ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΕΞΟΡΥΞΗ ΓΝΩΣΗΣ ΑΠΟ ΙΣΤΟΡΙΚΑ ΔΕΔΟΜΕΝΑ ΚΙΝΗΣΗΣ (GOOGLE MAPS HISTORY)

Μπούρας Γρηγόριος

ΑΜ: 21239

Αθήνα

Σεπτέμβριος 2017

ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

Τριμελής Εξεταστική Επιτροπή

Επιβλέπων : **Βαρλάμης Ηρακλής,** Επίκουρος Καθηγητής

Μέλη Τριμελούς Επιτροπής:

Μιχαήλ Δημήτριος, Επίκουρος Καθηγητής

Τσερπές Κωνσταντίνος, Επίκουρος Καθηγητής

Ο Μπούρας Γρηγόριος δηλώνω υπεύθυνα ότι:

1) Είμαι ο κάτοχος των πνευματικών δικαιωμάτων της πρωτότυπης αυτής εργασίας και από όσο γνωρίζω η εργασία μου δε συκοφαντεί πρόσωπα, ούτε προσβάλει τα πνευματικά δικαιώματα τρίτων.

2) Αποδέχομαι ότι η ΒΚΠ μπορεί, χωρίς να αλλάξει το περιεχόμενο της εργασίας μου, να τη διαθέσει σε ηλεκτρονική μορφή μέσα από τη ψηφιακή Βιβλιοθήκη της, να την αντιγράψει σε οποιοδήποτε μέσο ή/και σε οποιοδήποτε μορφότυπο καθώς και να κρατά περισσότερα από ένα αντίγραφα για λόγους συντήρησης και ασφάλειας.

Ευχαριστίες

Η παρούσα πτυχιακή εργασία εκπονήθηκε στα πλαίσια των προπτυχιακών σπουδών του τμήματος Πληροφορικής και Τηλεματικής του Χαροκοπείου Πανεπιστημίου. Σε αυτό το σημείο, θα ήθελα να εκφράσω τις ευχαριστίες μου σε όλους όσους συνέβαλαν άμεσα ή έμμεσα στην ολοκλήρωση της πτυχιακής εργασίας μου.

Αρχικά θα ήθελα να εκφράσω τις ιδιαίτερες ευχαριστίες μου στον επιβλέποντα καθηγητή της πτυχιακής μου εργασίας. Επίκουρο Καθηγητή Ηρακλή Βαρλάμη για την συνεχή του καθοδήγηση και τις πολύτιμες συμβουλές που μου έδωσε κατά την διάρκεια εκπόνησης της πτυχιακής μου εργασίας.

Στην συνέχεια θα ήθελα να ευχαριστήσω τα μέλη της τριμελούς εξεταστικής επιτροπής, τον Επίκουρο Καθηγητή, Μιχαήλ Δημήτριο και τον Επίκουρο Καθηγητή, Τσερπέ Κωνσταντίνο για την αξιολόγηση της προσπάθειας μου αλλά και για την συμβολή των πολύτιμων γνώσεων τους κατά διάρκεια των σπουδών μου.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου, καθώς και το κοντινό μου περιβάλλον για την καθημερινή τους υποστήριξη καθώς και για την ενθάρρυνση τους κατά την διάρκεια της εκπόνησης της παρούσας πτυχιακής εργασίας, αλλά και για την υποστήριξη τους καθόλη τη διάρκεια των σπουδών μου.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Περίληψη στα Ελληνικά σ.7

