης Χρήστη



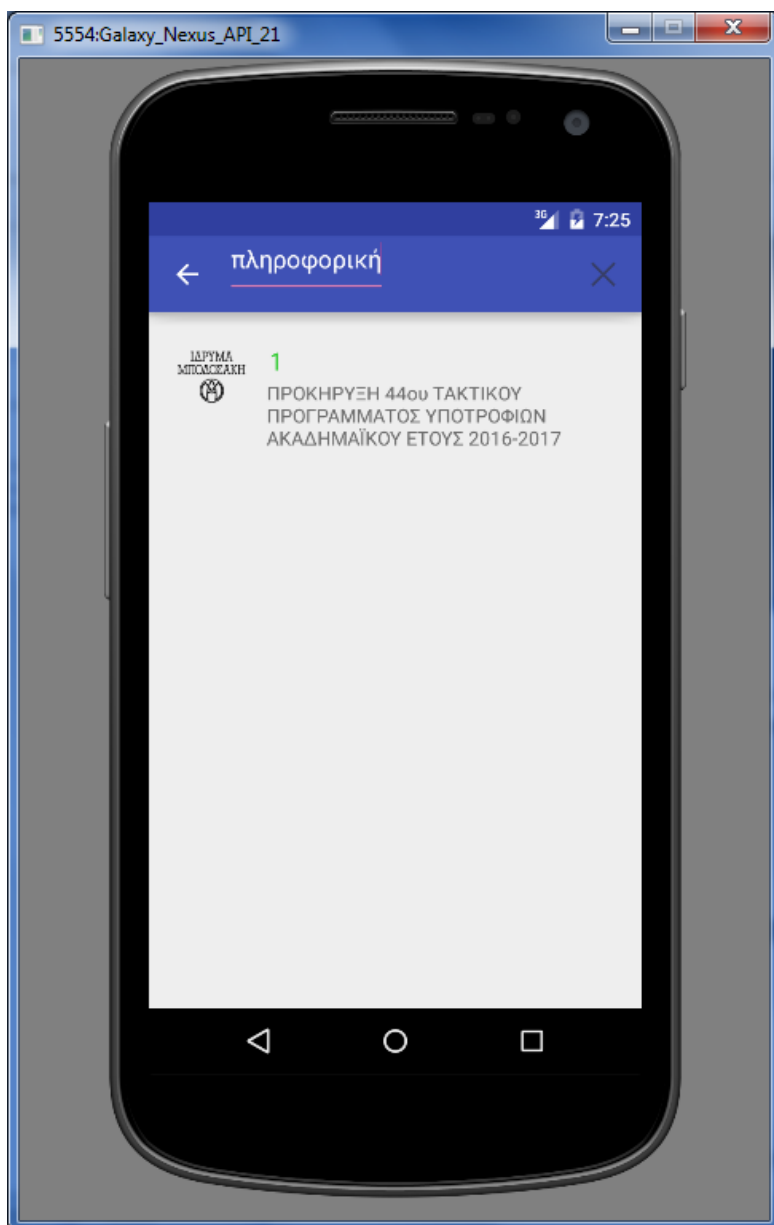
#### 4.2.4.3 Οθόνη Εγγραφής Χρήστη



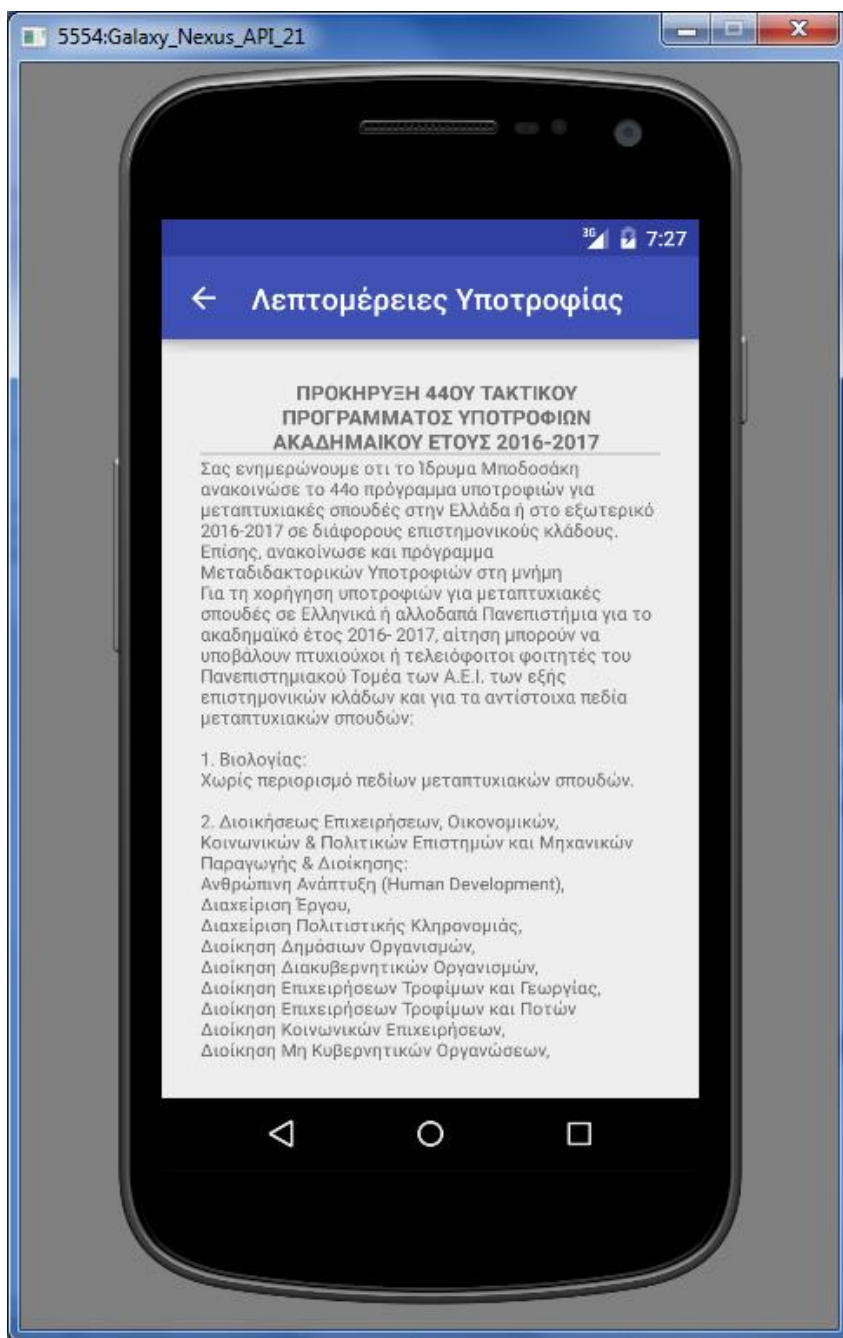
#### 4.2.4.4 Οθόνη Φίλτρων - Κριτηρίων Αναζήτησης



#### 4.2.4.5 Οθόνη Αναζήτησης



#### 4.2.4.6 Οθόνη Λεπτομερειών Υποτροφίας



#### 4.2.4.7 Περιγραφή Οθονών Android App

«**Αρχική Οθόνη**» : Είναι το Main Activity της εφαρμογής και πρακτικά η οθόνη που βλέπει ο χρήστης με το που ανοίγει την εφαρμογή. Πατώντας click πάνω αριστερά εμφανίζεται το κεντρικό μενού της εφαρμογής με τις διαθέσιμες επιλογές. Η μόνη επιλογή την οποία μπορεί να επιλέξει ο χρήστης αρχικά είναι το Logon ή το Register.

**«Σύνδεσης Χρήστη»** : Πρόκειται για το Login Activity που εμφανίζεται αφού ο χρήστης επιλέξει Login από το αρχικό μενού του Main Activity. Σε αυτό ο χρήστης καλείται να δώσει ένα Username (email κατά το registration) και ένα Password ώστε να κάνει sign on στο κεντρικό σύστημα.

**«Εγγραφής Χρήστη»** : Πρόκειται για το Register Activity που εμφανίζεται αφού ο χρήστης επιλέξει Register από το αρχικό μενού του Main Activity. Σε αυτό ο χρήστης καλείται να δώσει κάποια στοιχεία (username, password, surname, name) ώστε να κάνει εγγραφή ως νέος χρήστης στο κεντρικό σύστημα.

**«Φίλτρων – Κριτηρίων Αναζήτησης»** : Πρόκειται για το Search Filters Activity που εμφανίζεται αφού ο χρήστης επιλέξει Search Filters από το αρχικό μενού του Main Activity. Σε αυτό ο χρήστης καλείται να επιλέξει από μια λίστα επιλογών τα κριτήρια που επιθυμεί με τα οποία θα γίνονται οι αναζητήσεις υποτροφιών στο κεντρικό σύστημα. Οι επιλογές του χρήστη αποθηκεύονται τοπικά και ισχύουν για το ενεργό session, δηλαδή κάθε φορά που ο χρήστης ανοίγει την εφαρμογή ισχύουν οι προκαθορισμένες επιλογές από την εφαρμογή. Αφού ο χρήστης κάνει τις επιλογές που επιθυμεί στη συνέχεια πατάει Εφαρμογή Φίλτρων ώστε να αποθηκευτούν οι επιλογές του.

**«Απλής και Σύνθετης Αναζήτησης»** : Πρόκειται για το Simple και Clever Search Activities που εμφανίζεται αφού ο χρήστης επιλέξει αντίστοιχα Simple Search ή Clever Search από το αρχικό μενού του Main Activity. Σε αυτά ο χρήστης καλείται να εισάγει τις λέξεις κλειδιά (προς αναζήτηση) που επιθυμεί αφού πρώτα πατήσει το Search εικονίδιο (πάνω αριστερά στην οθόνη). Το αποτέλεσμα και των δύο αναζητήσεων να εμφανιστούν στο activity λίστες με τις υποτροφίες που θα στείλει το κεντρικό σύστημα.

**«Λεπτομέρειες Υποτροφίας»** : Το συγκεκριμένο Scholar Ship Details Activity ενεργοποιείται κάθε φορά που ο χρήστης κάνει παρατεταμένο click σε κάποια υποτροφία από αυτές που εμφανίζονται στα Activity Simple Search και Clever Search. Το αποτέλεσμα είναι να εμφανιστεί μια σελίδα με αρκετές λεπτομέρειες για την επιλεγμένη υποτροφία ώστε ο χρήστης να δει όλα τα στοιχεία (δικαιολογητικά, προαπαιτούμενα κ.α.)

# Κεφάλαιο 5

## Έλεγχος και επαλήθευση της εφαρμογής

### 5.1 Ανάλυση Ερωτηματολογίου Χρησιμότητας – Λειτουργικότητας από Μαθητές

Για την χρησιμότητα της εφαρμογής, τα κριτήρια αναζήτησης αλλά και το σύστημα συστάσεων δημιουργήθηκαν δύο ερωτηματολόγια τα οποία κοινοποιήθηκαν μέσω Facebook προς συμπλήρωση.

#### 5.1.1 1<sup>ο</sup> Ερωτηματολόγιο Χρησιμότητας Υποψήφιων Υποτρόφων

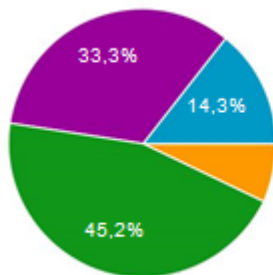
Συνολικά συμπληρώθηκαν 42 ερωτηματολόγια στην πλειοψηφία τους από άνδρες με μεγάλο ποσοστό στις ηλικίες 35-55. Πράγμα αρκετά φυσιολογικό αφού η κοινοποίηση έγινε από Facebook και απευθύνεται σε γνωστούς και φίλους συνομηλίκους ή συνεργάτες από τον εργασιακό χώρο. Συγκεντρωτικά ακολουθούν οι απαντήσεις που ελήφθησαν :

Δημογραφικά Στοιχεία

#### A1. Αρχικά Ονοματεπώνυμου

Δεν εμφανίζεται.

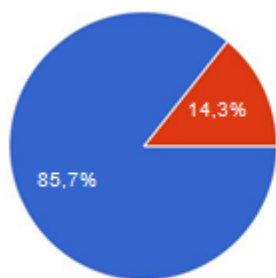
#### A2. Η ηλικία σας



< 19	0	0%
19-24	0	0%
25-34	3	7.1%
35-44	19	45.2%
45-54	14	33.3%
55-64	6	14.3%
65+	0	0%



### Α3. Φύλλο



Ανδρας	36	85.7%
Γυναίκα	6	14.3%

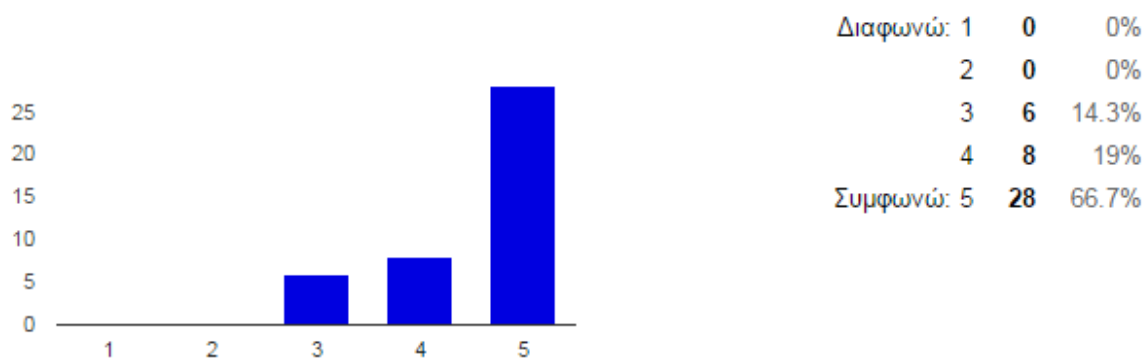
### Α4. Χώρα κατοικίας



Ελλάδα	42	100%
Κύπρος	0	0%
Άλλο	0	0%

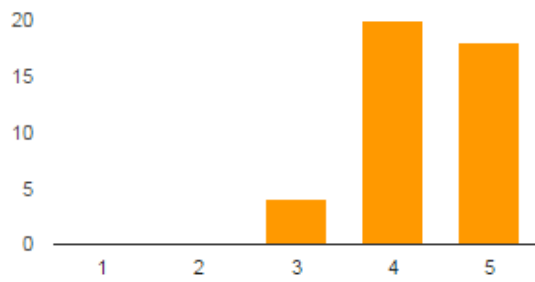
### Χρησιμότητα Εφαρμογής

Β1. Κατά πόσο συμφωνείτε ή διαφωνείτε με την δημιουργία μιας εφαρμογής κινητών συσκευών η οποία θα σας δίνει την δυνατότητα εύρεσης των διαθέσιμων υποτροφιών.



