



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

**Εφαρμογή γεωαναφορών και σχεδιασμού διαδρομών για  
την διαχείριση στατιστικών ερευνών**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

**ΚΩΣΤΟΥΛΑ ΝΙΚΟΛΑΟΥ**

**Επιβλέπων :** Τιμολέων Σελλής  
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2010





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

## Εφαρμογή γεωαναφορών και σχεδιασμού διαδρομών για την διαχείριση στατιστικών ερευνών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

**ΚΩΣΤΟΥΛΑ ΝΙΚΟΛΑΟΥ**

**Επιβλέπων :** Τιμολέων Σελλής  
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 21<sup>η</sup> Οκτωβρίου 2010.

.....  
Τιμολέων Σελλής  
Καθηγητής Ε.Μ.Π.

.....  
Γιάννης Σταύρακας  
Ερευνητής Ι.Π.ΣΥ.Π.

.....  
Θοδωρής Δαλαμάγκας  
Ερευνητής Ι.Π.ΣΥ.Π.

Αθήνα, Οκτώβριος 2010

.....  
**ΚΩΣΤΟΥΛΑΣ ΝΙΚΟΛΑΟΣ**

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Κωστούλας Νικόλαος, 2010

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

## Περίληψη

Ο σκοπός της διπλωματικής εργασίας ήταν η ανάπτυξη εφαρμογής για ομαδοποίηση διευθύνσεων, με βάση τις γεωγραφικές τους συντεταγμένες, και μια σειρά από περιορισμούς και, στην συνέχεια, η κατάταξή τους. Η διπλωματική εργασία βασίζεται σε ένα σενάριο χρήσης που προέρχεται από ένα επιμέρους πρόβλημα της ΕΛ.ΣΤΑΤ. (Ελληνική Στατιστική Αρχή). Η ΕΛ.ΣΤΑΤ. διεξάγει τακτικά έρευνες στις οποίες ένα σύνολο ερευνώμενων μοιράζεται σε έναν αριθμό ερευνητών. Το σύστημα που αναπτύχθηκε διευκολύνει την κατανομή των ερευνώμενων σε ερευνητές και προτείνει μια διαδρομή επίσκεψης σε κάθε ερευνητή τηρώντας τις προτιμήσεις του καθενός σε αριθμό ερευνώμενων και σε δήμους.

Συγκεκριμένα μελετήθηκε τρόπος εύρεσης συντεταγμένων από διευθύνσεις και χρησιμοποιήθηκε η υπηρεσία Google Geocoding. Επίσης, μελετήθηκε το πρόβλημα της ομαδοποίησης διευθύνσεων και χρησιμοποιήθηκε χάρτης για την δυναμική ανάθεση ερευνώμενων σε ερευνητές. Για την πρόταση διαδρομής υλοποιήθηκε ο αλγόριθμος του Χριστοφίδη, ο οποίος λύνει με προσέγγιση  $3/2$  το πρόβλημα του πλανόδιου πωλητή.

Το σύστημα που υλοποιήθηκε μπορεί να χρησιμοποιηθεί για να λύσει προβλήματα παρεμφερή με αυτά της ΕΛ.ΣΤΑΤ. όπου, για παράδειγμα, τη θέση των ερευνητών μπορεί να έχουν φορτηγά και την θέση των ερευνώμενων καταστήματα.

**Λέξεις Κλειδιά:** γεωγραφικές συντεταγμένες, γεωκωδικοποίηση, ερευνητής, ερευνώμενος, στατιστική υπηρεσία, ομαδοποίηση, κατάταξη, πρόβλημα πλανόδιου πωλητή.



## **Abstract**

The scope of this thesis was to develop an application to group addresses based on their coordinates and on a series of constraints and then classify them. The thesis was developed to solve a problem of EL.STAT. (Hellenic Statistical Authority). The EL.STAT. conduct regular surveys. In every survey surveyed are allocated among a number of surveyors. The developed system facilitates the attribution of the surveyed to surveyors and suggests a path for each surveyor bearing in mind his preferences in terms of number of surveyed and location.

Specifically, ways of finding the addresses' coordinates were studied. The application uses the Google Geocoding API. Then, the problem of clustering addresses was studied. A map was used for the dynamic assignment of surveyed to surveyors. The proposed route was carried out by Christofides algorithm, which solves with  $3/2$  approximation the travelling salesman problem.

The application implemented can be used to solve problems similar to that of EL.STAT. For example in the position of the surveyors may be trucks and in the position of the surveyed may be shops.

**Keywords:** coordinates, geocoding, survey, surveyor, statistic authority, clustering, classification, tsp problem.

Στην οικογένειά μου...



## Ευχαριστίες

Θα ήθελα να ευχαριστήσω όλους όσοι με βοήθησαν στην εκπόνηση της διπλωματικής εργασίας.

Ευχαριστώ τον επιβλέποντα Καθηγητή κ. Τιμολέοντα Σελλή, ο οποίος μου έδωσε την ευκαιρία να ασχοληθώ με αυτή την διπλωματική εργασία.

Επίσης θέλω να ευχαριστήσω θερμά τον συν-επιβλέποντα ερευνητή κ Γιάννη Σταύρακα για την πολύτιμη καθοδήγηση του καθ' όλη τη διάρκεια εκπόνησης της διπλωματικής.

Τέλος θέλω να ευχαριστήσω ολόψυχα τους γονείς μου και τα αδέρφια μου Γιώργο και Κατερίνα για την ουσιαστική στήριξή τους καθώς και όσους με στήριξαν κατά την διάρκεια των σπουδών μου.



## Πίνακας περιεχομένων

<b>1</b>	<b>Εισαγωγή.....</b>	<b>1</b>
1.1	Εύρεση και διαχείριση συντεταγμένων από διευθύνσεις.....	1
1.2	Αντικείμενο διπλωματικής.....	2
1.2.1	Περιγραφή προβλήματος.....	2
1.3	Οργάνωση κειμένου.....	3
<b>2</b>	<b>Υπόβαθρο .....</b>	<b>5</b>
2.1	Το πρόβλημα του πλανόδιου πωλητή .....	5
2.2	Ομαδοποίηση .....	6
2.3	Μερκατορική προβολή .....	7
2.4	Εύρεση Συντεταγμένων .....	11
2.5	Χάρτης .....	12
2.6	Διεπαφή για εύρεση αποστάσεων .....	14
<b>3</b>	<b>Ανάλυση Απαιτήσεων Συστήματος.....</b>	<b>15</b>
3.1	Απαιτήσεις συστήματος.....	15
3.2	Περιγραφή Λειτουργιών .....	16
3.2.1	Διαχείριση ερευνητών.....	16
3.2.2	Διαχείριση ερευνώμενων.....	20
3.2.3	Εύρεση συντεταγμένων.....	21
3.2.4	Ανάθεση ερευνώμενων σε ερευνητές.....	23
3.2.5	Ταξινόμηση ερευνώμενων.....	25
3.2.6	Το περιβάλλον της εφαρμογής.....	25
3.3	Διαγράμματα ροής δεδομένων.....	26
3.4	Μοντέλο Οντοτήτων Συσχετίσεων .....	29
<b>4</b>	<b>Σχεδίαση Συστήματος .....</b>	<b>33</b>
4.1	Αρχιτεκτονική.....	33
4.2	Περιγραφή Κλάσεων .....	35
4.2.1	Ερευνητές.....	35
4.2.2	Διευθύνσεις .....	37
4.2.3	Αποτελέσματα.....	39
4.2.4	Κεντρικό Μενού .....	40

4.2.5	<i>Πακέτο geocode</i> .....	41
4.2.6	<i>Πακέτα Print - sql</i> .....	42
4.2.7	<i>Πακέτο tsp</i> .....	42
4.3	<b>Βάση Δεδομένων</b> .....	43
4.4	<b>Κωδικοποίηση αρχείων</b> .....	45
4.4.1	<i>Αρχείο Ερευνώμενων</i> .....	45
4.4.2	<i>Αρχείο πρότασης διαδρομής</i> .....	46
<b>5</b>	<b>Υλοποίηση</b> .....	<b>47</b>
5.1	<b>Ενδιαφέροντα σημεία υλοποίησης</b> .....	47
5.1.1	<i>Εύρεση συντεταγμένων</i> .....	47
5.1.2	<i>Διαδραστικός χάρτης</i> .....	48
5.1.3	<i>Ταξινόμηση ερευνώμενων</i> .....	52
5.1.4	<i>Διαχείριση Βάσης Δεδομένων</i> .....	54
5.2	<b>Πλατφόρμες και προγραμματιστικά εργαλεία</b> .....	55
5.2.1	<i>Πλατφόρμες ανάπτυξης</i> .....	55
5.2.2	<i>Εργαλεία ανάπτυξης συστήματος</i> .....	57
5.3	<b>Εγκατάσταση συστήματος</b> .....	58
5.3.1	<i>Εγκατάσταση Java SE</i> .....	58
5.3.2	<i>Εγκατάσταση MySQL</i> .....	59
5.4	<b>Περιεχόμενα CD</b> .....	62
<b>6</b>	<b>Έλεγχος</b> .....	<b>65</b>
6.1	<b>Μεθοδολογία ελέγχου</b> .....	65
6.2	<b>Αναλυτική παρουσίαση ελέγχου</b> .....	65
6.2.1	<i>Εισαγωγή, προβολή, αλλαγή και διαγραφή ερευνητών</i> .....	66
6.2.2	<i>Εισαγωγή ερευνώμενων από αρχείο</i> .....	66
6.2.3	<i>Συγχώνευση δήμων, προτιμήσεις σε δήμους και ερευνώμενους</i> .....	67
6.2.4	<i>Εμφάνιση στον χάρτη</i> .....	68
6.2.5	<i>Εξαγωγή αποτελεσμάτων</i> .....	68
<b>7</b>	<b>Επίλογος</b> .....	<b>71</b>
7.1	<b>Σύνοψη και συμπεράσματα</b> .....	71
7.2	<b>Μελλοντικές επεκτάσεις</b> .....	72

**8**      **Βιβλιογραφία.....73**

## Εικόνες

Εικόνα 2.1 Περιγραφή μιας θέσης στην γη χρησιμοποιώντας γεωγραφικές συντεταγμένες [Ait09].	8
Εικόνα 2.2 Μερκατορική προβολή. [Lar03]	8
Εικόνα 2.3 Χάρτης Μερκατορικής προβολής. [Ait09]	10
Εικόνα 3.1 Μπλοκ διάγραμμα για την περιγραφή των υποσυστημάτων.	16
Εικόνα 3.2. Παράθυρο προσθήκης ερευνητή. Εικόνα 3.3 Προειδοποίηση σε περίπτωση που δεν δοθεί όνομα και επώνυμο.	17
Εικόνα 3.4. Στοιχεία ερευνητή.	17
Εικόνα 3.5. Επιβεβαίωση διαγραφής.	17
Εικόνα 3.6. Προειδοποίηση , όταν επιχειρείται προβολή στοιχείων ή διαγραφή.	18
Εικόνα 3.7. Συγχώνευση δήμων	19
Εικόνα 3.8. Παράθυρο διαχείρισης ερευνητών. Φαίνονται οι επιλογές που έχει κάνει ο χρήστης.	19
Εικόνα 3.9. Προσθήκη ερευνώμενου 3.10. Προειδοποίηση μη εισαγωγής όλων των απαραίτητων στοιχείων.	21
Εικόνα 3.11. Στοιχεία ερευνώμενου.	21
Εικόνα 3.12. Ο χάρτης με τις geocoded διευθύνσεις μαρκαρισμένες με μπλε καρφίτσες και την επιλεγμένη διεύθυνση με κόκκινη καρφίτσα. Επίσης φαίνονται μια προβληματική διεύθυνση και 4 χωρίς καρφίτσα στον χάρτη.	23
Εικόνα 3.13. Ανάθεση ερευνητών.	25
Εικόνα 3.14 Μοντέλο Οντοτήτων Συσχετίσεων.	31
Εικόνα 4.1 Block Διάγραμμα της εφαρμογής - Οι συνδέσεις της κλάσης JMap.	35
Εικόνα 4.2 Σχεσιακό σχήμα.	44
Εικόνα 4.3 Παράδειγμα αρχείου εισαγωγής ερευνώμενων.	45
Εικόνα 4.4 Παράδειγμα αρχείου προτιμήσεων για κάθε ερευνητή.	46
5.1 Διαδικασία εστίασης.	52
Εικόνα 5.2 Τεχνολογίες Java και APIs [ORA1].	56
5.3 Διαδικασία εγκατάστασης Java SE	58
5.4 Διαδικασία εγκατάστασης MySQL	60
Εικόνα 5.5 Περιεχόμενα CD.	62
Εικόνα 6.1 Νέα έρευνα.	66
Εικόνα 6.2 Εισαγωγή, προβολή και αλλαγή στοιχείων και διαγραφή ερευνητή.	66
Εικόνα 6.3 Εισαγωγή ερευνώμενων από αρχείο.	67

Εικόνα 6.4 Αριστερά: Συγχώνευση δήμων, Δεξιά: Δήλωση προτιμήσεων.....	67
Εικόνα 6.5 Εμφάνιση στον χάρτη. ....	68
Εικόνα 6.6 Κατανομή ερευνώμενων. ....	68
Εικόνα 6.7 Τοποθεσία εξαγωγής αποτελεσμάτων.....	69
Εικόνα 6.8 Προτεινόμενη σειρά επίσκεψης για την Δήμητρα Δ.....	69
Εικόνα 6.9 Προτεινόμενη διαδρομή όπως φαίνεται από το Google Maps.....	70
Εικόνα 6.10 Αναδιαταγμένη διαδρομή.....	70

## Πίνακες

Πίνακας 4.1 Δομή του κώδικα .....	34
Πίνακας 5.1 Τεχνολογίες ανάπτυξης.....	55
Πίνακας 5.2 Δημιουργία Βάσης Δεδομένων.....	62



# 1

## *Εισαγωγή*

### *1.1 Εύρεση και διαχείριση συντεταγμένων από διευθύνσεις*

Μια ιδιαίτερα ενδιαφέρουσα και χρήσιμη υπηρεσία διαθέσιμη στο διαδίκτυο είναι η εύρεση διευθύνσεων σε χάρτες. Η εύρεση διαδρομών και τρόπων μετάβασης στις επιθυμητές περιοχές έχει γίνει μια συνηθισμένη διαδικασία. Το πρόβλημα όμως δυσκολεύει όταν θέλουμε να σχεδιάσουμε διαδρομές που να περιλαμβάνουν περισσότερους από έναν άγνωστους προορισμούς. Και γίνεται αρκετά πιο πολύπλοκο όταν οι προορισμοί πρέπει να χωριστούν σε ομάδες, την καθεμία από τις οποίες θα επισκεφτεί διαφορετικός άνθρωπος. Όπως, για παράδειγμα, όταν 5 σχολικά λεωφορεία πρέπει να παραλάβουν 100 μαθητές ενός σχολείου. Προφανώς, η λύση της εύρεσης των προορισμών για τον καθέναν ξεχωριστά και στην συνέχεια του κατάλληλου χωρισμού τους είναι αρκετά πιο δύσκολη από την απλή υπόθεση.

Η διπλωματική εργασία λοιπόν ασχολείται ακριβώς με αυτό το πρόβλημα, την ομαδοποίηση διευθύνσεων με βάση:

- α. τις χωρικές συντεταγμένες στις οποίες αντιστοιχούν και
- β. μια σειρά από περιορισμούς.

Και σε δεύτερο βαθμό την κατάταξη των διευθύνσεων σε μια προτεινόμενη σειρά.

## **1.2 Αντικείμενο διπλωματικής**

Σκοπός της διπλωματικής εργασίας είναι η υλοποίηση μιας εφαρμογής που να δίνει λύση σε ένα επιμέρους πρόβλημα της ΕΛ.ΣΤΑΤ.(Ελληνική Στατιστική Αρχή). Η ΕΛ.ΣΤΑΤ. διεξάγει τακτικά έρευνες στο πληθυσμό της χώρας για συγκέντρωση δεδομένων σχετικά με διάφορα θέματα (πχ. απογραφή πληθυσμού, χρήση γης , καλλιέργειες , κτηνοτροφία, κτλ). Σε κάθε έρευνα, ένα σύνολο ερευνώμενων μοιράζεται σε έναν αριθμό ερευνητών οι οποίοι τους επισκέπτονται και συμπληρώνουν ερωτηματολόγια με βάση τις απαντήσεις τους. Κάθε ερευνητής μπορεί να αναλάβει ένα συγκεκριμένο αριθμό ερευνώμενων, και έχει προτίμηση σε κάποιες περιοχές.

Η εφαρμογή:

- Δέχεται ως είσοδο έναν κατάλογο από διευθύνσεις ερευνώμενων και έναν κατάλογο ερευνητών.
- Βρίσκει τις χωρικές συντεταγμένες των ερευνώμενων.
- Διευκολύνει την κατανομή των ερευνώμενων σε ερευνητές.
- Για κάθε ερευνητή κατατάσσει τους ερευνώμενους σύμφωνα με μια προτεινόμενη σειρά επίσκεψης.

### **1.2.1 Περιγραφή προβλήματος**

Στα πλαίσια της διπλωματικής εργασίας αναπτύχθηκε εφαρμογή η οποία:

- Διαχειρίζεται ερευνητές και ερευνώμενους. Δέχεται δηλαδή ως είσοδο ερευνητές και τις προτιμήσεις τους και ένα σύνολο ερευνώμενων.

Η εφαρμογή διαχειρίζεται μια βάση δεδομένων που περιέχει τους ερευνητές, τις προτιμήσεις τους, τους ερευνώμενους, τις συσχετίσεις μεταξύ τους και τα αποτελέσματα που εξάγει η εφαρμογή. Έχει έτσι την δυνατότητα να αποθηκεύει όλα τα δεδομένα που λαμβάνει στη διάρκεια της λειτουργίας της.

- Βρίσκει γεωγραφικές συντεταγμένες διευθύνσεων. Η εφαρμογή για να βρει τις συντεταγμένες μιας διεύθυνσης χρησιμοποιεί τη διασύνδεση της υπηρεσίας Google Geocoding.
- Χρησιμοποιεί χάρτη για να δείχνει τις διευθύνσεις. Ο χάρτης σχηματίζεται από στατικές εικόνες της υπηρεσίας Google Static Maps. Ο χάρτης είναι διαδραστικός. Ο χρήστης μπορεί να τον κυλίσει, να αλλάξει επίπεδο εστίασης, να επιλέξει διευθύνσεις πάνω σ' αυτόν, να δει πληροφορίες για υπάρχουσες διευθύνσεις και τέλος να κάνει κατανομή των ερευνώμενων σε ερευνητές.
- Όσον αφορά στην κατανομή ερευνώμενων ο χρήστης μπορεί με γραφικό τρόπο (χρήση του χάρτη – ανταπόκριση σε συμβάντα του ποντικιού) να αναθέσει

ερευνώμενους σε ερευνητές ενώ παράλληλα μπορεί να ελέγχει κατά πόσο τηρούνται οι προτιμήσεις του ερευνητή.

- Κατατάσσει με χρήση αλγορίθμου τους ερευνώμενους για κάθε ερευνητή. Χρησιμοποιήθηκε ο αλγόριθμος του Χριστοφίδη ο οποίος λύνει σε πολυωνυμικό χρόνο το πρόβλημα του πλανόδιου πωλητή με προσέγγιση  $3/2$ . Τα αποτελέσματα του αλγορίθμου εξάγονται σε αρχεία. Ένα για κάθε ερευνητή. Επίσης για κάθε ερευνητή του δίνεται η δυνατότητα να δει την διαδρομή στην τοποθεσία [maps.google.com](https://maps.google.com), γρήγορα και ανά 25 ερευνώμενους, με συνδέσμους που δίνονται στο ίδιο αρχείο των αποτελεσμάτων.

### ***1.3 Οργάνωση κειμένου***

Το επόμενο κεφάλαιο συζητά προβλήματα, υπηρεσίες και μεθοδολογίες που μελετήθηκαν κατά την διάρκεια της διπλωματικής. Το κεφάλαιο 3 αναλύει τις απαιτήσεις λειτουργίας του συστήματος. Η σχεδίαση του συστήματος περιγράφεται στο κεφάλαιο 4. Το κεφάλαιο 5 ασχολείται με ενδιαφέροντα σημεία της υλοποίησης και την διαδικασία εγκατάστασής της. Στο κεφάλαιο 6 ακολουθεί η μεθοδολογία ελέγχου του συστήματος. Στο κεφάλαιο 7 συνοψίζονται τα συμπεράσματα της διπλωματικής. Στο τελευταίο κεφάλαιο δίνεται η βιβλιογραφία και οι ιστοσελίδες που χρησιμοποιήθηκαν για την εκπόνηση της διπλωματικής.



# 2

## *Υπόβαθρο*

Κατά την ανάπτυξη της διπλωματικής μελετήθηκαν προβλήματα, υπηρεσίες και μεθοδολογίες. Κάποια από αυτά αξιοποιήθηκαν για την υλοποίηση της εφαρμογής.

### *2.1 Το πρόβλημα του πλανόδιου πωλητή*

Το πρόβλημα του πλανόδιου πωλητή είναι το παρακάτω δύσκολο πρόβλημα (ανήκει στην κλάση NP-hard): δοθέντων ενός συνόλου πόλεων και των αποστάσεων μεταξύ τους να βρεθεί η μικρότερη δυνατή διαδρομή που περνάει από κάθε πόλη ακριβώς μία φορά.

Το πρόβλημα, που μπορεί να μοντελοποιηθεί με χρήση γράφων, λύνεται με δυναμικό προγραμματισμό. Η λύση του αυτή απαιτεί:  $\text{Space} = (n - 1) * 2^{n-2}$  και είναι πολυπλοκότητας:  $O(n) = n * 2^{n-2}$  [Zach07].

Το γενικό πρόβλημα με γράφους έχει αποδειχτεί ότι δεν επιδέχεται κανένα χρήσιμο παράγοντα προσέγγισης. Στην ειδική περίπτωση όμως που ισχύει η τριγωνική ανισότητα για κάθε 3 κόμβους του (metric TSP), τότε το πρόβλημα ανήκει στην κλάση APX (approximable), δηλαδή μπορεί να βρεθεί λύση που απέχει από την βέλτιστη ένα σταθερό παράγοντα.

Για ευκλείδειο TSP<sup>1</sup> πρόβλημα έχει βρεθεί αλγόριθμος που να το λύνει και να ανήκει στο PTAS(Polynomial Time Approximation Scheme). Δηλαδή μπορεί να προσεγγιστεί σε

---

<sup>1</sup> Ευκλείδειο TSP: για τις αποστάσεις μεταξύ των κόμβων του ισχύει η σχέση του Πυθαγόρα για τις αποστάσεις.

πολυωνυμικό χρόνο με όση ακρίβεια είναι επιθυμητή, με τον χρόνο βέβαια να αυξάνεται γρήγορα όσο αυξάνεται η ακρίβεια [Γαλ09].

Για το metric TSP υπάρχουν αλγόριθμοι [Γαλ09], [Zac06]:

- με σταθερά προσέγγισης 2 και πολυπλοκότητα  $O(m + n \log n)$
- με σταθερά προσέγγισης  $3/2$  και πολυπλοκότητα  $O(n^3)$  [Γαλ09]. Σε αυτή την κατηγορία ανήκει ο αλγόριθμος του Χριστοφίδη, τον οποίο περιγράφουμε αναλυτικά παρακάτω:

Αλγόριθμος του Χριστοφίδη [WP1], [Min08], [Zac06]:

Έστω γράφος  $G$  για τον οποίο θέλουμε να βρούμε την μικρότερη διαδρομή που περνά από κάθε κόμβο του μια φορά.

1. Βρίσκουμε το ελαχίστου κόστους συνδετικό δέντρο  $T$  του γράφου  $G$
2.  $V'$  είναι οι κορυφές περιττού βαθμού στο δέντρο  $T$
3. Βρίσκουμε ένα minimum cost perfect matching  $M$  του  $V'$ .
4. Έστω ο γράφος  $G' = T \cup M$ . Ο  $G'$  έχει κύκλο Euler (γιατί όλες του οι κορυφές έχουν άρτιο βαθμό)
5. Βρίσκουμε έναν τέτοιο κύκλο Euler  $\pi$  στον  $G'$ .
6. Βρίσκουμε την διαδρομή  $C$  που επισκέπτεται τις κορυφές με την σειρά που πρωτοεμφανίζονται στο  $\pi$ .

Ο παραπάνω αλγόριθμος χρησιμοποιήθηκε για να προτείνει σειρά επίσκεψης των ερευνώμενων στους ερευνητές.

Για το ευκλείδειο TSP υπάρχουν διάφοροι αλγόριθμοι PTAS, ο πρώτος από τους οποίους αρχικά αναπτύχθηκε από τον Arora το 1996 [Zac06]. Ο αλγόριθμος αυτός είναι αρκετά θεωρητικός και η περιγραφή του αρκετά μεγάλη και ξεφεύγει από τους σκοπούς της παρούσας διπλωματικής.

## 2.2 Ομαδοποίηση

Η ομαδοποίηση (clustering) είναι πρόβλημα χωρισμού δεδομένων σε ομάδες, ανάλογα με τις ιδιότητες τους. Όσον αφορά χωρικά δεδομένα το πρόβλημα είναι το εξής. Να χωριστούν τα δεδομένα σε έναν αριθμό από ομάδες με τέτοιο τρόπο ώστε κάθε δεδομένο να αντιπροσωπεύεται από το κέντρο της ομάδας του.

Το πρόβλημα είναι εύκολο και επιδέχεται πολλούς αλγόριθμους επίλυσης. Μελετήθηκαν ο K-means και ένας hierarchical αλγόριθμος.

Ο αλγόριθμος K-means αποτελείται από τα παρακάτω 6 βήματα:

1. Δίνουμε τον αριθμό των επιθυμητών ομάδων (clusters)  $k$ .
2. Τυχαίαμαντεύει  $k$  κέντρα των ομάδων.
3. Κάθε δεδομένο βρίσκει σε ποιο κέντρο είναι πιο κοντά.

4. Κάθε κέντρο βρίσκει το κέντρο βάρους των δεδομένων που του ανήκουν.
5. Το κέντρο βάρους τίθεται το νέο κέντρο της ομάδας.
6. Επαναλαμβάνεται η διαδικασία από το βήμα 3 μέχρι να μην γίνονται αλλαγές.

Ο αλγόριθμος δεν δίνει πάντα το καλύτερο αποτέλεσμα. Για να δώσει το βέλτιστο αποτέλεσμα χρησιμοποιούνται διάφοροι τρόποι καθορισμού των κέντρων  $k$  του βήματος 2. Ένας από αυτούς είναι: επιλέγεται το πρώτο τυχαία, το δεύτερο όσο το δυνατόν πιο μακριά από το πρώτο, ... το  $n$  όσο το δυνατόν πιο μακριά από αυτό που είναι πιο κοντά του...

Ο αλγόριθμος hierarchical clustering αποτελείται από τα παρακάτω βήματα:

1. Κάθε σημείο είναι μια ομάδα.
2. Βρίσκει τις πιο κοντινές ομάδες.
3. Τα ενώνει σε μία πατρική ομάδα.
4. Επαναλαμβάνεται από το βήμα 2.

Υπάρχουν πολλοί τρόποι για να αξιολογηθεί αν δυο ομάδες είναι κοντά. Ένας είναι η ελάχιστη απόσταση μεταξύ δύο δεδομένων των clusters. Ένας άλλος η μέγιστη απόσταση. Και ένας τρίτος η μέση απόσταση [Moo1], [Dei].

Στην διπλωματική μας ένας από τους παραπάνω αλγόριθμους θα μπορούσε να χρησιμοποιηθεί για να χωρίζει τους ερευνώμενους σε ομάδες, μία για κάθε ερευνητή που συμμετέχει στην έρευνα.

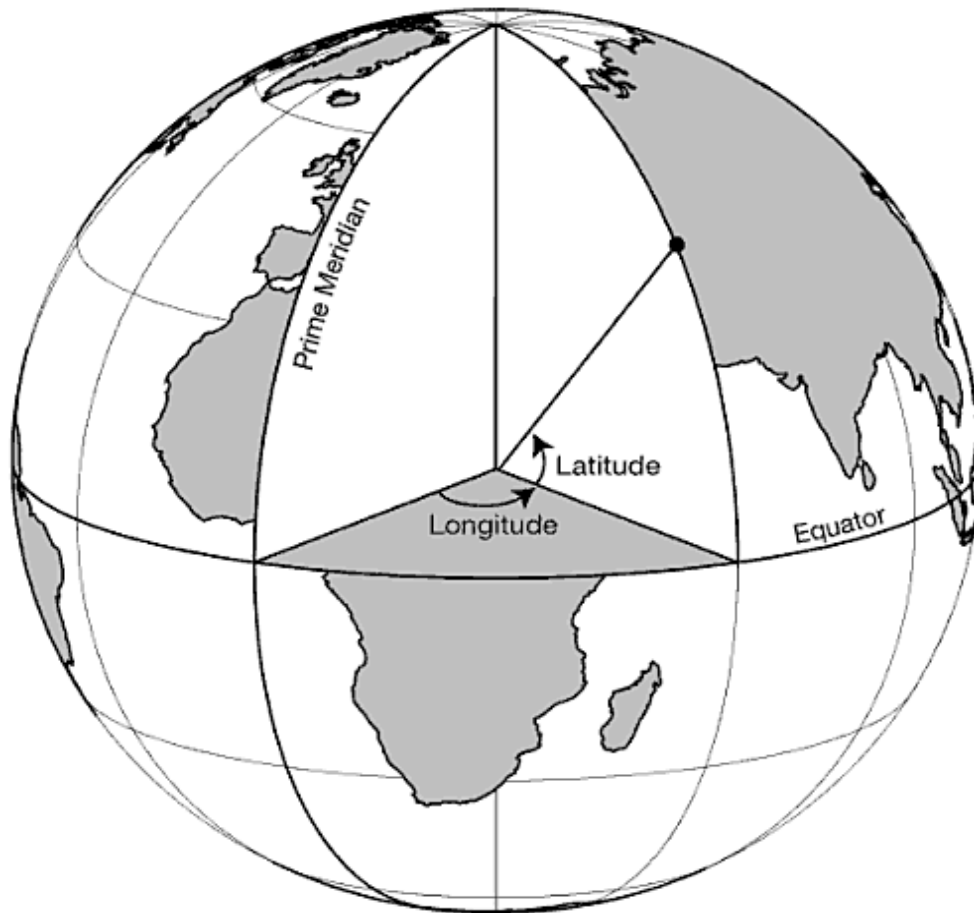
Επειδή όμως ο κάθε ερευνητής έχει προτιμήσεις στον αριθμό των ερευνώμενων και στην τοποθεσία τους, οι παραπάνω αλγόριθμοι πολύ δύσκολα θα μπορούσαν να χρησιμοποιηθούν για να βγάλουν χρήσιμα αποτελέσματα. Γι' αυτό τον λόγο προτιμήθηκε η παροχή της δυνατότητας στον χρήστη να αναθέτει με γραφικό τρόπο ερευνώμενους σε ερευνητές.