Abstract σ.8

Κατάλογος Εικόνων σ.9

Κατάλογος Πινάκων. σ.10

Κατάλογος Σχημάτων σ.11

Κεφάλαιο1: Εισαγωγή σ.12

1.1 Στόχος της εργασίας σ.12

1.2 Προτεινόμενη προσέγγιση σ.13

Κεφάλαιο 2: Θεωρητικό υπόβαθρο σ.15

2.1 Σχετικές έρευνες σ.15

2.1.1 Παράγοντες επιλογής ενός σημείου ενδιαφέροντος από τον χρήστη. σ.15

2.1.2 Ανίχνευση μοτίβων από δεδομένα κίνησης σ.16

2.1.3 Πρόβλεψη των δραστηριοτήτων του χρήστη σ.17

2.1.4 Συμπεράσματα σ.19

2.2 Σχετικές υλοποιήσεις σ.19

2.2.1 Βιβλιοθήκη Traminer σ.19

Κεφάλαιο 3: Σχεδιασμός σ.22

3.1 Αρχιτεκτονική εφαρμογής σ.22

3.2 Μεθοδολογία υλοποίησης εφαρμογής σ.24

3.2.1 Ανίχνευση συχνών τοποθεσιών του χρήστη σ.25

3.2.2 Ανίχνευση τύπου κινήσεων του χρήστη σ.26

3.2.3 Ανίχνευση συχνών διαδρομών του χρήστη σ.27

3.2.4 Εξαγωγή των συνηθειών του χρήστη σ.28

3.3 Εργαλεία υλοποίησης της εφαρμογής σ.28

3.3.1 Δομή του αρχείου σε μορφή JSON σ.28

3.3.2 Βιβλιοθήκη Jackson σ.29

3.3.3 Βιβλιοθήκη Commons Math σ.29

3.3.4 JMapViewer σ.30

3.3.5 OpenStreetMap σ.30

3.3.6 Overpass-API σ.30

3.4 Εργαλεία Ανάπτυξης της Εφαρμογής σ.31

3.4.1 Netbeans-IDE σ.31
Κεφάλαιο 4: Υλοποίηση σ.32
4.1 Επεξεργασία δεδομένων του ιστορικού τοποθεσιών σ.32
4.1.1 Περιγραφή του κώδικα της επεξεργασίας σ.32
4.1.2 Ο κώδικας για την μετατροπή JSON σε Αντικείμενο σ.33
4.2 Υλοποίηση ανίχνευσης συχνών τοποθεσιών σ.33
4.2.1 Περιγραφή υλοποίησης ανίχνευσης σημείων ενδιαφέροντος σ.33
4.2.2 Ο Κώδικας για την ανίχνευση σημείων ενδιαφέροντος σ.34
4.2.3 Περιγραφή υλοποίησης ανίχνευσης συχνών τοποθεσιών σ.35
4.2.4 Ο Κώδικας για την ανίχνευση συχνών τοποθεσιών σ.36
4.3 Υλοποίηση ανίχνευσης τύπων κινήσεων σ.37
4.4 Υλοποίηση ανίχνευσης συχνών διαδρομών σ.39
4.4.1 Δημιουργία διαδρομών σ.40
4.4.2 Περιγραφή αλγόριθμου για την ανίχνευση συχνών διαδρομών σ.40
4.5 Εξαγωγή αρχείου με τις συνήθειες του χρήστη σ.43
4.6 Δημιουργία Γραφικού Περιβάλλοντος Χρήστη σ.44
Κεφάλαιο 5 : Αποτελέσματα σ.47
5.1 Ανάλυση αποτελεσμάτων συχνών τοποθεσιών σ.47
5.2 Ανάλυση αποτελεσμάτων ανίχνευσης τύπων κινήσεων σ.49
5.3 Ανάλυση αποτελεσμάτων συχνών διαδρομών σ.51
5.4 Ανάλυση αποτελεσμάτων εξαγωγής συνηθειών του χρήστη σ.53
Κεφάλαιο 6: Συμπεράσματα σ.55
6.1 Συμπεράσματα εφαρμογής σ.55
Βιβλιογραφία σ.56
Άλλες πηγές σ.57

Περίληψη

Η ανάπτυξη των έξυπνων κινητών τηλεφώνων (smartphones) παρέχει στους χρήστες τους ένα τεράστιο πλήθος πληροφοριών. Μέσα σε αυτές τις πληροφορίες περιέχονται και αυτές της καταγραφής γεωγραφικών συντεταγμένων με χρονοσφραγίδα. Η καταγραφή αυτή μπορεί να γίνει με διάφορους τόπους, είτε στη συσκευή, είτε σε κάποιο κεντρικό server. Στη περίπτωση που μελετάμε, γίνεται με την χρήση της αντίστοιχης υπηρεσίας της Google. Τα δεδομένα αποθηκεύονται σε ένα αρχείο, το οποίο ο χρήστης μπορεί να το αποκτήσει μέσα από τον ιστότοπο του λογαριασμού του στην Google. Η παρούσα εργασία στοχεύει την ανάπτυξη μιας εφαρμογής για την ανάλυση των δεδομένων κίνησης από τα αρχεία αυτής της μορφής, με στόχο την εξαγωγή χρήσιμων πληροφοριών σχετικά με την συμπεριφορά του χρήστη καθώς και τις συνήθειες του. Η ανάπτυξη της εφαρμογής αυτής θα γίνει σε γλώσσα Java.

Αναλυτικότερα, στο πρώτο επίπεδο της εργασίας η ανάλυση στοχεύει στον εντοπισμό τοποθεσιών όπου περνά αρκετή ώρα ο χρήστης. Στην συνέχεια εντοπίζει τους τύπους κινήσεων που προτιμά ο χρήστης για τις μετακινήσεις τους μεταξύ αυτών των τοποθεσιών. Τέλος βρίσκει τις διαδρομές του χρήστη και εξάγει αυτές που κάνει πιο συχνά.

Στο δεύτερο επίπεδο της εργασίας χρησιμοποιούμε την πληροφορία για τις συχνές τοποθεσίες που περνά την ώρα του ο χρήστης και με την βοήθεια του OpenStreetMap API βρίσκουμε τι υπάρχει σε αυτήν την τοποθεσία. Τέλος εξάγουμε την γνώση αυτή σε ένα αρχείο, το οποίο περιέχει της συνήθειες του χρήστη.

Θεματική περιοχή: Εξόρυξη δεδομένων

Λέξεις κλειδιά: Σημεία ενδιαφέροντος, Συσταδοποίηση, Συχνές διαδρομές, OpenStreetMaps

Abstract

Nowadays, the development of smartphones provides their users with an immense amount of information including the recording of time-stamped geographical coordinates. There are a few different ways this can be done but some major ones are either on the device or in a remote accessed server. In this current case this is done by using a google service in which all the data are stored in a file accessible through the user's google account. This thesis aims to develop an application for analyzing traffic data from such records while the goal in this is to extract useful information about the user's behavior and habits. The development of this application will be done with the Java programming language.

In more detail, the first level of this project pinpoints all of the locations the user spends an adequate amount of time to and then it calculates all of the various types of movement the user prefers to do. The application then proceeds to find the user's trajectories and exports these who does most often.

For the second part of this project, it uses the information of the frequent locations that user spends time on and with the use of the OpenStreetMap API it finds automatically the type of place that exists there. Finally, the last step is for these data to be extracted into a file, which will contain the user's habits for future process.