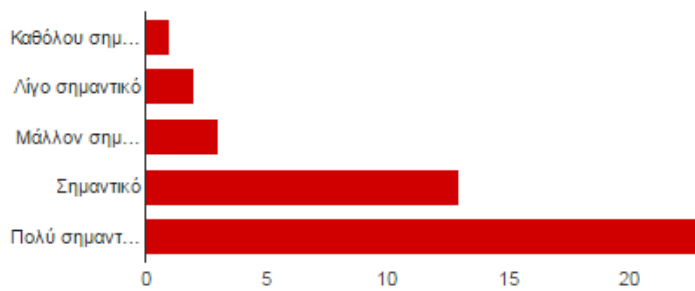
Διαφωνώ: 1	0	0%
2	0	0%
3	6	14.3%
4	8	19%
Συμφωνώ: 5	28	66.7%

Β2. Θεωρείτε ότι θα σας διευκόλυνε στην αναζήτηση σας από τον παραδοσιακό τρόπο αναζήτησης μέσω του Google.



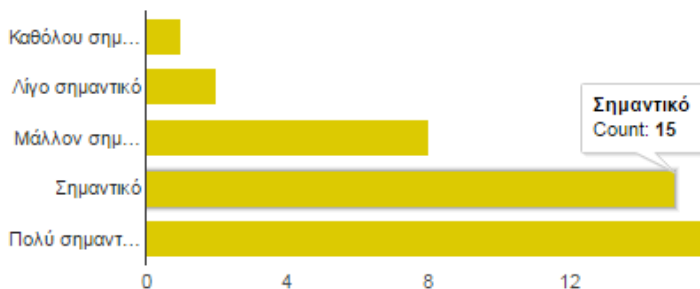
Διαφωνώ:	1	0	0%
	2	0	0%
	3	4	9.5%
	4	20	47.6%
Συμφωνώ:	5	18	42.9%

Ίδρυμα Σπουδών [C1. Βαθμολογήστε τα παρακάτω κριτήρια αναζήτησης σε σχέση με την σημαντικότητα τους.]



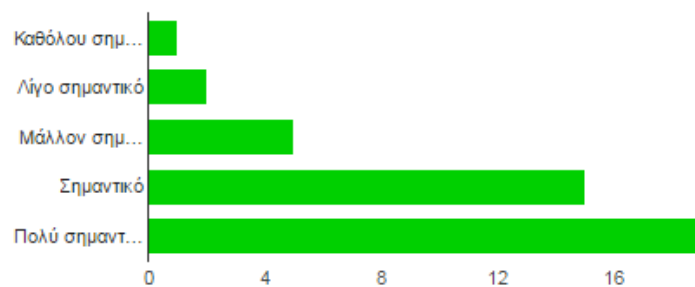
Καθόλου σημαντικό	1	2.4%
Λίγο σημαντικό	2	4.8%
Μάλλον σημαντικό	3	7.1%
Σημαντικό	13	31%
Πολύ σημαντικό	23	54.8%

Απόσταση από τόπο διαμονής [C1. Βαθμολογήστε τα παρακάτω κριτήρια αναζήτησης σε σχέση με την σημαντικότητα τους.]



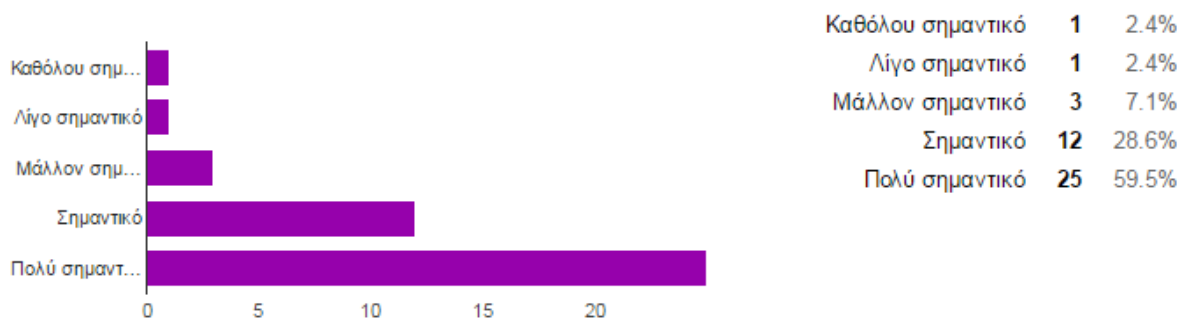
Καθόλου σημαντικό	1	2.4%
Λίγο σημαντικό	2	4.8%
Μάλλον σημαντικό	8	19%
Σημαντικό	15	35.7%
Πολύ σημαντικό	16	38.1%

Ποσό υποτροφίας [C1. Βαθμολογήστε τα παρακάτω κριτήρια αναζήτησης σε σχέση με την σημαντικότητα τους.]

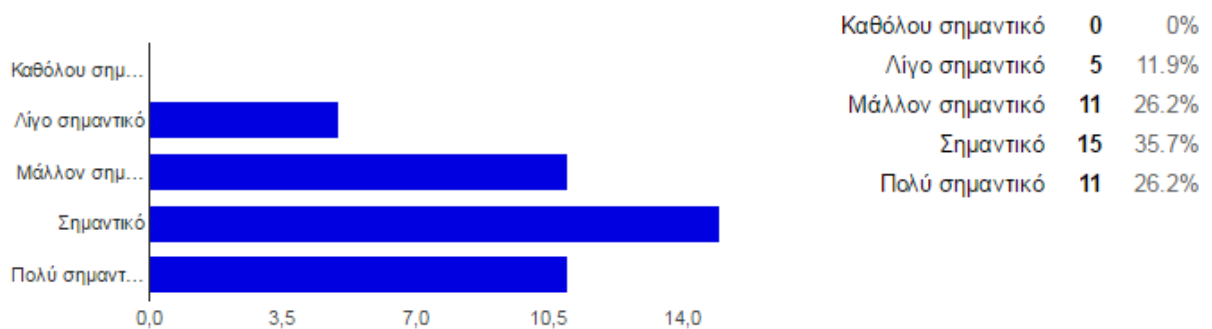


Καθόλου σημαντικό	1	2.4%
Λίγο σημαντικό	2	4.8%
Μάλλον σημαντικό	5	11.9%
Σημαντικό	15	35.7%
Πολύ σημαντικό	19	45.2%

Κλάδος Σπουδών [C1. Βαθμολογήστε τα παρακάτω κριτήρια αναζήτησης σε σχέση με την σημαντικότητα τους.]



Απόσταση από τόπο διαμονής [C1. Βαθμολογήστε τα παρακάτω κριτήρια αναζήτησης σε σχέση με την σημαντικότητα τους.]



C2. Προτείνετε κριτήρια που θεωρείτε σημαντικά να περιλαμβάνονται στην εφαρμογή και δεν αναφέρονται στην προηγούμενη ερώτηση.

ηλικιακά κριτήρια

Αριθμός διαθέσιμων υποτροφιών σε κάθε ίδρυμα

Διάρκεια σπουδών

Διάρκεια σπουδών που καλύπτει η υποτροφία & συνολικά έτη φοίτησης.

1. Παγκόσμια κατάταξη ιδρύματος γενικά και ανά κλάδο σπουδών. 2. Δείκτης απασχόλησης μετά το τέλος των σπουδών. 3. Πιθανά κριτήρια επιλεξιμότητας

Διάρκεια σπουδών, προϋποθέσεις, κριτήρια αξιολόγησης

Περίοδος Σπουδών

Δίδακτρα

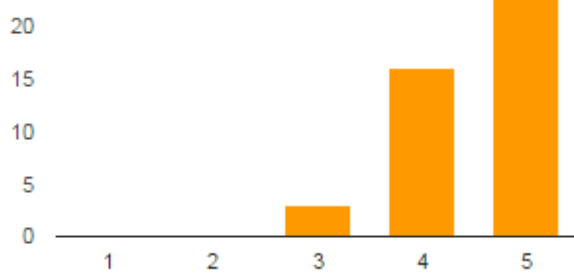
Χρόνος σπουδών

Χρονική διάρκεια προγράμματος σπουδών

Κανένα

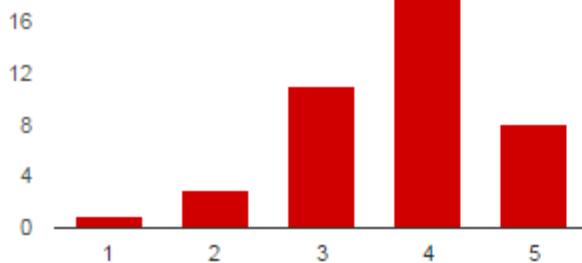
Διάρκεια σπουδών.

C3. Κατά πόσο συμφωνείτε ή διαφωνείτε με την αναγκαιότητα εμφάνισης των προαπαιτούμενων κριτηρίων κατάθεσης αίτησης.



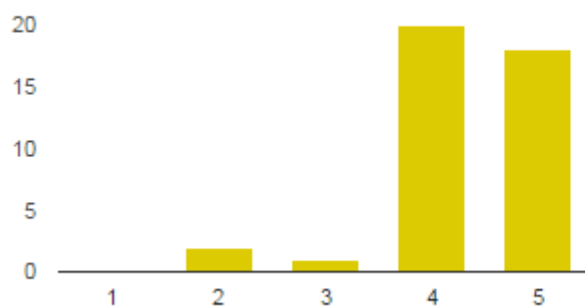
Διαφωνώ:	1	0	0%
	2	0	0%
	3	3	7.1%
	4	16	38.1%
Συμφωνώ:	5	23	54.8%

C4. Κατά πόσο συμφωνείτε ή διαφωνείτε με την δυνατότητα της εφαρμογής να μπορεί να πραγματοποιεί αναζήτηση βασισμένη σε Σύστημα Συστάσεων (Recommendation System).



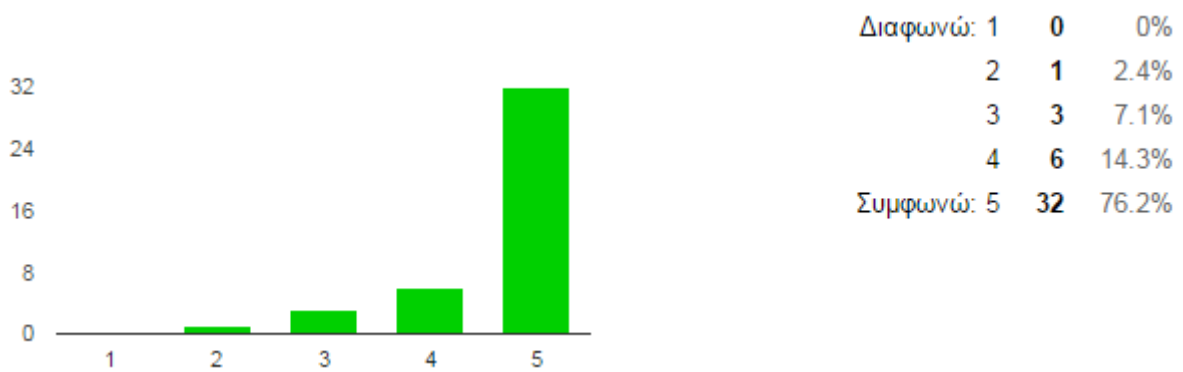
Διαφωνώ:	1	1	2.4%
	2	3	7.1%
	3	11	26.2%
	4	19	45.2%
Συμφωνώ:	5	8	19%

C5. Θεωρείτε ότι πρέπει να υπάρχει δυνατότητα στην εφαρμογή Εκδήλωσης ενδιαφέροντος.



Διαφωνώ:	1	0	0%
	2	2	4.9%
	3	1	2.4%
	4	20	48.8%
Συμφωνώ:	5	18	43.9%

C6. Θεωρείτε ότι η εφαρμογή πρέπει να παρέχει την δυνατότητα ενημέρωσης μέσω email για νέες υποτροφίες των χρηστών που το επιθυμούν.



C7. Προτείνετε κάποια πρόσθετη λειτουργικότητα που θεωρείτε σκόπιμο να υπάρχει στην εφαρμογή.