Επίσης θα μπορούσε να χρησιμοποιηθεί αλγόριθμος clustering για τον χωρισμό των ερευνώμενων που θα επισκεφτεί ένας ερευνητής σε πολλές μέρες. Κάτι τέτοιο, βέβαια, δεν ήταν στις απαιτήσεις της παρούσας διπλωματικής.

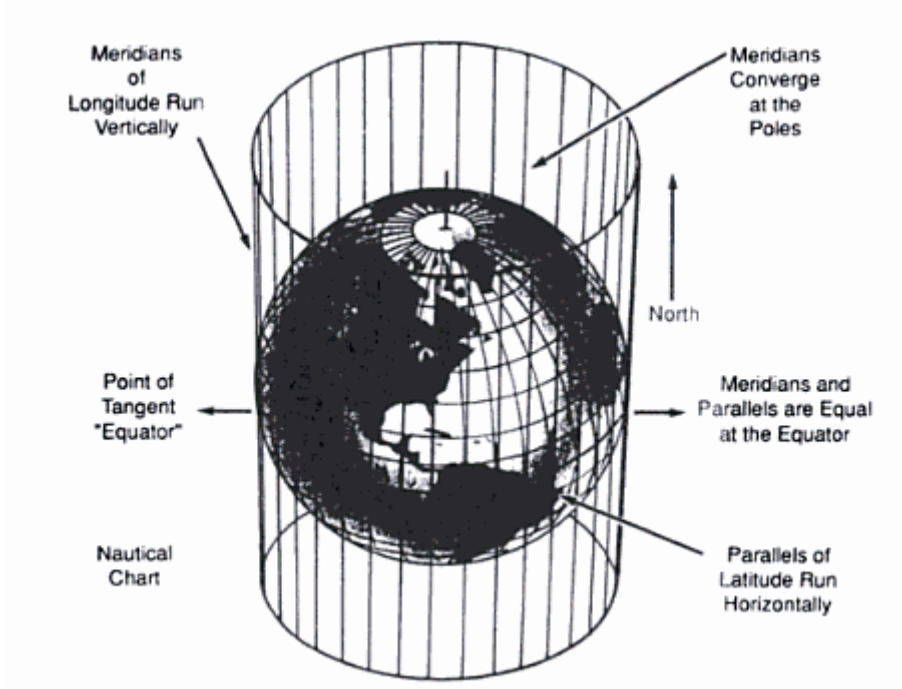
## **2.3 Μερκατορική προβολή**

Σε έναν χάρτη υπάρχουν οριζόντιες και κάθετες γραμμές. Οι κάθετες γραμμές λέγονται γραμμές γεωγραφικού μήκους ή μεσημβρινοί (σημεία πάνω στην γραμμή έχουν ίδιο γεωγραφικό μήκος - longitude) και οι οριζόντιες γραμμές λέγονται γραμμές γεωγραφικού πλάτους (σημεία πάνω στην γραμμή έχουν ίδιο γεωγραφικό πλάτος - latitude).

Αυτά τα δύο είδη γραμμών δημιουργούν ένα παγκόσμιο πλέγμα. Χρησιμοποιούμε αυτές τις γραμμές για να εξηγήσουμε που βρίσκονται συγκεκριμένα σημεία στον πλανήτη μας. Όταν συνδυάζονται ο αριθμός της γραμμής γεωγραφικού πλάτους με τον αριθμό της γραμμής γεωγραφικού μήκους προκύπτουν οι συντεταγμένες του σημείου [Zur05].



Εικόνα 2.1 Περιγραφή μιας θέσης στην γη χρησιμοποιώντας γεωγραφικές συντεταγμένες [Ait09].



Εικόνα 2.2 Μερκατορική προβολή. [Lar03]



Η Μερκατορική προβολή δημιουργείται μετατρέποντας την σφαιρική επιφάνεια της γης σε κύλινδρο. Η διαδικασία φαίνεται στην εικόνα 2.2

Οι μεσημβρινοί συγκλίνουν πάνω στην γη στον βόρειο και στον νότιο πόλο. Επειδή συγκλίνουν στους πόλους οι γραμμές αυτές προβάλλουν πάνω στον κύλινδρο αποστάσεις με ακρίβεια μόνο πάνω στον ισημερινό. Ένα λεπτό της μοίρας (γωνία) γεωγραφικό μήκος είναι ίσο με ένα ναυτικό μίλι μόνο στον ισημερινό. Σε κάθε άλλο σημείο είναι μικρότερο από ένα ναυτικό μίλι.

Το γεωγραφικό μήκος μετριέται μέχρι 180 μοίρες ανατολικά και δυτικά από τον μεσημβρινό που περνά από το Γκρίνουιτς της Αγγλίας. 360 μοίρες συνολικά. Οι μεσημβρινοί συναντώνται σε έναν μεσημβρινό που λέγεται διεθνής γραμμή αλλαγής ημερομηνίας στο αντίθετο σημείο της γης (σε σχέση με τον μεσημβρινό του Γκρίνουιτς). Οπότε το γεωγραφικό μήκος χαρακτηρίζεται είτε ανατολικό [E] ή δυτικό [W]. Κάθε μοίρα γεωγραφικού μήκους έχει 60 λεπτά και κάθε λεπτό 60 δευτερόλεπτα.

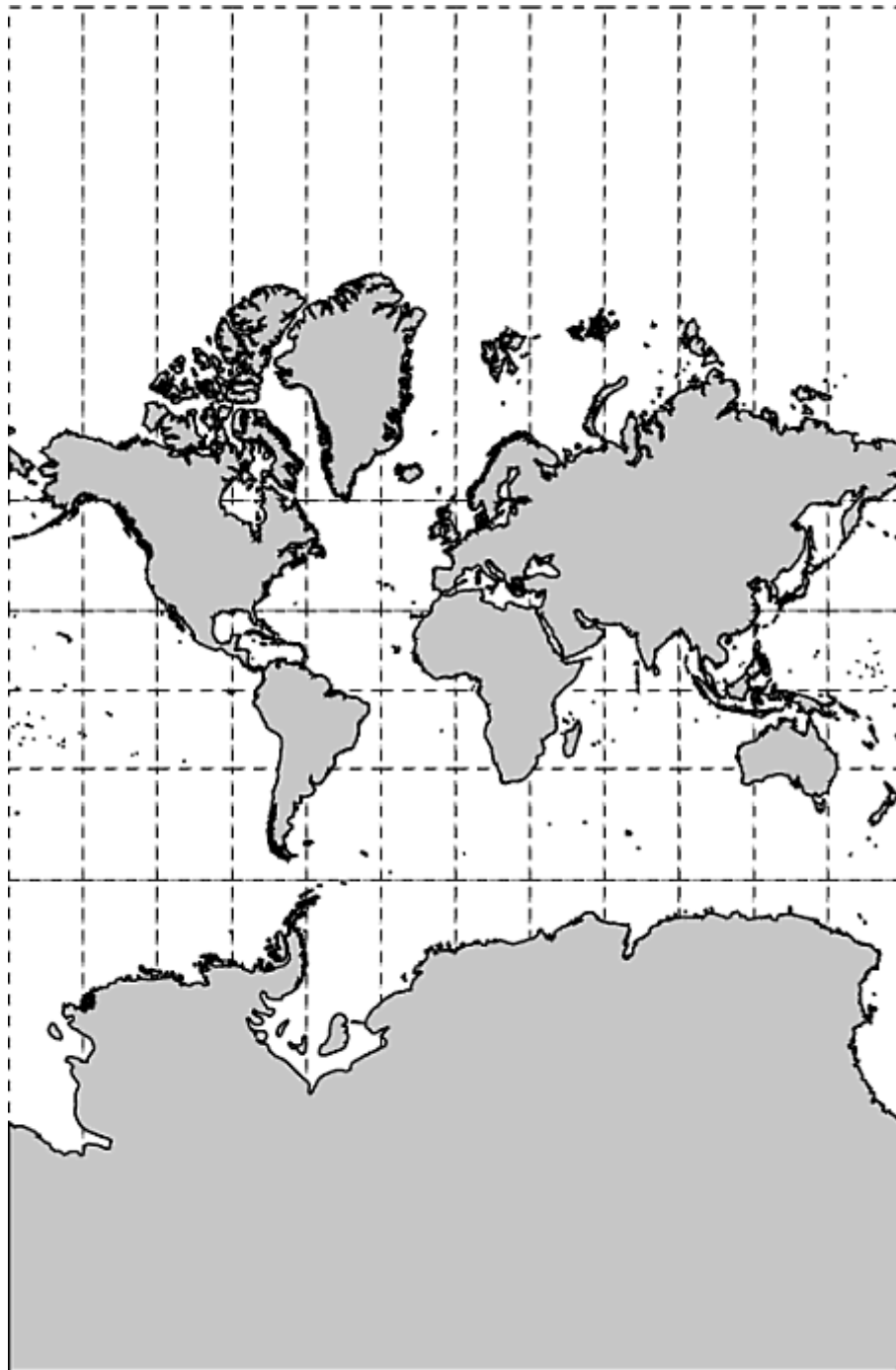
Οι γραμμές γεωγραφικού πλάτους ονομάζονται παράλληλοι πλάτους γιατί ισαπέχουν η μία από την άλλη. Εφόσον οι παράλληλοι πλάτους δεν συγκλίνουν, όπως οι μεσημβρινοί, μπορούν να χρησιμοποιηθούν για να μετρηθούν αποστάσεις σε έναν ναυτικό χάρτη. Ο παράλληλος πλάτους που λέγεται ισημερινός έχει 0 μοίρες γεωγραφικό πλάτος. Οι παράλληλοι αυξάνουν μοίρες καθώς κινούνται προς τους πόλους στους οποίους έχουν 90 μοίρες. 180 μοίρες για κάθε ημισφαίριο και όλο μαζί είναι 360 μοίρες. Κάθε μοίρα γεωγραφικού πλάτους έχει 60 λεπτά και κάθε λεπτό 60 δευτερόλεπτα. Το γεωγραφικό πλάτος χαρακτηρίζεται ως βόρειο [N] αν είναι βόρεια του ισημερινού ή νότιο [S] αν είναι νότια του ισημερινού [Lar03].

Η Μερκατορική προβολή είναι ένα παράδειγμα χάρτη σύμμορφης προβολής ( αλλιώς ορθομορφική). Ένας χάρτης σύμμορφης προβολής διατηρεί το τοπικό σχήμα των αντικειμένων στον συνολικό χάρτη.

Η Μερκατορική προβολή αναπτύχθηκε πρώτη φορά το 1569 από τον Βέλγο χαρτογράφο Gerardus Mercator και χρησιμοποιείται ευρέως από τότε. Χρησιμοποιείται κυρίως στην ναυτική πλοήγηση, γιατί όταν χρησιμοποιείς έναν χάρτη από Μερκατορική προβολή η διαδρομή ενός πλοίου που ακολουθεί μια σταθερή πορεία απεικονίζεται στον χάρτη με ευθεία γραμμή.

Ένας χάρτης από Μερκατορική προβολή προβάλλει με ακρίβεια όλα τα σημεία που είναι πάνω στον ισημερινό. Ωστόσο όσο απομακρυνόμαστε από τον ισημερινό η παραμόρφωση γίνεται όλο και μεγαλύτερη. Μια κριτική σ' αυτή την προβολή είναι ότι, λόγω της θέσης των διάφορων κρατών στον χάρτη, ανεπτυγμένες χώρες φαίνονται να είναι πολύ μεγαλύτερες από αντίστοιχου μεγέθους αναπτυσσόμενες χώρες. Για παράδειγμα η βόρεια Αμερική (19 εκατομμύρια τετραγωνικά χιλιόμετρα) και η Αφρική (30 εκατομμύρια τετραγωνικά χιλιόμετρα) φαίνονται να έχουν περίπου το ίδιο μέγεθος (βλέπε εικόνα 2.3).

Παρά την όποια κριτική χάρτες Μερκατορικής προβολής χρησιμοποιούνται ακόμα πολύ συχνά από πολλές εφαρμογές, συμπεριλαμβανομένου της ιστοσελίδας του Google Maps (<http://maps.google.com/>) [Ait09].



Εικόνα 2.3 Χάρτης Μερκατορικής προβολής. [Ait09]

## 2.4 Εύρεση Συντεταγμένων

Ένα από τα βασικότερα κομμάτια της παρούσας διπλωματικής είναι η εύρεση των χωρικών συντεταγμένων των ερευνώμενων. Χωρίς αυτό το κομμάτι κάθε άλλη ανάλυση θα ήταν αδύνατη.

Η εύρεση συντεταγμένων έχει καταστεί δυνατή τα τελευταία χρόνια μέσω υπηρεσιών που παρέχουν εταιρείες στο Διαδίκτυο. Κάποιες από αυτές τις υπηρεσίες είναι : Google Geocoding [GOG1], Yahoo Geocoding<sup>2</sup>, Bing Maps Geocoding<sup>3</sup>.

Εμείς χρησιμοποιήσαμε το API της Google, οπότε θα το παρουσιάσουμε με περισσότερες λεπτομέρειες στην συνέχεια. Η Google έχει JavaScript API για χρήση σε ιστοσελίδες , το οποίο δεν θα μπορούσαμε να το χρησιμοποιήσουμε , καθώς η εφαρμογή αναπτύχθηκε σε γλώσσα java και όχι σε ιστοσελίδα.

Χρησιμοποιήθηκε λοιπόν το static Geocoding API <sup>4</sup> που παρέχει η Google. Το API αυτό χρησιμοποιεί κάποιες παραμέτρους που στέλνονται με HTTP αίτημα. Για να σταλεί το αίτημα χρησιμοποιείται ένα URL της μορφής:

`http://maps.google.com/maps/api/geocode/output?parameters,`

όπου output μπορεί να είναι είτε json είτε xml και δείχνει σε τι μορφή μας γυρνά τα αποτελέσματα.

Ορισμένες παράμετροι είναι υποχρεωτικές ενώ άλλες είναι προαιρετικές. Όπως σε όλα τα HTTP αιτήματα οι παράμετροι διαχωρίζονται με τον χαρακτήρα &. Ακολουθεί η λίστα των παραμέτρων και εξηγείται η χρήση τους:

- address (υποχρεωτική). Η διεύθυνση για να μας γυρίσει τις συντεταγμένες.
- latlng (υποχρεωτική). Οι συντεταγμένες γεωγραφικό πλάτος/γεωγραφικό μήκος της οποίας θέλουμε να βρούμε την κοντινότερη διεύθυνση (reverse Geocoding-αντίστροφη γεωκωδικοποίηση). Σημειώνεται πως μια από τις δύο παραμέτρους είναι υποχρεωτική, ή η address ή η latlng.
- bounds (προαιρετική). Η περιοχή στην οποία κυρίως να ψάξει να βρει την διεύθυνση.
- region(προαιρετική). Ο κώδικας της περιοχής( πχ. gr)
- language(προαιρετική). Η γλώσσα στην οποία θα γυρίσει τα αποτελέσματα.
- sensor(υποχρεωτική). Καθορίζει αν η αίτηση γίνεται από συσκευή που έχει αναγνώριση θέσης.

Στο αρχείο που μας επιστρέφεται σημαντικό είναι το status code. Με βάση αυτό μπορούμε να καταλάβουμε αν όλα πήγαν καλά και βρέθηκε αυτό που ζητήσαμε. Συγκεκριμένα τα πιθανά status codes είναι:

---

<sup>2</sup> <http://developer.yahoo.com/geo/placefinder/>

<sup>3</sup> <http://msdn.microsoft.com/en-us/library/dd877180.aspx>

<sup>4</sup> Σχετικά με το API βλέπε [GOG1]

- “OK” δείχνει ότι δεν έγινε κανένα λάθος. Η διεύθυνση πέρασε επιτυχώς από συντακτική ανάλυση και γύρισε τουλάχιστον ένα αποτέλεσμα.
- “ZERO\_RESULTS” δείχνει ότι η διαδικασία ήταν επιτυχής αλλά δεν γύρισε κανένα αποτέλεσμα. Αυτό μπορεί να συμβεί αν περάσει σαν παράμετρος μια μη υπαρκτή διεύθυνση ή συντεταγμένες σε απομακρυσμένη περιοχή.
- “OVER\_QUERY\_LIMIT” δείχνει ότι έχουμε υπερβεί τα όρια χρήσης.
- “REQUEST\_DENIED” δείχνει ότι δεν έγινε δεκτό το αίτημά μας , λόγω έλλειψης της παραμέτρου sensor.
- “INVALID\_REQUEST” δείχνει ότι λείπει είτε η παράμετρος address είτε η παράμετρος latlng.

Το επόμενο σημαντικό στοιχείο είναι η ακρίβεια που έχει το επιστρεφόμενο αποτέλεσμα. Υπάρχουν 19 διαφορετικά είδη ακρίβειας (τα οποία δε θα τα αναφέρουμε εδώ. Είναι διαθέσιμα στην ιστοσελίδα του API) . Από όλα εκείνο που μας ενδιαφέρει είναι το “street\_address”, το οποίο αντιστοιχεί σε ακρίβεια διεύθυνσης.

Στο σημείο αυτό πρέπει να σημειώσουμε ότι κατά την διάρκεια εκπόνησης της διπλωματικής άλλαξε το συγκεκριμένο API. Στο deprecated API ,που μπορεί να βρεθεί στην παρακάτω ιστοσελίδα: <http://code.google.com/apis/maps/documentation/geocoding/v2/index.html> , ήταν απαραίτητη η απόκτηση API Key . Από τις 8 Μαρτίου του 2010 καταργήθηκε και πλέον το καινούριο API έχει διαφορετικούς περιορισμούς χρήσης από το προηγούμενο.

Το παρόν API μπορεί να δεχτεί από μια διεύθυνση IP μέχρι 2500 αιτήσεις την ημέρα (Χρήστες της υπηρεσίας Google Maps API Premier μπορούν να κάνουν μέχρι και 100.000 αιτήσεις την ημέρα).

## 2.5 Χάρτης

Τα αποτελέσματα της υπηρεσίας Geocoding πρέπει να χρησιμοποιηθούν (με βάση την πολιτική ορθής χρήσης)<sup>5</sup> σε συνεργασία με χάρτη της Google. Πέραν αυτού, η παρουσίαση των συντεταγμένων σε χάρτες είναι απολύτως επιθυμητή για την εφαρμογή μας.

Η λειτουργία των χαρτών της εφαρμογής βασίζεται στο Google Static Maps.

Η υπηρεσία δέχεται HTTP αιτήματα σε URL της μορφής:

<http://maps.google.com/maps/api/staticmap?parameters>,

και μας επιστρέφει μια εικόνα που αντιστοιχεί στον επιθυμητό χάρτη.

Όπως και προηγουμένως άλλες παράμετροι είναι υποχρεωτικές και άλλες προαιρετικές.

Παραθέτουμε λίστα με αυτές:

---

<sup>5</sup> Λεπτομέρειες για την πολιτική ορθής χρήσης της υπηρεσίας:  
[http://code.google.com/apis/maps/terms.html#section\\_10\\_12](http://code.google.com/apis/maps/terms.html#section_10_12)

- center (υποχρεωτική εκτός αν υπάρχει η παράμετρος markers). Καθορίζει το κέντρο του χάρτη, το οποίο ισαπέχει απ' όλα τα άκρα του χάρτη. Η παράμετρος δέχεται μια τοποθεσία είτε ως ένα ζεύγος {γεωγραφικό πλάτος, γεωγραφικό μήκος} χωρισμένο με κόμμα είτε ως διεύθυνση, που προσδιορίζει μονοσήμαντα ένα σημείο στην υδρογείο.
- zoom (υποχρεωτική εκτός αν υπάρχει η παράμετρος markers). Καθορίζει το επίπεδο εστίασης του χάρτη δηλαδή την κλίμακα του χάρτη. Αυτή η παράμετρος παίρνει σαν όρισμα μια αριθμητική τιμή που ανταποκρίνεται στο επίπεδο εστίασης της επιθυμητής τοποθεσίας. Τα επίπεδα εστίασης είναι από 0 (ολόκληρη η γη σε ένα χάρτη) μέχρι 21 (Η μεγαλύτερη δυνατή λεπτομέρεια). Κάθε επίπεδο εστίασης διπλασιάζει την ακρίβεια και οριζόντια και κάθετα του προηγούμενου επιπέδου. Σημειώνεται ότι δεν είναι διαθέσιμα όλα τα επίπεδα εστίασης σε όλα τα σημεία της υδρογείου.
- size (υποχρεωτική). Καθορίζει τις διαστάσεις του χάρτη. Η τιμή της παραμέτρου είναι μια συμβολοσειρά με την εξής μορφή: οριζόντια διάσταση x κατακόρυφη διάσταση. Ο αριθμοί αντιστοιχούν σε pixels. Μέγιστη επιτρεπτό μέγεθος : 640x640.
- format (προαιρετική). Ορίζει τον τύπο αρχείου της εικόνας (gif, jpeg, png). Προεπιλογή είναι το png.
- maptype (προαιρετική). Ορίζει τον τύπο του χάρτη. Διαθέσιμοι τύποι είναι : δορυφορικός, εδαφικός, υβριδικός και οδικός (προεπιλεγμένος).
- language (προαιρετική). Η γλώσσα στην οποία θα γυρίσει τα αποτελέσματα.
- markers (προαιρετική). Ορίζει σημεία στον χάρτη όπου θα εμφανιστούν καρφίτσες. Οι παράμετροι που δέχεται για τον καθορισμό του χρώματος , του συμβόλου της καρφίτσας και της διεύθυνσης που θα εμφανιστεί χωρίζονται με τον χαρακτήρα: | . Στην περίπτωση που δοθεί αυτή η παράμετρος δεν είναι απαραίτητο να δοθούν οι center και zoom.
- path (προαιρετική). Δίνει τη δυνατότητα να σχηματιστούν πολυγωνικές γραμμές (μονοπάτια) στον χάρτη.
- visible (προαιρετική). Καθορίζει τοποθεσίες που πρέπει οπωσδήποτε να φαίνονται στον χάρτη. Στην περίπτωση αυτή δεν χρειάζεται να δοθεί το zoom level.
- sensor (υποχρεωτική). Καθορίζει αν η αίτηση γίνεται από συσκευή που έχει αναγνώριση θέσης.

Οι αιτήσεις που μπορεί να εξυπηρετήσει η υπηρεσία που προέρχονται από το ίδιο IP δεν μπορούν να υπερβαίνουν τις 1000 την ημέρα [GOG2].

## 2.6 Διεπαφή για εύρεση αποστάσεων

Όπως και στις προηγούμενες περιπτώσεις υπάρχει API για αναζήτηση αποστάσεων μεταξύ δύο διευθύνσεων. Το αίτημα στέλνεται σε URL της μορφής:

<http://maps.google.com/maps/api/directions/output?parameters>

Και ανάλογα με τις παραμέτρους λαμβάνονται οι αποστάσεις μεταξύ αφετηρίας και προορισμού ανάλογα αν η διαδρομή θα γίνει με τα πόδια ή οδηγώντας. Επίσης λαμβάνονται οι αναλυτικές οδηγίες της διαδρομής [GOG3].

Αυτή η υπηρεσία θα μπορούσε να χρησιμοποιηθεί στο κομμάτι της κατάταξης των ερευνώμενων για κάθε ερευνητή. Όμως λόγω του περιορισμού χρήσης της (μέχρι 2500 χιλιάδες αιτήσεις την ημέρα) αυτό θα ήταν υπερβολικά χρονοβόρο. Για τον σχηματισμό του πλήρους γράφου 70 μόνο διευθύνσεων θα σπαταλούταν όλο το ημερήσιο όριο της υπηρεσίας.

( $\frac{70 \times 71}{2} = 2485$ ) Θα μπορούσε βέβαια να χρησιμοποιηθεί αφού έχει βγει μια κατάταξη για να προτείνει την καλύτερη διαδρομή από ερευνώμενο σε ερευνώμενο. Αντί για αυτό όμως τα αποτελέσματα δίνουν έναν σύνδεσμο για γρήγορη προβολή της προτεινόμενης διαδρομής στη σελίδα <http://maps.google.com>.

# 3

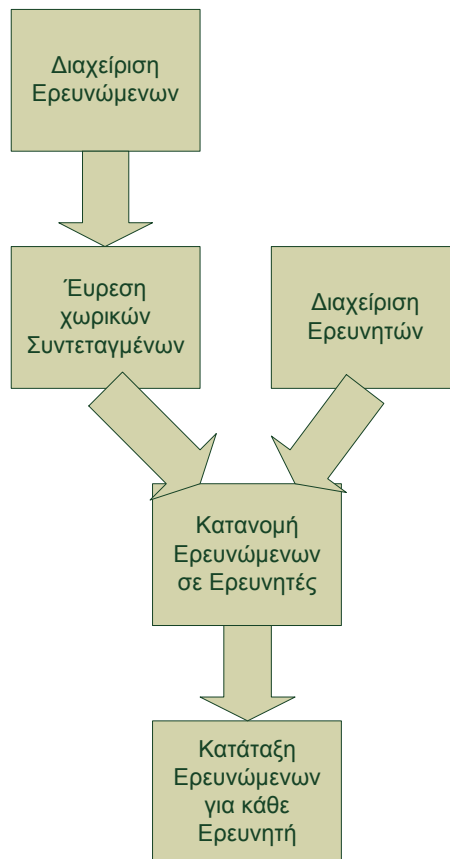
## *Ανάλυση Απαιτήσεων Συστήματος*

Ακολουθεί η περιγραφή της αρχιτεκτονικής του συστήματος και η ανάλυση των απαιτήσεων του.

### *3.1 Απαιτήσεις συστήματος*

Το σύστημά μας αποτελείται από 5 επιμέρους σημαντικά κομμάτια:

- Τη διαχείριση ερευνητών. Σε αυτό το κομμάτι εισάγονται στο σύστημα οι ερευνητές, που θα συμμετάσχουν στην παρούσα έρευνα. Επίσης δηλώνονται οι προτιμήσεις που έχουν σε δήμους και σε αριθμό ερευνώμενων που θέλουν να επισκεφτούν.
- Τη διαχείριση ερευνώμενων. Σε αυτό το κομμάτι εισάγονται οι διευθύνσεις των ερευνώμενων και τυχόν άλλες πληροφορίες για αυτούς.
- Την εύρεση χωρικών συντεταγμένων. Οι διευθύνσεις αντιστοιχίζονται σε συντεταγμένες και φαίνονται γραφικά σε κάποιον χάρτη.
- Την κατανομή των ερευνώμενων σε ερευνητές. Εδώ με βάση τις προτιμήσεις των ερευνητών σε δήμους και αριθμό επισκέψεων και με βάση τα αποτελέσματα των διευθύνσεων αντιστοιχίζεται κάθε ερευνώμενος σε έναν ερευνητή.
- Την κατάταξη των ερευνώμενων για κάθε ερευνητή. Το κομμάτι αυτό προτείνει σε κάθε ερευνητή την σειρά με την οποία να επισκεφτεί τους ερευνώμενους.



Εικόνα 3.1 Μπλοκ διάγραμμα για την περιγραφή των υποσυστημάτων.

## 3.2 Περιγραφή Λειτουργιών

Περιγράφονται παρακάτω αναλυτικά οι λειτουργίες που εκτελεί το κάθε υποσύστημα.

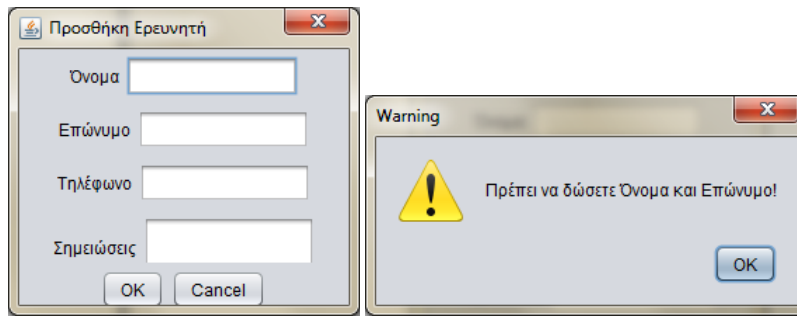
### 3.2.1 Διαχείριση ερευνητών

Αυτό το κομμάτι δίνει την δυνατότητα στον χρήστη να εισάγει τους υπάρχοντες ερευνητές, να εισάγει τους δήμους που γίνονται έρευνες, να επιλέξει ερευνητές για την παρούσα έρευνα, να επιλέξει προτιμώμενους δήμους για κάθε ερευνητή και να δώσει τον αριθμό των ερευνώμενων που επιθυμεί να επισκεφτεί ο κάθε ερευνητής.