Keywords: points of interest, clustering, frequent trajectories, OpenStreetMaps

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικ.4.1. Η μέθοδος για την αντιστοίχιση του αρχείου JSON στις κλάσεις σ.33
Εικ.4.2. Η μέθοδος που επιστρέφει την λίστα με τα σημεία ενδιαφέροντος σ.34
Εικ.4.3. Χαρακτηριστικά κλάσης Habit σ.35
Εικ.4.4. Η μέθοδος που επιστρέφει την λίστα με τις συνήθειες του χρήστη σ.36
Εικ.4.5. Ο ψευδοκώδικας για την εύρεση σημείων στάσης του χρήστη σ.38
Εικ.4.6. Υλοποίηση του αλγόριθμου για τα σημεία στάσης σ.39
Εικ.4.7. Τα βήματα 1-12 του αλγορίθμου CBM σ.41
Εικ.4.8. Τα βήματα 13-31 του αλγορίθμου CBM σ.42
Εικ.4.9. Τα βήματα 32-50 του αλγορίθμου CBM σ.43
Εικ.4.10. Απεικόνιση του γραφικού περιβάλλοντος χρήστη σ.45
Εικ.5.1. Στιγμιότυπα πριν και μετά από την ανίχνευση σημείων ενδιαφέροντος σ.48
Εικ.5.2. Στιγμιότυπο από το αρχείο με τις συνήθειες του χρήστη σ.48
Εικ.5.3. Αρχείο που περιέχει πληροφορίες για τους τύπους κινήσεων σ.50
Εικ.5.4. Στιγμιότυπο μετά την ανίχνευση συχνών διαδρομών σ.51
Εικ.5.5. Στιγμιότυπο από το αρχείο που δημιουργήθηκε με τις συνήθειες του χρήστη σ.54

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίν.5.1: Χαρακτηριστικά υπολογιστή που στηρίχτηκαν τα αποτελέσματα σ.47
Πίν.5.2: Χρόνοι εκτέλεσης σημείων ενδιαφέροντος σ.49
Πίν.5.3: Χρόνοι εκτέλεσης της διαδικασίας των συχνών διαδρομών σ.51
Πίν.5.4: Χρόνοι εκτέλεσης συχνών διαδρομών με διαφορετικό S σ.52
Πίν.5.5: Μελέτη της μεταβλητής threshold σ.53

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχ.3.1: Η βασική αρχιτεκτονική του προγράμματος σ.23
Σχ.3.2: Η Γενική αρχιτεκτονική της εφαρμογής σ.24
Σχ.3.3: Διαδικασία ανίχνευσης σημείων ενδιαφέροντος σ.25
Σχ.3.4: Διαδικασία ανίχνευσης συχνών διαδρομών σ.27
Σχ.3.5: Δομή του αρχείου ιστορικού δεδομένων σ.29
Σχ.4.2: Διαδικασία ανίχνευσης τύπου κινήσεων σ.37

Κεφάλαιο 1

Εισαγωγή

Η συνεχής ανάπτυξη των έξυπνων τηλεφώνων (smartphones) σε λογισμικό αλλά και υλικό προσφέρει στον χρήστη αποδοτικότερες υπηρεσίες και εφαρμογές. Μαζί με αυτήν την ανάπτυξη έρχεται και ένας τεράστιος αριθμός από πληροφορίες. Από την αύξηση αυτή δημιουργείται η ανάγκη εξόρυξης γνώσης από αυτές, για να γίνουν τα έξυπνα τηλέφωνα πιο χρήσιμα στον χρήστη. Ένα μέρος αυτών των πληροφοριών μπορεί να έρχεται από την καταγραφή χωρικών στιγμιότυπων που γίνονται από τους αισθητήρες GPS.

Οι αισθητήρες αυτοί, αν το επιθυμεί ο χρήστης του τηλεφώνου, καταγράφουν τις συντεταγμένες του και στην συνέχεια αποθηκεύονται, μαζί με τον χρόνο που καταγραφτήκαν σε μια βάση δεδομένων. Τα δεδομένα αυτά μετά από επεξεργασία μέσω κάποιον μεθόδων και τεχνικών μπορούν δώσουν αρκετές πληροφορίες σχετικές με τις συνήθειες του χρήστη. Κάποιες από αυτές τις πληροφορίες έχουν να κάνουν με τις συχνές τοποθεσίες που επισκέπτεται ο χρήστης, ενώ κάποιες άλλες είναι σχετικές με τα μοτίβα κίνησης του. Όμως σε αυτά τα δεδομένα υπάρχουν και σημεία που έχουν καταγραφεί από σφάλμα. Αυτά είναι θόρυβος για τα αποτελέσματα.

Για αυτό λοιπόν δημιουργηθήκαν τεχνικές για την απομάκρυνση θορύβου και με έναν πιο γενικό όρο τεχνικές για τον καθαρισμό τον δεδομένων από πληροφορίες που δεν είναι χρήσιμες. Ένα άλλο είδος μεθόδων που ερευνούνται έχει να κάνει με την συμπίεση των δεδομένων χωρίς να χαθεί η πληροφορία. Έτσι αν για παράδειγμα έχουμε ένα σύνολο από κάποιες διαδρομές, γίνεται προσπάθεια συμπίεσης των δεδομένων με τέτοιο τρόπο ώστε να μην χαθούν οι πληροφορίες που είναι σχετικές με τις διαδρομές. Η προσπάθεια της συμπίεσης γίνεται καθώς τα δεδομένα θα έχουν πρόβλημα αποθήκευσης όταν ο αριθμός τους γίνει πολύ μεγάλος.