Δεν έχω να προτείνω κάτι επιπλέον

Τρόποι εξόφλησης / ευκολίες πληρωμής (υπόλοιπου ποσού σπουδών)

Εκτός από email θα μπορούσε να υπάρξει αποστολή μέσω sms ή whatsapp ή κάποιας άλλης εφαρμογής αποστολής μηνυμάτων

Δυνατότητα αποθήκευσης στοιχείων μου για submit σε φόρμες Δυνατότητα εμφάνισης παλαιών αιτημάτων, αν έχω κάνει δυνατότητα download η save των αποτελεσμάτων μετά από ένα search με κριτήρια

ΝΑ ΥΠΑΡΧΕΙ ΔΥΝΑΤΟΤΗΤΑ ΠΑΡΑΛΛΗΛΗΣ ΕΠΙΛΟΓΗΣ ΕΝΗΜΕΡΩΣΗΣ ΜΕ SMS ΣΥΜΦΩΝΑ ΜΕ ΚΡΙΤΗΡΙΑ ΠΟΥ ΟΡΙΖΕΙ Ο ΧΡΗΣΤΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Μεταπτυχιακά παραπλήσιου υποβάθρου να εμφανίζονται

Alert για υποψήφιους με ενδιαφέρον σε συγκεκριμένα ιδρύματα - κλάδο σπουδών

Κανένα

Οι ερωτούμενοι απάντησαν θετικά για την χρησιμότητα της εφαρμογής (B1) σε ποσοστό 85,7% και στην ερώτηση (B2) για την διευκόλυνση σε σχέση με τους παραδοσιακούς τρόπους αναζήτησης σε ποσοστό 90,5%. Συμφώνησαν σε ποσοστό 92,9% για την εμφάνιση των προαπαιτούμενων για κάθε υποτροφία και τέλος για την ύπαρξη του συστήματος συστάσεων σε ποσοστό 64,2%, εδώ φαίνεται ότι κάποιοι από αυτούς δεν ξέρουν τι είναι τα συστήματα συστάσεων και πιθανότατα θα έπρεπε η ερώτηση να είχε διαμορφωθεί πιο κατάλληλα και περισσότερες διευκρινήσεις.

### 5.1.2 2<sup>ο</sup> Ερωτηματολόγιο Χρησιμότητας Υποψήφιων Υποτρόφων

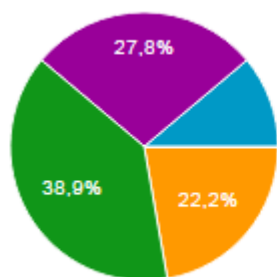
Ο λόγος που αναρτήθηκε και 2<sup>ο</sup> ερωτηματολόγιο χρησιμότητας – λειτουργικότητας ήταν για την τροποποίηση κριτηρίων κατά την αναζήτηση. Συνολικά συμπληρώθηκαν 18 και συγκεντρωτικά ακολουθούν οι απαντήσεις που ελήφθησαν :

Δημογραφικά Στοιχεία

#### A1. Αρχικά Ονοματεπώνυμου

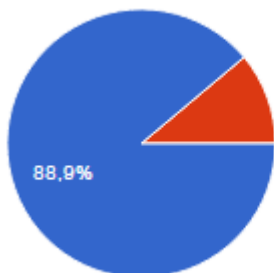
Δεν εμφανίζεται.

#### A2. Η ηλικία σας



< 19	0	0%
19-24	0	0%
25-34	4	22.2%
35-44	7	38.9%
45-54	5	27.8%
55-64	2	11.1%
65+	0	0%

#### A3. Φύλλο



Άνδρας	16	88.9%
Γυναίκα	2	11.1%

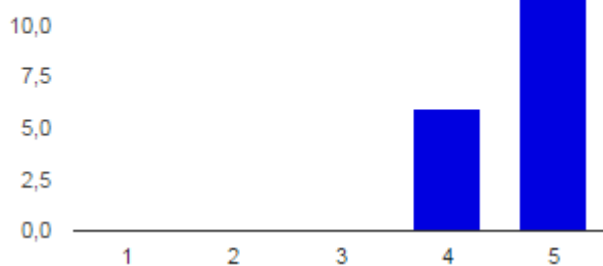
#### A4. Χώρα κατοικίας



Ελλάδα	18	100%
Κύπρος	0	0%
Άλλο	0	0%

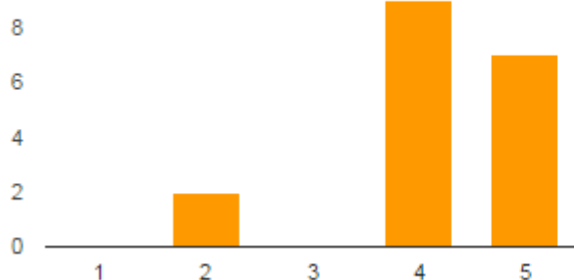
#### Χρησιμότητα Εφαρμογής

B1. Κατά πόσο συμφωνείτε ή διαφωνείτε με την δημιουργία μιας εφαρμογής κινητών συσκευών η οποία θα σας δίνει την δυνατότητα εύρεσης των διαθέσιμων υποτροφιών.



Διαφωνώ:	1	0	0%
	2	0	0%
	3	0	0%
	4	6	33.3%
Συμφωνώ:	5	12	66.7%

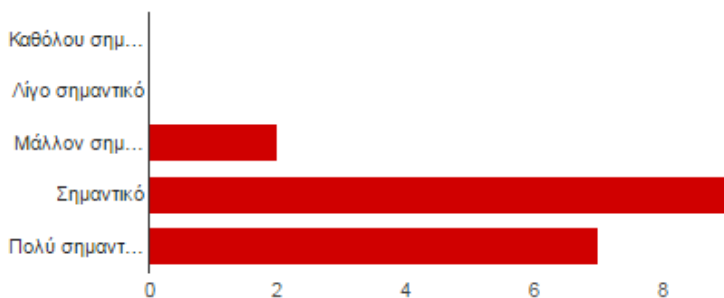
B2. Θεωρείτε ότι θα σας διευκόλυνε στην αναζήτηση σας από τον παραδοσιακό τρόπο αναζήτησης μέσω του Google.



Διαφωνώ:	1	0	0%
	2	2	11.1%
	3	0	0%
	4	9	50%
Συμφωνώ:	5	7	38.9%

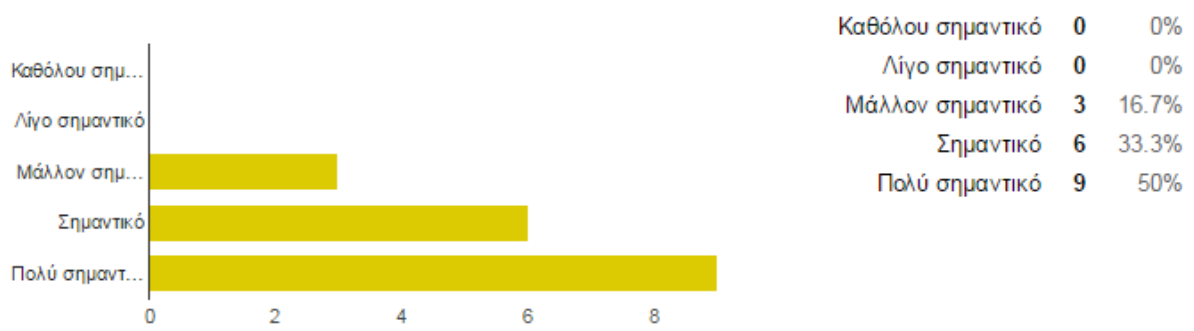
### Λειτουργικότητα Εφαρμογής

Φορέας Υποτροφίας (Πανεπιστήμιο, κολλέγιο, Ίδρυμα) [C1. Βαθμολογήστε τα παρακάτω κριτήρια αναζήτησης σε σχέση με την σημαντικότητα τους.]

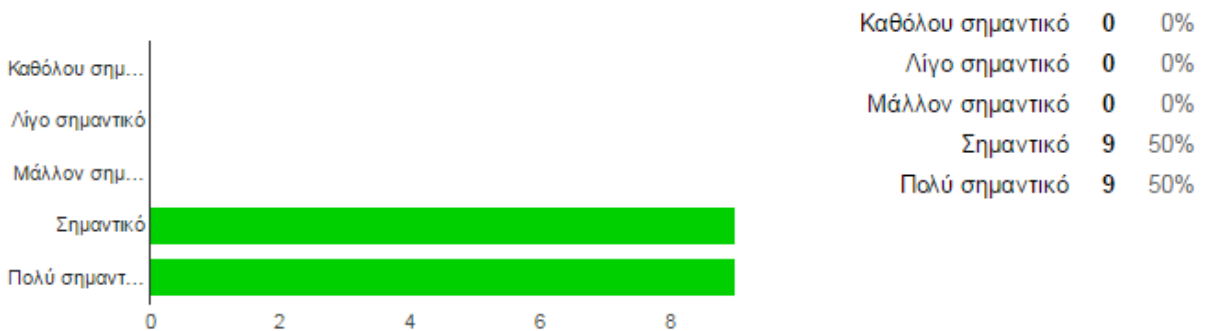


Καθόλου σημαντικό	0	0%
Λίγο σημαντικό	0	0%
Μάλλον σημαντικό	2	11.1%
Σημαντικό	9	50%
Πολύ σημαντικό	7	38.9%

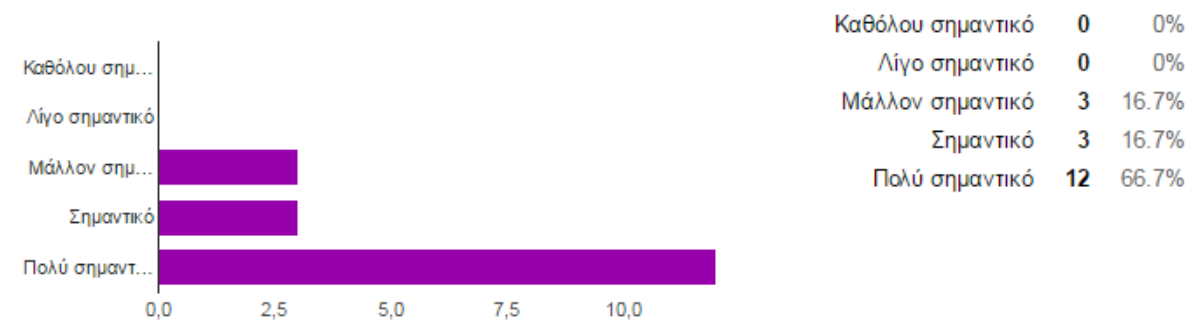
Γλώσσα Προγράμματος Σπουδών (Ελληνικά, Αγγλικά) [C1. Βαθμολογήστε τα παρακάτω κριτήρια αναζήτησης σε σχέση με την σημαντικότητα τους.]



Χώρα Υποτροφίας (Ελλάδα, Κύπρος, Εξωτερικό) [C1. Βαθμολογήστε τα παρακάτω κριτήρια αναζήτησης σε σχέση με την σημαντικότητά τους.]



Επίπεδο Σπουδών Υποτροφίας (Δευτεροβάθμια, Προπτυχιακό, Μεταπτυχιακό, Διδακτορικό, Μεταδιδακτορικό) [C1. Βαθμολογήστε τα παρακάτω κριτήρια αναζήτησης σε σχέση με την σημαντικότητά τους.]



C2. Προτείνετε κριτήρια που θεωρείτε σημαντικά να περιλαμβάνονται στην εφαρμογή και δεν αναφέρονται στην προηγούμενη ερώτηση.

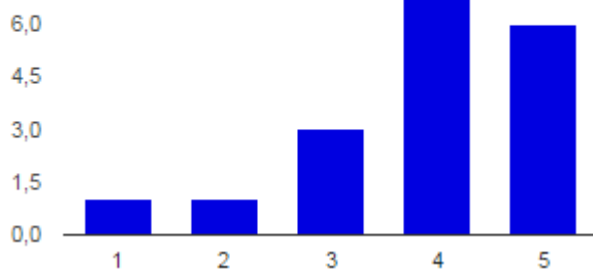
#### ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΣ ΥΠΟΤΡΟΦΙΑΣ

Ύψος υποτροφίας. Συνολική διάρκεια Τύπος υποτροφίας

Τίτλος πτυχίου

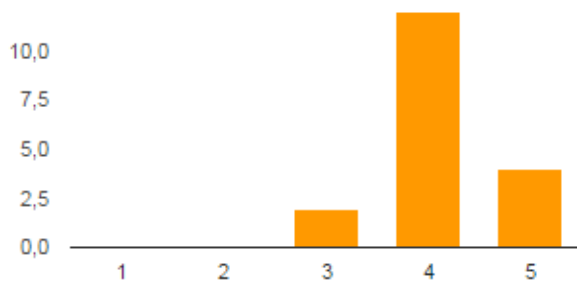
C3. Κατά πόσο συμφωνείτε ή διαφωνείτε με την αναγκαιότητα εμφάνισης των προαπαιτούμενων κριτηρίων κατάθεσης αίτησης.





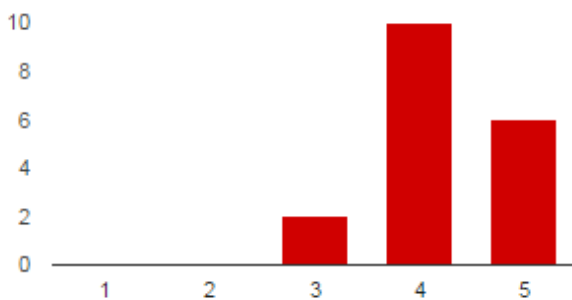
Διαφωνώ:	1	1	5.6%
	2	1	5.6%
	3	3	16.7%
	4	7	38.9%
Συμφωνώ:	5	6	33.3%

C4. Κατά πόσο συμφωνείτε ή διαφωνείτε με την δυνατότητα της εφαρμογής να μπορεί να πραγματοποιεί αναζήτηση βασισμένη σε Σύστημα Συστάσεων (Recommendation System).



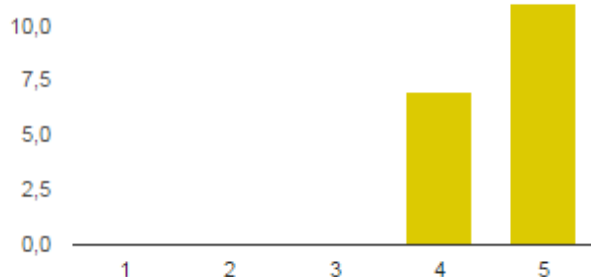
Διαφωνώ:	1	0	0%
	2	0	0%
	3	2	11.1%
	4	12	66.7%
Συμφωνώ:	5	4	22.2%

C5. Θεωρείτε ότι πρέπει να υπάρχει δυνατότητα στην εφαρμογή Εκδήλωσης ενδιαφέροντος.