#### 3.2.1.1 Ερευνητές

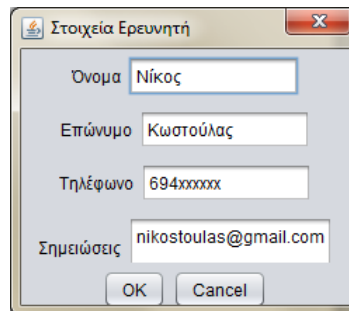
Ο χρήστης μπορεί να εισάγει νέους ερευνητές χειροκίνητα συμπληρώνοντας τα στοιχεία τους: Όνομα, Επώνυμο, Τηλέφωνο και τυχόν Σημειώσεις. Απαραίτητα στοιχεία για να γίνει η προσθήκη είναι το Όνομα και το Επώνυμο του ερευνητή, χωρίς τα οποία δεν μπορεί να γίνει αναγνώριση του ερευνητή.





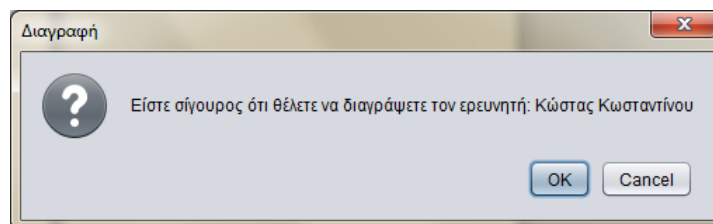
**Εικόνα 3.2. Παράθυρο προσθήκης ερευνητή. Εικόνα 3.3 Προειδοποίηση σε περίπτωση που δεν δοθεί όνομα και επώνυμο.**

Κάθε ερευνητής που προστίθεται φαίνεται αμέσως και στην λίστα «Επιλογή ερευνητών». Στην λίστα όμως δεν φαίνονται όλες οι πληροφορίες για κάποιον ερευνητή. Φαίνεται μόνο το ονοματεπώνυμό του. Ο χρήστης μπορεί να ξαναδεί το τηλέφωνο και τις σημειώσεις για έναν ερευνητή αν τον επιλέξει από την λίστα και πατήσει το κουμπί «Στοιχεία». Από αυτό το παράθυρο μπορεί επίσης να αλλάξει κάποια πληροφορία για τον ερευνητή.



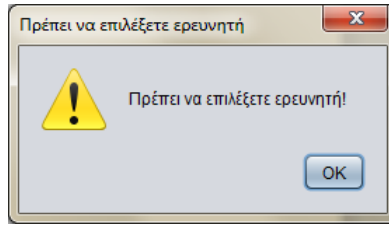
**Εικόνα 3.4. Στοιχεία ερευνητή.**

Μπορεί κάποιος να σταματήσει να εργάζεται ως ερευνητής, οπότε υπάρχει η δυνατότητα διαγραφής από την λίστα. Προτού ολοκληρωθεί η διαγραφή ένα παράθυρο ζητά επιβεβαίωση για την διαγραφή.



**Εικόνα 3.5. Επιβεβαίωση διαγραφής.**

Το άνοιγμα των στοιχείων και η διαγραφή ενός ερευνητή γίνονται με βάση το ποιος ερευνητής είναι επιλεγμένος στην λίστα επιλογής ερευνητών. Αν δεν είναι κανένας επιλεγμένος, τότε ένα παράθυρο προειδοποιεί τον χρήστη.



**Εικόνα 3.6. Προειδοποίηση , όταν επιχειρείται προβολή στοιχείων ή διαγραφή.**

### 3.2.1.2 Δήμοι

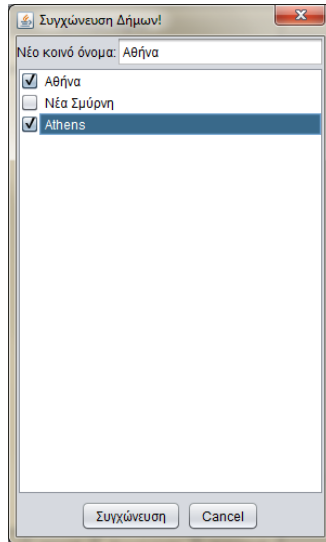
Όμοια με τους ερευνητές ο χρήστης μπορεί να διαχειριστεί τους δήμους. Μπορεί να εισάγει καινούριους δήμους, να τροποποιήσει έναν ήδη υπάρχοντα ή να διαγράψει κάποιον. Η μόνη πληροφορία που διαχειρίζεται ο χρήστης είναι το όνομα του δήμου.

Εδώ όμως, ο χρήστης μπορεί επιπλέον να συγχωνεύσει δύο ή περισσότερους δήμους σε έναν. Όπως θα δούμε σε επόμενη ενότητα είναι πιθανό να υπάρχει ο ίδιος δήμος με διαφορετικά ονόματα. Όπως π.χ. Αθήνα και Athens. Σε αυτήν την περίπτωση, επειδή στην ουσία είναι ο ίδιος δήμος, είναι επιθυμητό να συγχωνευτούν σε έναν. Στην συγχώνευση δίνεται το όνομα που θα αποκτήσουν οι επιλεγμένοι δήμοι. Ο πρώτος δήμος αποκτά το όνομα που δόθηκε και οι υπόλοιποι δήμοι διαγράφονται.

Αν δεν έχει συμπληρωθεί κάποιο όνομα όταν επιλεγεί ο πρώτος δήμος, τότε αυτό συμπληρώνεται με το όνομα αυτού του δήμου. Ωστόσο πάντα υπάρχει η δυνατότητα αλλαγής του.

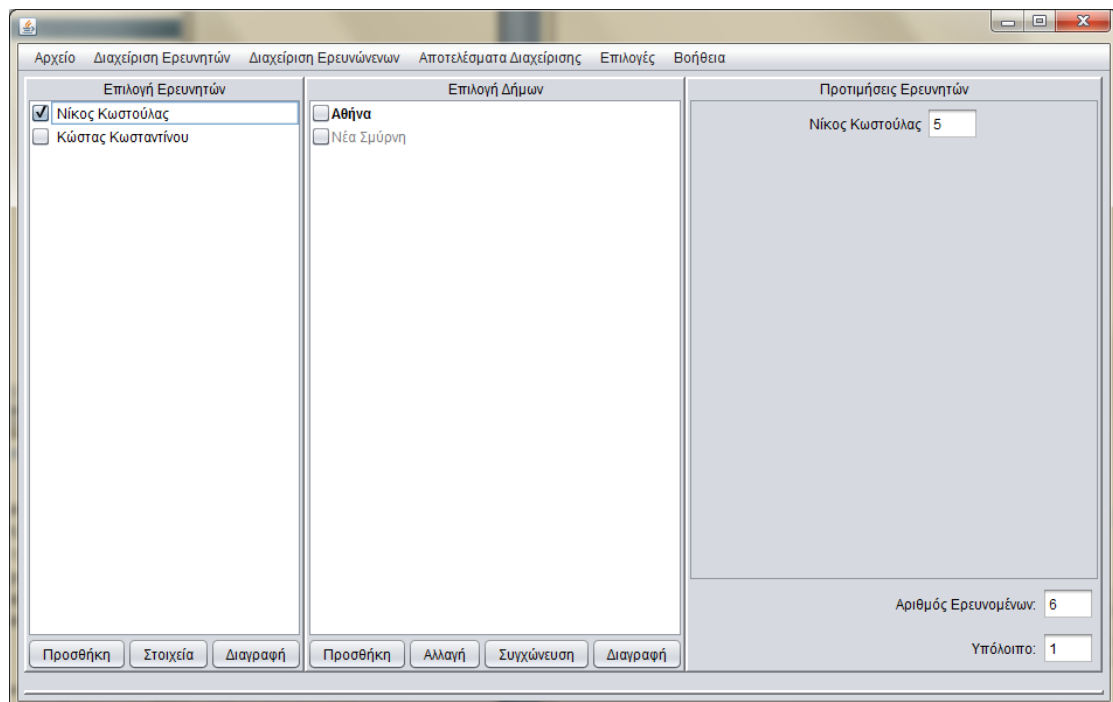
Στην συγχώνευση και στην αλλαγή ονόματος δήμων, όπως θα δούμε και παρακάτω, οι διευθύνσεις που αντιστοιχούν σε αυτούς τους δήμους αναπροσαρμόζονται ώστε να αντιστοιχούν στον νέο δήμο. Επίσης, διαγράφονται από την βάση τυχόν άλλες πληροφορίες σχετικές με τις διευθύνσεις, όπως αν είχαν συντεταγμένες ή αν ήταν προβληματικές. Προφανώς αν άλλαξε ο δήμος της διεύθυνσης μπορεί κάλλιστα να είναι διαφορετικά και τα αποτελέσματα των συντεταγμένων της ή το αν είναι προβληματική ή όχι. Για παράδειγμα η διεύθυνση Δουκίσσης Πλακεντίας 16 Αθήνα, υπάρχει στην Αττική, αλλά η διεύθυνση Δουκίσσης Πλακεντίας 16 Αιγάλεω, δεν υπάρχει, και όταν επιχειρηθεί να βρεθούν οι συντεταγμένες της θα μαρκαριστεί ως προβληματική.

Παρόμοια όταν διαγράφεται ένας δήμος διαγράφονται και όλες οι πληροφορίες για τις διευθύνσεις που ήταν σε αυτόν τον δήμο. Αυτό είναι βολικό γιατί σε μια έρευνα μπορεί να αποφασιστεί στην πορεία να μην συμμετέχει κάποιος δήμος.



**Εικόνα 3.7. Συγχώνευση δήμων**

Ο χρήστης επιλέγοντας έναν ερευνητή μπορεί στην διπλανή λίστα να σημειώσει ποιοι δήμοι ανήκουν στην προτίμησή του. Η λίστα αυτή δείχνει τους δήμους που δεν έχουν επιλεγθεί από κανέναν ερευνητή με γκρι χρώμα και απλά γράμματα αντί για bold γράμματα που έχουν οι υπόλοιποι δήμοι. Με αυτό τον τρόπο ο χρήστης μπορεί να διαπιστώσει ποιοι δήμοι δεν έχουν την προτίμηση κανενός και να προβεί στις κατάλληλες ενέργειες όπως να βρει καινούριους ερευνητές. Να σημειωθεί ότι κατά την συγχώνευση δήμων πρέπει να ξανά-ελεγχθούν οι προτιμήσεις σε δήμους γιατί μπορεί κάποιες να έχουν χαθεί. Για την ακρίβεια πέραν του πρώτου δήμου προς συγχώνευση οι προτιμήσεις για τους επόμενους δήμους θα έχουν χαθεί.



**Εικόνα 3.8. Παράθυρο διαχείρισης ερευνητών. Φαίνονται οι επιλογές που έχει κάνει ο χρήστης.**

Όσες αλλαγές έχουν αναφερθεί ως τώρα θεωρούνται μόνιμες, αποθηκεύονται και το σύστημα μόλις ξανά-ανοίξει έχει ακριβώς την ίδια εικόνα που είχε όταν έκλεισε. Επίσης διατηρούνται σε όλες τις έρευνες. Δεν θεωρείται ότι συνδέονται μόνο με την παρούσα έρευνα σε αντίθεση με τις υπόλοιπες πληροφορίες που εισάγει ο χρήστης.

### *3.2.1.3 Προτιμήσεις για την παρούσα έρευνα*

Στην λίστα με τους ερευνητές έχει θεωρηθεί ότι υπάρχουν όλοι οι διαθέσιμοι ερευνητές, είτε θα συμμετάσχουν στην παρούσα έρευνα είτε όχι. Οπότε ο χρήστης επιλέγει ποιοι από αυτούς θα συμμετάσχουν στην παρούσα έρευνα κάνοντας «κλικ» σε ένα κουτάκι δίπλα στον ερευνητή.

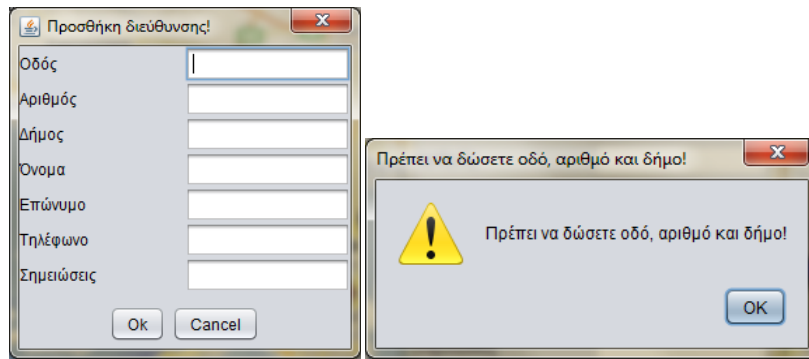
Οι ερευνητές που είναι επιλεγμένοι για αυτή την έρευνα φαίνονται στην τελευταία λίστα δεξιά αυτού του παραθύρου. Εκεί ο χρήστης μπορεί να δώσει και τον αριθμό των ερευνώμενων που θέλει να επισκεφτεί ο κάθε ερευνητής. Κάτω από αυτήν τη λίστα φαίνεται το σύνολο των ερευνώμενων που έχει βρεθεί στον χάρτη και κάτω από αυτό ο αριθμός ερευνώμενων που δεν έχει ανατεθεί σε ερευνητή.

Οι παραπάνω πληροφορίες συνδέονται με την παρούσα έρευνα. Αποθηκεύονται μέχρι να επιλέξει ο χρήστης καινούρια έρευνα.

### *3.2.2 Διαχείριση ερευνώμενων*

Το κομμάτι αυτό δίνει την δυνατότητα στον χρήστη να διαχειριστεί ερευνώμενους. Υπάρχει η δυνατότητα εισαγωγής ερευνώμενων. Δηλαδή των διευθύνσεων τους που προαιρετικά μπορεί να έχουν και άλλα στοιχεία, όπως όνομα και τηλέφωνο. Η εισαγωγή αυτή μπορεί να γίνει μαζικά από αρχείο. Η μορφή του αρχείου περιγράφεται στο κεφάλαιο 4.

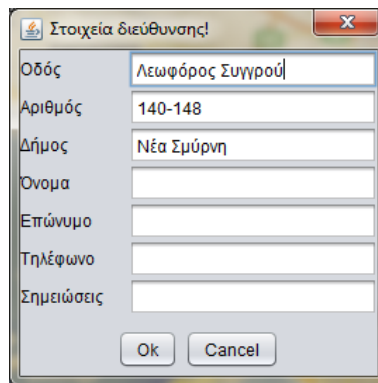
Πέραν της εισαγωγής από αρχείο μπορεί να γίνει εισαγωγή μέσα από την εφαρμογή, καθώς και διαγραφής ή αλλαγής κάποιας εγγραφής. Για την εισαγωγή μιας διεύθυνσης απαραίτητο είναι να δοθεί οδός, αριθμός και δήμος. Για κάθε διεύθυνση που εισάγεται, αν ο δήμος στον οποίο ανήκει δεν υπάρχει στην προηγούμενη λίστα δήμων, τότε ο δήμος αυτός εισάγεται χωρίς ειδοποίηση. Επίσης για αλλαγή ή διαγραφή απαραίτητο είναι να έχει επιλεγεί μια εγγραφή από την λίστα.



**Εικόνα 3.9. Προσθήκη ερευνώμενου 3.10. Προειδοποίηση μη εισαγωγής όλων των απαραίτητων στοιχείων.**

Όπως στην περίπτωση των ερευνητών έτσι και εδώ στην λίστα δεν δίνονται όλες οι πληροφορίες για τον ερευνώμενο αλλά μόνο η διεύθυνσή του. Στα στοιχεία της διεύθυνση φαίνονται όλα και δίνεται η δυνατότητα να αλλάξουν.

Σημειώνουμε επίσης ότι στον αριθμό διεύθυνσης η εφαρμογή δέχεται οποιοδήποτε συμβολοσειρά και όχι απαραίτητα αριθμό, καθώς υπάρχουν διευθύνσεις όπως στην Εικόνα 3.11 που έχουν ένα σύνολο αριθμών.



**Εικόνα 3.11. Στοιχεία ερευνώμενου.**

Όταν αλλάζει μια διεύθυνση ή όταν διαγράφεται, (όπως και πριν όταν αλλάζει ένας δήμος ή διαγράφεται) διαγράφονται οι πληροφορίες που αφορούν στις συντεταγμένες της διεύθυνσης ή στο αν είναι προβληματική. Εφ' όσον υπάρχουν αυτές οι πληροφορίες.

### **3.2.3 Εύρεση συντεταγμένων**

Η εύρεση συντεταγμένων είναι στο ίδιο παράθυρο με την διαχείριση ερευνώμενων. Στην ουσία υπάρχουν δύο κουμπιά. Το ένα «εμφάνισε επιλεγμένη διεύθυνση» και το άλλο «εμφάνισε όσες δεν είναι εμφανισμένες». Η διαφορά τους είναι ότι το πρώτο ψάχνει να βρει τις συντεταγμένες ενός μόνο ερευνώμενου, ενώ το άλλο ψάχνει για όλους όσους δεν υπάρχουν συντεταγμένες. Το δεύτερο κουμπί όταν πατηθεί εμφανίζει μια μπάρα στο κέντρο του παραθύρου που δείχνει την πρόοδο της διαδικασίας. Η εφαρμογή δεν «κολλάει» όση ώρα

ψάχνει να βρει τις συντεταγμένες των ερευνώμενων. Ενώ στην περίπτωση μιας διεύθυνσης η εφαρμογή «περιμένει» να βρεθούν οι συντεταγμένες της.

### 3.2.3.1 Χάρτης

Σε αυτό το παράθυρο δίπλα στην λίστα με τους ερευνώμενους (τις διευθύνσεις τους) υπάρχει ένας διαδραστικός χάρτης. Στον χάρτη αυτό δίνεται η δυνατότητα κύλισης προς όλες τις κατευθύνσεις με την βοήθεια του ποντικιού και η δυνατότητα εστίασης με τον τροχό κύλισης του ποντικιού. Προεπιλεγμένα εμφανίζεται η Αθήνα στο κέντρο του. Επίσης η εστίαση είναι περιορισμένη να γίνεται μόνο μέχρι το επίπεδο 16.

Ο χάρτης αυτός είναι οδικός και πάνω του εμφανίζονται με μπλε καρφίτσα, όσες διευθύνσεις έχουν βρεθεί οι συντεταγμένες τους. Μάλιστα όταν επιλέγεται μια διεύθυνση, που έχουν βρεθεί οι συντεταγμένες της, στην λίστα διευθύνσεων τότε η αντίστοιχη καρφίτσα κοκκινίζει. Επίσης, αν λόγω επιπέδου εστίασης, δεν είναι ορατή στον χάρτη, τότε αυτός αυτόματα μετακινείται, ώστε να έρθει στο κέντρο του, διατηρώντας το ίδιο επίπεδο εστίασης. Το ανάποδο συμβαίνει όταν επιλέγουμε μια διεύθυνση στον χάρτη («αριστερό κλικ» πάνω σε μια μπλε καρφίτσα). Τότε επιλέγεται από την λίστα η διεύθυνση στην οποία αντιστοιχεί η καρφίτσα, και τροποποιείται η λίστα αν αυτή δεν είναι ορατή. ( Η λίστα πηγαίνει σε σημείο που να γίνεται ορατή η διεύθυνση).

Τέλος από μια διεύθυνση στο χάρτη είναι δυνατόν να δει κανένας τα στοιχεία στα οποία αντιστοιχεί: διεύθυνση και πληροφορίες ερευνώμενου με «δεξί κλικ» πάνω στην καρφίτσα.

### 3.2.3.2 Εμφάνιση διεύθυνση

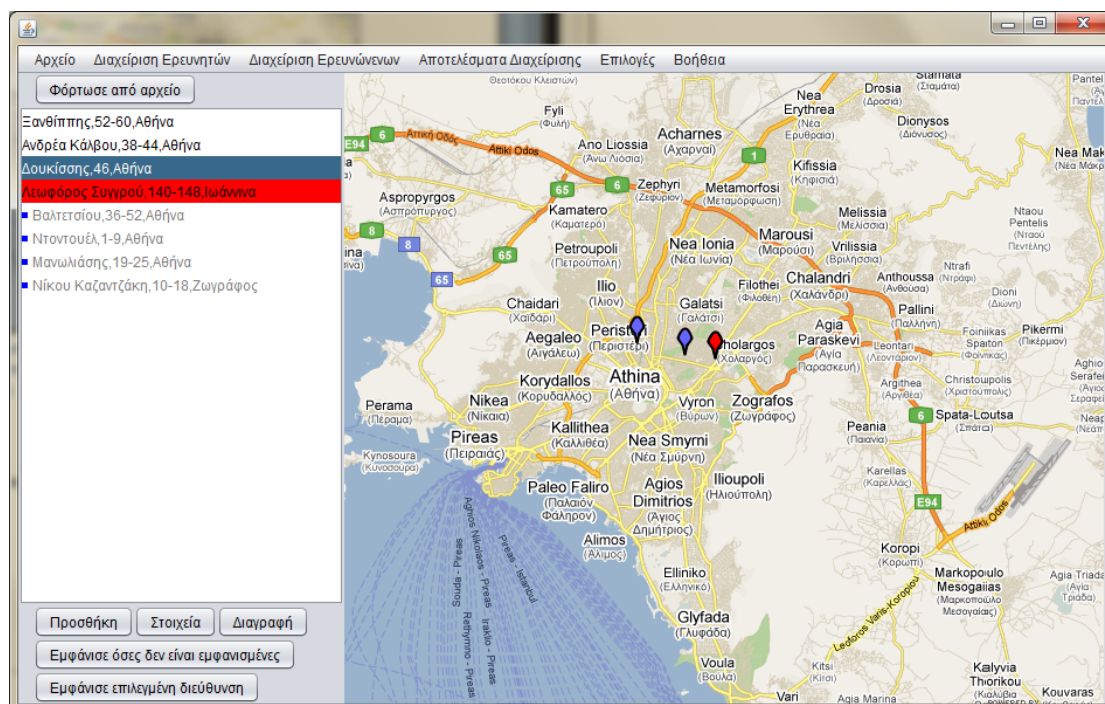
Όταν η εφαρμογή ψάχνει να βρει μια διεύθυνση υπάρχουν δύο περιπτώσεις, είτε να την βρει είτε να μην την βρει. Στην πρώτη περίπτωση εμφανίζεται μια καρφίτσα στον χάρτη στην τοποθεσία που βρίσκεται. Στην περίπτωση που δεν βρεθεί όμως, τότε η αντίστοιχη εγγραφή στην λίστα γίνεται κόκκινη για να τονιστεί ότι κάποιο πρόβλημα έχει η διεύθυνση. Επίσης μια προειδοποίηση λάθους προτρέπει τον χρήστη να ελέγξει την συνδεσιμότητα του στο Διαδίκτυο και να ξαναπροσπαθήσει.

Αν δεν βρεθεί η διεύθυνση τότε ή είναι λάθος γραμμένη, ή δεν υπάρχει, ή υπάρχει κάποιο πρόβλημα προσωρινό με την υπηρεσία εύρεσης συντεταγμένων. Σε κάθε περίπτωση ο χρήστης θα κρίνει ποιο είναι το πρόβλημα και είτε θα ξαναπροσπαθήσει είτε θα αλλάξει την διεύθυνση.

Για όσες διευθύνσεις δεν έχει γίνει ακόμα η εύρεση των θέσεων τους στον χάρτη εμφανίζονται με μια μπλε κουκίδα δίπλα τους.

Αντίστροφα με την παραπάνω διαδικασία είναι δυνατόν να επιλεγεί ένα σημείο στον χάρτη με «δεξί κλικ» και να βρεθεί η ακριβή διεύθυνσή του. Τότε ο χρήστης ερωτάται αν επιθυμεί να εισαχθεί αυτή η διεύθυνση και στη λίστα διευθύνσεων.

Όταν πρόκειται να γίνει γεωκωδικοποίηση (εύρεση συντεταγμένων) πολλών διευθύνσεων τότε προτείνεται να επιλεγεί από το μενού επιλογές η «αναμονή μεταξύ αναζητήσεων συντεταγμένων». Με αυτή την επιλογή οι αναζητήσεις γίνονται με ένα χρονικό κενό μεταξύ τους της τάξης των 35 δευτερολέπτων, έτσι ώστε η εφαρμογή να μην υπερβαίνει τον επιτρεπτό ρυθμό αιτήσεων στην υπηρεσία ( βλέπε 2.411).



Εικόνα 3.12.Ο χάρτης με τις geocoded διευθύνσεις μαρκαρισμένες με μπλε καρφίτσες και την επιλεγμένη διεύθυνση με κόκκινη καρφίτσα. Επίσης φαίνονται μια προβληματική διεύθυνση και 4 χωρίς καρφίτσα στον χάρτη.

### 3.2.4 Ανάθεση ερευνώμενων σε ερευνητές

Η ανάθεση ερευνώμενων είναι το πρώτο κομμάτι που παράγει αποτελέσματα που μπορούν να χρησιμοποιηθούν από τον διαχειριστή της έρευνας. Σε αυτό το κομμάτι την ανάθεση την κάνει ο ίδιος ο χρήστης και όχι κάποιος αλγόριθμος. Στο τελευταίο παράθυρο έχουμε δύο λίστες. Αυτή των επιλεγμένων ερευνητών και αυτή των δήμων. Δεξιά όπως και πριν υπάρχει ο ίδιος διαδραστικός χάρτης που τώρα έχει κάποιες παραπάνω λειτουργίες.

#### 3.2.4.1 Επιλογή δήμου

Όταν επιλέγεται ένας δήμος από την λίστα τότε όλες οι διευθύνσεις (οι καρφίτσες στον χάρτη) που αντιστοιχούν σε αυτό τον δήμο αποκτούν το χρώμα κίτρινο. Οι διευθύνσεις που ανήκουν στον δήμο αλλά έχουν ανατεθεί σε κάποιον ερευνητή παίρνουν το χρώμα

γκριζοκίτρινο σε αντίθεση με τις υπόλοιπες διευθύνσεις που δεν ανήκουν στον δήμο και έχουν ανατεθεί κάπου, που έχουν γκριζο χρώμα. Οι μη ανατεθειμένες διευθύνσεις έχουν μπλε χρώμα.

Η λίστα των δήμων μπορεί να έχει πολλούς δήμους επιλεγμένους.

#### 3.2.4.2 Επιλογή ερευνητή

Κατά την επιλογή ερευνητή από την λίστα ερευνητών συμβαίνουν δύο πράγματα. Επιλέγονται αυτόματα οι δήμοι που έχει στην προτίμησή του και δίνεται η δυνατότητα να επιλεγθούν με «αριστερό κλικ» ή με «δεξί κλικ και σύρσιμο» οι διευθύνσεις οι οποίες θα του ανατεθούν. Με δεύτερο «αριστερό κλικ» από-επιλέγονται. Οι διευθύνσεις που έχουν επιλεγθεί για τον εκάστοτε ερευνητή φαίνονται με ένα ελαφρύ ροζ χρώμα.

Επειδή έχουν επιλεγεί οι δήμοι που έχει ως προτίμηση ο ερευνητής, οι αντίστοιχες διευθύνσεις που προτιμά να του ανατεθούν θα έχουν κίτρινο χρώμα. Αυτό, όμως, δεν αποκλείει να του ανατεθούν και διευθύνσεις από άλλους δήμους.

Τέλος, οι διευθύνσεις που έχουν ανατεθεί σε κάποιον άλλο ερευνητή έχουν γκρι χρώμα ή γκριζοκίτρινο και δεν μπορούν να ανατεθούν στον επιλεγμένο.

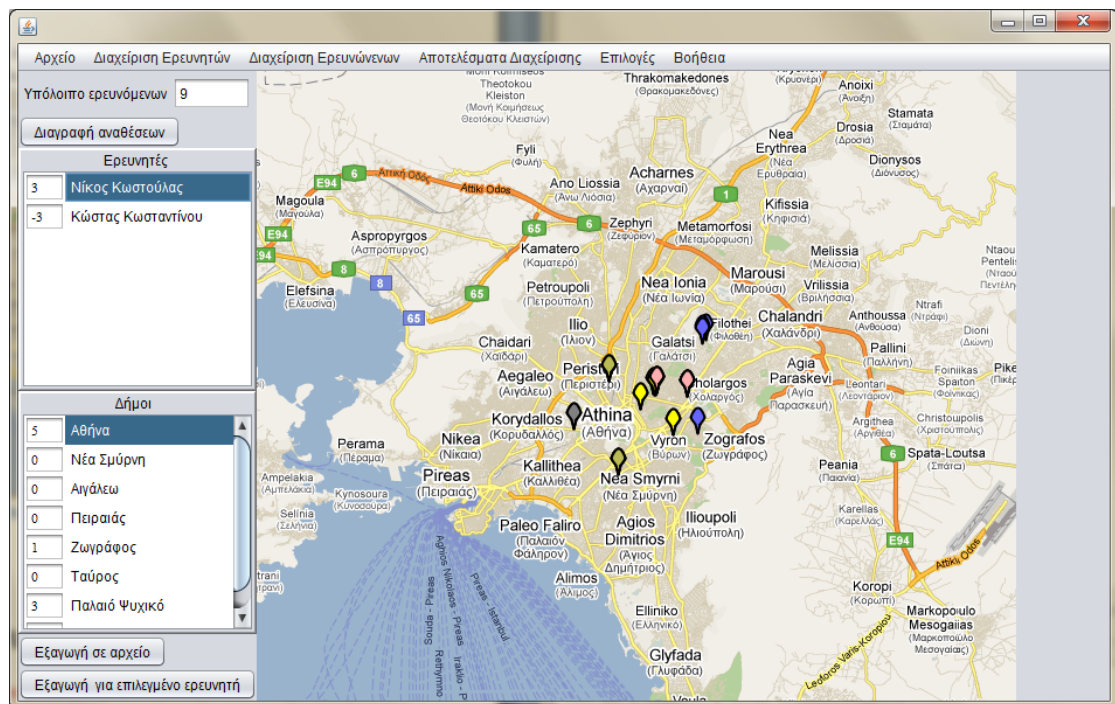
Δηλαδή στις καρφίτσες των διευθύνσεων χρησιμοποιούνται 5 διαφορετικά χρώματα για να ξεχωρίζει σε ποια κατάσταση είναι η διεύθυνση. Ανακεφαλαιώνοντας τα χρώματα αυτά είναι:

- Μπλε: Διεύθυνση που δεν έχει ανατεθεί σε κανέναν και δεν ανήκει σε επιλεγμένο δήμο από την λίστα δήμων.
- Κίτρινο: Διεύθυνση που δεν έχει ανατεθεί σε κανέναν και ανήκει σε κάποιον από τους επιλεγμένους δήμους της λίστας δήμων.
- Γκρι: Διεύθυνση που έχει ανατεθεί σε κάποιον ερευνητή και δεν ανήκει σε επιλεγμένο δήμο από την λίστα δήμων.
- Γκριζοκίτρινο: Διεύθυνση που έχει ανατεθεί σε κάποιον ερευνητή και ανήκει σε κάποιον από τους επιλεγμένους δήμους της λίστας δήμων.
- Ροζ: Διεύθυνση που έχει ανατεθεί στον επιλεγμένο ερευνητή. Είναι αδιάφορο σε ποιον δήμο ανήκει.