1.1 Στόχος της εργασίας

Ο στόχος της εργασίας είναι η προσπάθεια ανίχνευσης συνηθειών του χρήστη από το ιστορικό τοποθεσιών της Google maps. Συγκεκριμένα θα γίνει η προσπάθεια ανίχνευσης συχνών τοποθεσιών που βρέθηκε ο χρήστης κατά την διάρκεια της καταγραφής του ιστορικού.

Στην συνέχεια θα γίνει προσπάθεια για την εξαγωγή συμπερασμάτων, για τους τύπους κίνησης που προτιμάει ο χρήστης να κάνει. Και τέλος θα γίνει η προσπάθεια ανίχνευσης συχνών διαδρομών του χρήστη.

Όλες αυτές οι πληροφορίες που θα εξάγουμε βοηθάνε να κατανοήσουμε καλύτερα την συμπεριφορά του χρήστη. Έτσι ερχόμαστε ένα βήμα πιο κοντά στην κατανόηση των αναγκών του χρήστη. Η κατανόηση αυτή προσφέρει πληροφορίες σχετικά με το τι χρειάζεται ο χρήστης καθώς και το τι θέλει να κάνει. Οπότε αυτές τις πληροφορίες μπορεί να της χρησιμοποιήσει κάποιος για να αναπτύξει εφαρμογές και συστήματα που θα οδηγήσουν στην ανάπτυξη των έξυπνων τηλεφώνων σε ακόμη πιο χρήσιμα συστήματα για τον χρήστη.

1.2 Προτεινόμενη προσέγγιση

Για τον σκοπό αυτόν δημιουργήθηκε η παρούσα εφαρμογή, η οποία έχει φτιαχτεί για την εκτέλεση της σε έναν προσωπικό υπολογιστή. Η εφαρμογή έχει υλοποιηθεί σε γλώσσα Java. Η προτεινόμενη προσέγγιση της παρούσας πτυχιακής εργασίας έχει ως αρχικό στόχο την ανίχνευση συχνών τοποθεσιών του χρήστη, μέσα από το ιστορικό τοποθεσιών της Google maps που έχει ορίσει ο χρήστης. Αυτό γίνεται αφού έχει αποκτηθεί το αρχείο με το ιστορικό καταγραφής δεδομένων κίνησης από την Google maps. Ύστερα αναλύονται ο τύπος κίνησης που προτιμάει ο χρήστης. Στην συνέχεια γίνεται η προσπάθεια ανίχνευσης για συχνές διαδρομές που εκτελεί ο χρήστης που προκύπτουν από τα δεδομένα του ιστορικού. Τέλος με βάση τις συχνές τοποθεσίες του χρήστη προκύπτουν κάποιες συνήθειες του για τις τοποθεσίες. Μια από αυτές είναι η συχνότητα που εμφανίζεται η τοποθεσία σε μια χρονική περίοδο. Επίσης μέσα σε αυτές τις πληροφορίες έχουν προστεθεί και πληροφορίες σχετικές με την συγκεκριμένη τοποθεσία. Όπως για παράδειγμα αν είναι ένα εστιατόριο, η εφαρμογή βρίσκει το όνομα του.

Αφού έχουν υλοποιηθεί όλα τα παραπάνω για τις ανάγκες της παρούσας εφαρμογής έχει αναπτυχθεί ένα απλό Γραφικό περιβάλλον χρήστη, ώστε να μπορεί να γίνει μια συλλογική επεξεργασία των δεδομένων με βάση τις παραπάνω επιλογές. Ουσιαστικά παρέχεται στον χρήστη της εφαρμογής η ανάλυση των δεδομένων με διαφορετικές επιλογές. Ο χρήστης έχει την ευκαιρία να ρυθμίζει κάποιες παραμέτρους που επηρεάζουν τα αποτελέσματα. Ακόμη δίνεται η δυνατότητα απεικόνισης των αποτελεσμάτων στις αντίστοιχες περιοχές σε έναν χάρτη της Open Street Map.

Σχετικά με την ανίχνευση συχνών τοποθεσιών τα αποτελέσματα προκύπτουν μέσα από μεθόδους αναζήτησης σημείων ενδιαφέροντος καθώς και την ανάλυση αυτών. Οι προτιμήσεις του χρήστη που αφορούν των τύπο κίνησης προκύπτει από ανάλυση των πληροφοριών που μας παρέχει το ιστορικό καταγραφής δεδομένων κίνησης η Google. Στις πληροφορίες που είναι σχετικές με τον τύπο κίνησης γίνεται και η προσπάθεια ανάλυσης για τον τύπο περιοχών που εξελίσσονται αυτές οι κινήσεις. Για παράδειγμα αν ο χρήστης δραστηριοποιείται περισσότερο σε ένα πάρκο από ότι σε μια παραλία. Αυτό γίνεται με χρήση του OverPass-API που παρέχεται από την OpenStreetMap.

Για τις συχνές διαδρομές ο τρόπος που προκύπτουν είναι μέσα από μια διαδικασία επεξεργασίας των δεδομένων καθώς και από την χρήση αποτελεσματικών μεθόδων για την ανίχνευση των διαδρομών αυτών. Η πρώτη προσπάθεια γίνεται για την ανίχνευση των διαδρομών. Για να γίνει αυτό γίνεται προσπάθεια απομάκρυνσης του θορύβου, όπως επίσης η προσπάθεια αναγνώρισης των διαδρομών μέσα από τα δεδομένα ιστορικού. Αφού πραγματοποιηθεί αυτό και έχει γίνει η συλλογή των διαδρομών, για μια χρονική περίοδο, τότε εφαρμόζονται σε αυτές οι τεχνικές και μέθοδοι για την ανίχνευση των πιο συχνών διαδρομών.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

2.1 Σχετικές έρευνες

Ο τεράστιος αριθμός δεδομένων που συλλέγεται καθημερινά από τους αισθητήρες GPS έχει ανοίξει δρόμους για πολλές έρευνες και εφαρμογές. Αρκετές είναι οι έρευνες που έχουν να κάνουν με την κατανόηση της συμπεριφοράς χρήστη μέσα από τα δεδομένα κίνησης που έχουν καταγραφτεί. Οι έρευνες εστιάζουν στην προσπάθεια εύρεσης πληροφοριών που βοηθάνε στην κατανόηση των συμπεριφορών αυτών.