Διαφωνώ:	1	0	0%
	2	0	0%
	3	2	11.1%
	4	10	55.6%
Συμφωνώ:	5	6	33.3%

C6. Θεωρείτε ότι η εφαρμογή πρέπει να παρέχει την δυνατότητα ενημέρωσης μέσω email για νέες υποτροφίες των χρηστών που το επιθυμούν.



Διαφωνώ: 1	0	0%
2	0	0%
3	0	0%
4	7	38.9%
Συμφωνώ: 5	11	61.1%

C7. Προτείνετε κάποια πρόσθετη λειτουργικότητα που θεωρείτε σκόπιμο να υπάρχει στην εφαρμογή.

#### ΠΡΟΣΩΠΟΠΟΙΗΜΕΝΗ ΕΝΗΜΕΡΩΣΗ ΜΕΣΩ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Μέσο κόστος διαβίωσης ανά χώρα και πανεπιστήμιο

Νομίζω το σημαντικότερο είναι να φτάνει εύκολα και απλά στο κινητό ή το tablet η βασική πληροφορία και στην συνέχεια να γίνεται η ενημέρωση για τις επιπλέον πληροφορίες ώστε να μην είναι δύσχρηστη η εφαρμογή.

Οι ερωτούμενοι απάντησαν θετικά για την χρησιμότητα της εφαρμογής (B1) σε ποσοστό 100% και στην ερώτηση (B2) για την διευκόλυνση σε σχέση με τους παραδοσιακούς τρόπους αναζήτησης σε ποσοστό 88,9%. Συμφώνησαν σε ποσοστό 72,2% για την εμφάνιση των προαπαιτούμενων για κάθε υποτροφία και τέλος για την ύπαρξη του συστήματος συστάσεων σε ποσοστό 88,9%. Στο 2<sup>ο</sup> ερωτηματολόγιο τα κριτήρια αναζήτησης υποτροφιών συγκεντρώνουν μεγαλύτερα ποσοστά σε σχέση με το 1<sup>ο</sup>.

## 5.2 Ανάλυση Ερωτηματολογίου Χρησιμότητας από Εκπαιδευτικά Ιδρύματα

Εδώ υπήρξε σημαντικό πρόβλημα συμμετοχής, από 50 περίπου email που στάλθηκαν σε γραμματείες διαφόρων Πανεπιστημιακών Ιδρυμάτων μόνο δύο απάντησαν το ερωτηματολόγιο.

Παρακάτω φαίνονται οι απαντήσεις που δόθηκαν από τους συμμετέχοντες, αλλά δεν μπορεί να αντληθεί κάποιο ασφαλές συμπέρασμα λόγω της χαμηλής συμμετοχής.

A1. Αρχικά Ονοματεπώνυμου Λειτουργού Ιδρύματος - Κολλεγίου

Δεν εμφανίζεται

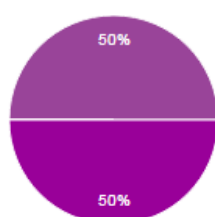
## A2. Χώρα Ιδρύματος



Ελλάδα	2	100%
Κύπρος	0	0%
Άλλο	0	0%

## Επιλογή Ιδρύματος - Κολλεγίου στην Ελλάδα

### A3. Επωνυμία Ιδρύματος - Κολλεγίου



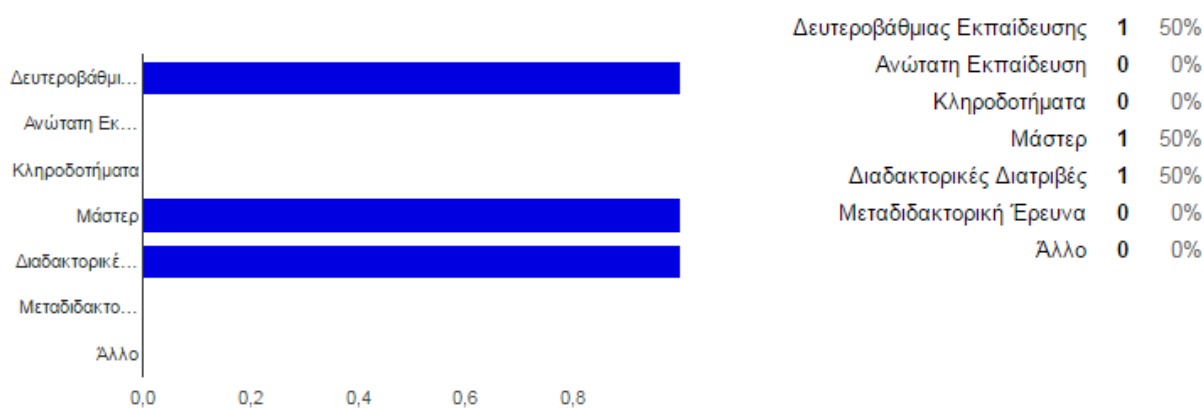
Ανώτατη Σχολή Παιδαγωγικής και Τεχνολογικής Εκπαίδευσης	0	0%
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης	0	0%
Γεωπονικό Πανεπιστήμιο Αθηνών	0	0%
Δημοκρίτειο Πανεπιστήμιο Θράκης	0	0%
Διεθνές Πανεπιστήμιο Ελλάδος	1	50%
Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών	0	0%
Εθνικό Μετσόβιο Πολυτεχνείο	0	0%
Ελληνικό Ανοικτό Πανεπιστήμιο	0	0%
Ιόνιο Πανεπιστήμιο	0	0%
Οικονομικό Πανεπιστήμιο Αθηνών	0	0%
Πανεπιστήμιο Αιγαίου	1	50%
Πανεπιστήμιο Δυτικής Ελλάδας	0	0%
Πανεπιστήμιο Δυτικής Μακεδονίας	0	0%
Πανεπιστήμιο Θεσσαλίας	0	0%
Πανεπιστήμιο Ιωαννίνων	0	0%
Πανεπιστήμιο Κρήτης	0	0%
Πανεπιστήμιο Μακεδονίας	0	0%
Πανεπιστήμιο Πειραιώς	0	0%
Πανεπιστήμιο Πελοποννήσου	0	0%
Πανεπιστήμιο Στερεάς Ελλάδας	0	0%
Πάντειο Πανεπιστήμιο	0	0%
Πολυτεχνείο Κρήτης	0	0%
Στρατιωτική Σχολή Αξιωματικών Σωμάτων	0	0%
Σχολή Ικάρων	0	0%
Σχολή Ναυτικών Δοκίμων	0	0%
Χαροκόπειο Πανεπιστήμιο	0	0%
Άλλο	0	0%

### A4. Σύνδεσμος (URL) Ιδρύματος - Κολλεγίου

[www.aegean.gr](http://www.aegean.gr)

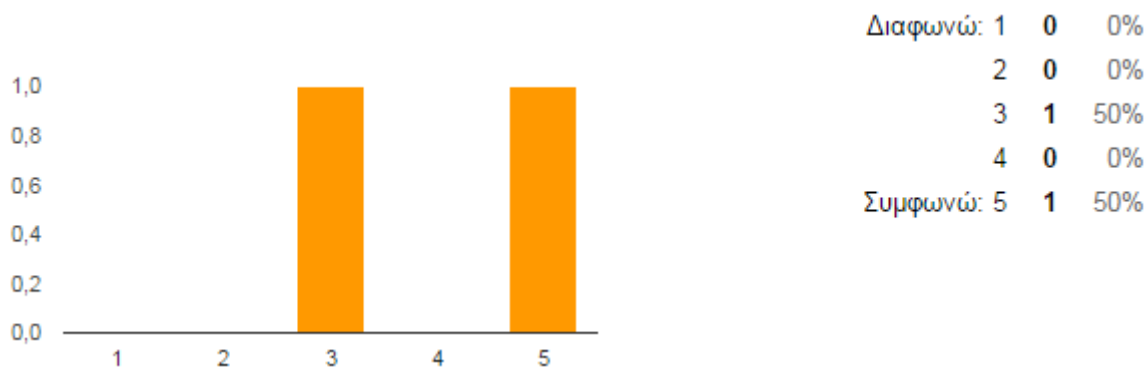
[www.ihu.edu.gr](http://www.ihu.edu.gr)

### A5. Παρεχόμενες κατηγορίες υποτροφιών Ιδρύματος - Κολλεγίου

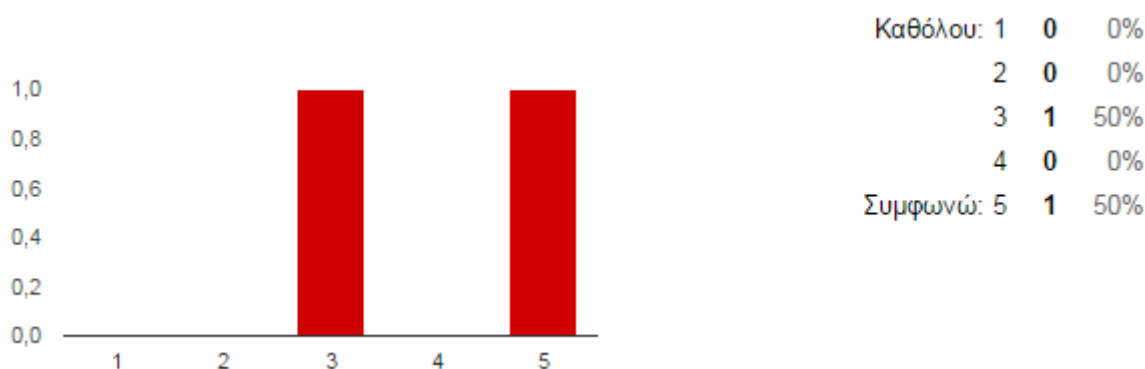


### Χρησιμότητα Εφαρμογής

B1. Κατά πόσο συμφωνείτε ή διαφωνείτε με την δημιουργία μιας εφαρμογής κινητών συσκευών η οποία θα δίνει την δυνατότητα στους υποψήφιους υποτρόφους εύρεσης των διαθέσιμων υποτροφιών.

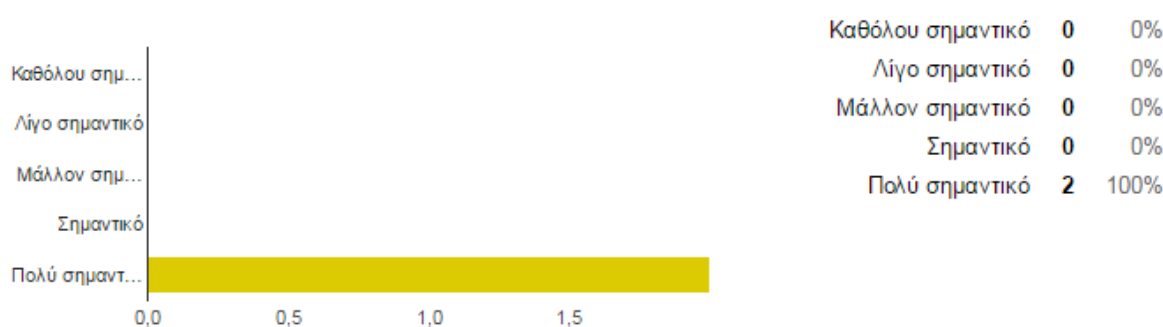


B2. Κατά πόσο συμφωνείτε ή διαφωνείτε με την συμμετοχή του Ιδρύματος - Κολλεγίου στην διαχείριση - ενημέρωση του κεντρικού συστήματος υποτροφιών.

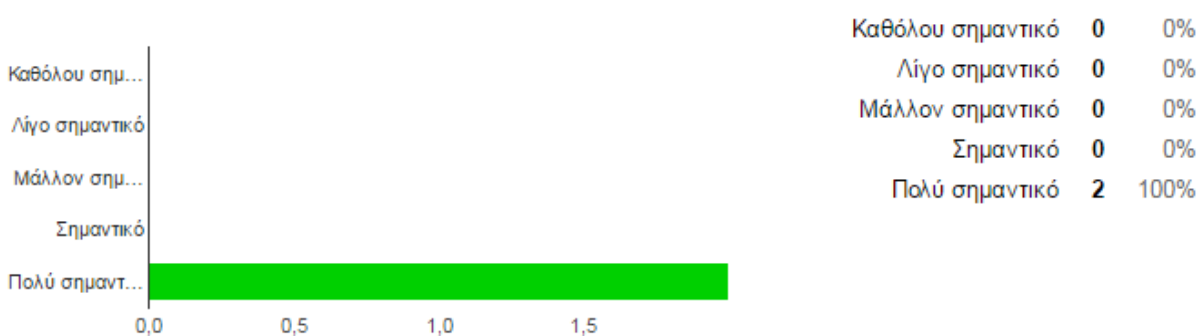


### Σύστημα καταχώρησης - ενημέρωσης υποτροφιών

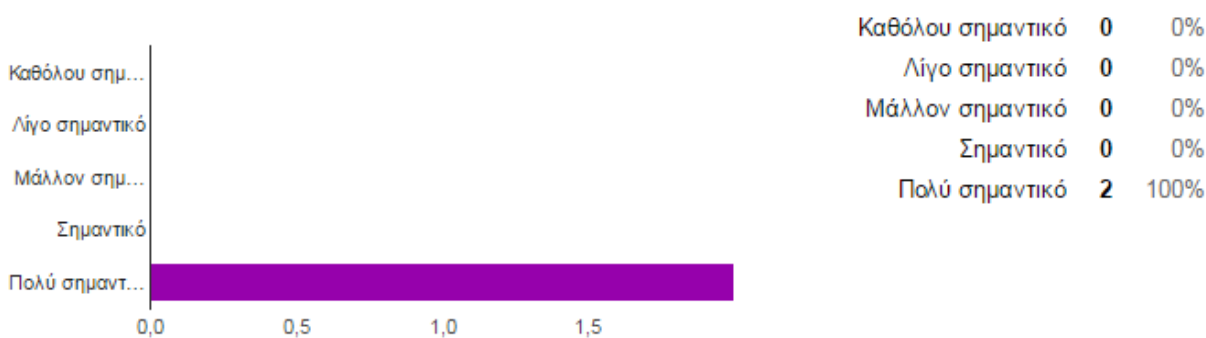
Επωνυμία Ιδρύματος [C1. Βαθμολογήστε τα παρακάτω στοιχεία του Ιδρύματος - Κολλεγίου σε σχέση με την σημαντικότητα τους κατά την διαδικασία Καταχώρησης - Ενημέρωσης στο σύστημα.]