Δίπλα σε κάθε ερευνητή υπάρχει ένας αριθμός που δηλώνει πόσες διευθύνσεις, σύμφωνα με την προτίμησή του, μένουν για να του ανατεθούν. Ο αριθμός αυτός μπορεί να είναι αρνητικός στην περίπτωση που σε έναν ερευνητή ανατεθούν περισσότεροι ερευνώμενοι.

Αντίστοιχα δίπλα από κάθε δήμο υπάρχει ένας αριθμός που δηλώνει πόσοι ερευνώμενοι που ανήκουν σε αυτόν τον δήμο δεν έχουν αντιστοιχηθεί σε κανέναν. Ο αριθμός αυτός δεν μπορεί να είναι αρνητικός.





Εικόνα 3.13. Ανάθεση ερευνητών.

### 3.2.5 Ταξινόμηση ερευνώμενων

Μόλις τελειώσει η ανάθεση των ερευνώμενων σε ερευνητές τότε τα αποτελέσματα μπορεί να ταξινομηθούν για κάθε ερευνητή, ώστε να του προταθεί μια βέλτιστη διαδρομή. Αυτό γίνεται με αλγόριθμο που τρέχει και παράγει αποτελέσματα τα οποία αποθηκεύονται σε αρχεία.

Υπάρχουν δύο επιλογές. Να βγουν αποτελέσματα για τον επιλεγμένο ερευνητή ή για όλους. Σε κάθε περίπτωση ο χρήστης ρωτάται σε ποιον φάκελο να αποθηκευτούν τα αποτελέσματα και σε αυτόν τον φάκελο δημιουργούνται αρχεία «txt», το καθένα με όνομα το ονοματεπώνυμο του εκάστοτε ερευνητή («Όνομα\_Επώνυμο.txt»).

Το κάθε αρχείο έχει τους ερευνώμενους με τις διευθύνσεις τους στην προτεινόμενη σειρά. Στο τέλος του αρχείου δίνεται ένας σύνδεσμος για γρήγορη προβολή της προτεινόμενης διαδρομής (ανά 25 διευθύνσεις) στη σελίδα <http://maps.google.com>. Περισσότερα για τη μορφή του αρχείου στο κεφάλαιο 4.

### 3.2.6 Το περιβάλλον της εφαρμογής

Η εφαρμογή πέραν των παραπάνω παραθύρων έχει μενού, ένα κεντρικό παράθυρο και ένα παράθυρο για δημιουργία νέας έρευνας.

Όσον αφορά το μενού, η εφαρμογή έχει 6 καρτέλες: Αρχείο, Διαχείριση Ερευνητών, Διαχείριση Ερευνώμενων, Αποτελέσματα Διαχείρισης, Επιλογές, Βοήθεια. Τα υπομενού είναι ως εξής:

**Αρχείο:** Νέα Έρευνα, Έξοδος

**Διαχείριση Ερευνητών:** Παράθυρο Διαχείρισης Ερευνητών, Διαγραφή όλων των προτιμήσεων

**Διαχείριση Ερευνώμενων:** Παράθυρο Διαχείρισης Ερευνώμενων, Διαγραφή όλων των διευθύνσεων, Διαγραφή όλων των Συντεταγμένων

**Αποτελέσματα Διαχείρισης:** Παράθυρο Αποτελεσμάτων Διαχείρισης, Διαγραφή όλων των συσχετίσεων

**Επιλογές:** Προβολή γραμμής «τρέχουσα έρευνα», Προβολή γραμμής πληροφοριών, Αναμονή μεταξύ αναζητήσεων συντεταγμένων, Αποθήκευση χαρτών, Φόρτωση χαρτών από την μνήμη όταν ζητούνται

**Βοήθεια:** about

Κάθε κουμπί των μενού εκτελεί την προφανή από την ονομασία του λειτουργία. Όσον αφορά στο μενού Επιλογές αυτό διαχειρίζεται 5 σταθερές του συστήματος. Ανάλογα με το αν είναι αληθής ή ψευδής κάθε μια από αυτές, η εφαρμογή λειτουργεί και λίγο διαφορετικά, ή εμφανίζει επιπλέον πάνελ.

Όσον αφορά το μενού «Βοήθεια» αυτό εμφανίζει ένα παράθυρο με πληροφορίες για την εφαρμογή.

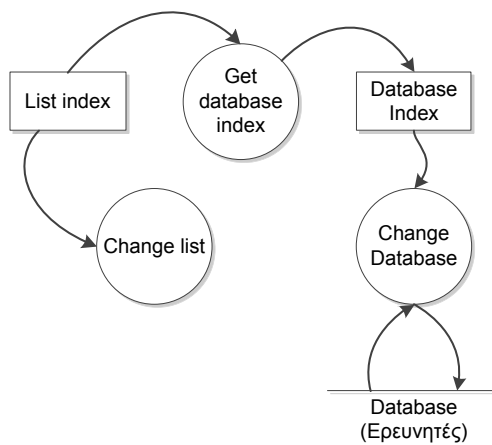
Τέλος το κεντρικό παράθυρο της εφαρμογής έχει 3 κουμπιά που οδηγούν στα παραπάνω βασικά πάνελ της εφαρμογής (Διαχείριση ερευνητών, Διαχείριση ερευνώμενων, Αποτελέσματα).

### 3.3 Διαγράμματα ροής δεδομένων

Ακολουθούν διαγράμματα ροής δεδομένων για τις ακόλουθες περιπτώσεις:

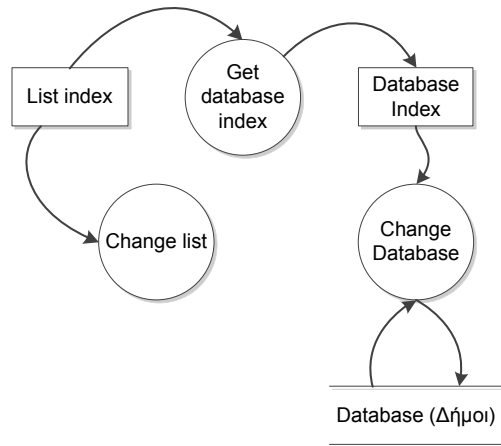
#### 3.3.1.1 Εισαγωγή-αλλαγή-διαγραφή ερευνητή

όπου change είναι ανάλογα με την διαδικασία add, update, delete from.

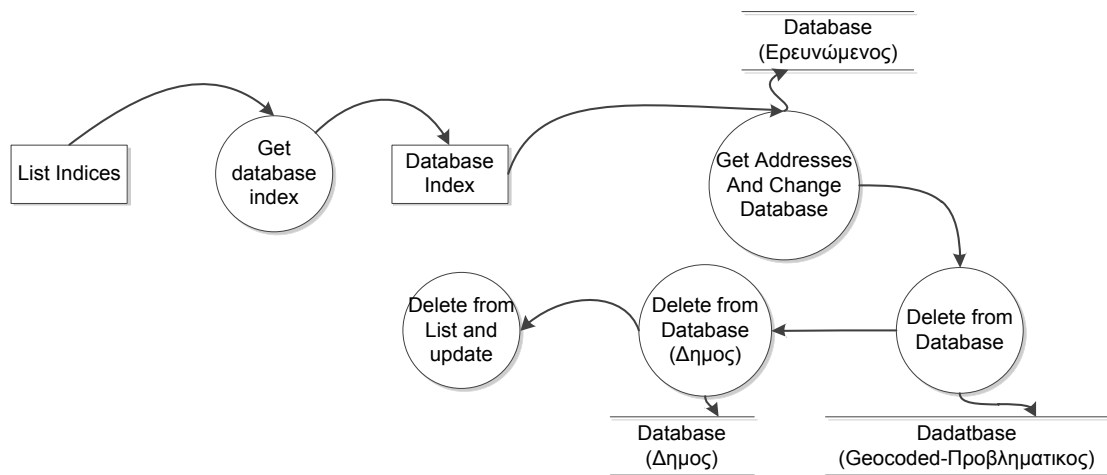


### 3.3.1.2 Εισαγωγή-αλλαγή-διαγραφή δήμου

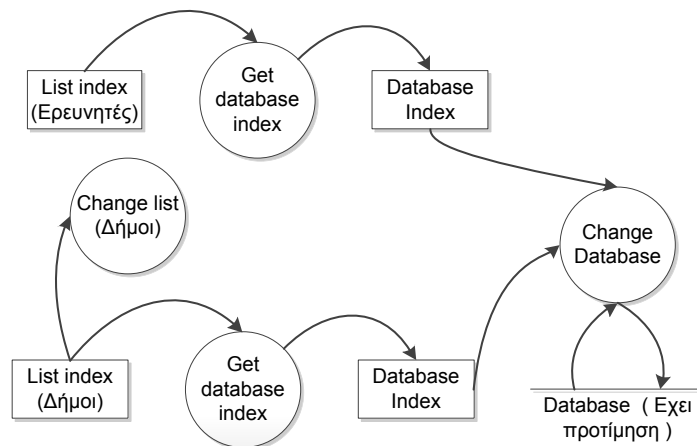
όμοια με πριν:



### 3.3.1.3 Συγχώνευση δήμων

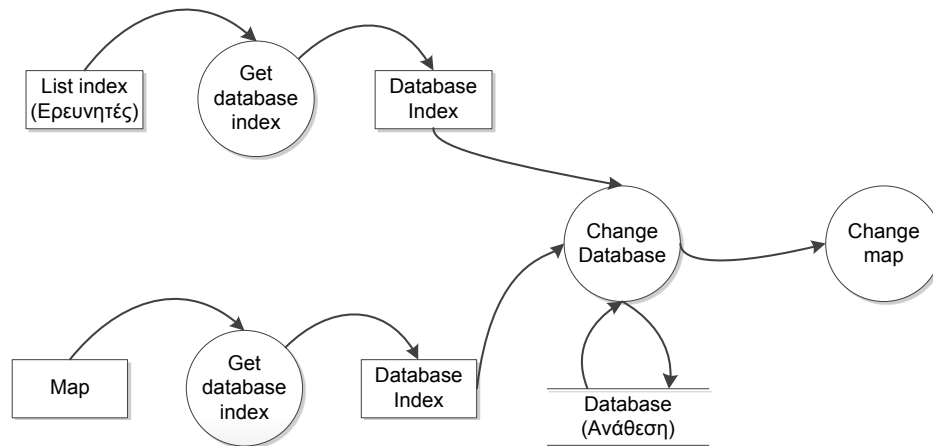


### 3.3.1.4 Επιλογή δήμου για ερευνητή

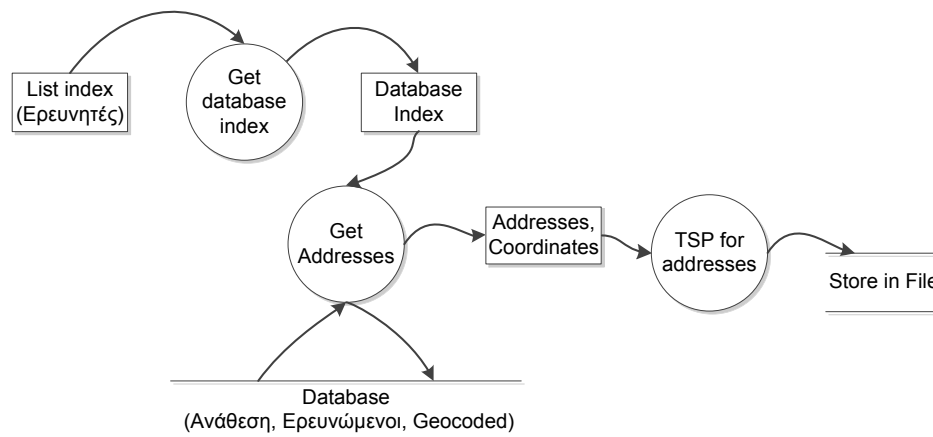




### 3.3.1.8 Ανάθεση-διαγραφή ερευνώμενου σε ερευνητή



### 3.3.1.9 Ταξινόμηση Ερευνώμενων για ερευνητή



## 3.4 Μοντέλο Οντοτήτων Συσχετίσεων

Η εφαρμογή μας χρησιμοποιεί μια βάση δεδομένων για την μόνιμη αποθήκευση των στοιχείων που θέλουμε. Οι πληροφορίες που θέλουμε να αποθηκεύονται μόνιμα είναι πολλές και συνδέονται μεταξύ τους, οπότε η χρήση βάσης δεδομένων για την διαχείριση τους κρίθηκε απαραίτητη.

Τα ΣΥΝΟΛΑ ΟΝΤΟΤΗΤΩΝ της βάσης δεδομένων που σχεδιάσαμε είναι τα εξής:

- **Ερευνητής.** Η οντότητα ερευνητής. Κάθε ερευνητής προσδιορίζεται μοναδικά από το γνώρισμα id (πρωτεύον κλειδί). Για κάθε ερευνητή έχουμε τα γνωρίσματα Όνομα, Επώνυμο, Τηλέφωνο και Σημειώσεις.
- **Επιλεγμένος.** Εξειδίκευση της οντότητας Ερευνητής. Είναι το υποσύνολο των ερευνητών που έχουν επιλεγεί να συμμετέχουν στην παρούσα έρευνα.

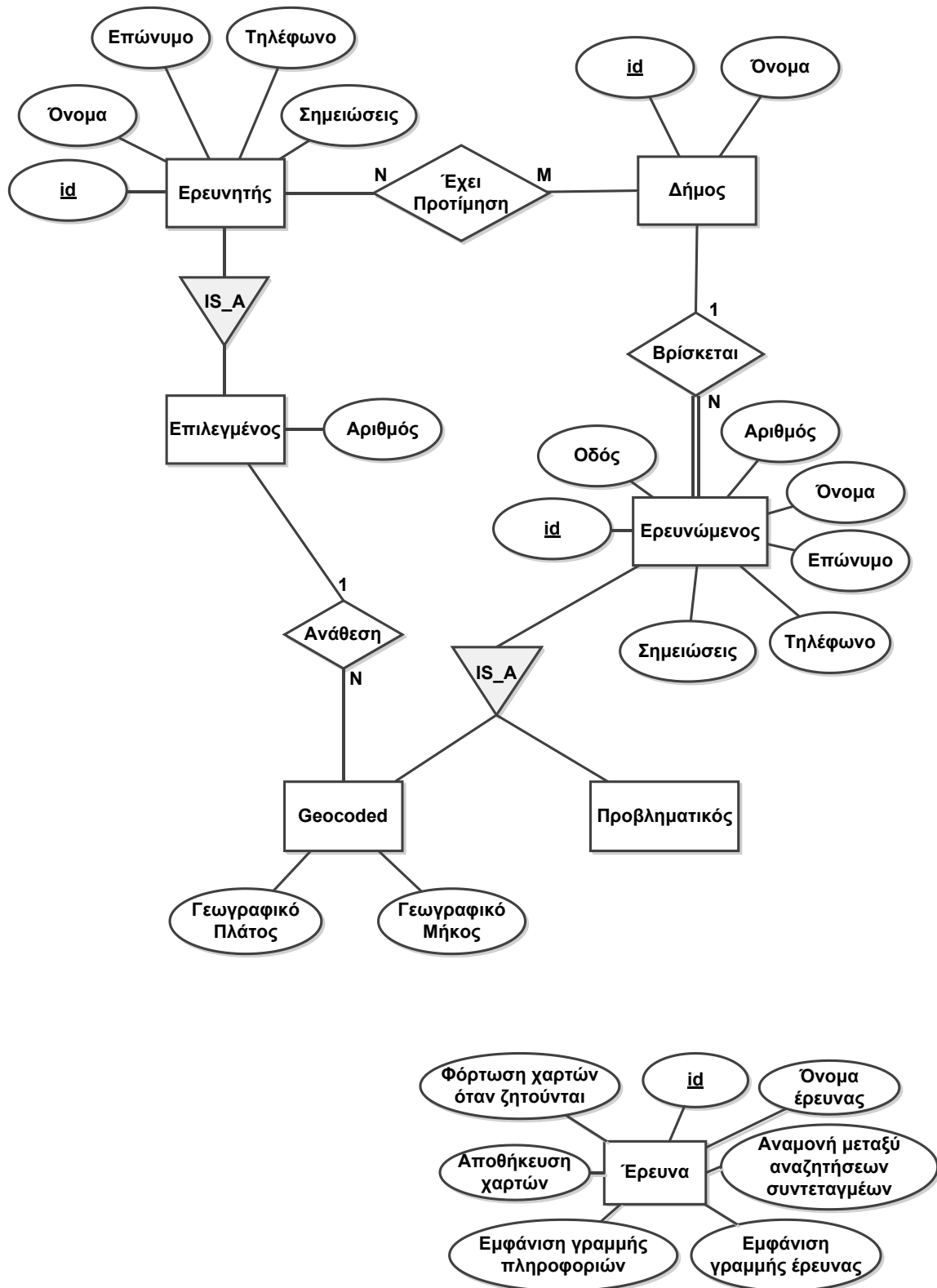
Χαρακτηρίζεται από το γνώρισμα Αριθμός, που είναι ο αριθμός των ερευνώμενων που επιθυμεί να επισκεφτεί ο ερευνητής.

- **Δήμος.** Η οντότητα δήμος αναφέρεται στους πιθανούς δήμους που έχουν δηλώσει προτίμηση οι ερευνητές, και στους δήμους που ανήκουν οι ερευνώμενοι. Πρωτεύον κλειδί εδώ είναι το id του δήμου. Μόνο του γνώρισμα είναι το Όνομα του.
- **Ερευνώμενος.** Οι ερευνώμενοι που συμμετέχουν σε μια έρευνά μας. Προσδιορίζονται μοναδικά από το γνώρισμα id, όπως και οι άλλες οντότητες. Τα γνωρίσματα που φυλάσσονται για κάθε ερευνώμενο είναι η διεύθυνσή του: οδός και αριθμός, Όνομα, Επώνυμο, Τηλέφωνο και Σημειώσεις.
- **Προβληματικός.** Πρόκειται για εξειδίκευση της οντότητας Ερευνώμενος. Εδώ ανήκουν οι ερευνώμενοι των οποίων η διεύθυνση δεν μπόρεσε να αντιστοιχηθεί από την εφαρμογή μας με συντεταγμένες. Πιθανότατα υπάρχει κάποιο λάθος στην διεύθυνση του εν λόγω ερευνώμενου.
- **Geocoded.** Εξειδίκευση της οντότητας Ερευνώμενος. Εδώ ανήκουν οι ερευνώμενοι για τους οποίους έχουν βρεθεί οι συντεταγμένες. Έχει γνωρίσματα τα Γεωγραφικό Πλάτος ( longitude) και Γεωγραφικό Μήκος (latitude).
- **Έρευνα.** Αυτή η οντότητα δεν συνδέεται με την υπόλοιπη βάση δεδομένων. Έχει πάντα μόνο μια εγγραφή. Τα χαρακτηριστικά της είναι το όνομα της παρούσας έρευνας και κάποιες άλλες μεταβλητές που αντιστοιχούν σε κάποιες επιλογές λειτουργίας της εφαρμογής. Οι μεταβλητές αυτές αποθηκεύονται στην βάση, έτσι ώστε η εφαρμογή να έχει την εικόνα που είχε την τελευταία φορά που έκλεισε.

Τα ΣΥΝΟΛΑ ΣΥΣΧΕΤΙΣΕΩΝ είναι:

- **Έχει Προτίμηση.** Συνδέει τους ερευνητές με τους δήμους και δηλώνει την προτίμηση κάθε ερευνητή σε δήμους. Η σχέση είναι πολλά προς πολλά. Η συμμετοχή δεν είναι ολική για καμία από τις οντότητες.
- **Βρίσκεται.** Συνδέει ερευνώμενους με δήμους. Στην ουσία είναι η σχέση που συμπληρώνει την διεύθυνση κάθε ερευνώμενου με βάση τον δήμο στον οποίο βρίσκεται. Η σχέση είναι ολική από την πλευρά των ερευνώμενων καθώς όλοι πρέπει να ανήκουν σε ένα δήμο. Επίσης η σχέση είναι ένα προς πολλά, αφού κάθε ερευνώμενος ανήκει μόνο σε ένα δήμο, ενώ σε ένα δήμο μπορεί να ανήκουν πολλοί ερευνώμενοι.
- **Ανάθεση.** Είναι η σχέση των αποτελεσμάτων. Ενώνει τους επιλεγμένους ερευνητές με τους ερευνώμενους που έχουν βρεθεί οι συντεταγμένες τους. Η σχέση είναι ένα προς πολλά, αφού κάθε επιλεγμένος ερευνητής αντιστοιχείται σε πολλούς ερευνώμενους, αλλά κάθε ερευνώμενος μόνο σε έναν ερευνητή. Η σχέση δεν είναι

ολική για καμία οντότητα, αφού εξαρτάται από τον χρήστη αν και που θα αντιστοιχηθούν οι ερευνώμενοι. Χαρακτηρίζεται από το γνώρισμα Σειρά Επίσκεψης.



Εικόνα 3.14 Μοντέλο Οντοτήτων Συσχετίσεων.





# 4

Σ

## χεδίαση Συστήματος

Ακολουθεί η περιγραφή της σχεδίασης του συστήματος.

### 4.1 Αρχιτεκτονική

Η εφαρμογή μας αποτελείται από 47 κλάσεις οι οποίες χωρίζονται σε 5 βασικά πακέτα:

- πακέτο Geocode: Έχει τις κατάλληλες κλάσεις για την γεωκωδικοποίηση διευθύνσεων και την διαχείριση χαρτών.
- πακέτο Gui: Το γραφικό περιβάλλον της εφαρμογής
- πακέτο Tsp: Πακέτο με κλάσεις που υλοποιούν τον αλγόριθμο του Χριστοφίδη και τυπώνουν τα αποτελέσματα σε αρχείο.
- πακέτο Print: Ασχολείται με το τι τυπώνεται κατά το debugging στη γραμμή πληροφοριών της εφαρμογής.
- πακέτο Sql: Ασχολείται με την επικοινωνία με την βάση δεδομένων.

Αναλυτικά η δομή του κώδικά μας είναι η εξής:

```
thesis
  geocode
    Geocode.java
    GoogleStaticImage.java
    Urlmake.java
  gui
    Footer.java
    GuiManager.java
    Header.java
    MainPanel.java
    NeaEreuna.java
  addresses
    GeocodeThread.java
    GuiAddresses.java
```

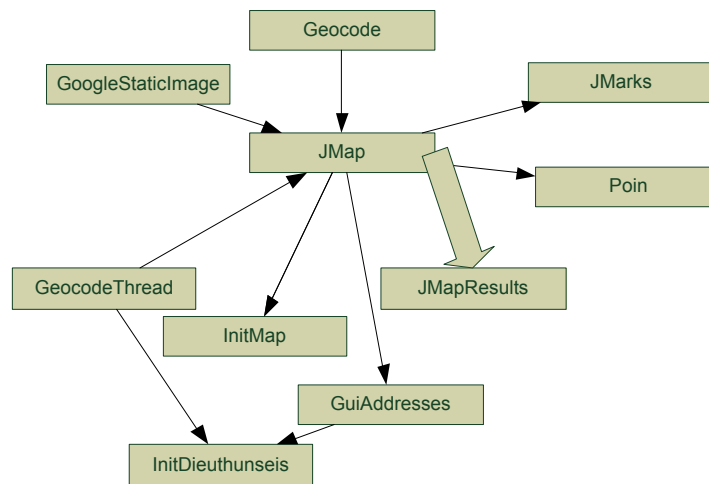
```

    init
        FileLoad.java
        InitDieuthunseis.java
        InitMap.java
    map
        JMap.java
        JMarks.java
        Marks.java
        Poin.java
    popupWindows
        AddAddress.java
        ChangeAddress.java
ereunhtes
    CheckBoxRendererer.java
    DhmoiRendererer.java
    Get.java
    GuiEreunhtes.java
    init
        InitDhmoi.java
        InitEreunhtes.java
    popupWindows
        AddDhmoi.java
        AddEreunhth.java
        ChangeDhmo.java
        ChangeEreunhth.java
        MergeDhmo.java
    results
        GuiResults.java
        JMapResults.java
    init
        InitEreunhtesAddresses.java
        InitResults.java
    init
        Global.java
print
    Print.java
sql
    MySqlConnection.java
tsp
    Edge.java
    Euler.java
    FromCoordinatesToFile.java
    MinimumCostPerfectMatching.java
    Tsp.java

```

**Πίνακας 4.1** Δομή του κώδικα

Τα αρχεία GuiEreunhtes, GuiAddresses, GuiResults περιέχουν 2, 3 και 3 κλάσεις αντίστοιχα.



Εικόνα 4.1 Block Διάγραμμα της εφαρμογής - Οι συνδέσεις της κλάσης JMap.

## 4.2 Περιγραφή Κλάσεων

Όπως είδαμε στο κεφάλαιο 3 η εφαρμογή αποτελείται από 5 βασικά κομμάτια. Αυτά τα κομμάτια, όπως παρουσιάστηκαν, μπορούν να χωριστούν σε 3 παράθυρα:

- Ένα παράθυρο διαχείρισης ερευνητών.
- Ένα παράθυρο διαχείρισης ερευνώμενων και των διεθνύσεών τους.
- Ένα παράθυρο για την παραγωγή των αποτελεσμάτων. Την κατανομή και την κατάταξη δηλαδή των ερευνώμενων σε ερευνητές.

Τα παράθυρα αυτά μαζί υλοποιούνται στα υποπακέτα: `gui.addresses`, `gui.ereunhtes`, `gui.results`.

Πέραν του πακέτου `gui`, που υλοποιεί τα γραφικά της εφαρμογής, υπάρχουν, όπως προαναφέρθηκε, άλλα 4 πακέτα: `tsp`, `sql`, `geocode`, `print`.

Παρακάτω παρουσιάζονται συνοπτικά οι λειτουργίες/μέθοδοι/συναρτήσεις των κλάσεων της εφαρμογής.

### 4.2.1 Ερευνητές

Εδώ θα μελετήσουμε το πακέτο `gui.ereunhtes`. Το κομμάτι αυτό αποτελείται από 12 κλάσεις:

<pre> <b>ereunhtes</b>   CheckBoxRenderer.java   DhmoiRenderer.java   Get.java   GuiEreunhtes.java ( 2 κλάσεις ) <b>init</b>   InitDhmoi.java   InitEreunhtes.java <b>popupWindows</b>   AddDhmoi.java   AddEreunthth.java   ChangeDhmo.java   ChangeEreunthth.java   </pre>
--

#### 4.2.1.1 *CheckBoxRenderer*

Η κλάση αυτή διαχειρίζεται το κομμάτι επιλογών της λίστας με τους δήμους.

#### 4.2.1.2 *DhmoiRenderer*

Διαχειρίζεται τη λίστα των δήμων. Φροντίζει να δοθεί έντονο μαύρο χρώμα στο όνομα των δήμων που ανήκουν στην προτίμηση κάποιου ερευνητή και γκρι χρώμα στο όνομα των δήμων που δεν ανήκουν στην προτίμηση κανενός ερευνητή.

#### 4.2.1.3 *Get*

Κλάση για τη διαχείριση του πάνελ με τους ερευνητές που έχουν επιλεγθεί. Διαθέτει κατάλληλες μεθόδους για την εισαγωγή του αριθμού των ερευνώμενων που επιθυμεί να επισκεφθεί ο κάθε ερευνητής.

#### 4.2.1.4 *GuiEreunhtes*

Η βασική κλάση που υλοποιεί το πάνελ για την διαχείριση ερευνητών και δήμων. Διαθέτει κατάλληλες μεθόδους για την απόκριση στο πάτημα των κουμπιών καθώς και τις κατάλληλες ενέργειες όταν επιλέγονται δήμοι ή ερευνητές από την λίστα. Στην ουσία αυτή η κλάση διασυνδέεται με όλες τις υπόλοιπες του πακέτου. Στο αρχείο *GuiEreunhtes.java* υπάρχει και μία 2<sup>η</sup> κλάση για την διαχείριση συμβάντων του ποντικιού.

#### 4.2.1.5 *InitDhmoi*

Κλάση που χρησιμοποιείται για την αρχικοποίηση της λίστας των δήμων. Κάθε φορά που ανοίγει η εφαρμογή την αρχικοποιεί στην κατάσταση της βάσης δεδομένων. Και κάθε φορά που κάποιος δήμος προστίθεται ή διαγράφεται εκτελεί τις κατάλληλες ενέργειες στην λίστα και στην βάση δεδομένων. Επίσης, έχει τις μεθόδους για την εμφάνιση των κατάλληλων δήμων επιλεγμένων, όταν επιλέγεται ένας ερευνητής και την εισαγωγή ή διαγραφή από την βάση δεδομένων ανάλογα με το αν επιλέγεται ή από-επιλέγεται ένας δήμος για έναν ερευνητή.

#### 4.2.1.6 *InitEreunhtes*

Όμοια με την παραπάνω κλάση αυτή η κλάση έχει μεθόδους για την αρχικοποίηση της λίστας των ερευνητών. Εισάγει τους ερευνητές που υπάρχουν στην βάση δεδομένων. Αλλάζει την βάση δεδομένων όταν προστίθενται, διαγράφονται, αλλάζουν ή επιλέγονται ερευνητές.