Σε αυτήν την ενότητα θα παρουσιαστούν σχετικές έρευνες που έχουν γίνει που είναι πολύ κοντά με το αντικείμενο της παρούσας πτυχιακής εργασίας. Πρώτα θα αναφερθούν κάποιες σχετικές έρευνες που έχουν να κάνουν με την ανίχνευση σημείων ενδιαφέροντος του χρήστη. Στην συνέχεια έρευνες που ασχολούνται με την ανάλυση μεθόδων ανίχνευσης μοτίβων από τα δεδομένα κίνησης. Ακόμη θα γίνει και μια αναφορά σε έρευνες όπου η προσπάθεια τους επικεντρώνεται στην πρόβλεψη των δραστηριοτήτων του χρήστη. Μέσα από αυτές τις έρευνες θα αναλυθούν τα συμπεράσματα που προκύπτουν. Τέλος θα παρουσιαστούν μερικές σχετικές υλοποιήσεις που είναι πολύ κοντά στην παρούσα εφαρμογή.

2.1.1 Παράγοντες επιλογής ενός σημείου ενδιαφέροντος από τον χρήστη

Αρκετές είναι οι έρευνες που έχουν ασχοληθεί με την ανίχνευση σημείων ενδιαφέροντος του χρήστη. Τα σημεία αυτά αποτελούν μια σημαντική πληροφορία για την κατανόηση της συμπεριφοράς του χρήστη. Η πληροφορία αυτή δίνει την ευκαιρία για ανάπτυξη εφαρμογών που έχουν στόχο να προτείνουν στον χρήστη σημεία που τον ενδιαφέρουν. Για τον στόχο αυτό πολλές έρευνες έχουν γίνει για την συσχέτιση μεταξύ διάφορων παραγόντων και των σημείων αυτών. Κάποιες από αυτές που έχουν γίνει αφορούν την συσχέτιση των σημείων αυτών με βάση τον κοινωνικό παράγοντα, την ατομική προτίμηση του χρήστη και δημοτικότητα του σημείου (Ying et al., 2012). Κάποιες άλλες η προσπάθεια συσχέτισης που μελετήθηκε έγινε με

βάση την ώρα και τα σημεία αυτά (Yuan et al., 2013). Δηλαδή αν υπάρχει κάποια επίδραση της ώρας με την επιλογή των σημείων.

Η πρόταση των Ying et al. (2012) είναι μια προσέγγιση που λέγεται UPOI-Mine που ενσωματώνει τα κοινωνικά δίκτυα που βασίζονται στην τοποθεσία (Location-based social Network) για την σύσταση αστικών σημείων ενδιαφέροντος του χρήστη με βάση τις προτιμήσεις του (όπως για παράδειγμα ένα εστιατόριο) αλλά και τις ιδιότητες της συγκεκριμένης τοποθεσίας. Συγκεκριμένα στην προτεινόμενη προσέγγιση αυτό που γίνεται είναι η εφαρμογή ενός αλγορίθμου δυο φάσεων. Η πρώτη φάση φροντίζει την εξαγωγή χαρακτηριστικών ενώ η δεύτερη φάση είναι υπεύθυνη για την σύσταση του εστιατορίου. Για την πρόταση τους ερευνήθηκε ο κοινωνικός παράγοντας, η ατομική προτίμηση και η δημοτικότητα του σημείου ενδιαφέροντος για την εξαγωγή περιγραφικών χαρακτηριστικών. Τα δεδομένα που χρησιμοποιήθηκαν για τον λόγο αυτό ήταν αυτά από τα κοινωνικά δίκτυα που βασίζονται στην τοποθεσία. Τα πειράματα τους έδειξαν ότι η πρόταση τους έχει εξαιρετικές αποδόσεις κάτω από διάφορες συνθήκες.

Μια άλλη προσέγγιση του θέματος έγινε από τους Yuan et al. (2013). Σε αυτήν έγινε η μελέτη για την συσχέτιση της ώρας και των σημείων ενδιαφέροντος του χρήστη. Συγκεκριμένα αυτό που κάνανε είναι η δημιουργία μιας μεθόδου που χρησιμοποιεί την χρονική επιρροή για συστάσεις σημείων ενδιαφέροντος. Στη συνέχεια προτείνουν μια νέα προσέγγιση που διερευνά τη χωρική επίδραση στα σημεία αυτά. Και τέλος, συνδυάζουν τις δύο προσεγγίσεις μέσω μιας ενοποιημένης δομής. Τα πειραματικά αποτελέσματα έδειξαν ότι οι προτεινόμενες μέθοδοι καλύπτουν όλες τις βασικές ανάγκες του προβλήματος και βελτιώνουν την ακρίβεια των συστάσεων για τα σημεία ενδιαφέροντος κατά περισσότερο από 37% έναντι της τελευταίας τεχνολογίας μεθόδου. Η μέθοδος τους έχει να κάνει με τις ώρες της ημέρας. Εκτός από την ώρα της ημέρα θα μπορούσαν να γίνουν αρκετές μελέτες για την συσχέτιση της ημέρας της εβδομάδας, του μήνα και του έτους. Δηλαδή αν συσχετίζονται αυτοί οι παράγοντες με τις επιλογές των σημείων ενδιαφέροντος.