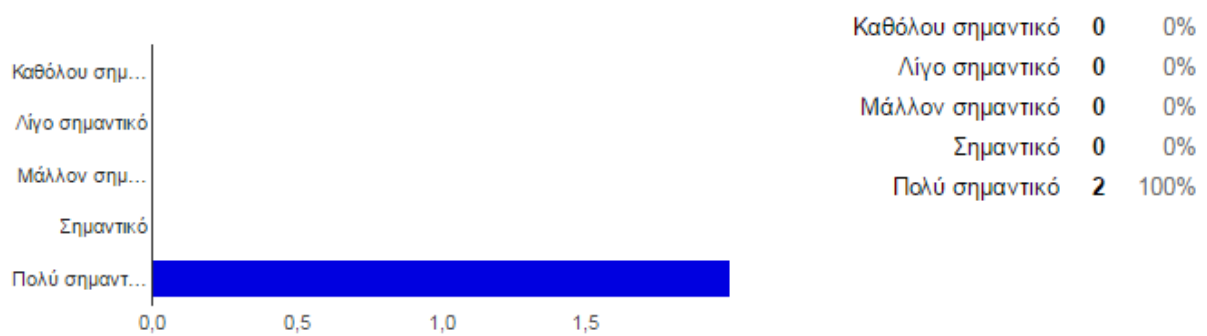
Διεύθυνση Ιδρύματος [C1. Βαθμολογήστε τα παρακάτω στοιχεία του Ιδρύματος - Κολλεγίου σε σχέση με την σημαντικότητα τους κατά την διαδικασία Καταχώρησης - Ενημέρωσης στο σύστημα.]



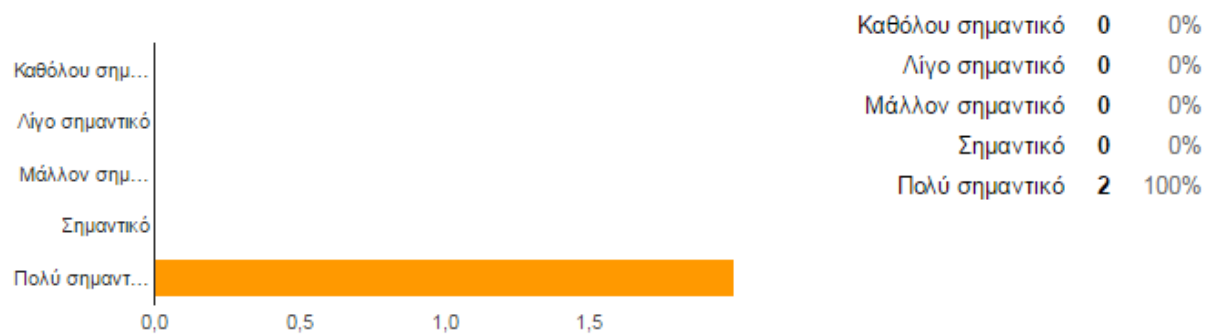
Παρεχόμενες κατηγορίες υποτροφιών [C1. Βαθμολογήστε τα παρακάτω στοιχεία του Ιδρύματος - Κολλεγίου σε σχέση με την σημαντικότητα τους κατά την διαδικασία Καταχώρησης - Ενημέρωσης στο σύστημα.]



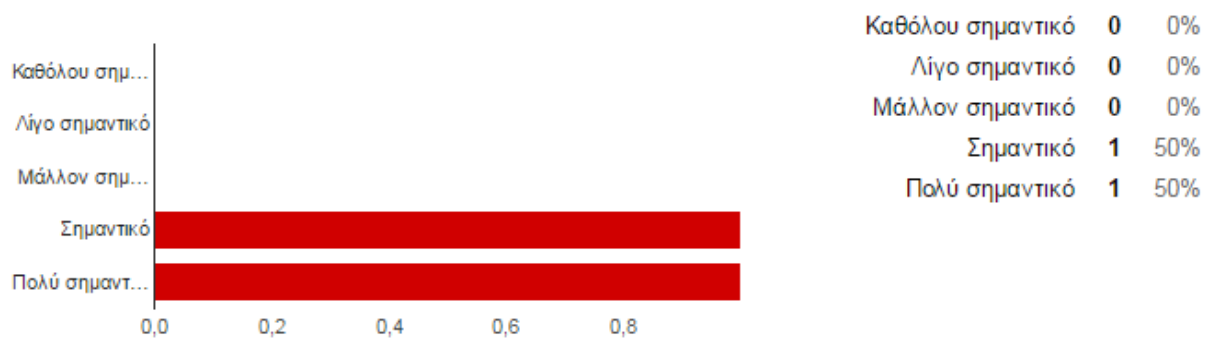
Περιγραφή Υποτροφίας [C1. Βαθμολογήστε τα παρακάτω στοιχεία του Ιδρύματος - Κολλεγίου σε σχέση με την σημαντικότητα τους κατά την διαδικασία Καταχώρησης - Ενημέρωσης στο σύστημα.]



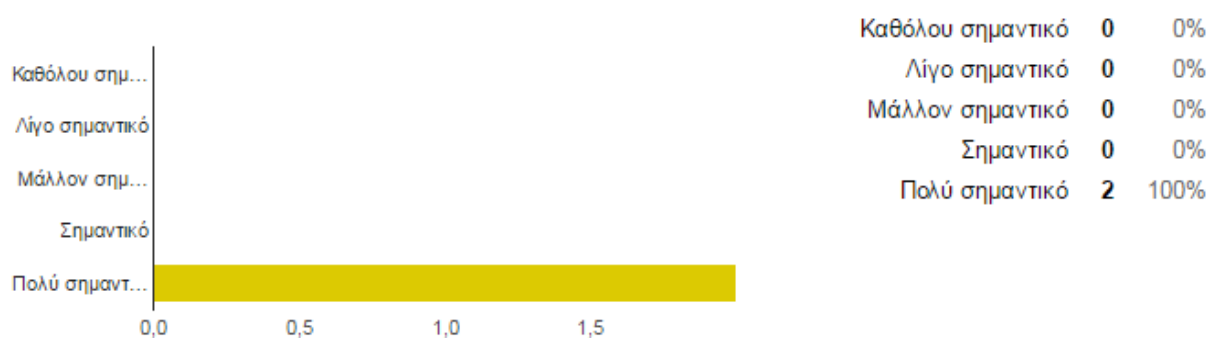
Ημερομηνία Λήξης Αιτήσεων [C1. Βαθμολογήστε τα παρακάτω στοιχεία του Ιδρύματος - Κολλεγίου σε σχέση με την σημαντικότητα τους κατά την διαδικασία Καταχώρησης - Ενημέρωσης στο σύστημα.]



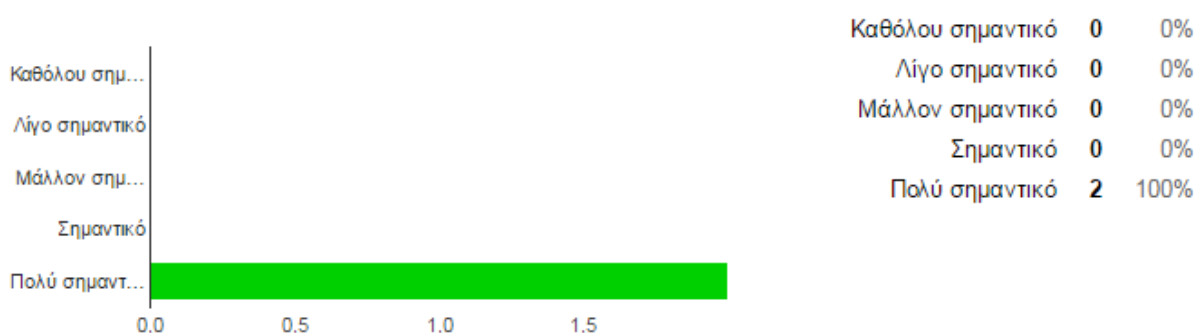
Κλάδος Σπουδών [C1. Βαθμολογήστε τα παρακάτω στοιχεία του Ιδρύματος - Κολλεγίου σε σχέση με την σημαντικότητα τους κατά την διαδικασία Καταχώρησης - Ενημέρωσης στο σύστημα.]



Περίοδος Χρηματοδότησης (Μηνιαία - Εξάμηνη - Ετήσια) [C1. Βαθμολογήστε τα παρακάτω στοιχεία του Ιδρύματος - Κολλεγίου σε σχέση με την σημαντικότητα τους κατά την διαδικασία Καταχώρησης - Ενημέρωσης στο σύστημα.]



Ποσό Χρηματοδότησης [C1. Βαθμολογήστε τα παρακάτω στοιχεία του Ιδρύματος - Κολλεγίου σε σχέση με την σημαντικότητα τους κατά την διαδικασία Καταχώρησης - Ενημέρωσης στο σύστημα.]



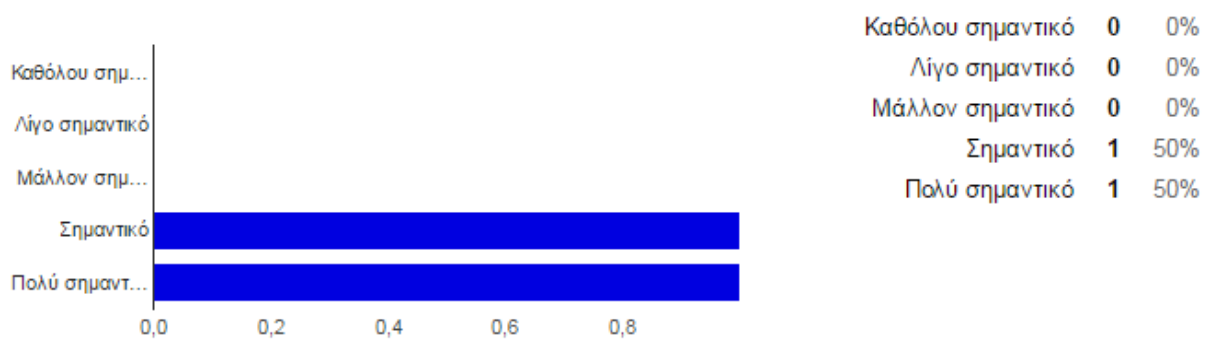
C2. Προτείνετε τα δικά σας στοιχεία Ιδρύματος - Κολλεγίου που θεωρείτε σημαντικό να περιλαμβάνονται κατά την διαδικασία Καταχώρησης - Ενημέρωσης του συστήματος.

-  
κατάταξη Ιδρύματος Μέλη ΔΕΠ ιδρύματος, αριθμός, βαθμίδα Αντικείμενο- καθήκοντα υποτρόφων- παραδοτέα

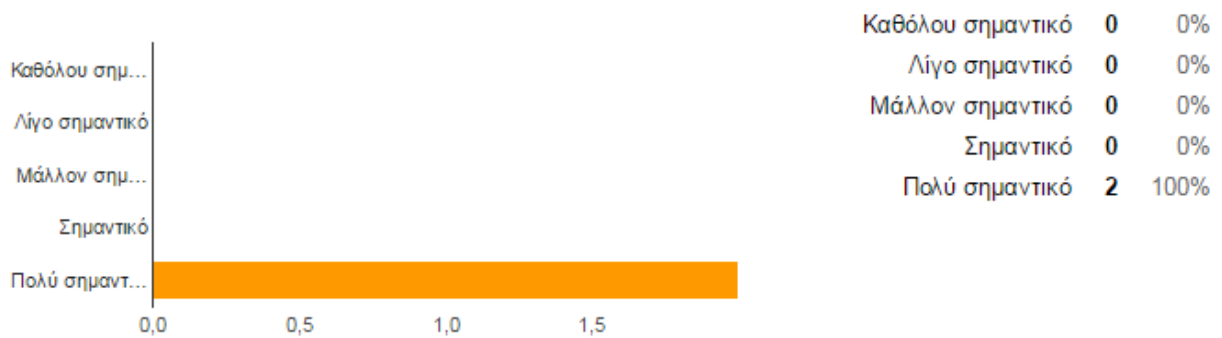
C3. Κατά πόσο συμφωνείτε ή διαφωνείτε με την αναγκαιότητα εμφάνισης των προαπαιτούμενων κριτηρίων κατάθεσης αίτησης.



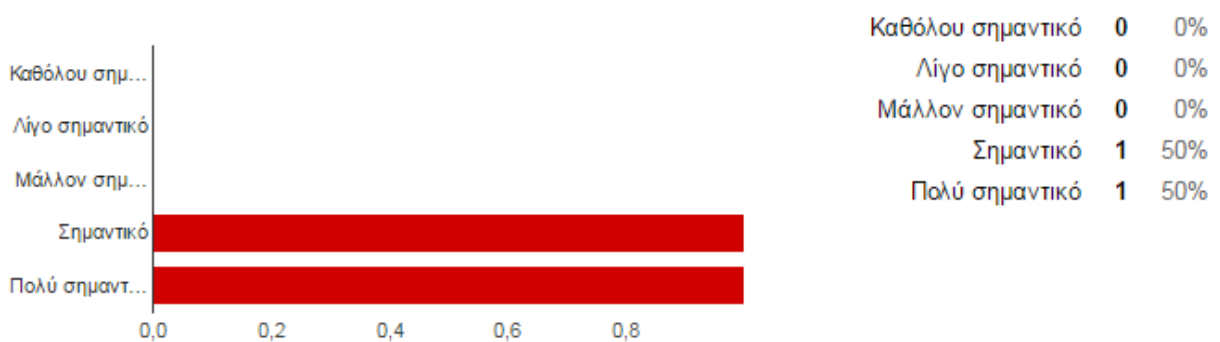
Αριστεία [C4. Βαθμολογήστε τα παρακάτω προαπαιτούμενα στοιχεία μιας υποτροφίας του Ιδρύματος - Κολλεγίου σε σχέση με την σημαντικότητά τους.]