#### 4.2.1.7 *AddDhmoi*

Κλάση που υλοποιεί ένα παράθυρο διαλόγου για την εισαγωγή κάποιου δήμου. Ελέγχει το όνομα του δήμου ώστε να μην υπάρχει ήδη και στην συνέχεια το εισάγει στην βάση δεδομένων και στην λίστα χρησιμοποιώντας μεθόδους από την κλάση *InitDhmoi*.

#### 4.2.1.8 *AddEreunhth*

Κλάση που υλοποιεί ένα παράθυρο διαλόγου για την εισαγωγή κάποιου ερευνητή. Ελέγχει το όνομα του ερευνητή ώστε να μην υπάρχει ήδη και στην συνέχεια το εισάγει στην βάση δεδομένων και στην λίστα χρησιμοποιώντας μεθόδους από την κλάση *InitEreunhtes*.

#### 4.2.1.9 *ChangeDhmo*

Κλάση που υλοποιεί ένα παράθυρο διαλόγου για την προβολή του ονόματος ενός δήμου και την πιθανή αλλαγή του.

#### 4.2.1.10 *ChangeEreunhth*

Κλάση που υλοποιεί ένα παράθυρο για την προβολή των στοιχείων ενός ερευνητή και την πιθανή αλλαγή κάποιων από αυτά.

#### 4.2.1.11 *MergeDhmo*

Κλάση για την συγχώνευση δύο δήμων σε έναν με καινούριο όνομα. Η κλάση ανανεώνει και τις αντίστοιχες εγγραφές των ερευνώμενων ώστε να ανήκουν στον σωστό δήμο.

### 4.2.2 *Διευθύνσεις*

Εδώ θα μελετήσουμε το πακέτο *gui.addresses*. Το κομμάτι αυτό αποτελείται από 13 κλάσεις.

<pre>addresses   GeocodeThread.java   GuiAddresses.java ( 3 κλάσεις )   <b>init</b>     FileLoad.java     InitDieuthunseis.java     InitMap.java   <b>map</b>     JMap.java     JMarks.java     Marks.java     Poin.java   <b>popupWindows</b>     AddAddress.java     ChangeAddress.java</pre>
---

#### 4.2.2.1 *GeocodeThread*

Η κλάση αυτή δημιουργεί ένα νήμα για την εύρεση των συντεταγμένων όλων των διευθύνσεων (των ερευνώμενων) που δεν έχουν ήδη βρεθεί. Επειδή υπάρχει η επιλογή τα αιτήματα να γίνονται με καθυστέρηση 35 δευτερόλεπτα το ένα από το άλλο, αυτή η κλάση επιτρέπει οι αιτήσεις να γίνονται σε άλλο νήμα και να μην σταματάει η εφαρμογή. Επίσης μια μπάρα δείχνει την πρόοδο της διαδικασίας στο κέντρο της εφαρμογής.

#### 4.2.2.2 *GuiAddresses*

Η κύρια κλάση που υλοποιεί και διαχειρίζεται το πάνελ για την διαχείριση των ερευνώμενων και των διευθύνσεων τους. Έχει τις κατάλληλες μεθόδους για εισαγωγή διευθύνσεων από αρχείο, εισαγωγή διαγραφή, αλλαγή ερευνώμενων. Συνδέεται στην ουσία με όλες τις υπόλοιπες κλάσεις του πακέτου. Το αρχείο *GuiAddresses.java* περιέχει ακόμα τις κλάσεις: *BoxRenderer* και *NotFilled*. Η *BoxRenderer* διαχειρίζεται την λίστα με τις διευθύνσεις των ερευνώμενων. Δηλαδή, καθορίζει ποια θα είναι επιλεγμένη και τη χρώμα θα έχει. Η *NotFilled* είναι στην ουσία το εικονίδιο που εμφανίζεται δίπλα στις διευθύνσεις για τις οποίες δεν έχουν βρεθεί συντεταγμένες.

#### 4.2.2.3 *FileLoad*

Κλάση για την εισαγωγή των διευθύνσεων (ερευνώμενων) από αρχείο.

#### 4.2.2.4 *InitDieuthunseis*

Κλάση για την αρχικοποίηση της λίστας των ερευνώμενων. Περιέχει κατάλληλες μεθόδους για προσθήκη, διαγραφή και αλλαγή ερευνώμενου. Επίσης διαχειρίζεται τους πίνακες Προβληματικός και Ερευνώμενος της βάσης δεδομένων.

#### 4.2.2.5 *InitMap*

Κλάση για την αρχικοποίηση των διευθύνσεων που εμφανίζονται στον χάρτη. Περιέχει μεθόδους για προσθήκη ή αφαίρεση διευθύνσεων από τον χάρτη.

#### 4.2.2.6 *JMap*

Η κλάση αυτή υλοποιεί τον διαδραστικό χάρτη. Η κλάση περιέχει 34 μεθόδους, οι οποίες βοηθούν στο να εμφανίζεται σωστά ο χάρτης, να είναι διαδραστικός, να αντιδρά σε συμβάντα του ποντικιού και να εμφανίζονται στο κατάλληλο σημείο οι καρφίτσες των διευθύνσεων.

#### 4.2.2.7 *JMarks*

Περιέχει μεθόδους που δημιουργούν το γραφικό (την καρφίτσα) που εμφανίζεται στον χάρτη για να δείχνει διευθύνσεις.

#### 4.2.2.8 *Marks*

Μια δεύτερη κλάση που δημιουργεί διαφορετικό γραφικό από την προηγούμενη.

#### 4.2.2.9 *Poin*

Κλάση αντίστοιχη με την κλάση Point της java που υλοποιεί όμως σημεία με ακρίβεια πραγματικού αριθμού. Χρησιμοποιείται για την αναπαράσταση πραγματικών γεωγραφικών συντεταγμένων.

#### 4.2.2.10 *AddAddress*

Παράθυρο διαλόγου για την προσθήκη ερευνώμενου. Απαραίτητο στοιχείο για την προσθήκη η διεύθυνσή του. Κατά την εισαγωγή ελέγχεται ότι δεν υπάρχει πανομοιότυπος ερευνώμενος.

#### 4.2.2.11 *ChangeAddress*

Παράθυρο διαλόγου για την προβολή των στοιχείων ενός ερευνώμενου και πιθανή αλλαγή κάποιων από αυτά.

### 4.2.3 *Αποτελέσματα*

Εδώ θα μελετήσουμε το πακέτο gui.results. Το κομμάτι αυτό αποτελείται από 6 κλάσεις.

<pre>results   GuiResults.java (3 κλάσεις)   JMapResults.java   init     InitEreunhtesAddresses.java     InitResults.java</pre>
---

#### 4.2.3.1 *GuiResults*

Είναι η κλάση που διαχειρίζεται το παράθυρο για την παραγωγή αποτελεσμάτων. Έχει ένα χάρτη μία λίστα των επιλεγμένων ερευνητών και μια λίστα των δήμων. Στο ίδιο αρχείο βρίσκονται και οι κλάσεις: EreunhtesRender, DhmoiRender οι οποίες διαχειρίζονται τα κουτάκια με τους αριθμούς δίπλα σε κάθε στοιχείο των δύο λιστών.

#### 4.2.3.2 *JMapResults*

Ο χάρτης του παραθύρου. Κληρονομεί από την κλάση *Jmap*. Προσθέτει κάποιες λειτουργίες, όπως διαφορετικά χρώματα για τις διευθύνσεις (καρφίτσες), την δυνατότητα επιλογής διευθύνσεων (ερευνώμενων) και την ανάθεσή τους σε ερευνητές.

#### 4.2.3.3 *InitEreunhtesAddresses*

Κλάση που βοηθά στην αρχικοποίηση των διευθύνσεων και καθορίζει το χρώμα με το οποίο θα εμφανίζεται η καθεμία. Ποια έχει ανατεθεί και ποια όχι. Διαθέτει μεθόδους για ανάθεση και διαγραφή των διευθύνσεων σε και από ερευνητές.

#### 4.2.3.4 *InitResults*

Αρχικοποιεί και διαχειρίζεται τις 2 λίστες του παραθύρου. Τους δήμους και τους επιλεγμένους ερευνητές δηλαδή.

### 4.2.4 **Κεντρικό Μενού**

Εδώ περιλαμβάνονται οι κλάσεις που δημιουργούν το βασικό μενού της εφαρμογής. Οι κλάσεις είναι οι εξής:

```
gui
    Footer.java
    GuiManager.java
    Header.java
    MainPanel.java
    NeaEreuna.java
    init
        Global.java
```

#### 4.2.4.1 *Footer*

Η κλάση δημιουργεί το πάνελ που εμφανίζεται στο κάτω μέρος της εφαρμογής και τυπώνει τι κάνει η εφαρμογή κατά την διάρκεια λειτουργίας της.

#### 4.2.4.2 *GuiManager*

Δημιουργεί το παράθυρο της εφαρμογής με τα μενού του. Είναι η κλάση που διαχειρίζεται πιο πάνελ εμφανίζει η εφαρμογή και πως αυτά εναλλάσσονται.

#### 4.2.4.3 *Header*

Κλάση που δημιουργεί ένα βοηθητικό πάνελ στο οποίο περιέχονται κουμπιά συντομεύσεις για το πάνελ δημιουργίας νέας έρευνας και το κυρίως μενού. Επίσης περιέχει και το όνομα της παρούσας έρευνας.



#### 4.2.4.4 *MainPanel.java*

Το κυρίως πάνελ της εφαρμογής. Έχει 3 κουμπιά που οδηγούν στα πάνελ : διαχείρισης ερευνητών, διαχείρισης διευθύνσεων και αποτελεσμάτων.

#### 4.2.4.5 *NeaEreuna.java*

Πάνελ για την δημιουργία νέας έρευνας. Αρχικοποιεί την εφαρμογή για την νέα έρευνα.

#### 4.2.4.6 *Global.java*

Κλάση που κρατά της μεταβλητές της εφαρμογής που χρησιμοποιούνται από πολλές άλλες κλάσεις.

### 4.2.5 **Πακέτο geocode**

Το πακέτο αυτό χρησιμοποιείται από το παράθυρο διαχείρισης ερευνώμενων για την δημιουργία του χάρτη και την εύρεση συντεταγμένων των ερευνώμενων.

Το πακέτο περιλαμβάνει 3 κλάσεις.

<b>geocode</b> Geocode.java GoogleStaticImage.java Urlmake.java
--

#### 4.2.5.1 *Geocode*

Η κλάση αυτή περιέχει μεθόδους για την εύρεση συντεταγμένων από διευθύνσεις (geocode). Επίσης περιέχει μεθόδους για την εύρεση διευθύνσεων από συντεταγμένες (reverse geocode). Κάποιες από τις μεθόδους της έχουν παρθεί από ιστοσελίδα (βλέπε <sup>6</sup> παρακάτω).

#### 4.2.5.2 *GoogleStaticImage*

Διαχείριση των εικόνων που αποτελούν τον χάρτη της εφαρμογής. Η παρούσα κλάση περιέχει τις μεθόδους που αναζητούν τον κατάλληλο χάρτη είτε από το διαδίκτυο (υπηρεσία της Google) είτε από την μνήμη ή τον σκληρό δίσκο. Περιέχει επίσης μεθόδους για το καθορισμό των ιδιοτήτων των εικόνων : που θα εμφανίζονται στον χάρτη και σε ποιο επίπεδο εστίασης.

#### 4.2.5.3 *Urlmake*

Περιέχει μεθόδους για την δημιουργία HTTP αιτήσεων. Ο κώδικάς της έχει παρθεί από το ιστοσελίδα (βλέπε υποσημείωση <sup>6</sup> παρακάτω)).

#### 4.2.6 Πακέτα *Print - sql*

Τα πακέτα `print` και `sql` έχουν από μία κλάση . Γι' αυτό τα αναφέρουμε μαζί.

<pre><b>print</b>     Print.java <b>sql</b>     MySqlConnection.java</pre>
--

##### 4.2.6.1 *Print*

Η κλάση αυτή διαχειρίζεται ότι τυπώνει η εφαρμογή για debugging. Στην εφαρμογή υπάρχει ένα παράθυρο το οποίο εμφανίζει ότι πρέπει να τυπώσει η εφαρμογή.

##### 4.2.6.2 *MySqlConnection*

Η κλάση διαχειρίζεται όλη την επικοινωνία με την βάση δεδομένων. Κάθε επιλογή , αλλαγή ή διαγραφή που συμβαίνει στην βάση δεδομένων κατά την λειτουργία της εφαρμογής γίνεται με κλήση κάποιας μεθόδου της κλάσης αυτής.

#### 4.2.7 Πακέτο *tsp*

Το πακέτο αυτό έχει τις απαραίτητες κλάσεις για την υλοποίηση του αλγορίθμου του Χριστοφίδη. Περιέχει τις παρακάτω 5 κλάσεις:

<pre><b>tsp</b>     Edge.java     Euler.java     FromCoordinatesToFile.java     MinimumCostPerfectMatching.java     Tsp.java</pre>
--

##### 4.2.7.1 *Edge*

Κλάση που αναπαριστά μια ακμή ενός γράφου.

##### 4.2.7.2 *Euler*

Κλάση που υλοποιεί αλγόριθμο εύρεσης κύκλου του Euler σε γράφο που έχει. Έχει παρθεί από βιβλίο (βλέπε <sup>8</sup> παρακάτω).

##### 4.2.7.3 *FromCoordinatesToFile.java*

Κλάση που δέχεται ως είσοδο έναν ερευνητή και εξάγει σε αρχείο την προτεινόμενη σειρά ερευνώμενων.

#### 4.2.7.4 *MinimumCostPerfectMatching.java*

Υλοποιεί αλγόριθμο εύρεσης του ελαχίστου κόστους perfect matching ενός πλήρους γράφου με άρτιο αριθμό κορυφών. Έχει παρθεί από βιβλίο (βλέπε <sup>8</sup> παρακάτω)..

#### 4.2.7.5 *Tsp.java*

Κλάση που υλοποιεί τον αλγόριθμο του Χριστοφίδη για την επίλυση του προβλήματος του πλανόδιου πωλητή. Χρησιμοποιεί και τους άλλους αλγορίθμους που υπάρχουν στο παρόν πακέτο.

### 4.3 *Βάση Δεδομένων*

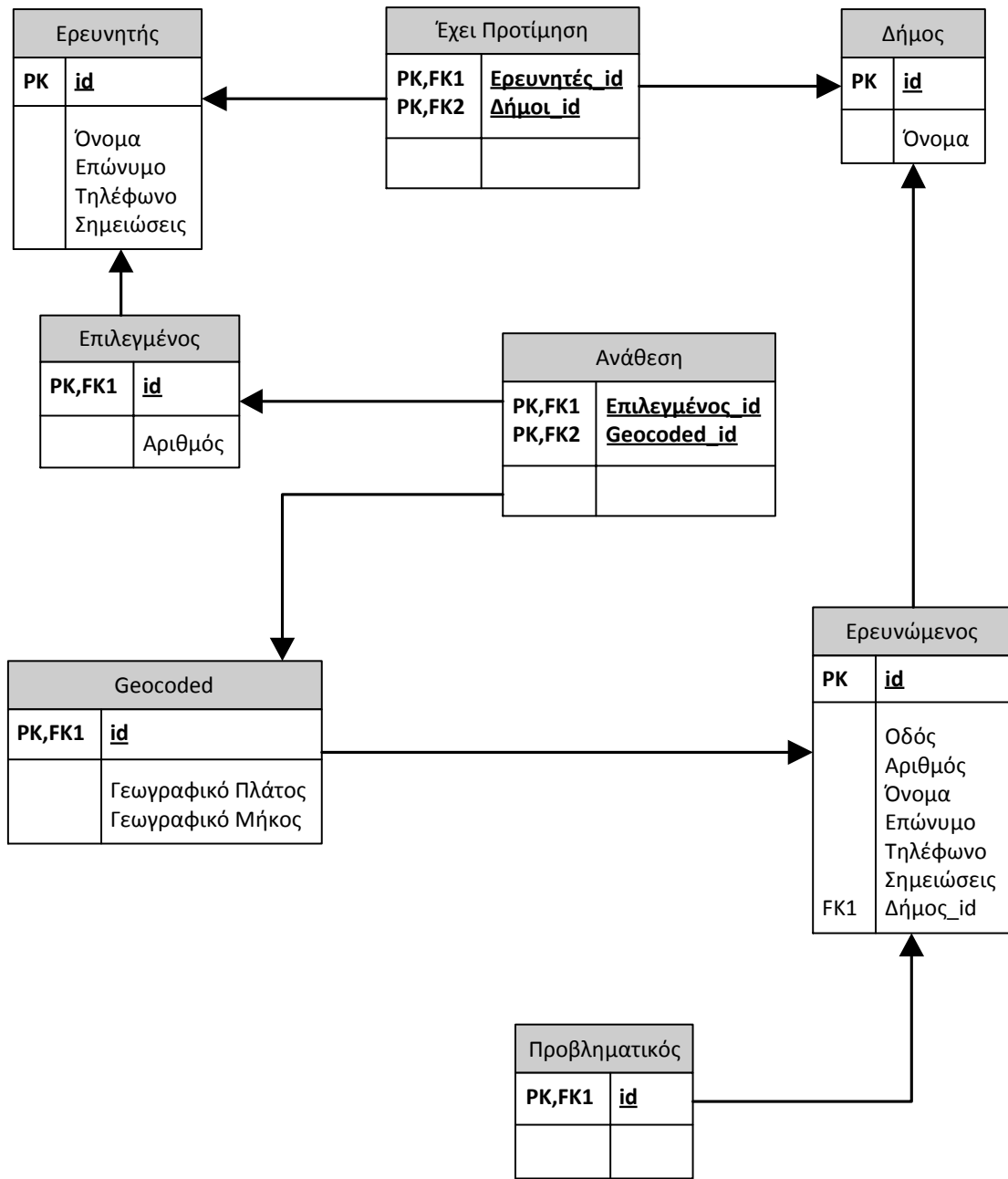
Δίνεται εδώ το σχεσιακό σχήμα της βάσης δεδομένων που χρησιμοποιεί η εφαρμογή μας. Κάθε σύνολο οντοτήτων του μοντέλου οντοτήτων-συσχετίσεων (βλέπε 3.4) έχει μετατραπεί σε σχέση.

- **Ερευνητής** (Κωδικός, Όνομα, Επώνυμο, Τηλέφωνο, Σημειώσεις)
- **Επιλεγμένος** (Κωδικός, Αριθμός)
- **Δήμος** (Κωδικός, Όνομα)
- **Ερευνώμενος** (Κωδικός, Οδός, Αριθμός, Όνομα, Επώνυμο, Τηλέφωνο, Σημειώσεις, Κωδικός\_Δήμου)
- **Προβληματικός** (Κωδικός\_Ερευνώμενου)
- **Geocoded** (Κωδικός\_Ερευνώμενου, Γεωγραφικό πλάτος, Γεωγραφικό μήκος)
- **Έρευνα** (Κωδικός, Όνομα έρευνας, Εμφάνιση γραμμής έρευνας, Εμφάνιση γραμμής πληροφοριών, Αναμονή μεταξύ αναζητήσεων συντεταγμένων, Αποθήκευση χαρτών, Φόρτωση χαρτών όταν ζητούνται)

Να σημειώσουμε ότι η συσχέτιση **Βρίσκεται** που συνδέει τους ερευνώμενους με δήμους έχει γίνει ένα επιπλέον πεδίο Κωδικός\_δήμου στην σχέση ερευνώμενος. Αυτό το πεδίο δείχνει στον αντίστοιχο δήμο.

Τα ΣΥΝΟΛΑ ΣΥΣΧΕΙΣΕΩΝ μετατρέπονται στις παρακάτω σχέσεις:

- **Έχει Προτίμηση** (Κωδικός\_Ερευνητή, Κωδικός\_Δήμου)
- **Ανάθεση** (Κωδικός\_Επιλεγμένου, Κωδικός\_Geocoded)



Έρευνα	
<b>PK</b>	<u><b>id</b></u>
	Όνομα Έρευνας Εμφάνιση γραμμής έρευνας Εμφάνιση γραμμής πληροφοριών Αναμονή μεταξύ αναζητήσεων συντεταγμένων Αποθήκευση χαρτών Φορτωση χαρτών όταν ζητούνται

Εικόνα 4.2 Σχεσιακό σχήμα.

## 4.4 Κωδικοποίηση αρχείων

Στο σύστημά μας γίνεται χρήση δύο ειδών αρχείων. Πρώτον είναι δυνατόν να εισαχθούν ερευνώμενοι από αρχείο. Δεύτερον η πρόταση διαδρομής για κάθε ερευνητή εξάγεται σε αρχείο.

### 4.4.1 Αρχείο Ερευνώμενων

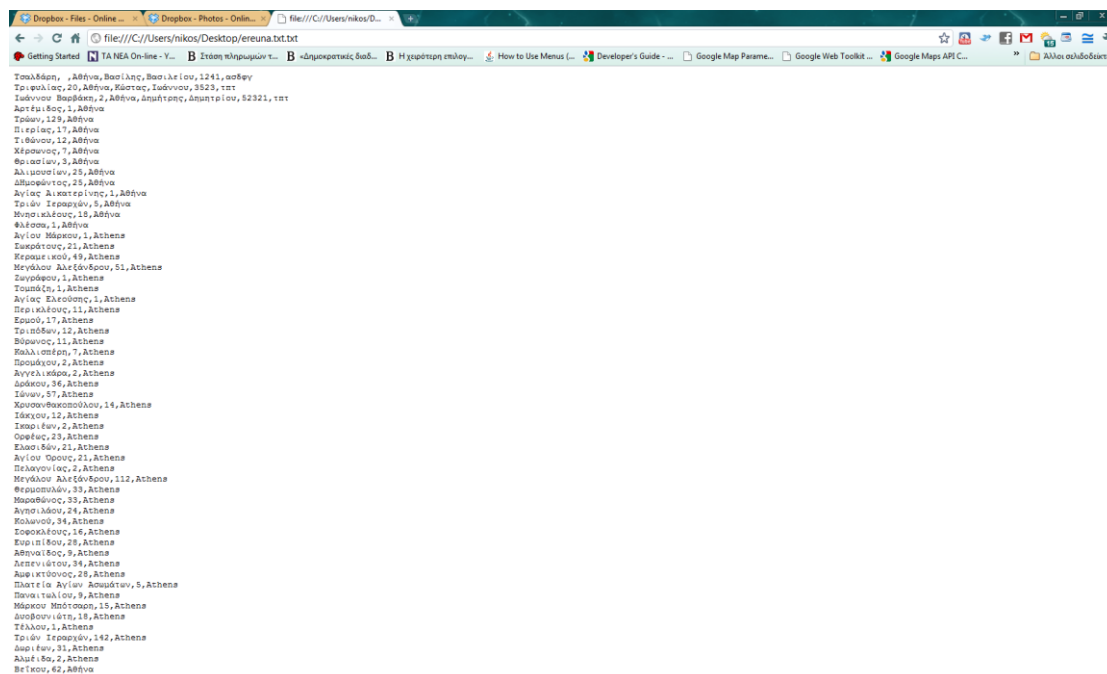
Το αρχείο αυτό περιέχει ερευνώμενους, έναν σε κάθε γραμμή. Η δομή που δίνουμε, τα στοιχεία του κάθε ερευνώμενου, είναι η εξής:

Οδός, Αριθμός, Δήμος, <Όνομα>, <Επώνυμο>, <Τηλέφωνο>, <Σημειώσεις>

όπου είναι απαραίτητο να δοθούν η οδός, ο αριθμός και ο δήμος, δηλαδή η διεύθυνση του κάθε ερευνητή, ενώ όλα τα υπόλοιπα πεδία είναι προαιρετικά.

Αν λείπει μια πληροφορία όπως πχ. το «Τηλέφωνο» αλλά υπάρχει μια επόμενη όπως οι «Σημειώσεις» πρέπει να αφηθεί κενό μεταξύ των δύο κομμάτων. Διαφορετικά οι «Σημειώσεις» θα θεωρηθούν ως το «Τηλέφωνο».

Το αρχείο μπορεί να τοποθετηθεί οπουδήποτε στον σκληρό δίσκο με οποιοδήποτε όνομα. Πρέπει όμως να είναι τύπου txt. Διαφορετικά ενδέχεται να μην αναγνωριστεί σωστά από την εφαρμογή. Όταν επιλεγεί εισαγωγή από αρχείο τότε ζητείται από τον χρήστη να βρεθεί το επιθυμητό αρχείο.



Εικόνα 4.3 Παράδειγμα αρχείου εισαγωγής ερευνώμενων.

#### 4.4.2 Αρχείο πρότασης διαδρομής

Όταν ο χρήστης (αφού έχει αναθέσει τους ερευνώμενους) επιλέξει να κάνει εξαγωγή σε αρχείο των αποτελεσμάτων τότε δημιουργούνται από ένα αρχείο για κάθε ερευνητή (ή μόνο ένα αν έχει επιλεγεί εξαγωγή μόνο για επιλεγμένο ερευνητή). Ο χρήστης στη συνέχεια δίνει τον φάκελο στον οποίο θέλει να δημιουργηθούν τα αρχεία.

Το κάθε αρχείο που δημιουργείται έχει όνομα το ονοματεπώνυμο του ερευνητή και είναι αρχείο txt. Η δομή που έχει το αρχείο είναι η εξής:

```

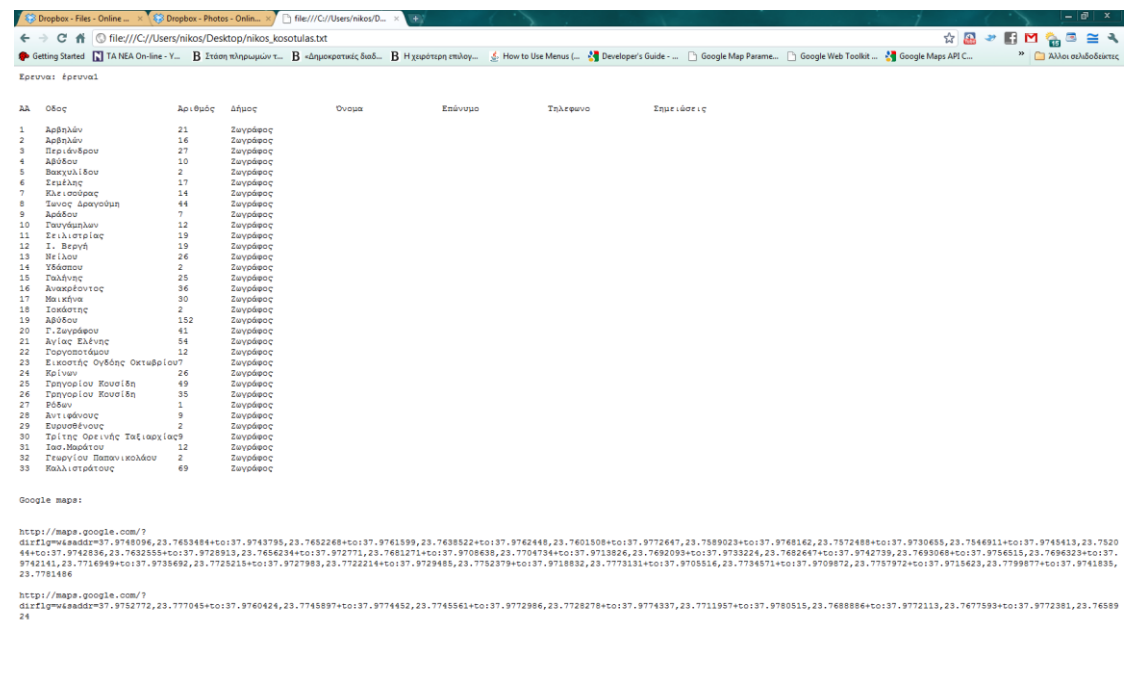
Ερευνα: Όνομα Έρευνας

AA Οδός Αριθμός Δήμος Όνομα Επώνυμο Τηλέφωνο Σημειώσεις
1 ..... ..... ..... ..... ..... ..... .....
2 ..... ..... ..... ..... ..... ..... .....
...
n ..... ..... ..... ..... ..... ..... .....

Google Maps

Link1
Link2
...
    
```

Δίνονται με προτεινόμενη σειρά οι ερευνώμενοι και στο τέλος ανά 25 ερευνώμενους δίνεται ένας σύνδεσμος στην υπηρεσία: [maps.google.com](https://maps.google.com) με κατάλληλες παραμέτρους ώστε να εμφανιστεί στον χάρτη η προτεινόμενη σειρά επίσκεψης των 25 (το πολύ) αυτών ερευνώμενων. Σε αυτή την σελίδα ο κάθε ερευνητής μπορεί να αλλάξει τη σειρά.



Εικόνα 4.4 Παράδειγμα αρχείου προτιμήσεων για κάθε ερευνητή.

# 5

## *Υλοποίηση*

Στο κεφάλαιο αυτό θα συζητήσουμε λεπτομερώς θέματα υλοποίησης του συστήματος και την διαδικασία εγκατάστασής του.

### *5.1 Ενδιαφέροντα σημεία υλοποίησης*

Ακολουθούν μερικά σημαντικά σημεία της υλοποίησης που αξίζουν περισσότερη προσοχή.

#### *5.1.1 Εύρεση συντεταγμένων*

Η διαδικασία εύρεσης συντεταγμένων, όπως έχουμε ήδη αναφέρει, αποτελεί ένα πολύ σημαντικό κομμάτι του συστήματος. Στο κομμάτι αυτό χρησιμοποιείται η διασύνδεση της υπηρεσίας της Google, η Google Geocoding (βλέπε 2.4).

Για κάθε διεύθυνση που θέλουμε να πάρουμε τις συντεταγμένες της, στέλνουμε ένα HTTP αίτημα και επεξεργαζόμαστε τα αποτελέσματα που μας γυρνάνε.

Αρχικά θέλουμε μια μέθοδο η οποία να παίρνει τις παραμέτρους που θα περάσουμε στο αίτημα, και να τις προσαρμόζει σε μια αποδεκτή μορφή για HTTP αίτημα. Σε αυτή την μορφή δεν επιτρέπονται ελληνικοί χαρακτήρες και κενά. Επιτρέπονται μόνο αγγλικοί χαρακτήρες, αριθμοί και τα σύμβολα: - , . , ~. Οι υπόλοιποι χαρακτήρες πρέπει να αναπαρασταθούν στην αντίστοιχη δεκαεξαδική μορφή τους: %xx. Για παράδειγμα το κενό αναπαρίσταται ως: %20 .