2.1.2 Ανίχνευση μοτίβων από δεδομένα κίνησης

Μια συνήθεια του χρήστη που παίζει σημαντικό ρόλο στην κατανόηση της συμπεριφοράς του είναι και αυτή των συχνών διαδρομών. Μέσω αυτής της πληροφορίας μπορούν να αναπτυχθούν αρκετές εφαρμογές για την βοήθεια του χρήστη σε ότι έχει να κάνει με τις καθημερινές του κινήσεις.

Μια μέθοδος που προτάθηκε από τους Shaw και Gopalan (2011), είναι η ανίχνευση συχνών διαδρομών από ένα σύνολο με διαδρομές με την χρήση του αλγορίθμου Apriori. Ο αλγόριθμος αυτός χρειάζεται μόνο τις συντεταγμένες, αγνοώντας την παράμετρο του χρόνου. Περιληπτικά ο αλγόριθμος δέχεται ως είσοδο ένα σύνολο από διαδρομές και μια οριακή τιμή ξ. Σαν πρώτο βήμα βρίσκει την συχνότητα εμφάνισης από κάθε σημείο που υπάρχει στην λίστα. Αν η συχνότητα αυτή είναι μεγαλύτερη ή ίση με την τιμή ξ τότε το σημείο αποθηκεύεται σε μια συλλογή L. Στην συνέχεια για τα σημεία που βρίσκονται στην λίστα L δημιουργεί ζεύγη με το επόμενο σημείο από την διαδρομή που ανήκουν. Για κάθε ζεύγος βρίσκει την συχνότητα εμφάνισης του στις διαδρομές και όποια ζεύγη έχουν συχνότητα εμφάνισης μεγαλύτερη ή ίση από τα ξ τότε καταχωρούνται σε μια νέα λίστα L2. Στην συνέχεια η ίδια διαδικασία ακολουθείται μέχρι να βρεθούν οι συχνές διαδρομές. Ο αλγόριθμος αυτός είναι αρκετά απλός και τα αποτελέσματα του αποδοτικά ως προς το περιεχόμενο.

Η ανίχνευση μοτίβων από δεδομένα κίνησης είναι χρήσιμη για πολλές άλλες θεματικές περιοχές πέρα από την ανάλυση συνηθειών του χρήστη. Όπως για παράδειγμα για την ανάλυση των συμπεριφορών άγριων ζώων. Μια τέτοια μελέτη έγινε από τους Chen et al. (2010). Συγκεκριμένα έγινε μια προσέγγιση της ανίχνευσης συχνών διαδρομών μέσα από δεδομένα που συλλέχθηκαν από αισθητήρες GPS, οι οποίοι ήταν συνδεδεμένοι σε κάποια πτηνά. Στην συνέχεια χρησιμοποίησαν τον αλγόριθμο DBSCAN για να απομακρύνουν τον θόρυβο και να εντοπίσουν κάποια μοτίβα κίνησης των πτηνών. Φυσικά αυτά τα δεδομένα μπορεί να τα χρησιμοποιήσει κάποιος ορνιθολόγος για τους σκοπούς μιας έρευνας.

2.1.3 Πρόβλεψη των δραστηριοτήτων του χρήστη

Τα τελευταία χρόνια έχουν αυξηθεί αρκετά τα κοινωνικά δίκτυα που βασίζονται στην τοποθεσίας (Location- based Social Network). Μια υπηρεσία των δικτύων αυτών είναι η κοινοποίηση της παρουσίας του χρήστη σε κάποια τοποθεσία. Πολλές είναι η έρευνες που έχουν ασχοληθεί με την ανάλυση αυτών των κοινοποιήσεων για να εξάγουν συνήθειες και προτιμήσεις του χρήστη.

Μια από αυτές της έρευνες είναι και αυτή των Noulas et al. (2011). Στο κείμενο τους γίνεται μελέτη σε μεγάλη κλίμακα για ένα από τα πιο γνωστά κοινωνικά δίκτυα που βασίζονται στην τοποθεσία, το Foursquare. Παρουσιάζεται μια ανάλυση από τις δραστηριότητες των χρηστών και δείχνουν τον τρόπο με τον οποίο οι κοινοποιήσεις παρουσίας παρέχουν πληροφορία σχετικά με τα καθημερινά και εβδομαδιαία σχέδια τους καθώς και

επαναλαμβανόμενες μεταβάσεις μεταξύ διαφορετικών δραστηριοτήτων. Το API του Foursquare παρέχει περιορισμένη εξουσιοδότηση πρόσβασης στα δεδομένα. Για την ανάγκη αυτή χρησιμοποιήθηκαν τα δημόσια δεδομένα που είναι διαθέσιμα σε μεγάλες ποσότητες από μηνύματα από το Twitter που περιέχουν κοινοποιήσεις παρουσίας Foursquare . Το γεγονός ότι τα δεδομένα αυτά έχουν προκύψει από τον ίδιο το χρήστη και όχι από κάποιο ιστορικό με συντεταγμένες κάνει την πληροφορία πιο δυνατή.