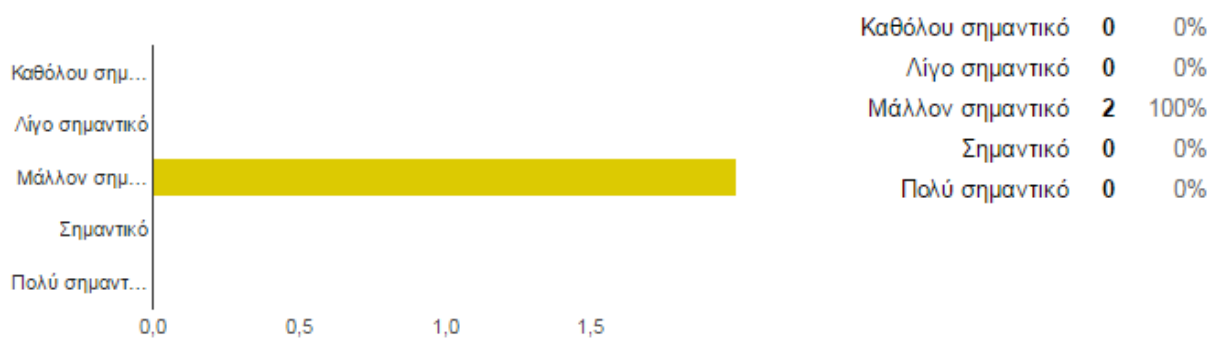
Οικονομικές δυσκολίες [C4. Βαθμολογήστε τα παρακάτω προαπαιτούμενα στοιχεία μιας υποτροφίας του Ιδρύματος - Κολλεγίου σε σχέση με την σημαντικότητά τους.]



Κοινωνικές δυσκολίες [C4. Βαθμολογήστε τα παρακάτω προαπαιτούμενα στοιχεία μιας υποτροφίας του Ιδρύματος - Κολλεγίου σε σχέση με την σημαντικότητά τους.]



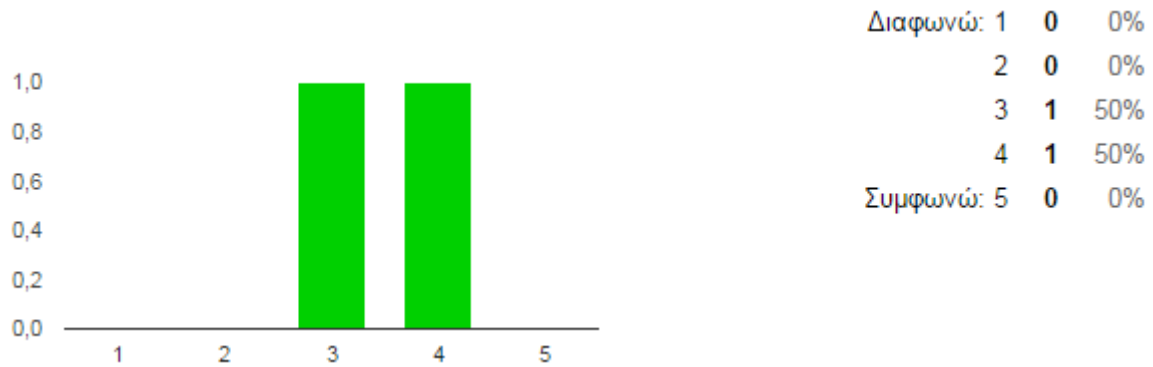
Τόπος καταγωγής [C4. Βαθμολογήστε τα παρακάτω προαπαιτούμενα στοιχεία μιας υποτροφίας του Ιδρύματος - Κολλεγίου σε σχέση με την σημαντικότητά τους.]



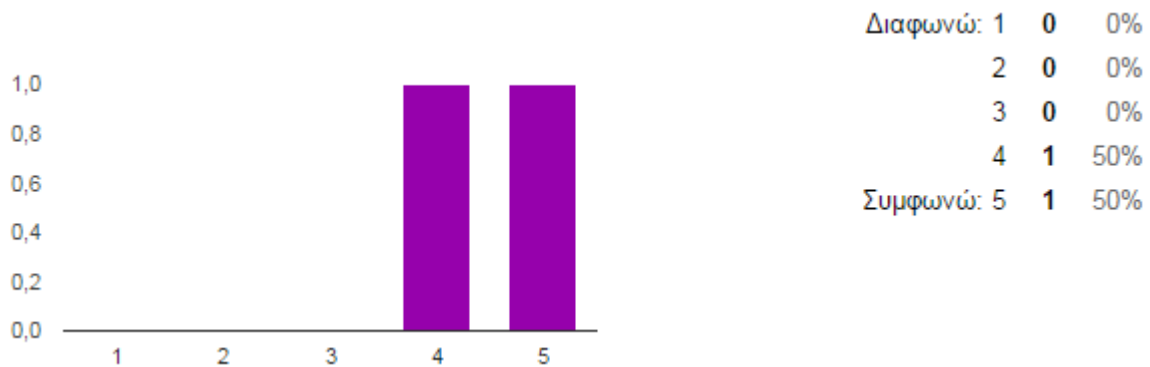
Προτείνετε προαπαιτούμενα στοιχεία μιας υποτροφίας του Ιδρύματος - Κολλεγίου τα οποία δεν αναφέρονται στην προηγούμενη ερώτηση.



C6. Κατά πόσο συμφωνείτε ή διαφωνείτε με την δυνατότητα της εφαρμογής να μπορεί να πραγματοποιεί αναζήτηση βασισμένη σε Σύστημα Συστάσεων (Recommendation System).



C7. Θεωρείτε ότι η εφαρμογή πρέπει να παρέχει την δυνατότητα ενημέρωσης του Ιδρύματος - Κολλεγίου με τους υποψηφίους οι οποίοι έχουν εκδηλώσει ενδιαφέρον για κάποια προσφερόμενη υποτροφία.



C8. Προτείνετε κάποια πρόσθετη λειτουργικότητα που θεωρείτε σκόπιμο να υπάρχει στην εφαρμογή και θα διευκολύνει το Ίδρυμα - Κολλέγιο.

Καμία

### 5.3 Ανάλυση Ερωτηματολογίου Ευχρηστίας

Η παρακάτω ανάλυση Ευχρηστίας της εφαρμογής κινητών συσκευών έδειξε σαφείς αδυναμίες στο User Interface. Πράγματι δόθηκε μεγάλη βάση στην ανάπτυξη του Middleware στο οποίο γίνεται και η περισσότερη δουλειά. Η client εφαρμογή είναι έτσι φτιαγμένη που μπορεί εύκολα να αλλάξει το User Interface ώστε να είναι πιο ελκυστικό στους χρήστες. Σκοπός μας είναι μετά την κατάθεση της παρούσας διατριβής αυτό να

αλλάζει μέχρι και την τελική παρουσίαση λαμβάνοντας υπόψη τις παρατηρήσεις του ερωτηματολογίου ευχρηστίας και την προσθήκη μηχανισμού αποστολής email και hyperlinks στις ιστοσελίδες των υποτροφιών.

## **5.4 Πειραματικές Δοκιμές Συστήματος Συστάσεων**

### **5.4.1 Σενάριο 1<sup>ο</sup>**

Πραγματοποιήθηκε Απλή Αναζήτηση από τον χρήστη «user1» με κριτήρια στο επίπεδο Υποτροφίας την επιλογή «Προπτυχιακά», και λέξη κλειδί «Πληροφορικής», το αποτέλεσμα ήταν η υποτροφία του Πανεπιστημίου Μακεδονίας για 4 προπτυχιακούς φοιτητές (κωδικός 6).

Στην συνέχεια ο χρήστης «user5» πραγματοποίησε Έξυπνη Αναζήτηση με τα ίδια κριτήρια και την ίδια λέξη κλειδί, το σύστημα του πρότεινε την ίδια Υποτροφία.

### **5.4.2 Σενάριο 2<sup>ο</sup>**

Πραγματοποιήθηκε Απλή Αναζήτηση από τον χρήστη «user1» με κριτήρια στο επίπεδο Υποτροφίας την επιλογή «Διδακτορικό», και λέξεις κλειδιά «+Εξόρυξη +γνώσης», το αποτέλεσμα ήταν η υποτροφία του Διεθνές Πανεπιστημίου Ελλάδας για απόκτηση διδακτορικού Διπλώματος (κωδικός 12).

Στην συνέχεια ο χρήστης «user5» πραγματοποίησε Έξυπνη Αναζήτηση με τα ίδια κριτήρια και τις ίδιες λέξεις κλειδιά, το σύστημα του πρότεινε την ίδια Υποτροφία.

Ο λόγος που έπρεπε να γίνουν συγκεκριμένα σενάρια δοκιμών για να φανεί ότι ο αλγόριθμος συστάσεων λειτουργεί επαρκώς είναι γιατί στο σύστημα δεν υπάρχουν καταχωρημένες πολλές υποτροφίες με αποτέλεσμα οι περισσότερες απλές αναζητήσεις να μην φέρνουν αποτελέσματα.

Σε όσα σενάρια δοκιμών πραγματοποιήθηκαν (αφού πρώτα ελέγχαμε ότι οι λέξεις κλειδιά βρισκόντουσαν στα κείμενα των υποτροφιών με τα συγκεκριμένα κριτήρια) τα αποτελέσματα του συστήματος συστάσεων ήταν τα αναμενόμενα.

# Κεφάλαιο 6

## Συμπεράσματα και Επεκτάσεις

### 6.1 Συμπεράσματα

Μπορούμε εύκολα να συμπεράνουμε την αναγκαιότητα μιας εφαρμογής εύρεσης υποτροφιών σε κινητές συσκευές, και μόνο η έλλειψη παρόμοιας εφαρμογής για την Ελλάδα και Κύπρο καθιστά αναγκαία μια τέτοια εφαρμογή, άλλωστε και τα ερωτηματολόγια με πολύ υψηλά ποσοστά αναδεικνύουν την ανάγκη αυτή στο 1<sup>ο</sup> ερωτηματολόγιο της τάξης του 86% ενώ στο 2<sup>ο</sup> 100%. Φυσικά μια τέτοια εφαρμογή εξυπηρετεί περισσότερο τις ανάγκες των υποψηφίων και η γνώμη αυτών περί χρησιμότητας έχει πιο πολύ σημασία από αυτή των Ιδρυμάτων (στα οποία δεν έχουμε και επαρκές δείγμα).

Το σύστημα συστάσεων δεν μπόρεσε να αποδειχθεί από το ερωτηματολόγιο ευχρηστίας κατά πόσο αποδίδει επαρκείς συστάσεις με βάση τις προτιμήσεις άλλων χρηστών. Για τον λόγο αυτό έγιναν πειραματικές δοκιμές με σενάρια προκαθορισμένα ώστε να ανταποκρίνονται στις καταχωρημένες υποτροφίες της Βάσης Δεδομένων. Οι δοκιμές αυτές όπως φαίνεται στην ενότητα 6.4 δείχνουν ότι ο αλγόριθμος συστάσεων αποδίδει ικανοποιητικά σε όλα τα σενάρια δοκιμών.

Το 1<sup>ο</sup> ερευνητικό ερώτημα της διατριβής «του κατά πόσο μια εφαρμογή έξυπνων κινητών συσκευών με τα κατάλληλα κριτήρια μπορεί να συνεισφέρει διευκολύνοντας την εύρεση διαθέσιμων υποτροφιών σε Ελλάδα και Κύπρο» φαίνεται να έχει απαντηθεί θετικά από πολύ μεγάλη πλειοψηφία των ερωτηθέντων.

Όσον αφορά το 2<sup>ο</sup> ερευνητικό ερώτημα «αν μπορεί να υπάρξει ένα μηχανισμός συστάσεων υποτροφιών σε χρήστες της εφαρμογής με παρόμοια ενδιαφέροντα» φαίνεται σαφώς η θετική στάση των ερωτηθέντων, αλλά δεν μπορούμε να είμαστε σίγουροι για το αν ο αλγόριθμος αποδίδει σε όλες τις περιπτώσεις αφού αξιολογήθηκε μόνο με πειραματικές δοκιμές.

Για να μπορούσαμε να καταλήξουμε με ασφάλεια στην απάντηση του 2<sup>ου</sup> ερευνητικού ερωτήματος θα έπρεπε το σύστημα να δουλέψει αρκετό καιρό με πολλές υποτροφίες

καταχωρημένες στην βάση δεδομένων μόνο με απλές αναζητήσεις. Στη συνέχεια από τα στοιχεία που θα συγκεντρώνονταν στην βάση θα μπορούσαμε να εκτελέσουμε πιο εκτεταμένες δοκιμές για το κατά πόσο αποδίδει ο αλγόριθμος συστάσεων.

Το ερωτηματολόγιο ευχρηστίας δεν είχε και τις καλύτερες απαντήσεις για το πόσο φιλική προς τον χρήστη είναι η εφαρμογή. Όμως αυτό δεν ήταν ο κύριος σκοπός της διατριβής αυτής και επιπλέον ο τρόπος σχεδιασμού της εφαρμογής είναι τέτοιος που επιτρέπει με απλές αλλαγές σε xml αρχεία εικόνων, χρωμάτων και μηνυμάτων να αλλάξει δραστικά προς το καλύτερο η εμφάνιση και η ευχρηστία της εφαρμογής.