Αυτή η μέθοδος (`urlencode`) υπάρχει στην κλάση `urlmake` (οι περισσότερες μέθοδοι που μελετούνται σ' αυτό το κεφάλαιο έχουν παρθεί από φόρουμ της NOKIA<sup>6</sup> και έχουν στην συνέχεια τροποποιηθεί κατάλληλα).

Στη συνέχεια το κατάλληλα μορφοποιημένο αίτημα πρέπει να αποσταλεί για να πάρουμε την απάντηση, που είναι ένα αρχείο `xml`. Αυτό το αναλαμβάνει μια άλλη μέθοδος της κλάσης `urlmake`, η `loadHttpFile`. Την απάντηση την διαχειρίζεται η μέθοδος `geocodeAddress`. Η μέθοδος αυτή αρχικά ελέγχει 2 πράγματα. Καταρχήν ότι η απάντηση έχει γυρίσει με `status_code` "OK". Δηλαδή, δεν έχει γίνει κανένα λάθος και έχει γυρίσει τουλάχιστον ένα αποτέλεσμα. Δεύτερον, ότι η ακρίβεια του αποτελέσματος είναι τύπου `street_address`. Αυτό σημαίνει ότι έχει βρεθεί ακριβώς σε ποιο σημείο είναι η διεύθυνση σε επίπεδο οδού. Κάθε άλλο επίπεδο ακρίβειας (δήμος, πόλη, χώρα κλπ.) είναι μη ικανοποιητικό για το σύστημά μας. Σε περίπτωση, λοιπόν, που κάποια από τις παραπάνω 2 παραμέτρους του αποτελέσματος δεν είναι η επιθυμητή, η μέθοδος σταματάει πετώντας εξαίρεση με το κατάλληλο μήνυμα.

Αν οι παράμετροι είναι οι επιθυμητές τότε η μέθοδος `geocodeAddress` προχωράει βρίσκοντας, στην δομή `xml`, στα αποτελέσματα τις συντεταγμένες (γεωγραφικό πλάτος, γεωγραφικό μήκος) και τις γυρνάει στην εφαρμογή που την κάλεσε.

Παρόμοια διαδικασία με την `geocodeAddress` ακολουθεί και η `reversegeocode`. Αυτή παίρνει ως παράμετρο τις συντεταγμένες ενός σημείου και γυρνάει την διεύθυνση του. Δεν χρειάζεται να ελέγξει τον τύπο του αποτελέσματος, αλλά μόνο το στάτους του.

### 5.1.2 Διαδραστικός χάρτης

Για το σύστημά μας υλοποιήθηκε ένας διαδραστικός χάρτης. Η κεντρική ιδέα είναι η χρήση μιας εικόνας της Αθήνας ως χάρτη. Χρησιμοποιήθηκε για την απόκτηση της εικόνας η διασύνδεση της υπηρεσίας Google Static Maps (βλέπε 2.5). Στο σημείο αυτό να τονιστεί ότι χρησιμοποιήθηκε αυτή η υπηρεσία επειδή δεν είναι διαθέσιμες βιβλιοθήκες για απευθείας χρήση μιας διαδραστικής υπηρεσίας χάρτη όπως για παράδειγμα δίνονται για χρήση σε ιστοσελίδα με `javascript` ή σε κινητό<sup>7</sup>. Με τη χρήση στατικών χαρτών γίνεται δυνατή και η προσωρινή αποθήκευσή τους για πιο γρήγορη λειτουργία του χάρτη.

Η υλοποίηση που πραγματοποιήθηκε επεκτείνει την κεντρική ιδέα και χρησιμοποιεί πολλές εικόνες αποκτημένες μέσω Google Static Maps API, έτσι ώστε να προσφέρει μεγαλύτερο συνολικό χάρτη. Ο χάρτης λοιπόν αποτελείται από πολλές εικόνες τοποθετημένες η μια δίπλα στην άλλη, με τέτοιο τρόπο, ώστε δημιουργείται η αίσθηση ότι δεν είναι πολλές εικόνες αλλά μόνο μια. Η χρήση πολλών εικόνων επιτρέπει στην εφαρμογή την χρήση πολλών επιπέδων εστίασης. Κάθε επίπεδο εστίασης έχει τις δικές του εικόνες. Για να γίνει αυτό δυνατό και ο χάρτης να λειτουργεί σωστά πρέπει κάθε εικόνα να έχει 2 ιδιότητες: 1) την θέση από όπου

<sup>6</sup> [http://wiki.forum.nokia.com/index.php/Google\\_Maps\\_API\\_in\\_Java\\_ME](http://wiki.forum.nokia.com/index.php/Google_Maps_API_in_Java_ME)

<sup>7</sup> <http://code.google.com/apis/maps/documentation/javascript/>



ξεκινάει στον χάρτη, και 2) σε ποιο επίπεδο εστίασης ανήκει και εμφανίζεται. Εννοείται πως όλες οι εικόνες πρέπει να έχουν τις ίδιες διαστάσεις. (Θυμίζουμε εδώ ότι η μέγιστη διάσταση που μπορεί να γυρίσει η υπηρεσία Google Static Maps είναι 640X640 pixels).

Εδώ περιγράφουμε όλες τις χρήσεις που έχει ο χάρτης, ώστε να αναλύσουμε αναλυτικά στην συνέχεια όλες τις λειτουργίες που πρέπει να έχει και πως αυτές υλοποιήθηκαν. Οι χρήσεις που έχει λοιπόν ο χάρτης είναι οι εξής:

- Προβολή των συντεταγμένων των διευθύνσεων
- Δυνατότητα επιλογής νέων διευθύνσεων
- Πληροφορίες σχετικές με τις διευθύνσεις
- Κατανομή των ερευνώμενων

Για να γίνουν οι παραπάνω χρήσεις του χάρτη δυνατές και εύχρηστες, ο χάρτης υλοποιεί τις παρακάτω λειτουργίες:

- Ύπαρξη πολλών επιπέδων εστίασης. Έτσι είναι δυνατή η διάκριση των δρόμων σε μεγάλο επίπεδο εστίασης και η προβολή ολόκληρης της Αθήνας σε μικρό επίπεδο εστίασης.
- Κάθε σημείο του χάρτη (pixel) μπορεί να αντιστοιχιστεί με γεωγραφικές συντεταγμένες και το αντίστροφο.
- Ο χάρτης αντιδρά ποικιλοτρόπως σε συμβάντα του ποντικιού. Μάλιστα υπάρχουν 2 είδη χαρτών. Αυτό της επιλογής και προβολής των διευθύνσεων (στο εξής χάρτης 1) και αυτό της ανάθεσης ερευνώμενων σε ερευνητές (στο εξής χάρτης 2). Οι δύο χάρτες έχουν κάποιες διαφορές στο πως αντιδρούν σε συμβάντα του ποντικιού και στο πως φαίνονται - με τι χρώμα - οι διευθύνσεις πάνω τους.
- Αντιστοίχιση κάθε διεύθυνσης που φαίνεται στον χάρτη με έναν ερευνώμενο στη βάση δεδομένων.
- Όταν αλλάζει το επίπεδο εστίασης ο δείκτης του ποντικιού συνεχίζει να δείχνει στο ίδιο σημείο στον χάρτη.

#### *5.1.2.1 Σωστή λειτουργία του χάρτη*

Ο χάρτης που υλοποιήθηκε λειτουργεί σωστά μόνο για την Αθήνα. Αυτό δεν αντιβαίνει στις απαιτήσεις της διπλωματικής καθώς η διαχείριση των ερευνών αφορά τον νομό της Αττικής.

Γιατί συμβαίνει όμως αυτό;

Για να λειτουργεί σωστά ο χάρτης χρησιμοποιούνται δύο σταθερές. Μία που δείχνει σε πόσες μοίρες γεωγραφικό πλάτος αντιστοιχεί ένα pixel του χάρτη σε επίπεδο εστίασης 19 (σε μικρότερο επίπεδο κάθε pixel αντιστοιχεί στον διπλάσιο αριθμό μοιρών σε σχέση με το αμέσως μεγαλύτερο του, γιατί υποδιπλασιάζεται η ακρίβειά του). Και μία που δείχνει σε πόσες μοίρες γεωγραφικό μήκος αντιστοιχεί ένα pixel του χάρτη σε επίπεδο εστίασης 19. Οι

τιμές αυτές έχουν βρεθεί πειραματικά για την Αθήνα. Όπως είδαμε στο κεφάλαιο 2.3 ο χάρτης της υπηρεσίας Google Maps (και επομένως και της Google Static Maps) είναι φτιαγμένος με Μερκατορική προβολή. Αυτό σημαίνει πως, παρότι ίδια κάθετη απόσταση στον χάρτη θα αντιστοιχεί σε ίδια διαφορά μοιρών (γεωγραφικού μήκους) σε οποιοδήποτε σημείο του χάρτη (άρα και για 1 pixel), ίδια οριζόντια απόσταση θα αντιστοιχεί σε διαφορετική διαφορά γωνίας (γεωγραφικού πλάτους) ανάλογα με το γεωγραφικό πλάτος που βρίσκεται το σημείο (βλέπε Εικόνα 2.3 Χάρτης Μερκατορικής προβολής.).

Το αντίστροφο βέβαια συμβαίνει στην γη. Οι οριζόντιες ισαπέχουν και οι κάθετες συγκλίνουν στους πόλους. Εφόσον λοιπόν η εφαρμογή έχει σταθερές που αντιστοιχούν στο γεωγραφικό πλάτος της Αθήνας λειτουργεί σωστά μόνο σ' αυτή. Πιο βόρεια στην Θεσσαλονίκη για παράδειγμα ο χάρτης χάνει πάνω και κάτω από τις εικόνες λίγα pixel (γύρω στα 5 ανά 500). Ενώ πιο νότια όπως στην Κρήτη ο χάρτης δείχνει δύο φορές τα σημεία στα άκρα των εικόνων.

Ο χάρτης μπορεί να κυλιέται πατώντας «αριστερό κλικ» και «σέρνοντάς τον». Αυτό που αλλάζει στην ουσία είναι το σημείο από το οποίο ξεκινάει ο χάρτης πάνω αριστερά. Για να εμφανίζεται σωστά ο χάρτης κάθε φορά που κυλιέται θα πρέπει να ελέγχεται ότι στο πάνελ που είναι ορατό στην οθόνη υπάρχουν όλες οι εικόνες που συμπληρώνουν τον χάρτη για αυτό το διάστημα. Αυτό το αναλαμβάνει μια μέθοδος της κλάσης GoogleStaticImage. Παρόμοια όταν το παράθυρο μεγαλώνει ή μικραίνει πρέπει να ελέγχεται ότι υπάρχουν όλες οι απαραίτητες εικόνες. Το ίδιο ισχύει και όταν αλλάζει το επίπεδο εστίασης.

Ο χάρτης αποτελείται σε κάθε επίπεδο εστίασης από έναν κεντρικό χάρτη που αρχίζει από το pixel (0,0) πάνω αριστερά και στην ουσία είναι το κέντρο της Αθήνας. Κάθε φορά που φαίνονται περιοχές εκτός αυτού του χάρτη( 600x600 pixels) η εφαρμογή βρίσκει τις κατάλληλες διπλανές εικόνες (υπολογίζοντας τα κέντρα τους) και τις φέρνει.

Κάθε φορά λοιπόν που ανοίγει ο χάρτης κάνει τα εξής: βλέπει σε ποιο σημείο είναι , σε ποιο επίπεδο εστίασης, ποιο είναι το μέγεθός του (πλάτος , μήκος) και ελέγχει αν υπάρχουν στην μνήμη όλες οι κατάλληλες εικόνες. Αν δεν υπάρχουν τις φέρνει είτε από το διαδίκτυο είτε από τον σκληρό δίσκο. Όταν μια παράμετρος από τις παραπάνω αλλάζει πρέπει να ξαναγίνει ο έλεγχος.

#### *5.1.2.2 Αντιστοίχιση κάθε σημείο με γεωγραφικές συντεταγμένες*

Στην εφαρμογή υπάρχουν σταθερές που αντιστοιχούν σε κάθε pixel και προσαρμόζονται για κάθε επίπεδο εστίασης μοίρες γεωγραφικού πλάτους και μήκους. Κάθε εικόνα λαμβάνεται με βάση τις συντεταγμένες του κέντρου της.

Γνωρίζοντας, λοιπόν, τις συντεταγμένες του κέντρου της πρώτης εικόνας είναι εύκολο να βρεθούν οι συντεταγμένες του κέντρου κάθε εικόνας που μπορεί να χρειαστεί, αλλά και οι

συντεταγμένες κάθε σημείου γενικά. Έτσι σε κάθε επίπεδο εστίασης σχεδιάζονται στο σωστό σημείο οι διευθύνσεις.

Όταν ζητείται με «δεξί κλικ» η διεύθυνση κάποιου σημείου στον χάρτη υπολογίζονται οι σωστές συντεταγμένες και στέλνεται το κατάλληλο αίτημα αντίστροφης γεωκωδικοποίησης. Στη συνέχεια αν ο χρήστης επιλέξει να εισάγει την διεύθυνση στο σύστημα τότε ξαναστέλνεται ένα αίτημα γεωκωδικοποίησης αυτή την φορά για εμφανιστεί στο κατάλληλο σημείο η διεύθυνση και όχι στο σημείο που πατήθηκε το ποντίκι (που μπορεί να ήταν αρκετά κοντά αλλά όχι πάνω στον δρόμο).

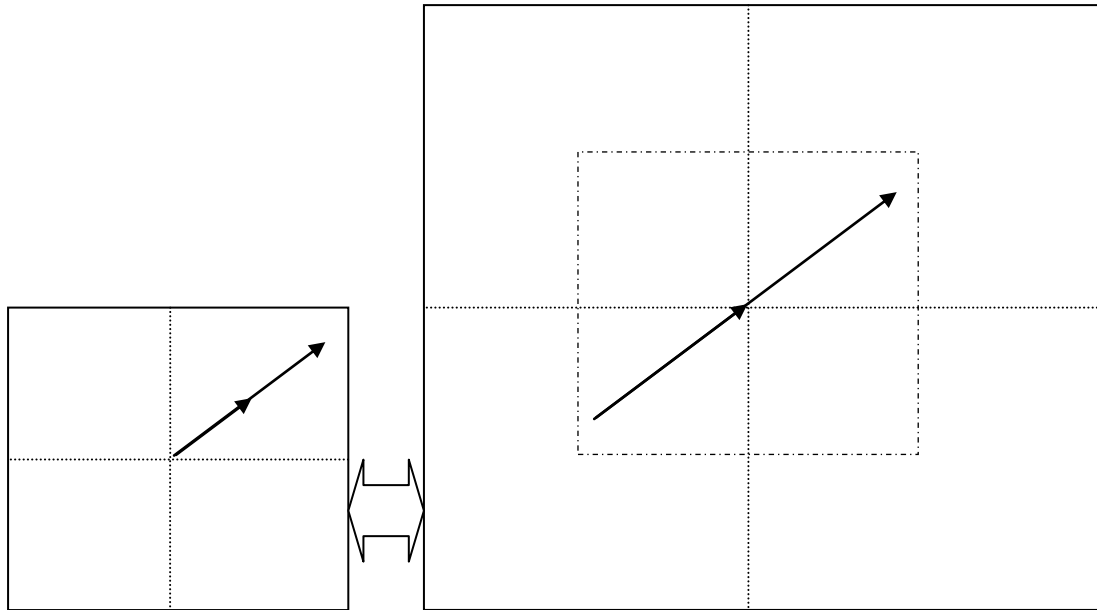
### 5.1.2.3 Συμβάντα του ποντικιού

Έχουν ήδη αναφερθεί κάποια από τα συμβάντα του ποντικιού. Εδώ τα αναφέρουμε όλα:

- Μετακίνηση του ποντικιού - Αλλαγή του δείκτη πάνω από διευθύνσεις. Ο δείκτης του ποντικιού γίνεται «χεράκι» πάνω από τις διευθύνσεις για να δείξει ότι ο χρήστης μπορεί να τις επιλέξει.
- Αριστερό κλικ - Επιλογή διεύθυνσης. Στον χάρτη 1 επιλογή της διεύθυνσης και σήμανσή της με κόκκινο. Αντίστοιχη αυτόματη επιλογή στην λίστα των διευθύνσεων. Στον χάρτη 2 αν είναι επιλεγμένος κάποιος ερευνητής ανάθεση της διεύθυνσης στον ερευνητή (αν δεν είναι ήδη ανατεθειμένη).
- Δεξί κλικ - μόνο χάρτης 1. Αποστολή αιτήματος για εύρεση της διεύθυνσης που έγινε το κλικ.
- Δεξί κλικ και σύρσιμο- μόνο χάρτης 2. Ανάθεση όλων των διευθύνσεων που περιλαμβάνονται στο νοητό τετράγωνο (το οποίο φαίνεται με ελαφρύ κίτρινο) από την αρχή του συρσίματος μέχρι το τέλος στον επιλεγμένο ερευνητή.
- Αριστερό κλικ και σύρσιμο. Κύλιση του χάρτη.
- Κύλιση του τροχού του ποντικιού. Αλλαγή εστίασης με το ποντίκι να δείχνει στο ίδιο σημείο. Το πώς επιτυγχάνεται αυτό αναλύεται αμέσως παρακάτω.

### 5.1.2.4 Εστίαση ποντικιού

Κατά την εστίαση σε υψηλότερο επίπεδο το σημείο που δείχνει το ποντίκι λόγω του διπλασιασμού του χάρτη θα απέχει την διπλάσια απόσταση από το τωρινό κέντρο του χάρτη. Για αυτό βρίσκουμε το σημείο (για την ακρίβεια τις συντεταγμένες του) που βρίσκεται στην μέση της απόστασης του τωρινού κέντρου από την θέση του ποντικιού και το θέτουμε ως νέο κέντρο. Η απόσταση του σημείου αυτού από την θέση του ποντικιού στον νέο επίπεδο εστίασης θα είναι όση ήταν και πριν από το παλιό κέντρο. Η διαδικασία αυτή φαίνεται σχηματικά στο παρακάτω σχήμα.



### 5.1 Διαδικασία εστίασης.

Η αντίστροφη διαδικασία πρέπει να ακολουθηθεί κατά την εστίαση σε μικρότερο επίπεδο. Τότε το νέο κέντρο θα είναι το σημείο που είναι στο αντίθετο σημείο σε σχέση με την θέση του ποντικιού αν θεωρηθούν σαν άξονες οι κάθετες που διέρχονται από το κέντρο της εικόνας, όπως φαίνεται στο σχήμα.

#### 5.1.3 Ταξινόμηση ερευνώμενων

Σε αυτό το σημείο θα αναλύσουμε τον αλγόριθμο που χρησιμοποιήθηκε για την ταξινόμηση των ερευνώμενων για κάθε ερευνητή.

Το πρόβλημα είναι παρόμοιο με το tsp που περιγράφηκε στο κεφάλαιο 2. Η μόνη διαφορά είναι ότι δεν είναι γνωστό αν ο ερευνητής θέλει να γυρίσει στο σημείο από όπου ξεκίνησε. Γενικά δεν ξέρουμε από πού θέλει να ξεκινήσει και που θέλει να τελειώσει, οπότε θα αγνοηθεί εντελώς αυτή η παράμετρος. Στον ερευνητή θα παρουσιαστεί μια προτεινόμενη σειρά την οποία ο ίδιος μπορεί να αλλάξει κυκλικά κατά τρόπο που να τον βολεύει. Επίσης δεν είναι σίγουρο ότι την διαδρομή θα την κάνει σε μια μέρα και όχι σε περισσότερες. Αυτό εξαρτάται από τον αριθμό των ερευνώμενων που έχει αναλάβει. Όμως αν το άθροισμα των διαδρομών που θα κάνει όλες τις μέρες είναι το ελάχιστο (αγνοώντας πάντα την διαδρομή προς την πρώτη διεύθυνση που επισκέπτεται και από την τελευταία) δηλαδή η διαδρομή του tsp, τότε και κάθε μέρα θα κάνει κατά μέσο όρο τη μικρότερη διαδρομή.

Ο αλγόριθμος που χρησιμοποιήθηκε είναι ο αλγόριθμος του Χριστοφίδη που περιγράφηκε στο κεφάλαιο 2. Θα τον αναλύσουμε εδώ σε περισσότερο βάθος. Θυμίζουμε ότι ο αλγόριθμος είναι πολυπλοκότητας  $O(n^3)$  και με σταθερά προσέγγισης  $3/2$ .

Αναλυτικά τα βήματα του αλγορίθμου είναι [Min08]:

1. Βρίσκουμε το ελαχίστου κόστους συνδετικό δέντρο  $T$  του γράφου  $G$

Δηλαδή ζητάμε ένα δέντρο που να περιλαμβάνει όλες τις κορυφές του  $G$  και να έχει το ελάχιστο συνολικό κόστος. Θα χρησιμοποιήσουμε τον αλγόριθμο του Prim για να το κάνουμε αυτό γιατί η πολυπλοκότητα ( $O(n^2)$ ) του εξαρτάται από τον αριθμό των κορυφών και είναι πιο αποδοτικός (σε σχέση με τον αλγόριθμο του Kruskal που εξαρτάται και από τον αριθμό των πλευρών) για πλήρεις γράφους όπως ο  $G$ . Ο αλγόριθμος του Prim: κάθε φορά διαλέγουμε την πλευρά που έχει το ελάχιστο κόστος έτσι ώστε ο νέος υπογράφος να παραμένει δέντρο [Zach07].

2.  $V'$  είναι οι κορυφές περιττού βαθμού στο δέντρο  $T$

Ο αριθμός των κορυφών περιττού βαθμού (έχει περιττό αριθμό πλευρών) σε κάθε γράφο είναι άρτιος αριθμός.

Απόδειξη:

Έστω ότι ισχύει για γράφο  $G$  και έστω ότι προσθέτω(ή αφαιρώ) μια πλευρά που ενώνει δύο κορυφές του. Τότε είτε μειώνω κατά δύο τις περιττές κορυφές, είτε τις αυξάνω κατά δύο, είτε μένουν αμετάβλητες. Άρα πάλι περιττός αριθμός. (παρόμοια αν προσθέσω κάποια κορυφή δεν αλλάζει ο αριθμός περιττών κορυφών γιατί η καινούρια κορυφή έχει βαθμό 0, άρα άρτιο).

3. Βρίσκουμε ένα minimum cost perfect matching  $M$  του  $V'$ .

Ταίριασμα (matching) είναι ένα υποσύνολο των ακμών ενός γράφου τέτοιο ώστε ποτέ δυο ακμές να μην έχουν κοινό άκρο. Perfect matching είναι το ταίριασμα όταν όλες οι κορυφές του γράφου ανήκουν στο ταίριασμα. Προφανώς ένας πλήρης γράφος με άρτιο βαθμό κορυφών (όπως αποδείξαμε πιο πριν οι κορυφές στο  $V'$  είναι άρτιες στον αριθμό) έχει πολλά τέτοια ταίριασματα. Για τον αλγόριθμό μας ζητάμε εκείνο που έχει το συνολικό ελάχιστο κόστος.

Το πρόβλημα παρότι είναι πολυπλοκότητας  $O(n^3)$  δηλαδή πολυωνυμικό, και είναι αυτό που δίνει και την πολυπλοκότητα σε όλον τον αλγόριθμο του Χριστοφίδη, έχει πολύ δύσκολη υλοποίηση.

Η υλοποίηση μας βασίστηκε σε έτοιμο κώδικα για Java από βιβλιογραφία<sup>8</sup> [Zach07], [WP2].

4. Έστω ο γράφος  $G' = T \cup M$ . Ο  $G'$  έχει κύκλο Euler (γιατί όλες του οι κορυφές έχουν άρτιο βαθμό)

Κύκλος Euler λέγεται ένας κύκλος που περνά ακριβώς μια φορά από κάθε πλευρά ενός γράφου [Zach07].

5. Βρίσκουμε έναν τέτοιο κύκλο Euler  $\pi$  στον  $G'$ .

---

<sup>8</sup> A Java Library of Graph Algorithms and Optimization Hang T. Lau

Για να βρούμε κύκλο Euler στον γράφο  $G'$  χρησιμοποιούμε πάλι έτοιμο κώδικα από βιβλιογραφία <sup>8</sup>.

6. Βρίσκουμε την διαδρομή  $C$  που επισκέπτεται τις κορυφές με την σειρά που πρωτοεμφανίζονται στο  $\pi$ .

Αυτό είναι εύκολο. Δημιουργούμε έναν πίνακα μεγέθους του αριθμού των διευθύνσεων. Εισάγουμε στον πίνακα αυτό τα στοιχεία του  $\pi$  που πρωτοεμφανίζονται σ' αυτό.

#### 5.1.4 Διαχείριση Βάσης Δεδομένων

Η διαχείριση της βάσης δεδομένων γίνεται από το πακέτο `sql`, που περιέχει την κλάση `MysqlConnection`. Η κλάση περιέχει τις παρακάτω μεθόδους:

- `MysqlConnection(String)`. Constructor μέθοδος της κλάσης. Ανοίγει την σύνδεση με την Βάση. Έχει το όνομα χρήστη και τον κωδικό της Βάσης Δεδομένων. Είναι το μόνο σημείο του κώδικα που θα αλλάξει αν χρησιμοποιηθεί διαφορετική βάση δεδομένων (όχι η MySQL) ή διαφορετικός χρήστης.
- `mysqlCreate(String)`. Δημιουργεί την Βάση δεδομένων την πρώτη φορά δημιουργίας της κλάσης.
- `mysqlTableCreate(String)`. Δημιουργεί τους πίνακες της Βάσης Δεδομένων την πρώτη φορά που εκτελείται μια ενέργεια σε κάθε πίνακα. Η κλάση έχει αποθηκευμένες τις εντολές για δημιουργία κάθε πίνακα και τις εξαρτήσεις μεταξύ των πινάκων, ώστε κάθε φορά που πάει να εισάγει έναν πίνακα πρώτα να εισάγει τους πίνακες από τους οποίους εξαρτάται.
- `mysqlClose()`. Κλείνει την σύνδεση με την Βάση Δεδομένων.
- `mysqlSelect(String, String, String)`. Ζητάει δεδομένα από την βάση από έναν πίνακα με βάση μια συνθήκη. Τα αποτελέσματα μπορεί να παρθούν με μια από τις δύο επόμενες μεθόδους.
- `mysqlSelectInt(String)`. Παίρνει τα αποτελέσματα ενός πεδίου που είναι ακέραιος. Δίνεται ως όρισμα το όνομα του πεδίου.
- `mysqlSelectString(String)`. Παίρνει τα αποτελέσματα ενός πεδίου που είναι συμβολοσειρά. Δίνεται ως όρισμα το όνομα του πεδίου.
- `mysqlSelect(String, String, String, String)`. Ζητάει και γυρνάει δεδομένα από έναν πίνακα.
- `mysqlDelete(String, String)`. Διαγράφει δεδομένα από πίνακα.
- `mysqlInsert(String, String, String)`. Εισάγει δεδομένα σε πίνακα.
- `mysqlCommand(String)`. Δίνει μια εντολή προς εκτέλεση στην βάση δεδομένων. Μπορεί να είναι οτιδήποτε.

Η χρήση της παραπάνω κλάσης για την διαχείριση της Βάσης Δεδομένων απλοποιεί πολύ τον κώδικα. Κάθε ενέργεια πάνω στην βάση γίνεται με 3-4 εντολές: δημιουργία ενός αντικειμένου της κλάσης, κλήση μιας μεθόδου, πιθανή κλήση μεθόδου που γυρνάει αποτελέσματα και κλείσιμο της σύνδεσης. Σε κάθε τέτοια ενέργεια γίνεται αντιμετώπιση λαθών και τυπώνεται η ενέργεια που γίνεται στην Βάση. Δημιουργούνται διαφανώς άμα χρειάζεται πίνακες.

## 5.2 Πλατφόρμες και προγραμματιστικά εργαλεία

Εδώ περιγράφονται τα χαρακτηριστικά της συγκεκριμένης υλοποίησης, όπως η πλατφόρμα ανάπτυξης και εκτέλεσης και τα προγραμματιστικά εργαλεία.

### 5.2.1 Πλατφόρμες ανάπτυξης

Στον πίνακα 5.1 καταγράφονται οι τεχνολογίες και οι πλατφόρμες που αναπτύχθηκε το σύστημά μας.

Εφαρμογή	Java SE
Βάση Δεδομένων	MySQL

Πίνακας 5.1 Τεχνολογίες ανάπτυξης

#### 5.2.1.1 Java Standard Edition

Η εφαρμογή του συστήματος έχει υλοποιηθεί στην γλώσσα προγραμματισμού Java. Η Java είναι μια γλώσσα προγραμματισμού η οποία αναπτύχθηκε αρχικά από τον James Gosling στην Sun Microsystems (η οποία πλέον είναι θυγατρική της Oracle Corporation) και εκδόθηκε το 1995 σαν ο πυρήνας της Java πλατφόρμας. Η γλώσσα έχει πάρει πολλά από την σύνταξη της C και της C++ αλλά έχει πιο απλό μοντέλο αντικειμένων και λιγότερες κατώτερου επιπέδου λειτουργίες. Οι Java εφαρμογές τυπικά μεταγλωττίζονται σε bytecode, το οποίο μπορεί να εκτελεστεί σε κάθε Java Virtual Machine ανεξάρτητα από την αρχιτεκτονική του υπολογιστή. Η Java είναι γενικού σκοπού, βασισμένη σε κλάσεις, αντικειμενοστραφής γλώσσα, η οποία είναι σχεδιασμένη ώστε να έχει όσο λιγότερες εξαρτήσεις υλοποίησης είναι δυνατόν. Εξυπηρετεί τον σκοπό να δίνει την δυνατότητα στους προγραμματιστές εφαρμογών «να γράφουν μία φορά για παντού». Η Java είναι μια από τις πιο δημοφιλείς γλώσσες προγραμματισμού σε χρήση και χρησιμοποιείται ευρέως για λογισμικό εφαρμογών και εφαρμογές διαδικτύου [WP3].