Συγκεκριμένα έγινε η μελέτη για τον αριθμό των κοινοποιήσεων για δυο κατηγορίες. Η μια είναι η καθημερινές μέρες και η άλλη τα Σαββατοκύριακα. Το χρονοδιάγραμμα για τον αριθμό των κοινοποιήσεων που γίνονται διαφορετικές χρονικές στιγμές της μέρας έδειξε ότι παρουσιάζονται τρεις κορυφές. Δηλαδή τρεις χρονικές περίοδοι που ο αριθμός των κοινοποιήσεων παρουσίας είναι μεγάλος σε σχέση με τις υπόλοιπες χρονικές στιγμές. Αυτές οι τρεις χρονικές περίοδοι είναι το πρωί όταν οι άνθρωποι πηγαίνουν στις δουλείες τους, το μεσημέρι και το βράδυ όταν γυρίζουν από τις δουλειές τους. Όσον αφορά τα Σαββατοκύριακα το χρονοδιάγραμμα είναι πιο ομαλό και δείχνει ότι ο αριθμός των κοινοποιήσεων είναι αρκετά μεγάλος για το χρονικό διάστημα μεταξύ 12πμ μέχρι 10μμ.

Στην συνέχεια γίνεται η ανάλυση για το πώς οι κοινοποιήσεις αυτές πραγματοποιούνται με την πάροδο του χρόνου και του χώρου. Αυτό που δείχνει η ανάλυση είναι ότι όσο πιο κοντά βρίσκονται οι κοινοποιήσεις χρονικά και χωρικά τόσο πιο πιθανό είναι οι δύο τοποθεσίες που βρέθηκε ο χρήστης να είναι διαδοχικές. Αυτό είναι λογικό αν σκεφτούμε ότι για αρκετά μεγάλες αποστάσεις μεταξύ τοποθεσιών ο χρήστης χρειάζεται αρκετό χρόνο. Ύστερα γίνεται η μελέτη για το αν υπάρχει κάποια συσχέτιση μεταξύ των δραστηριοτήτων του χρήστη. Για παράδειγμα αν μετά την δουλειά προτιμάει να πηγαίνει σε κάποιο εστιατόριο. Ακόμη η μελέτη αυτή εξηγεί πώς οι κοινοποιήσεις παρουσίας του χρήστη μπορούν να αποτελούν χώρο-χρονικά ίχνη και πως μπορεί αυτό μπορεί να βοηθήσει στη διερεύνηση των μεταβάσεων χρηστών από μια τοποθεσία ή δραστηριότητα σε μια άλλη. Η ανάλυση των κοινοποιήσεων παρουσίας του χρήστη μας ενημερώνει για την δραστηριότητα του χρήστη την ζητούμενη χρονική στιγμή και ζητούμενο χώρο. Μια εφαρμογή διαφημίσεων θα μπορούσε να λάβει την ανάλυση αυτή υπόψη της και να παρέχει τις αντίστοιχες υπηρεσίες.

Μια μεταβλητή που δεν υπάρχει στην μελέτη αυτή είναι ότι οι χρήστες μαζί με την κοινοποίηση της παρουσίας τους γράφουν συνήθως και κάποια σχόλια. Αυτό το κείμενο συνήθως περιέχει πληροφορίες σχετικές με την δραστηριότητα του χρήστη. Οι πληροφορίες αυτές ίσως είναι άξιες να μελετηθούν και να αναλυθούν για μελλοντικές εφαρμογές πάνω στο θέμα.

Ακόμη μια σχετική έρευνα είναι αυτή των Ye et al. (2013). Στόχος τους είναι πως γίνεται να βρεθεί η επόμενη δραστηριότητα του χρήστη καθώς και η επόμενη τοποθεσία, χρησιμοποιώντας μια ακολουθία από κοινοποιήσεις παρουσίας από ένα κοινωνικό δίκτυο. Χρησιμοποιήθηκε το μοντέλο hidden Markov για να μοντελοποιηθεί η εξάρτηση μεταξύ των κατηγοριών. Συγκεκριμένα για την καλύτερη απόδοση των αποτελεσμάτων χρησιμοποιήθηκε το μοντέλο mixed HMM. Ύστερα γίνεται η ανίχνευση της επόμενης τοποθεσίας με βάση της κατάταξης τους. Για να βρεθεί σε τι κατάταξη είναι μια τοποθεσία προτείνονται τα παρακάτω συστήματα. Το πρώτο αναφέρει ότι η κατάταξη της τοποθεσίας γίνεται με βάση το συνολικό αριθμό κοινοποιήσεων παρουσίας στην τοποθεσία αυτή. Το δεύτερο αναφέρει ότι η κατάταξη γίνεται με βάση το πλήθος των χρηστών που έχουν κοινοποιήσει την παρουσία τους εκεί. Το τρίτο προκύπτει με βάση το γινόμενο του πρώτου συστήματος και του δεύτερου. Τέλος το τέταρτο προκύπτει από τον μέγιστο αριθμό κοινοποιήσεων που έχουν γίνει από έναν χρήστη. Από τα πειράματα που πραγματοποιήθηκαν ανάλυση τους έδειξε ότι η προσέγγιση τους είναι αρκετά αποδοτική.

2.1.4 Συμπεράσματα

Η κατανόηση της συμπεριφοράς ενός χρήστη είναι απαραίτητη για την δημιουργία αποδοτικότερων εφαρμογών που καλύπτουν της ανάγκες του. Οι έρευνες και οι μελέτες που έχουν γίνει εντός του θέματος έχουν δημιουργήσει τεχνικές και μεθόδους αρκετά αποδοτικές. Φυσικά με την πάροδο του χρόνου οι τεχνικές αυτές αυξάνονται σε πλήθος, όπως επίσης γίνονται πιο αποδοτικές. Στο πλαίσιο της παρούσας πτυχιακής εργασίας γίνεται και μια προσπάθεια για την εξόρυξη χρήσιμων πληροφοριών, από ένα ιστορικό καταγραφής δεδομένων κίνησης, για την δυνατότητα ανάπτυξης μελλοντικών εφαρμογών.