## 6.2 Μελλοντικές Επεκτάσεις

Όπως είναι εύκολα κατανοητό ένα τέτοιο σύστημα είναι καταδικασμένο να αποτύχει σε περίπτωση που δεν υπάρχει ενεργή συμμετοχή των φορέων που παρέχουν τις υποτροφίες. Μια μελλοντική επέκταση της εφαρμογής είναι η κατασκευή ιστοσελίδας για τους φορείς των υποτροφιών ώστε μέσα από αυτή να γίνεται η προσθήκη – ενημέρωση των υποτροφιών. Η βάση δεδομένων και οι κλάσεις του κεντρικού συστήματος ήδη παρέχουν τις κατάλληλες μεθόδους για χρήση τους από τέτοια ιστοσελίδα, το δύσκολο είναι να ενημερωθούν και να αποδεχθούν οι διάφοροι φορείς την αναγκαιότητα του συστήματος. Αν σκεφτούμε ότι από ένα ερωτηματολόγιο που στάλθηκε μέσω email σε περίπου πενήντα γραμματείες Πανεπιστημιακών Ιδρυμάτων μόνο δύο μπήκαν στον κόπο να συμπληρώσουν το ερωτηματολόγιο καταλαβαίνουμε ότι χρειάζεται ένας τρόπος προσέγγισης και ενημέρωσης όλων αυτών των φορέων ώστε να αποδεχθούν την συγκεκριμένη διαδικασία.

Για το σύστημα συστάσεων που αυτή την στιγμή δεν αφήνει στον χρήστη να βαθμολογήσει μια υποτροφία (αυτό έγινε για λόγους ευχρηστίας) παρά την εκτελεί στο παρασκήνιο με την επιλογή των λεπτομερειών μιας υποτροφίας από τον χρήστη, ίσως θα πρέπει για να αποφευχθούν και οι από λάθος επιλογές των χρηστών να δίνεται η δυνατότητα βαθμολόγησης του πόσο σχετική ήταν η επιλεγμένη υποτροφία βάση του τι πραγματικά έψαχνε ο χρήστης.

# Παράρτημα Α

## SQL Full-Text Searches

### A.1 Η Χρησιμότητα των Full-text Searches

Οι ολοκληρωμένου κειμένου αναζητήσεις ( Full-text searches ) είναι γρηγορότερες από άλλες μεθόδους αναζήτησης όπως η wildcard ή οι βασισμένες στον χαρακτήρα αναζητήσεις, οι οποίες εκτελούνται συνήθως χρησιμοποιώντας την εντολή LIKE της MySQL. Εκτός βέβαια αυτού, οι Full-text searches είναι προτιμότερες και για τους λόγους που αναλύουμε παρακάτω.

- Η ολοκληρωμένου κειμένου αναζήτηση είναι ιδανική για τις εξαιρετικά μεγάλες βάσεις δεδομένων που περιέχουν χιλιάδες ή ακόμα και τα εκατομμύρια σειρές. Οι υπολογισμοί εκτελούνται γρηγορότερα και οι σειρές μπορούν να ταξινομηθούν βασισμένες στη σχετικότητα αναζήτησης, η οποία επιστρέφεται ως δεκαδικός αριθμός από την MySQL.
- Οι λέξεις θορύβου και οποιεσδήποτε λέξεις που είναι 3 χαρακτήρες ή λιγότεροι στο μήκος όπως οι the, and, κ.λπ... αφαιρούνται από την ερώτηση αναζήτησης. Αυτό σημαίνει ότι επιστρέφονται ακριβέστερα αποτελέσματα. Έτσι, εάν ψάξατε για τη φράση "as the people", οι λέξεις που προκαλούν θόρυβο "the" και "as" θα αφαιρεθούν αυτόματα από την αναζήτηση.
- Εκτός από τις απλές αναζητήσεις, οι ολοκληρωμένου κειμένου αναζητήσεις μπορούν επίσης να εκτελεστούν σε Boolean Mode που επιτρέπουν αναζητήσεις βασισμένες σε κριτήρια AND και OR, όπως για παράδειγμα "+person +Mitchell", τα οποία θα επέστρεφαν όλα τα αρχεία που περιέχουν και τις δύο λέξεις person και Mitchell.
- Η ερώτηση είναι case - insensitive, που σημαίνει πως η λέξη "cat" δε διαφέρει σε τίποτα από "CAT" ή από το "AT". Όπως γίνεται άλλωστε κατανοητό, η Full-text-

searching είναι γρήγορη, ισχυρή και έξυπνη εξαλείφοντας την ανάγκη για συγγραφή περίπλοκων αλγορίθμων αναζήτησης.

## A.2 Απαιτήσεις για αναζήτηση πλήρους κειμένου

Για να μπορεί να εκτελεστεί αναζήτηση πλήρους κειμένου θα πρέπει όταν δημιουργείται η βάση δεδομένων να ακολουθηθούν τα παρακάτω τρία βήματα ώστε αργότερα να είναι εφικτή η αναζήτηση με keywords:

- Δημιουργία ενός index τύπου FULLTEXT
- Οι Full-text indexes μπορούν να χρησιμοποιηθούν μόνο σε πίνακες τύπου MyIsam και για στήλες τύπου CHAR, VARCHAR ή TEXT.
- Ο χαρακτηρισμός ενός index ως Full text μπορεί να γίνει κατά τη δημιουργία του πίνακα στη δήλωση CREATE TABLE ή μπορεί να εισαχθεί αργότερα χρησιμοποιώντας είτε την εντολή ALTER TABLE είτε την εντολή CREATE INDEX.

### εντολή MATCH() ...AGAINST

Full-text αναζήτηση πραγματοποιείται χρησιμοποιώντας την εντολή:

**MATCH (col1, col2, ...)**

...

**AGAINST (expr [search\_modifier])**

*search\_modifier* : { IN BOOLEAN MODE | WITH QUERY EXPANSION }

όπου η εντολή MATCH δέχεται μια λίστα από ονόματα στηλών στις οποίες θα γίνει η αναζήτηση του string που ορίζεται στο σώμα της εντολής AGAINST. Προσοχή, στην τελευταία εντολή θα πρέπει να δοθεί αποκλειστικά string και όχι κάποια μεταβλητή ή όνομα στήλης κάποιου πίνακα. Υπάρχουν τρεις τύποι *search\_modifier* και αυτοί είναι οι ακόλουθοι:

Μια αναζήτηση τύπου Boolean ερμηνεύει το string που πρόκειται να αναζητηθεί χρησιμοποιώντας τους κανόνες μιας ειδικής γλώσσας διατύπωσης ερωτήσεων. Το string περιέχει τις λέξεις που θα αναζητηθούν καθώς επίσης και τελεστές (operators) που διευκρινίζουν αν μια λέξη πρέπει να είναι παρούσα ή απύσα στο ταίριασμα των σειρών, ή αν πρέπει να είναι υψηλότερης ή όχι σημασίας από μια άλλη. Μερικοί από αυτούς αναγράφονται στον Πίνακα 2.1. Λέξεις όπως "some " ή "then " είναι stopwords

και απαλείφονται. Αν πρόκειται να εκτελεσθεί μια αναζήτηση τύπου Boolean θα πρέπει να δηλωθεί ως *IN BOOLEAN MODE*.

+	Για AND
-	Για NOT
[χωρίς τελεστές ]	Για OR
> <	Αυξάνουν ή μειώνουν τη σημαντικότητα μιας λέξης

### Πίνακας 3.

Μια αναζήτηση φυσικής γλώσσας ( natural language search) ερμηνεύει το string προς αναζήτηση ως φράση σε φυσική γλώσσα (σε ελεύθερο κείμενο). Δεν υπάρχει κανένας ειδικός τελεστής και η λίστα με τα stopwords ισχύει και εδώ. Οι ολοκληρωμένου κειμένου αναζητήσεις είναι αναζητήσεις που γίνονται σε φυσική γλώσσα και δηλώνονται ως *IN NATURAL LANGUAGE MODE*. Σε περίπτωση που ο χρήστης δε αναφέρει τίποτα στο σώμα της εντολής τότε εξ ορισμού θεωρείται ως natural language search.

Μια αναζήτηση επέκτασης ερώτησης (query expansion search) είναι μια τροποποίηση της αναζήτησης σε φυσική γλώσσα. Το string αναζητείται αρχικά όπως περιγράψαμε στην αναζήτηση με φυσική γλώσσα. Κατόπιν οι λέξεις από τις πιο σχετικές σειρές που επιστρέφονται από την αναζήτηση προστίθενται στην αναζήτηση και η αναζήτηση γίνεται πάλι. Η ερώτηση επιστρέφει τις σειρές από τη δεύτερη αναζήτηση. Για γίνει μια τέτοιου είδους αναζήτηση θα πρέπει να δηλωθεί ως αναζήτηση *IN NATURAL LANGUAGE MODE WITH QUERY EXPANSION* ή *WITH QUERY EXPANSION*.

Εδώ κρίνεται απαραίτητο να σημειωθεί πως οι παραπάνω δυνατότητες προστέθηκαν από την έκδοση MySQL 5.1.7.

Ο τρόπος αναζήτησης που προτιμήθηκε για την ανάπτυξη του συστήματος ανάκτησης πληροφορίας και εγγράφων στην διατριβή αυτή είναι αναζήτηση σε *BOOLEAN MODE*.

#### A.2.1 Ολοκληρωμένου κειμένου αναζητήσεις τύπου Boolean

Ο MySQL μπορεί να εκτελέσει *BOOLEAN* αναζητήσεις πλήρους κειμένου με τη χρήση του modifier *IN BOOLEAN MODE*. Με αυτόν τον modifier, ορισμένοι χαρακτήρες έχουν

ιδιαίτερη σημασία στην αρχή ή στο τέλος των λέξεων στη συμβολοσειρά αναζήτησης. Στο παρακάτω ερώτημα, οι τελεστές + και - δείχνουν ότι μια λέξη απαιτείται να είναι παρών ή απών αντίστοιχα, για να προκύψει μια αντιστοίχιση. Έτσι, το ερώτημα ανάκτησης όλων των γραμμών που περιέχουν τη λέξη "MySQL", αλλά δεν περιέχουν τη λέξη "YourSQL"

**SELECT \* FROM articles WHERE MATCH (title,body)**

**AGAINST ('+MySQL -YourSQL' IN BOOLEAN MODE);**

id	title	Body
1	MySQL Tutorial	DBMS stands for Database ...
2	How To Use MySQL Well	After you went through a ...
3	Optimizing MySQL	In this tutorial we will show ...
4	1001 MySQL Tricks	1. Never run mysqld as root. 2. ...
5	MySQL Security	When configured properly, MySQL ...

#### Πίνακας 4.

Οι BOOLEAN αναζητήσεις πλήρους κειμένου έχουν αυτά τα παρακάτω χαρακτηριστικά:

- Δεν χρησιμοποιούν το όριο του 50%.
- Δεν ταξινομούνται αυτόματα σειρές κατά φθίνουσα σειρά σημασίας. Μπορείτε να το δείτε αυτό από το αποτέλεσμα του προηγούμενου ερωτήματος: Η γραμμή με το υψηλότερο ενδιαφέρον είναι αυτή που περιέχει το "MySQL" δύο φορές, αλλά είναι εισηγμένη τελευταία, δεν είναι η πρώτη.
- Μπορούν να εκτελεστούν ακόμη και χωρίς ένα index FULLTEXT, αλλά η αναζήτηση που εκτελείται με αυτό τον τρόπο θα είναι αρκετά αργή.
- Εφαρμόζονται οι παράμετροι με ελάχιστο και μέγιστο μήκος λέξης.
- Η λίστα stopwords ισχύει.

Η αναζήτηση BOOLEAN full-text Ο λογικός δυνατότητα αναζήτησης πλήρους κειμένου υποστηρίζει τις παρακάτω τελεστές:

+ «Το συν στα αριστερά μια λέξης δείχνει ότι αυτή η λέξη πρέπει να είναι παρούσα σε κάθε σειρά που επιστρέφεται.»



- «Το πλην στα αριστερά μια λέξης δείχνει ότι αυτή η λέξη πρέπει να είναι απύσχα σε κάθε σειρά που επιστρέφεται.»

Σημείωση: Το πλην ενεργεί μόνο για να αποκλείσει σειρές που έχουν διαφορετικά συνδυαστεί με άλλους όρους αναζήτησης. Έτσι, μια αναζήτηση Boolean-mode που περιέχει μόνο όρους - , επιστρέφει ένα κενό αποτέλεσμα. Δεν επιστρέφει "όλες τις γραμμές εκτός από εκείνες που περιέχουν τους εξαιρούμενους όρους."

**Κανέναν τελεστή αριστερά της λέξης** : «Η λέξη είναι προαιρετική, αλλά οι σειρές που την περιέχουν βαθμολογούνται με υψηλότερη βαθμολογία. Αυτό έχει ίδια συμπεριφορά με των MATCH () ... AGAINST() χωρίς τον modifier IN BOOLEAN MODE.»

> < : «Αυτές οι δύο φορείς που χρησιμοποιούνται για να αλλάξετε τη συμβολή μιας λέξης με την αξία ενδιαφέροντος που έχει εκχωρηθεί σε μια σειρά. Ο τελεστής > αυξάνει τη συνεισφορά και το < την μειώνει. Δείτε τα παραδείγματα που ακολουθούν σε αυτή την ενότητα.

() : «Οι παρενθέσεις ομαδοποιούν λέξεις σε υποεκφράσεις. Μπορεί να χρησιμοποιηθούν και για ομαδοποίηση ομάδων.»

~ : «Η περισπωμένη ενεργεί ως τελεστής άρνησης, προκαλώντας τη συμβολή της λέξης στο ενδιαφέρον της σειράς να είναι αρνητική. Αυτό είναι χρήσιμο για τη σήμανση λέξεων με "θόρυβο". Μια σειρά που περιέχει μια τέτοια λέξη έχει βαθμολογηθεί χαμηλότερη από άλλους, αλλά δεν αποκλείεται εντελώς, όπως θα ήταν με τον τελεστή - .»