Η πλατφόρμα Java SE στην ουσία είναι η Java Virtual Machine που πρέπει να χρησιμοποιηθεί για να τρέχει ένα πρόγραμμα σε Java, μαζί με τις βιβλιοθήκες (πακέτα) που χρειάζονται για να επιτρέπουν την χρήση συστημάτων αρχείων, δικτύων γραφικής διασύνδεσης και από αυτά τα προγράμματα [WP4].

Η πλατφόρμα Java SE δίνει την δυνατότητα να αναπτύσσει κανείς και να υλοποιήσει εφαρμογές Java σε επιτραπέζιους υπολογιστές και σε Servers καθώς και σε απαιτητικά Ενσωματωμένα συστήματα πραγματικού χρόνου.

Οι ακόλουθες τεχνολογίες Java και οι διασυνδέσεις προγραμματισμού εφαρμογών (API) είναι το θεμέλιο της πλατφόρμας Java, Standard Edition (Java SE). Παρέχουν όλα όσα χρειάζεται κανείς για τη δημιουργία εφαρμογών server, εφαρμογές γραφείου, και βοηθητικές εφαρμογές που τρέχουν σε όλα σχεδόν τα δημοφιλή λειτουργικά συστήματα, συμπεριλαμβανομένου των Linux, Macintosh, Solaris και Windows [ORA1].

Java™ SE Platform at a Glance	
<b>Java Language</b>	Java Language
<b>Tools &amp; Tool APIs</b>	java javac javadoc apt jar javap JPDA JConsole Java VisualVM Security Int'l RMI IDL Deploy Monitoring Troubleshoot Scripting JVM TI
<b>Deployment Technologies</b>	Deployment Java Web Start Java Plug-in
<b>User Interface Toolkits</b>	AWT Swing Java 2D Accessibility Drag n Drop Input Methods Image I/O Print Service Sound
<b>Integration Libraries</b>	IDL JDBC JNDI RMI RMI-IIOP
<b>Other Base Libraries</b>	Beans Intl Support Input/Output JMX JNI Math Networking Override Mechanism Security Serialization Extension Mechanism XML JAXP
<b>lang and util Base Libraries</b>	lang and util Collections Concurrency Utilities JAR Logging Management Preferences Ref Reflection Regular Expressions Versioning Zip Instrumentation API Objects
<b>Java Virtual Machine</b>	Java Hotspot Client VM Java Hotspot Server VM
<b>Platforms</b>	Solaris Linux Windows Other

Εικόνα 5.2 Τεχνολογίες Java και APIs [ORA1].

Στην εφαρμογή μας χρησιμοποιήθηκε η διασύνδεση AWT (βλέπε [ORA2]) και Swing (βλέπε [ORA3]) για την δημιουργία και διαχείριση του γραφικού της περιβάλλοντος.

Η διασύνδεση Abstract Window Toolkit (AWT) υποστηρίζει προγραμματισμό γραφικού περιβάλλοντος χρήστη (GUI). Χαρακτηριστικά του AWT περιλαμβάνουν:

- Μια σειρά από συστατικά, διεπαφής χρήστη
- Ένα ισχυρό μοντέλο εκδήλωση-χειρισμού
- Εργαλεία γραφικών και εικόνων, συμπεριλαμβανομένων του σχήματος, χρώματος, και κλάσεις γραμματοσειράς
- Διαχειριστές διάταξης, για ευέλικτες διατάξεις παράθυρου που δεν εξαρτώνται από ένα συγκεκριμένο μέγεθος παραθύρου ή ανάλυση οθόνης
- Κλάσεις μεταφοράς δεδομένων, για cut-and-paste μέσω των μεθόδων της παρούσας αρχιτεκτονικής



Οι κλάσεις Swing υλοποιούν ένα σύνολο στοιχείων για ανάπτυξη γραφικών διεπαφών χρήστη και προσθήκη πλούσιας λειτουργικότητας στα γραφικά και διαδραστικότητα στις εφαρμογές Java. Τα συστατικά του Swing υλοποιούνται εξ ολοκλήρου στη γλώσσα προγραμματισμού Java. Δίνει την δυνατότητα δημιουργίας γραφικών διεπαφών που φαίνονται ακριβώς το ίδιο σε διαφορετικές πλατφόρμες ή κληρονομούν την εμφάνιση του λειτουργικού συστήματος.

Επίσης πέραν των βιβλιοθηκών της Java χρησιμοποιήσαμε

- την βιβλιοθήκη XOM (XML Object Model) μια βιβλιοθήκη για την επεξεργασία xml αρχείων με Java.
- και το πρόγραμμα οδήγησης JDBC-MySQL για την σύνδεση της Java με την MySQL βάση δεδομένων που χρησιμοποιήσαμε.

### 5.2.1.2 MySQL



Η Βάση Δεδομένων που χρησιμοποιήθηκε στο σύστημα είναι η MySQL. Η βάση δεδομένων MySQL® έχει γίνει η πιο δημοφιλής βάση δεδομένων ανοιχτού κώδικα στον κόσμο, λόγω της συνεπούς γρήγορης απόδοσης της, της υψηλής αξιοπιστίας και της ευκολίας στη χρήση. Χρησιμοποιείται από μεμονωμένους προγραμματιστές Web, καθώς και από πολλούς από τους μεγαλύτερους και ταχύτερα αναπτυσσόμενους οργανισμούς στον κόσμο, για να εξοικονομήσουν χρόνο και χρήμα, τροφοδοτώντας τους μεγάλου όγκου δικτυακούς τόπους τους και σημαντικά συστήματα και πακέτα λογισμικού -- συμπεριλαμβανομένων των ηγετών της βιομηχανίας, όπως η Yahoo, η Alcatel-Lucent, η Google, η Nokia, το YouTube, καθώς και η Zappos.com.

Δεν είναι μόνο η MySQL η δημοφιλέστερη βάση δεδομένων ανοιχτού κώδικα στον κόσμο, έχει επίσης γίνει η βάση δεδομένων που χρησιμοποιείται για μια νέα γενιά εφαρμογών που βασίζεται στην LAMP stack (Linux, Apache, MySQL, PHP / Perl / Python.) Η MySQL τρέχει σε περισσότερες από 20 πλατφόρμες όπως Linux, Windows, Mac OS, Solaris, HP-UX, IBM AIX [MySQL].

### 5.2.2 Εργαλεία ανάπτυξης συστήματος

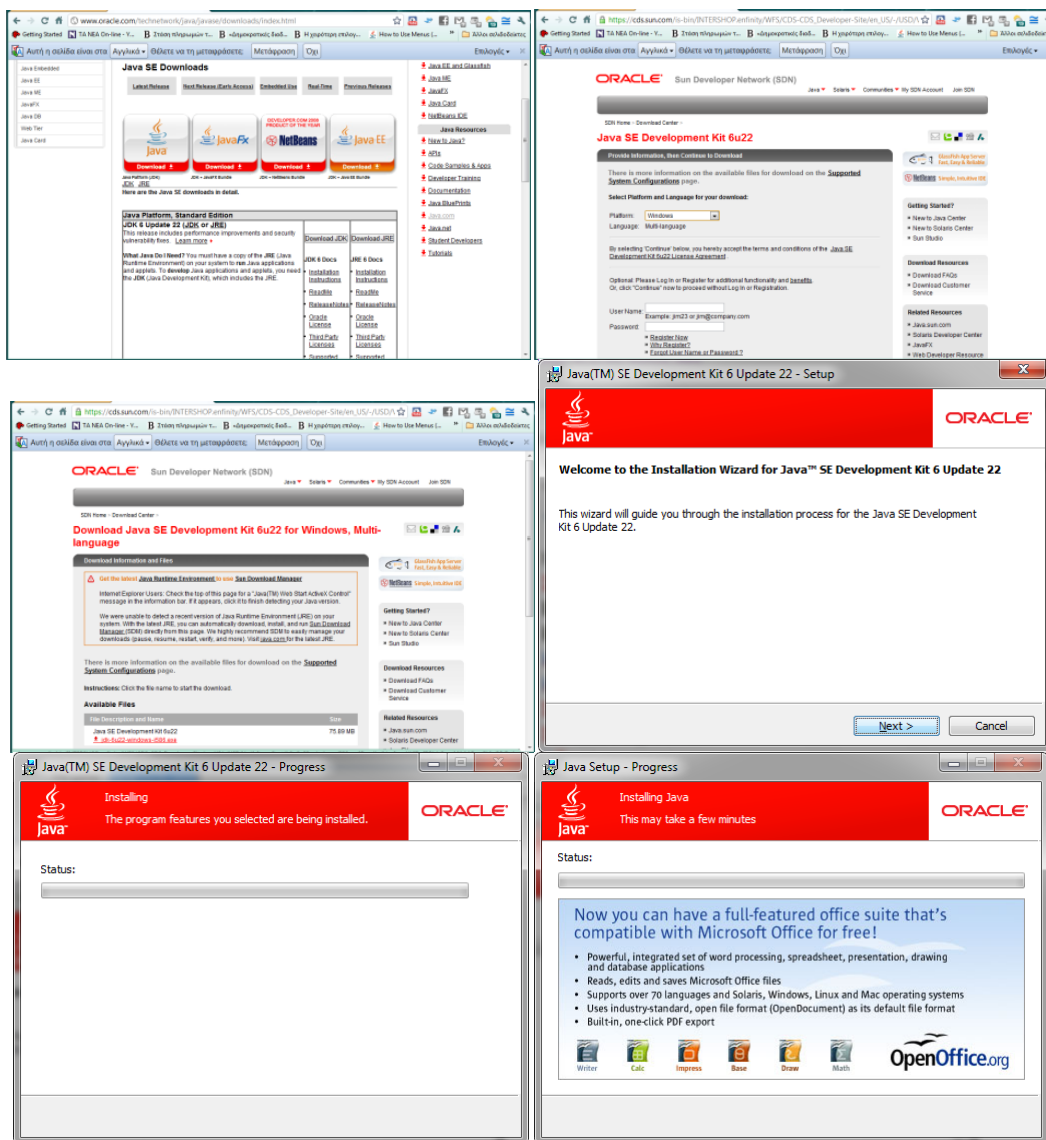
Για την ανάπτυξη του όλου συστήματος χρησιμοποιήθηκε το προγραμματιστικό εργαλείο Eclipse IDE for Java Developers. Το εργαλείο Eclipse είναι ένα πολύ-γλωσσικό περιβάλλον ανάπτυξης λογισμικού που περιλαμβάνει ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE), και είναι επεκτάσιμο με πρόσθετα σύστημα. Είναι γραμμένο κυρίως σε Java και μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών σε Java (και σε άλλες γλώσσες προγραμματισμού) [ECL], [WP5].

## 5.3 Εγκατάσταση συστήματος

Η εγκατάσταση του συστήματος περιλαμβάνει την εγκατάσταση της Java SE και την εγκατάσταση της MySQL. Παρακάτω περιγράφονται αναλυτικά τα βήματα για εγκατάσταση της εφαρμογής σε λειτουργικό σύστημα Windows. Παρόμοια διαδικασία ακολουθείται και σε άλλα λειτουργικά συστήματα.

### 5.3.1 Εγκατάσταση Java SE

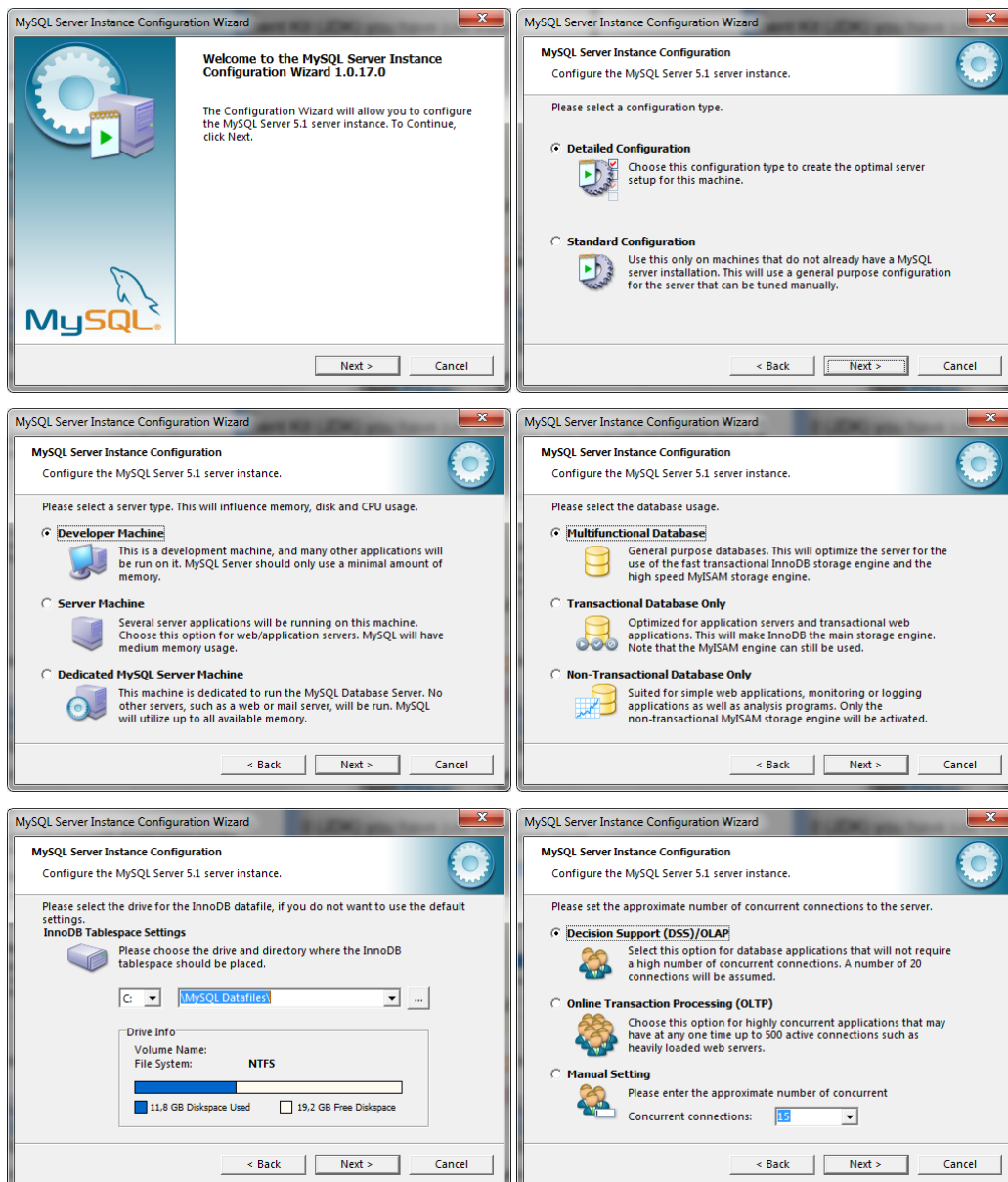
Μεταβαίνουμε στην τοποθεσία [www.oracle.com](http://www.oracle.com) και από εκεί μεταβαίνουμε στην σελίδα λήψης της τελευταίας έκδοσης Java SE. Στην συνέχεια επιλέγουμε την κατάλληλη έκδοση για το σύστημα προχωρούμε στην λήψη της και στην εγκατάστασή της. Η εγκατάσταση είναι απλή και δεν χρειάζεται να αλλάξουμε καμία επιλογή. Η διαδικασία φαίνεται συνοπτικά στις παρακάτω εικόνες.

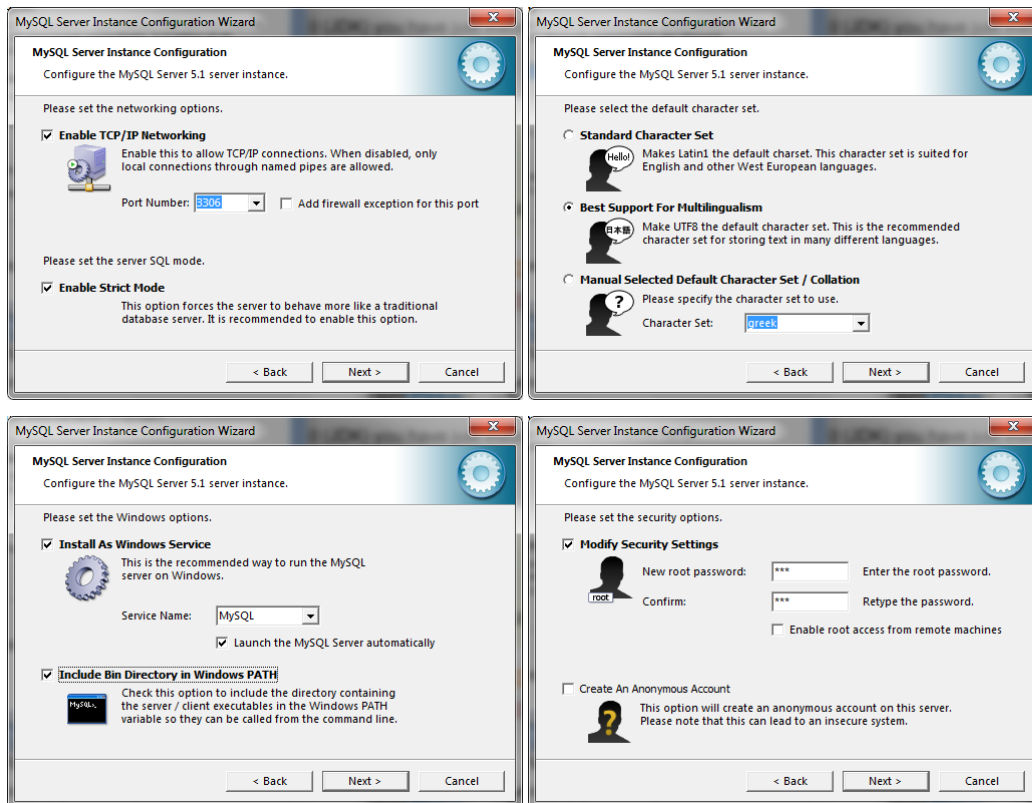


### 5.3 Διαδικασία εγκατάστασης Java SE

### 5.3.2 Εγκατάσταση MySQL

Μεταβαίνουμε στην τοποθεσία <http://mysql.com/> και κάνουμε λήψη της τελευταίας έκδοσης του MySQL Community Server. Ολοκληρώνουμε την εγκατάσταση επιλέγοντας τυπική εγκατάσταση. Στο τέλος της εγκατάστασης ρυθμίζουμε της παραμέτρους λειτουργίας του MySQL Server. Επιλέγουμε Detailed Configuration, Developer Machine, Multifunctional Database, Decision Support (DSS) /OLAP, Best Support For Multilingualism και τέλος δίνουμε ως κωδικό για τον χρήστη “root” το “123”. Η διαδικασία φαίνεται στις παρακάτω εικόνες.





## 5.4 Διαδικασία εγκατάστασης MySQL

Όπως έχει ήδη αναφερθεί η βάση δεδομένων του συστήματος **δημιουργείται αυτόματα** την πρώτη φορά που ανοίγει η εφαρμογή. Παρ' όλα αυτά εδώ θα δώσουμε την διαδικασία και τον κώδικα που ακολουθείται για την χειροκίνητη δημιουργία της.

```

DROP DATABASE IF EXISTS `ereuna`;
CREATE DATABASE `ereuna`;
CREATE TABLE `ereuna`.`ereuna` (
  `i` int(11) NOT NULL AUTO_INCREMENT,
  `Onoma` varchar(20) DEFAULT NULL,
  `Header` tinyint(1) DEFAULT NULL,
  `Footer` tinyint(1) DEFAULT NULL,
  `Geocode_Delay` tinyint(1) DEFAULT NULL,
  `Store_Map` tinyint(1) DEFAULT NULL,
  `Load_Map_On_Request` tinyint(1) DEFAULT NULL,
  PRIMARY KEY (`i`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;
CREATE TABLE `ereuna`.`dhmoi` (
  `i` int(11) NOT NULL AUTO_INCREMENT,
  `Onoma` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`i`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `ereuna`.`addresses` (
  `i` int(11) NOT NULL AUTO_INCREMENT,
  `Address` varchar(256) DEFAULT NULL,
  `Number` varchar(20) DEFAULT NULL,
  `Dhmos` int(11) DEFAULT NULL,
  `Onoma` varchar(20) DEFAULT NULL,
  `Epwnumo` varchar(20) DEFAULT NULL,

```

```

`Thlefwno` varchar(20) DEFAULT NULL,
`Shmeiwseis` varchar(256) DEFAULT NULL,
PRIMARY KEY (`i`),
KEY `Dhmos` (`Dhmos`),
CONSTRAINT `addresses_ibfk_1` FOREIGN KEY (`Dhmos`) REFERENCES
`dhmoi` (`i`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
CREATE TABLE `ereuna`.`coordinates` (
`i` int(11) NOT NULL,
`lng` double DEFAULT NULL,
`lat` double DEFAULT NULL,
PRIMARY KEY (`i`),
KEY `i` (`i`),
CONSTRAINT `coordinates_ibfk_1` FOREIGN KEY (`i`) REFERENCES
`addresses` (`i`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
CREATE TABLE `ereuna`.`ereunhtes` (
`i` int(11) NOT NULL AUTO_INCREMENT,
`Onoma` varchar(20) DEFAULT NULL,
`Epwnumo` varchar(20) DEFAULT NULL,
`Thlefwno` varchar(20) DEFAULT NULL,
`Shmeiwseis` varchar(256) DEFAULT NULL,
PRIMARY KEY (`i`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
CREATE TABLE `ereuna`.`selected` (
`i` int(11) NOT NULL,
`number` int(11) DEFAULT NULL,
PRIMARY KEY (`i`),
CONSTRAINT `selected_ibfk_1` FOREIGN KEY (`i`) REFERENCES
`ereunhtes` (`i`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
CREATE TABLE `ereuna`.`ereunhtes_addresses` (
`i` int(11) NOT NULL DEFAULT '0',
`j` int(11) NOT NULL DEFAULT '0',
`visit` int(11) DEFAULT NULL,
PRIMARY KEY (`i`,`j`),
KEY `i` (`i`),
KEY `j` (`j`),
CONSTRAINT `ereunhtes_addresses_ibfk_1` FOREIGN KEY (`i`)
REFERENCES `selected` (`i`) ON DELETE CASCADE,
CONSTRAINT `ereunhtes_addresses_ibfk_2` FOREIGN KEY (`j`)
REFERENCES `coordinates` (`i`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
CREATE TABLE `ereuna`.`ereunhtes_dhmoi` (
`i` int(11) NOT NULL DEFAULT '0',
`j` int(11) NOT NULL DEFAULT '0',
PRIMARY KEY (`i`,`j`),
KEY `i` (`i`),
KEY `j` (`j`),
CONSTRAINT `ereunhtes_dhmoi_ibfk_1` FOREIGN KEY (`i`) REFERENCES
`ereunhtes` (`i`) ON DELETE CASCADE,
CONSTRAINT `ereunhtes_dhmoi_ibfk_2` FOREIGN KEY (`j`) REFERENCES
`dhmoi` (`i`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
CREATE TABLE `ereuna`.`problematic` (
`i` int(11) NOT NULL,
PRIMARY KEY (`i`),
KEY `i` (`i`),
CONSTRAINT `problematic_ibfk_1` FOREIGN KEY (`i`) REFERENCES
`addresses` (`i`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

## Πίνακας 5.2 Δημιουργία Βάσης Δεδομένων.

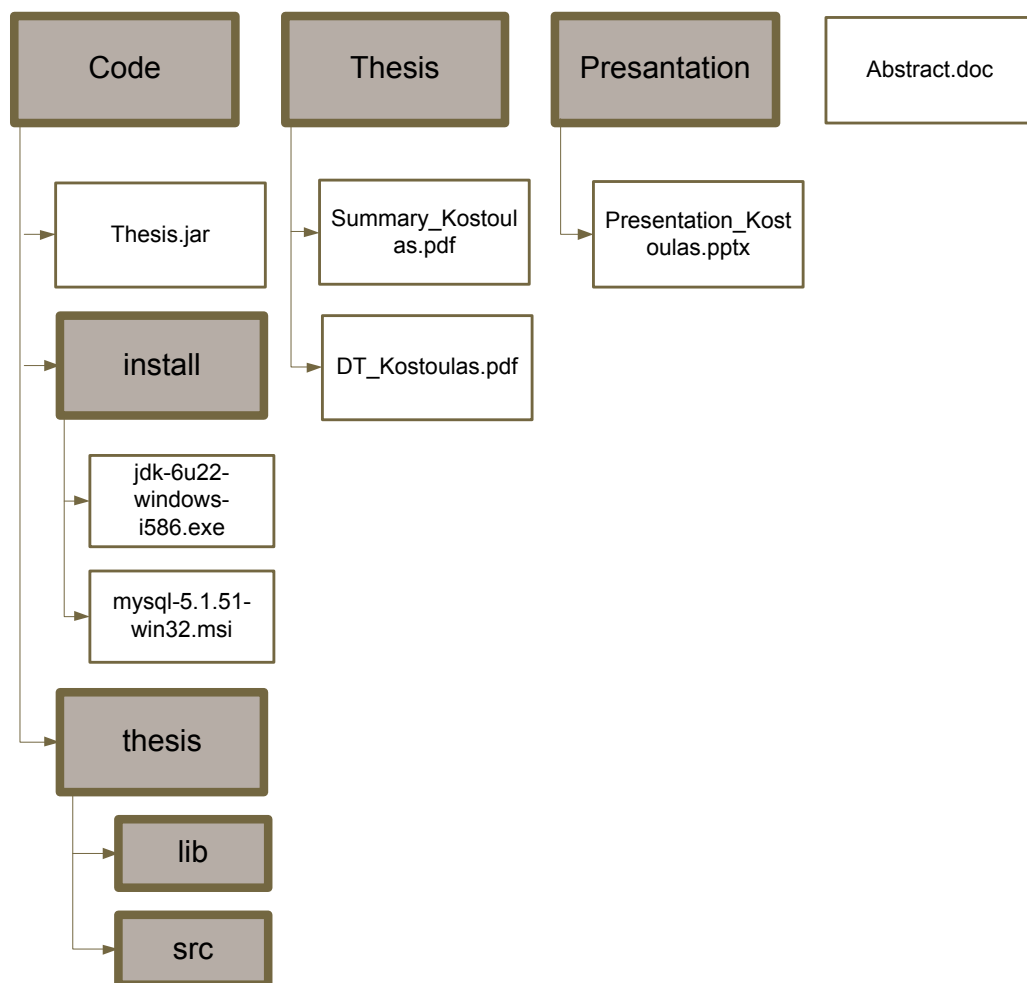
Από τον παραπάνω κώδικα είναι πολύ εύκολο να εισάγουμε την Βάση Δεδομένων. Αποθηκεύουμε τον παραπάνω κώδικα σε ένα αρχείο έστω στο ereuna.txt. Ανοίγουμε μία γραμμή εντολών και πληκτρολογούμε:

```
mysql -u root -p123 <ereuna.txt
```

Με αυτό τον τρόπο εισάγουμε την Βάση Δεδομένων ή την αρχικοποιούμε διαγράφοντάς την και ξαναδημιουργώντας την.

## 5.4 Περιεχόμενα CD

Ο τόμος συνοδεύεται από CD που περιέχει τα παρακάτω αρχεία:



Εικόνα 5.5 Περιεχόμενα CD.

- Φάκελος Code: Περιέχει το αρχείο Thesis.jar με το οποίο ανοίγει η εφαρμογή. Στην τοποθεσία που είναι το αρχείο δημιουργείται αυτόματα, κατά την πρώτη φορά που τρέχει η εφαρμογή, φάκελος lib στον οποίο αποθηκεύονται δεδομένα της εφαρμογής (εικόνες). Ο υπό-φάκελος install περιέχει τα δύο αρχεία που εγκαθιστούν την Java SE

και την MySQL. Ενώ τέλος ο υπό-φάκελος thesis περιέχει τον κώδικα της εφαρμογής μαζί με τις απαραίτητες βιβλιοθήκες.

- Φάκελος Thesis: Περιέχει το αρχείο DT\_Kostoulas.pdf, που είναι ο παρών τόμος, και το αρχείο Summary\_Kostoulas.pdf, που είναι μια περίληψη της διπλωματικής.
- Φάκελος Presentation: Περιέχει την παρουσίαση της διπλωματικής
- Abstract.doc: Αρχείο με το εξώφυλλο, την περίληψη και τις λέξεις κλειδιά της διπλωματικής.





# 6

## *Έλεγχος*

Ακολουθεί η αξιολόγηση και ο έλεγχος της σωστής λειτουργίας του συστήματος.

### *6.1 Μεθοδολογία ελέγχου*

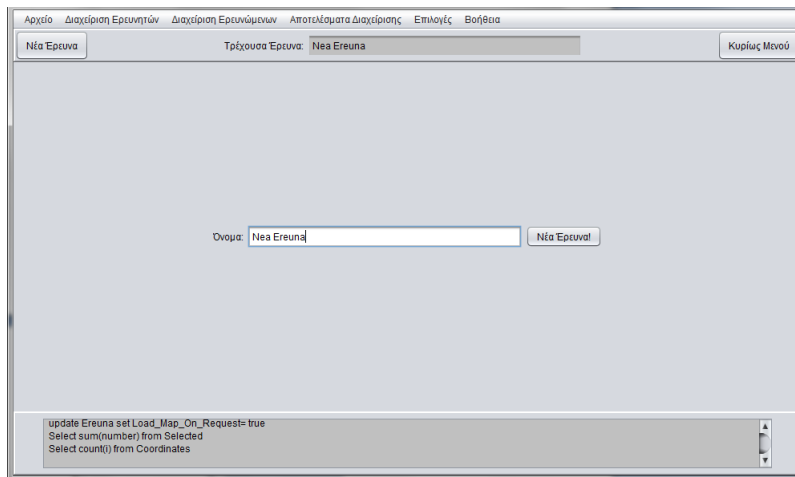
Ο έλεγχος πραγματοποιήθηκε με την χρήση του παρακάτω σεναρίου λειτουργίας:

Υπάρχουν 4 ερευνητές. Οι 3 έχουν προτίμηση τον δήμο Αθηναίων και ο 1 τον δήμο Ζωγράφου. Οι ερευνώμενοι είναι 131 από τους οποίους οι 36 στον δήμο Ζωγράφου και οι υπόλοιποι στον δήμο Αθηναίων. Οι δύο πρώτοι ερευνητές θέλουν να επισκεφτούν από 35 ερευνώμενους, ο τρίτος 25 και ο τελευταίος 36. Η εφαρμογή είναι αρχικοποιημένη, δεν έχει κανένα δεδομένο αποθηκευμένο. Ο χρήστης αρχικά προσθέτει του ερευνητές. Στην συνέχεια προσθέτει τους ερευνώμενους.

### *6.2 Αναλυτική παρουσίαση ελέγχου*

Στην ενότητα αυτή παρουσιάζουμε αναλυτικά τον έλεγχο του συστήματος σύμφωνα με το σενάριο που περιγράφηκε στην προηγούμενη ενότητα. Παρουσιάζουμε τον έλεγχο της εφαρμογής χρησιμοποιώντας εικόνες λειτουργίας της. Οι εικόνες που χρησιμοποιούμε για συντομία δείχνουν μόνο το κομμάτι της εφαρμογής που εξετάζουμε κάθε φορά. Πολλές από τις λειτουργίες που εξετάζουμε έχουν ήδη περιγραφεί στο κεφάλαιο 3.2.

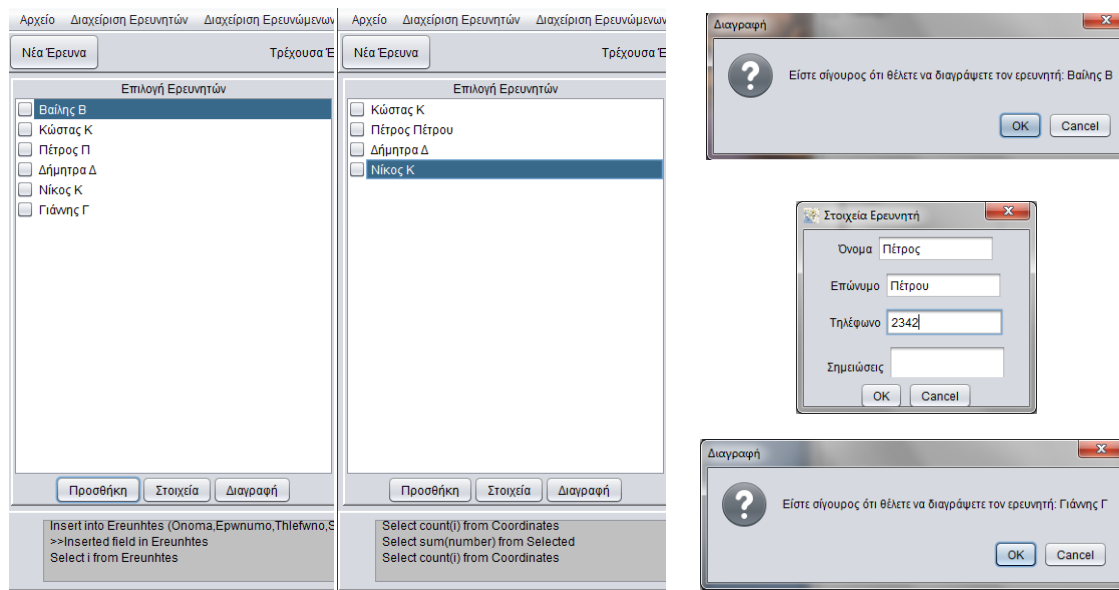
Αρχικά δημιουργούμε μια νέα έρευνα με όνομα Νέα Έρευνα.



Εικόνα 6.1 Νέα έρευνα.

### 6.2.1 Εισαγωγή, προβολή, αλλαγή και διαγραφή ερευνητών

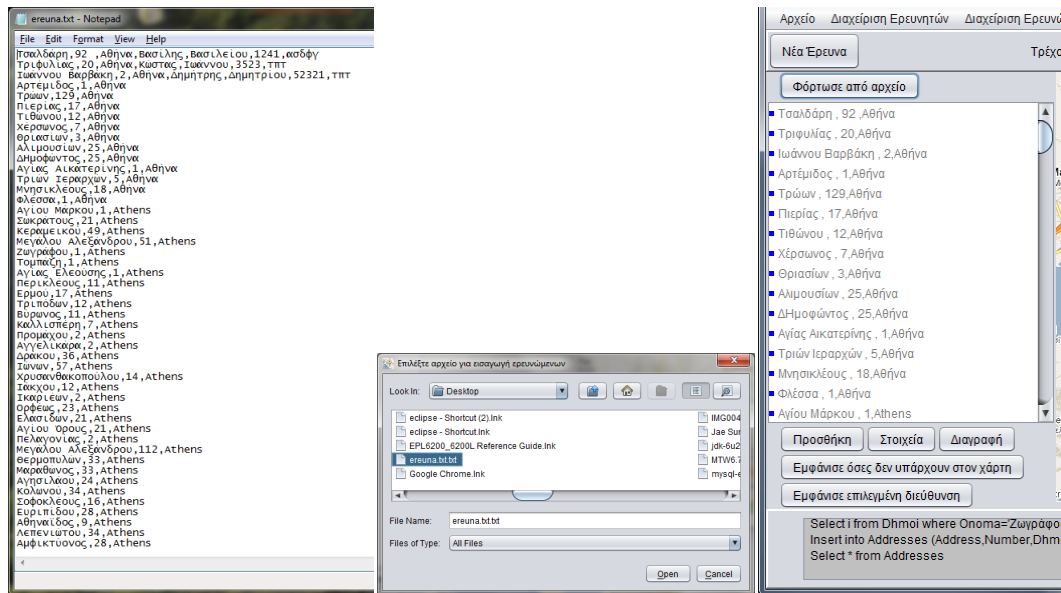
Προσθέτουμε 6 ερευνητές και στην συνέχεια θα διαγράψουμε τον πρώτο, θα προβάλλουμε τα στοιχεία του 2<sup>ου</sup> και θα αλλάξουμε το Όνομά του. Τέλος θα διαγράψουμε και τον τελευταίο ερευνητή.



Εικόνα 6.2 Εισαγωγή, προβολή και αλλαγή στοιχείων και διαγραφή ερευνητή.

### 6.2.2 Εισαγωγή ερευνώμενων από αρχείο

Στην συνέχεια εισάγουμε από αρχείο 131 ερευνώμενους.



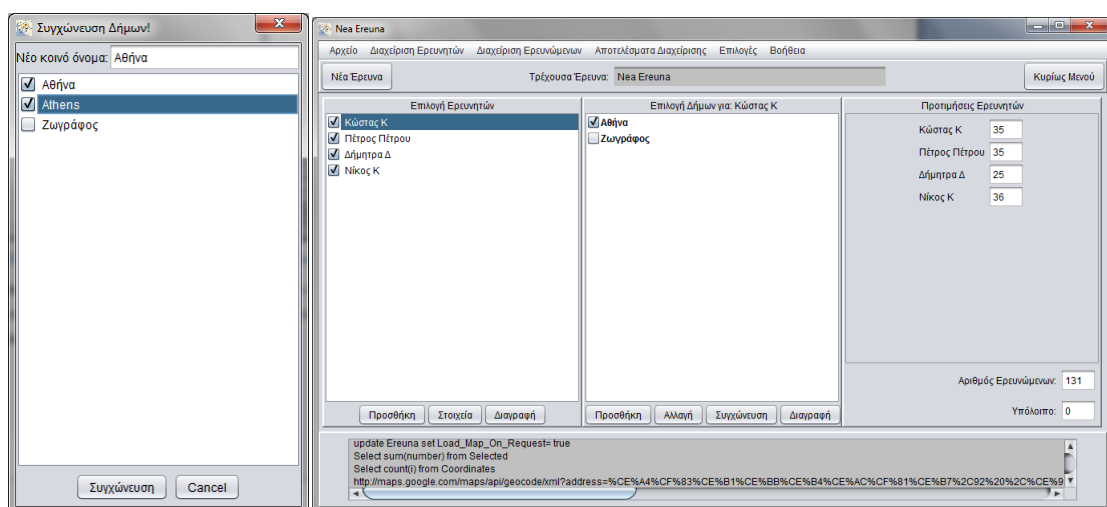
Εικόνα 6.3 Εισαγωγή ερευνώμενων από αρχείο.

### 6.2.3 Συγχώνευση δήμων, προτιμήσεις σε δήμους και ερευνώμενους

Για τον δήμο Αθηναίων, στο αρχείο, χρησιμοποιούνται οι ονομασίες Αθήνα και Athens. Γυρίζουμε στο παράθυρο των ερευνητών και βλέπουμε ότι στην λίστα με τους δήμους έχουν εισαχθεί 3 δήμοι. Αθήνα, Athens και Ζωγράφος.

Αρχικά θα συγχωνεύσουμε σε έναν δήμο τους Athens, Αθήνα (αυτή η ενέργεια θα αλλάξει και τις διευθύνσεις των ερευνώμενων).

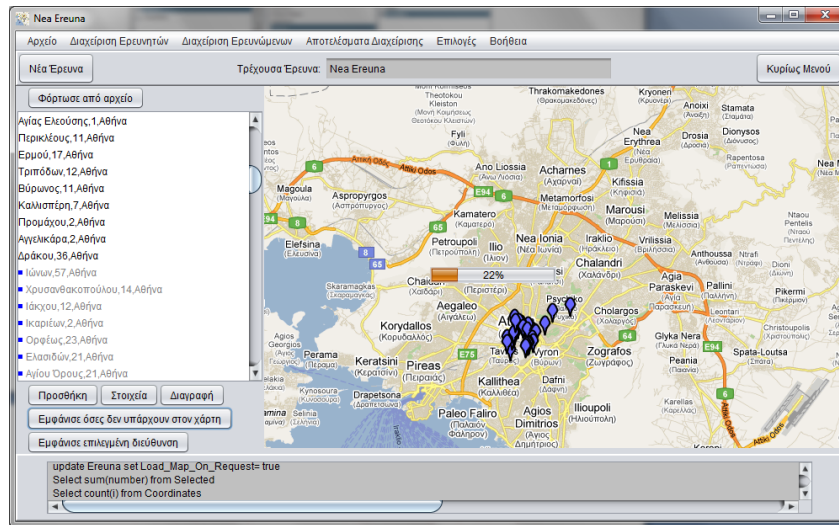
Στην συνέχεια δηλώνουμε τις προτιμήσεις των ερευνητών σύμφωνα με το σενάριο. Όλοι οι ερευνητές συμμετέχουν στην έρευνα (Την επιλογή του αριθμού των ερευνώμενων που θέλει ο κάθε ερευνητής την κάναμε μετά την εμφάνιση στον χάρτη των ερευνώμενων για να προκύψει το σύνολο των ερευνώμενων κάτω αριστερά).



Εικόνα 6.4 Αριστερά: Συγχώνευση δήμων, Δεξιά: Δήλωση προτιμήσεων.

## 6.2.4 Εμφάνιση στον χάρτη

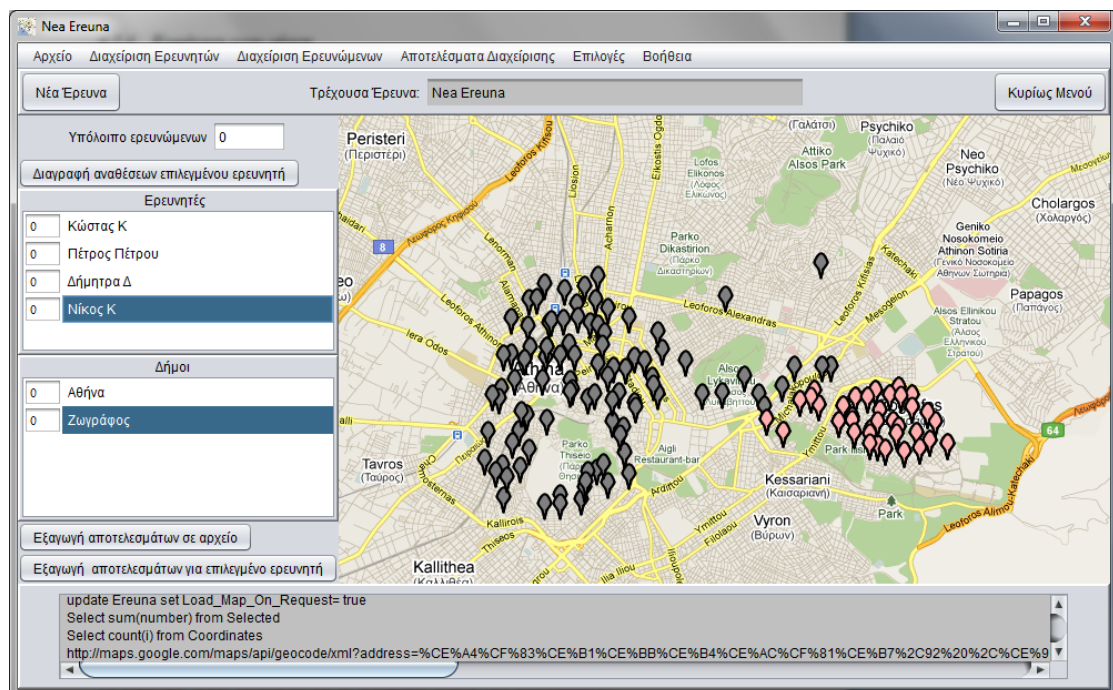
Στην συνέχεια εμφανίζουμε στον χάρτη τους ερευνώμενους.



Εικόνα 6.5 Εμφάνιση στον χάρτη.

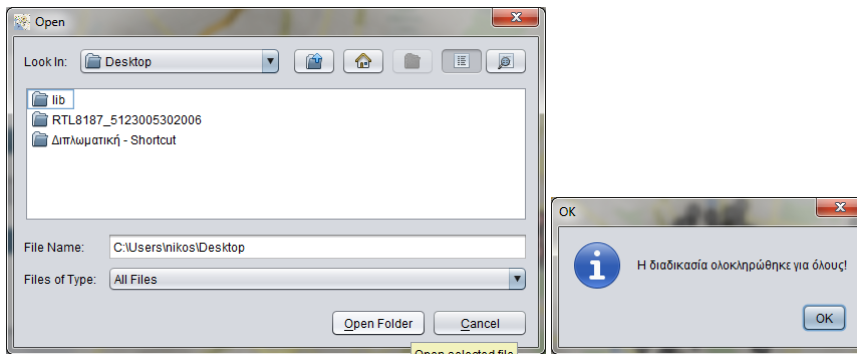
## 6.2.5 Εξαγωγή αποτελεσμάτων

Μετά πάμε στο παράθυρο επιλογής αποτελεσμάτων και κατανέμουμε τους ερευνητές σύμφωνα με τις προτιμήσεις τους.



Εικόνα 6.6 Κατανομή ερευνώμενων.

και αφού το κάνουμε αυτό εξαγάγουμε την σειρά προτίμησης τους σε αρχείο.



**Εικόνα 6.7 Τοποθεσία εξαγωγής αποτελεσμάτων**

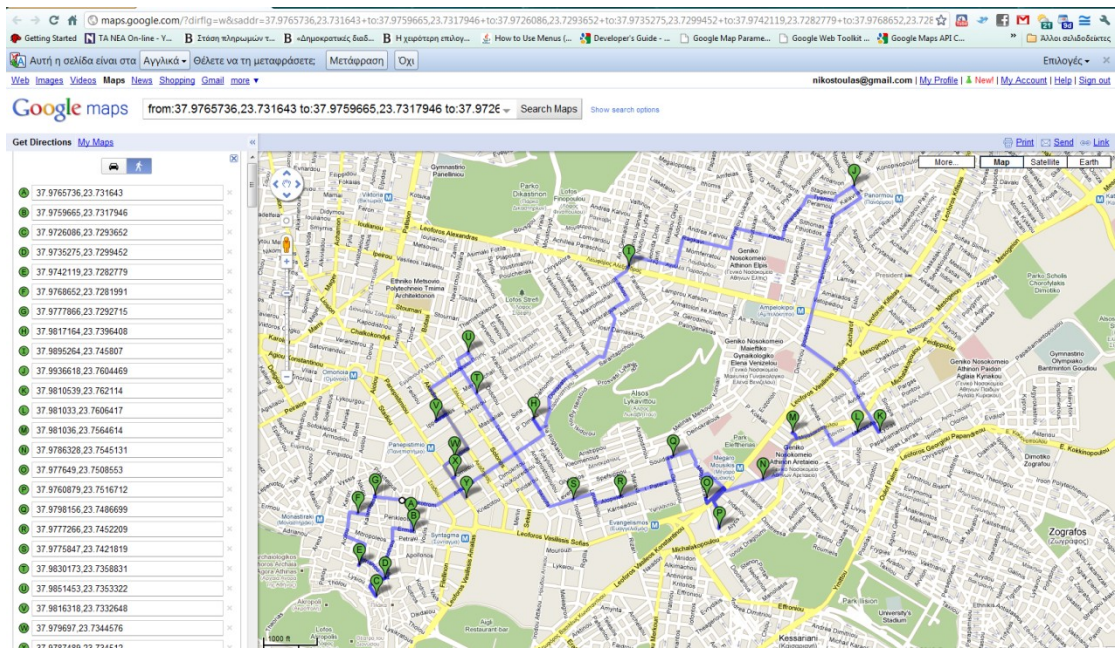
Βλέπουμε τα αποτελέσματα του αρχείου για έναν από αυτούς, έστω την Δήμητρα Δ.

ΑΑ	Οδός	Αριθμός	Δήμος	Όνομα	Επώνυμο	Τηλέφωνο	Σημειώσεις
1	Περικλέους	11	Αθήνα				
2	Ερμού	17	Αθήνα				
3	Τριπόδων	12	Αθήνα				
4	Φλέσσα	1	Αθήνα				
5	Μνησικλέους	18	Αθήνα				
6	Αθηνάϊδος	9	Αθήνα				
7	Αγίου Μάρκου	1	Αθήνα				
8	Χέρσουνας	7	Αθήνα				
9	Ιωάννου Βαρβάκη	2	Αθήνα	Δημήτρης	Δημητρίου	52321	τηπ
10	Τριφυλίας	20	Αθήνα	Κώστας	Ιωάννου	3523	τηπ
11	Λυδίας	2	Αθήνα				
12	Μικράς Ασίας	20	Αθήνα				
13	Λαμπάκου	7	Αθήνα				
14	Παπαδοιαμαντοπούλου	17	Αθήνα				
15	Μονής Πετράκη	2	Αθήνα				
16	Μετή Χατζηγιάννη	7	Αθήνα				
17	Σουηδίας	61	Αθήνα				
18	Αλαπεκλής	40	Αθήνα				
19	Πατριάρχου Ισακίου	12	Αθήνα				
20	Ναυαρίνου	5	Αθήνα				
21	Ανδρέα Μεταξά	7	Αθήνα				
22	Ιπποκράτους	15	Αθήνα				
23	Σινά	6	Αθήνα				
24	Ομήρου	10	Αθήνα				
25	Λεωφόρος Ελευθερίου Βενιζέλου		Αθήνα				

Google maps:  
<http://maps.google.com/?dirflg=w&saadl=37.9765736,23.731649+to:37.9759665,23.7317946+to:37.9726086,23.7293652+to:37.9735275,23.7299452+to:37.9742119,23.7282708+to:37.9835264,23.745507+to:37.9936618,23.7404469+to:37.9810339,23.762114+to:37.9810339,23.7606417+to:37.981036,23.7564614+to:37.976156,23.7486699+to:37.9777266,23.7452209+to:37.9775847,23.7421819+to:37.9830179,23.7358831+to:37.9851453,23.7353322+to:37.9816318,23.72254>

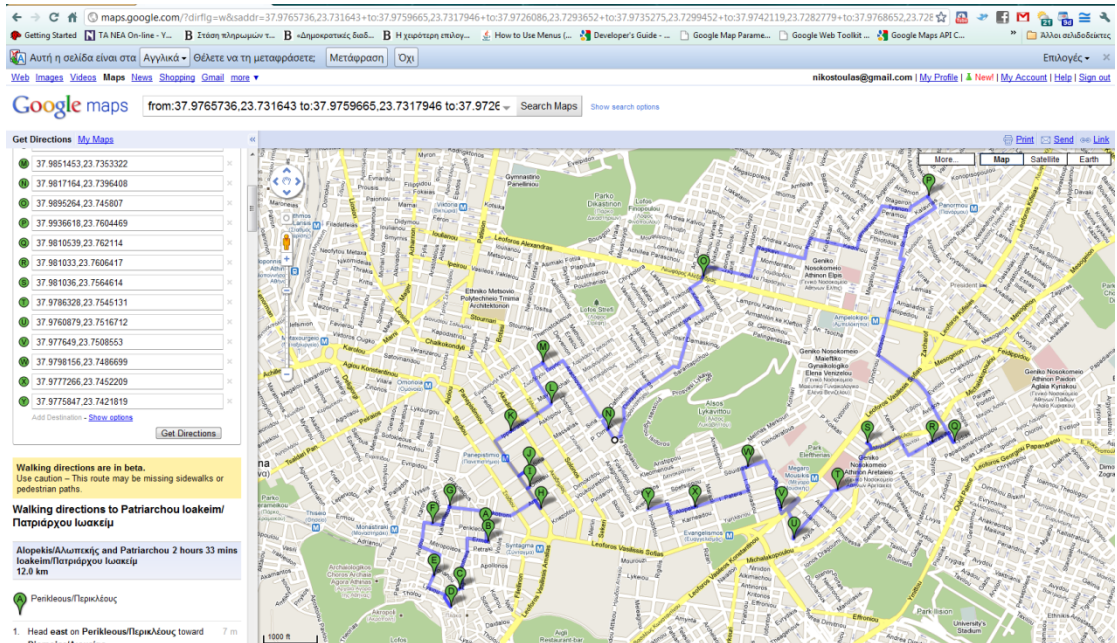
**Εικόνα 6.8 Προτεινόμενη σειρά επίσκεψης για την Δήμητρα Δ.**

Και στην συνέχεια ανοίγουμε τον σύνδεσμο του αρχείου, ώστε να δούμε την προτεινόμενη διαδρομή στην σελίδα maps.google.com.



**Εικόνα 6.9** Προτεινόμενη διαδρομή όπως φαίνεται από το Google Maps.

Τέλος αναδιατάσσουμε λίγο την διαδρομή και μειώνουμε το συνολικό της μήκος από 13.6 σε 12.3 χιλιόμετρα.



**Εικόνα 6.10** Αναδιαταγμένη διαδρομή.

# 7

## *Επίλογος*

Συνοψίζουμε στο σημείο αυτό την παρουσίαση της διπλωματικής.

### *7.1 Σύνοψη και συμπεράσματα*

Η υπηρεσία Google Maps πρωτοεμφανίστηκε στο διαδίκτυο τον Ιούνιο του 2005. (Παρόμοιες υπηρεσίες προσφέρονται έκτοτε και από άλλους οργανισμούς: Yahoo, Microsoft, OpensStreetMaps κ.α.) Η υπηρεσία προσφέρει εύκολη πρόσβαση σε χάρτες και γεωγραφικά δεδομένα οπουδήποτε, ακόμα και από κινητό, σε πολύ λίγο χρόνο, μέσω του διαδικτύου. Η αναζήτηση διευθύνσεων και τοποθεσιών καθώς και τρόπων μετάβασης έχει γίνει μέσω αυτών των υπηρεσιών υπόθεση δευτερολέπτων.

Πιο πολύπλοκα προβλήματα σχεδιασμού πολύπλοκων διαδρομών, διαχείριση πολλών διευθύνσεων, χρειάζονται τη χρήση επιπλέον λογισμικού και αλγορίθμων. Η παρούσα διπλωματική εργασία συμπληρώνει προς αυτή την κατεύθυνση προτείνοντας έναν τρόπο επίλυσης του επιμέρους προβλήματος της ΕΛ.ΣΤΑΤ.. Χρησιμοποιεί υπηρεσίες που έγιναν διαθέσιμες τα τελευταία χρόνια και αξιοποιεί αλγορίθμους που είναι γνωστοί εδώ και δεκαετίες (Αλγόριθμος του Χριστοφίδη).

Η ΕΛ.ΣΤΑΤ σήμερα αναθέτει ερευνώμενους σε ερευνητές με αλφαβητικό τρόπο χωρίς βέβαια να τους προτείνει σειρά επίσκεψης. Η χρήση της παρούσας διπλωματικής είναι προφανές πως μπορεί να γλιτώσει τους ερευνητές από αρκετό χαμένο χρόνο. Τους αναθέτει ερευνώμενους με βάση την χωρική τους τοποθεσία, τους προτείνει σειρά επίσκεψης και τους

δίνει την δυνατότητα να την δουν και να την τροποποιήσουν χρησιμοποιώντας την υπηρεσία του Google Maps στο διαδίκτυο.

Η εφαρμογή θα μπορούσε με τροποποιήσεις να χρησιμοποιηθεί σε παρόμοια προβλήματα με αυτό της ΕΛ.ΣΤΑΤ, όπου την θέση του ερευνητή μπορεί να έχουν φορτηγά, ταχυδρόμοι, μεταφορικές και του ερευνώμενου καταστήματα, αποδέκτες κλπ.

## **7.2 Μελλοντικές επεκτάσεις**

Μελλοντικές επεκτάσεις της εφαρμογής μπορεί να είναι να γίνει περισσότερο παραμετρική και ο χρήστης να μπορεί να επιλέξει και να καθορίσει τον τρόπο λειτουργίας της.

Επίσης θα μπορούσε να προτείνεται εκτός από διαδρομή και μια ανάθεση ερευνώμενων σε ερευνητές, ώστε ο χρήστης να χρειάζεται να κάνει ακόμα λιγότερα πράγματα, αφού θα χρειάζεται μόνο να τροποποιήσει την προτεινόμενη κατανομή.

Τέλος η εφαρμογή έχει την σκοπιά του διαχειριστή της έρευνας. Θα μπορούσε να αναπτυχθεί συμπληρωματική εφαρμογή που να έχει την σκοπιά του ερευνητή.

Αυτή η εφαρμογή θα έπαιρνε ως είσοδο την προτεινόμενη σειρά επίσκεψης που δίνει η εφαρμογή της παρούσας διπλωματικής για τον χρήστη-ερευνητή. Ο χρήστης εδώ θα μπορεί να ελέγχει την λίστα αυτή, να του προτείνονται ημερήσιες διαδρομές ανάλογα με τον διαθέσιμο χρόνο του, να διαγράφει από αυτή τους ερευνώμενους που έχει επισκεφτεί κλπ. Η νέα εφαρμογή θα μπορούσε να λειτουργεί και σε κινητό, έτσι ώστε ο ερευνητής να μην χρειάζεται να εκτυπώνει πληροφορίες.



# 8

## Βιβλιογραφία

- [Ait09] Aitchison, A. (2009). *Beginning Spatial with SQL Server 2008*. Apress.
- [Moo01] Moore, Andrew W. (2001, November 16). *K-means and hierarchical Clustering*. Retrieved from <http://www.autonlab.org/tutorials/kmeans11.pdf>
- [Min08] Ming-Yang, Kao. (2008). *Encyclopeida of Algorithms*.
- [Lar03] Larkin, F. J. (2003). *Coastal Navigation Using Gps: For Sail and Power*. Sheridan House Inc.
- [Zac06] Zachariasen, M. (2006, December 6). *Metric and Euclidean TSP*. Retrieved 9 2010, from [http://www.diku.dk/OLD/undervisning/2006-2007/2006-2007\\_b2\\_415/tspappr.pdf](http://www.diku.dk/OLD/undervisning/2006-2007/2006-2007_b2_415/tspappr.pdf)
- [Zur05] Zuravicky, O. (2005). *Map Math: Learning About Latitude And Longitude Using Coordinate Systems*. The Rosen Publishing Group, Inc.
- [Γαλ09] Γαλάνης, Ανδρέας. (2009, Απρίλιος 30). *Arora's PTAs for Euclidean TSP*. Retrieved from <http://corelab.ntua.gr/courses/approx-alg/material/Euclidean%20TSP.pdf>
- [Zαχ07] Ζάχος, Στάθης. (2007). *Αλγόριθμοι και πολυπλοκότητα*. Σημειώσεις Πολυτεχνείου.

- [WP1] [http://en.wikipedia.org/wiki/Christofides\\_algorithm](http://en.wikipedia.org/wiki/Christofides_algorithm)
- [WP2] [http://en.wikipedia.org/wiki/Matching\\_\(graph\\_theory\)](http://en.wikipedia.org/wiki/Matching_(graph_theory))
- [WP3] [http://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/Java_(programming_language))
- [WP4] [http://en.wikipedia.org/wiki/Java\\_Platform\\_Standard\\_Edition](http://en.wikipedia.org/wiki/Java_Platform_Standard_Edition)
- [WP5] [http://en.wikipedia.org/wiki/Eclipse\\_java](http://en.wikipedia.org/wiki/Eclipse_java)
- [Dei] [http://home.dei.polimi.it/matteucc/Clustering/tutorial\\_html/kmeans.html](http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html)
- [GOG1] <http://code.google.com/apis/maps/documentation/geocoding/>
- [GOG2] <http://code.google.com/apis/maps/documentation/staticmaps/>
- [GOG3] <http://code.google.com/apis/maps/documentation/directions/>
- [ORA1] <http://www.oracle.com/technetwork/java/javase/tech/index.html>
- [ORA2] <http://download.oracle.com/javase/6/docs/technotes/guides/swing/index.html>
- [ORA3] <http://download.oracle.com/javase/6/docs/technotes/guides/awt/index.html>
- [MySQL] <http://www.mysql.com/>
- [ECL] <http://www.eclipse.org/>