\* : «Ο αστερίσκος χρησιμεύει ως τελεστής περικοπής (ή wildcard). Σε αντίθεση με τους άλλους τελεστές, θα πρέπει να επισυνάπτεται στη λέξη που επηρεάζει. Οι λέξεις ταιριάζουν, αν υπάρχει η λέξη που προηγείται του τελεστή \*.»

" : «Μια φράση που περικλείεται μέσα σε διπλά εισαγωγικά ( "" ") χαρακτήρων αντιστοιχεί μόνο γραμμές που περιέχουν τη φράση κυριολεκτικά, όπως έχει πληκτρολογηθεί. Η full-text engine χωρίζει τη φράση σε λέξεις και εκτελεί μια αναζήτηση στο ευρετήριο FULLTEXT για τις λέξεις. Non word χαρακτήρες δεν χρειάζεται να ταιριάσουν ακριβώς: Η αναζήτηση φράσης απαιτεί μόνο ότι τα

ταιριάσματα περιέχουν ακριβώς τις ίδιες λέξεις με αυτά της φράσης και με την ίδια σειρά. Για παράδειγμα, το "test phrase" ταιριάζει με το "test, phrase".»

Αν η φράση δεν περιέχει λέξεις που είναι στο ευρετήριο, το αποτέλεσμα είναι κενό. Για παράδειγμα, αν όλες οι λέξεις είναι είτε stopwords ή μικρότερες από το ελάχιστο μήκος των λέξεων του ευρετηρίου, το αποτέλεσμα είναι κενό.

Τα παρακάτω παραδείγματα δείχνουν κάποια search strings που χρησιμοποιούν τελεστές BOOLEAN πλήρους κειμένου:

'apple banana'

Βρίσκει εγγραφές που περιέχουν τουλάχιστον μια από τις δύο λέξεις.

'+apple +juice'

Βρίσκει εγγραφές που περιέχουν και τις δύο λέξεις.

'+apple macintosh'

Βρίσκει εγγραφές που περιέχουν την λέξη "apple", αλλά βαθμολογεί περισσότερο τις εγγραφές που περιέχουν και την λέξη "macintosh".

'+apple -macintosh'

Βρίσκει τις εγγραφές που περιέχουν την λέξη "apple" αλλά δεν περιέχουν την λέξη "macintosh".

'+apple ~macintosh'

Βρίσκει τις εγγραφές που περιέχουν την λέξη "apple", αλλά αν η εγγραφή περιέχει και την λέξη "macintosh", την βαθμολογεί λιγότερο από μια εγγραφή που δεν περιέχει την δεύτερη. Είναι πιο "soft" από την αναζήτηση '+apple -macintosh', για την οποία η παρουσία της λέξης "macintosh" προκαλεί την μη επιστροφή της εγγραφής.

'+apple +(>turnover <strudel)'

Βρίσκει τις εγγραφές που περιέχουν την λέξη "apple" και "turnover", ή "apple" και "strudel" (με όποια σειρά), αλλά βαθμολογεί την "apple turnover" υψηλότερα από την "apple strudel".

'apple\*'

Βρίσκει εγγραφές που περιέχουν λέξεις όπως "apple", "apples", "applesauce", ή "applet".

""some words""

Βρίσκει τις εγγραφές που περιέχουν ακριβώς την φράση "some words" (για παράδειγμα, εγγραφές που περιέχουν την φράση "some words of wisdom" αλλά όχι εγγραφές με την φράση "some noise words").

### **A.3.1 Ολοκληρωμένου κειμένου αναζητήσεις φυσικής γλώσσας**

Μια αναζήτηση Natural Language Full-Text μπορεί να εκτελεστεί είτε επειδή έχει οριστεί ως *IN NATURAL LANGUAGE MODE* είτε επειδή έχει οριστεί εξ ορισμού. Η συνάρτηση MATCH() εκτελεί μια αναζήτηση ενός string, το οποίο δίνεται ως παράμετρος στη συνάρτηση AGAINST() ,πάνω σε μια συλλογή κειμένων . Για κάθε σειρά στον πίνακα της βάσης δεδομένων, η συνάρτηση MATCH() επιστρέφει μια τιμή σχετικότητας του έγγραφου και των λέξεων κλειδιών που έδωσε ο χρήστης .

Εξ ορισμού, η αναζήτηση εκτελείται σε case-insensitive mode . Εντούτοις, μπορεί να εκτελεστεί case-insensitive full-text search με τη χρησιμοποίηση μιας binary collation για τις στήλες . Παραδείγματος χάριν, σε μια στήλη που χρησιμοποιεί το LATIN1 μπορεί να οριστεί ως για να καταστήσει διακριτά τα κεφαλαία και τα μικρά γράμματα.

Κατά την εκτέλεση λοιπόν μιας εντολής που περιέχει τον όρο MATCH() θα επιστραφούν βαθμολογημένα όλα τα σχετικά έγγραφα από το πιο σχετικό προς στο λιγότερο. Οι τιμές σχετικότητας είναι μη αρνητικοί αριθμοί που ανήκουν στο σύνολο των δεκαδικών, με την τιμή 0 να κατέχει την μη σχετικότητα του εγγράφου.

## **A.3 Το φαινόμενο 50% Threshold**

Εκτός από το γεγονός ότι η MySQL αφαιρεί τις λέξεις που προκαλούν θόρυβο και τις σύντομες λέξεις, όπως άλλωστε αναφέρθηκε και προηγουμένως, αφαιρεί και τις λέξεις που εμφανίζονται πάνω από το 50% των σειρών ενός εγγράφου καθώς αντιμετωπίζονται ως stopwords. Έτσι αν για παράδειγμα μια λέξη υπάρχει σε περισσότερο από το 50% των αρχείων που ερευνάται, τότε αυτά τα αρχεία δεν θα πρέπει να επιστραφούν. Μια τέτοια συμπεριφορά είναι επιθυμητή σε μεγάλους πίνακες δεδομένων μιας και τα αποτελέσματα που θα εμφανιστούν θα είναι σχετικά κάτι τέτοιο όμως δεν είναι καθόλου επιθυμητό σε μικρούς πίνακες καθώς μπορεί να μην επιστρέψει σχετικά έγγραφα που τυχόν υπάρχουν ή να επιστρέψει πολλά άσχετα έγγραφα προς το ερώτημα του χρήστη.

# Παράρτημα Β

## UML – Unified Modeling Language

### B.1 Εισαγωγή

Η ενοποιημένη γλώσσα σχεδιασμού (*unified modeling language*) (UML) είναι μια γραφική γλώσσα για την οπτική παράσταση, τη διαμόρφωση προδιαγραφών και την τεκμηρίωση συστημάτων που βασίζονται σε λογισμικό. Η UML στοχεύει στο σχεδιασμό αντικειμενοστραφών συστημάτων. Το σχέδιο είναι μια απλοποιημένη παράσταση της πραγματικότητας.

Σχεδιάζουμε για να μπορέσουμε να καταλάβουμε το σύστημα που αναπτύσσουμε. Έτσι δημιουργώντας σχέδια επιτυγχάνουμε τέσσερις στόχους:

- παριστάνουμε οπτικά το σύστημα που έχουμε ή θέλουμε να κατασκευάσουμε,
- προσδιορίζουμε τη δομή και τη συμπεριφορά του συστήματος,
- δημιουργούμε ένα πρότυπο για να βασίσουμε την κατασκευή του συστήματος,
- τεκμηριώνουμε τις αποφάσεις που λάβαμε.

Σε όλους τους τεχνολογικούς τομείς ο σχεδιασμός βασίζεται σε τέσσερις βασικές αρχές:

- η επιλογή του είδους του σχεδίου έχει επίπτωση στον τρόπο και την μορφή επίλυσης του προβλήματος,
- όλα τα σχέδια εκφράζονται σε διαφορετικές βαθμίδες ακρίβειας,
- τα καλύτερα σχέδια σχετίζονται με την πραγματικότητα,
- ένα είδος σχεδίων δεν είναι ποτέ αρκετό.

Η UML περιλαμβάνει τρία βασικά στοιχεία:

- Οντότητες
- Σχέσεις
- Διαγράμματα

Η UML είναι μια πλήρης και πλούσια γλώσσα με εξαιρετικά ευρύ πεδίο εφαρμογής. Στο μάθημα αυτό θα εξετάσουμε εξαιρετικά συνοπτικά τον τρόπο παράστασης ορισμένων αντικειμενοστραφών δομών σε UML.

## B.2 Διαγράμματα

Η UML ορίζει τα παρακάτω διαγράμματα:

- Διάγραμμα περιπτώσεων χρήσης (*use case diagram*)
- Διαγράμματα δομής
  - Διάγραμμα κλάσεων (*class diagram*)
  - Διάγραμμα αντικειμένων (*object diagram*)
- Διαγράμματα συμπεριφοράς
  - Διάγραμμα καταστάσεων (*statechart diagram*)
  - Διάγραμμα δραστηριοτήτων (*activity diagram*)
  - Διαγράμματα αλληλεπίδρασης
    - Διάγραμμα ακολουθίας (*sequence diagram*)
    - Διάγραμμα συνεργασίας (*collaboration diagram*)
- Διαγράμματα δομής υλοποίησης
  - Διάγραμμα εξαρτημάτων (*component diagram*)
  - Διάγραμμα ανάπτυξης (*deployment diagram*)

# Βιβλιογραφία

- [1].Bell, Charles. (2013) *Expert MySQL*. New York: Apress.
- [2].Bock, H. (2011) *The Definitive Guide to NetBeans Platform 7*. New York: Apress.
- [3].Cinar, Onur. (2012) *Android Apps with Eclipse*. New York: Apress.
- [4].Daniel, D. Lee, Seung, HS. (2001) *Algorithms for Non-negative Matrix Factorization*. In Leen K., Dietterich, T., Tresp, V. *Advances in Neural Information Processing Systems* 13, 556-562. Massachusetts: MIT Press
- [5].Glifton, Ian. (2014) *Implementing Material Design for Developers, Android User Interface Design*. Boston : Addison Wesley
- [6].Google Android. (2016) *ADT Plugin Release Notes*.  
<http://developer.android.com/tools/sdk/eclipse-adt.html#>. [Πρόσβαση : 20.01.2016]
- [7].Google Android. (2016) *Android Supporting Different Languages*.  
<http://developer.android.com/training/basics/supporting-devices/languages.html#>. [Πρόσβαση : 20.02.2016]
- [8].Gotoh, O. (1990). Optimal Sequence Alignment allowing for long gaps. *Bulletin of Mathematical Biology*, Αριθμός Τόμου 52(3), Σελίδες 359-373.
- [9].Harris, Matthew. (2015) *Responsive or Adaptive Design - Which is Best for Mobile Viewing of Your Website?* <http://mediumwell.com/responsive-adaptive-mobile/#>. [Πρόσβαση : 12.02.2016]
- [10]. Juba, S., Vannahme, A., Volkov A. (2015) *Learning PostgreSQL*. Birmingham UK: Packt
- [11]. Karypis, G. (2001) Evaluation of Item-Based Top-N Recommendation Algorithms. *Proceedings of the tenth international conference on Information and knowledge management, CIKM 01*, Σελίδες 247-254.
- [12]. Konstantinidis, S. (2007) Computing the Edit Distance of a Regular Language. *Information and Computation*, Αριθμός Τόμου 205(9), Σελίδες 1307-1316.
- [13]. Lee, DD., Seung, HS. (1999) *Learning the Parts of Objects by Non-Negative Matrix Factorization*. *Nature* 401, 788-791.
- [14]. Miller, F., Vandome, A., McBrewster, J. (2009) *Levenshtein Distance*. Saarbruncken : VDM Publishing

- [15]. Mistry, R., Misner, S. (2014) *Introducing Microsoft SQL Server 2014 Technical Overview*. Redmond, Washington : Microsoft Press
- [16]. Oracle, Docs. (2009) *RESTful Web Services Developer's Guide*. Santa Clara : Sun Microsystems.
- [17]. Orfali, R., Harkey, D., Edwards, J. (2007) *Client/server survival guide*. New Jersey: John Wiley & Sons.
- [18]. Ott, P. (2008) Incremental matrix factorization for collaborative filtering. *Τεύχος 1 του Beiträge zur Wissenschaft, Technologie und Gestaltung*, Hochsch. Anhalt, Presse.
- [19]. Sciparasa Srinivas S. (2013) *JavaScript and JSON Essentials*. Birmingham UK: Packt
- [20]. Seragan, T. (2007) *Programming Collective Intelligence: Building Smart Web 2.0 Applications*. United States California : O'Reilly
- [21]. Singh, I., Leitch, J., Wilson, J. (2008) *Gson User Guide*.  
<https://github.com/google/gson/blob/master/UserGuide.md#>. [Πρόσβαση : 23.01.2016]
- [22]. Smith, James. Android Asynchronous Http Client, A Callback-Based Http Client Library for Android. <http://loopj.com/android-async-http/#>. [Πρόσβαση : 22.01.2016]
- [23]. Takács, G., Pilászy, I., Nemeth, B., Tikk, D. (2008) *Matrix Factorization and Neighbor Based Algorithms for the Netflix Prize Problem*, 2<sup>nd</sup> ACM Conference on Recommendations Systems, , 267-274, Lausanne Switzerland
- [24]. Visveswaran, S. (2000) *Dive into Connection Pooling with J2EE*.  
<http://www.javaworld.com/article/2076221/jndi/dive-into-connection-pooling-with-j2ee.html#> [Πρόσβαση: 15.01.2016]
- [25]. Wargo, John. (2015) *Apache Cordova 4 Programming*. Boston : Addison Wesley
- [26]. Zapata, Belen. (2013) *Android Studio Application Development*. Birmingham UK: Packt