2.2 Σχετικές υλοποιήσεις

2.2.1 Βιβλιοθήκη Traminer

Μια σχετική υλοποίηση που έχει γίνει είναι αυτή της βιβλιοθήκης Traminer. Η βιβλιοθήκη αυτή είναι για την γλώσσα Java. Αυτό που κάνει βιβλιοθήκη αυτή είναι η προ-επεξεργασία, η διαχείριση και η εξόρυξη χωρικών δεδομένων διαδρομών. Σκοπός της είναι να βοηθήσει τους ερευνητές, τους προγραμματιστές, επαγγελματίες αλλά και άλλους

επαγγελματίες που δεν ειδικεύονται στο αντικείμενο, να χειρίζονται πιο εύκολα ένα σύνολο από διαδρομές οχημάτων και πεζών χωρίς να εφαρμόζουν από την αρχή της πιο θεμελιώδης μεθόδους.

Η αρχιτεκτονική της βιβλιοθήκης αποτελείται από τρία επίπεδα και μια βιβλιοθήκη ανάγνωσης δεδομένων, τα οποία συλλογικά δημιουργούν μια πλήρη στοίβα από API, τα οποία βοηθάνε στην ανάλυση των δεδομένων για τις διαδρομές.

Τα δεδομένα κίνησης μπορεί να έχουν αποθηκευτεί με διάφορες μορφές, οπότε υπάρχει η ανάγκη μετατροπής των δεδομένων σε πιο αποδοτικές μορφές. Αυτό το κάνει η βιβλιοθήκη ανάγνωσης δεδομένων. Στόχος της είναι η μετατροπή των δεδομένων που έχουν αποθηκευτεί κάποιου είδους μορφή σε αντίστοιχες κλάσεις που έχουν φτιαχτεί. Αυτό γίνεται για να είναι πιο εύκολη η επεξεργασία των δεδομένων.

Το πρώτο επίπεδο της αρχιτεκτονικής είναι αυτό της προ-επεξεργασίας. Στο επίπεδο αυτό υπάρχουν βασικές μέθοδοι προ-επεξεργασίας διαδρομών. Αυτές έχουν να κάνουν με την απομάκρυνση θορύβου, την συμπίεση διαδρομών, την ανίχνευση σημείων στάσης του χρήστη, την βαθμονόμηση της διαδρομής, την αντιστοίχιση των διαδρομών σε κάποιο ψηφιακό χάρτη, καθώς και προσθήκη διαδρομών με βάση τα σημεία ενδιαφέροντος.

Το δεύτερο επίπεδο της αρχιτεκτονικής ασχολείται με συλλογές από διαδρομές. Συγκεκριμένα παρέχει υλοποιήσεις των πιο αντιπροσωπευτικών μέτρων ομοιότητας μεταξύ δυο διαδρομών. Για παράδειγμα κάποια από αυτά είναι το Longest Common Subsequence (LCSS), ή το Dynamic Time Warping (DTW). Τα δύο προηγούμενα αναλύονται και συγκρίνονται από τους Zhang et al. (2006).

Οι υπηρεσίες που προσφέρει αναφέρονται παρακάτω. Μια υπηρεσία είναι το μέτρο ομοιότητας μεταξύ δυο διαδρομών. Μια άλλη είναι ο χώρο-χρονικός δείκτης. Ακόμη μια υπηρεσία είναι τα χρονικά ερωτήματα που μπορούν να γίνουν. Άλλη υπηρεσία είναι τα ερωτήματα για την σημασιολογία της διαδρομής. Μια τελευταία υπηρεσία σε αυτό το επίπεδο είναι τα ερωτήματα για διαδρομές με βάση κάποιο δίκτυο. Αυτή η υπηρεσία χρησιμοποιεί δεδομένα από κάποιον χάρτη.

Τέλος το τρίτο επίπεδο της αρχιτεκτονικής της βιβλιοθήκης απευθύνεται και χρησιμοποιείται κυρίως από προγραμματιστές εφαρμογών και επιχειρηματικούς αναλυτές. Και αυτό επειδή το επίπεδο αυτό περιέχει ένα μεγάλο φάσμα εργασιών εξόρυξης δεδομένων από τα δεδομένα διαδρομών τα οποία μπορούν να τα χρησιμοποιήσουν απευθείας για το έργο τους. Κάποιες κατηγορίες από τις υπηρεσίες που περιέχει είναι η ανακάλυψη συχνών μοτίβων

κίνησης, εξόρυξη συχνών διαδρομών, συσταδοποίηση, αναγνώριση συμβάντος, καθώς και την σημασιολογική εξόρυξη διαδρομής.

Η βιβλιοθήκη αυτή είναι πολύ χρήσιμη και οι στόχοι σε μεγάλη κλίμακα είναι πολύ κοντά σε αυτούς της παρούσας εργασίας. Την βιβλιοθήκη αυτή μπορεί κάποιος να την ενσωματώσει στην εφαρμογή του για να καλύψει έτσι κάποιες ανάγκες που ίσως έχει η εφαρμογή του. Βέβαια θα μπορούσε να αναπτυχθεί και ένα γραφικό περιβάλλον που θα βοηθάει τον χρήστη να χρησιμοποιεί τις υπηρεσίες αυτές.

Κεφάλαιο 3

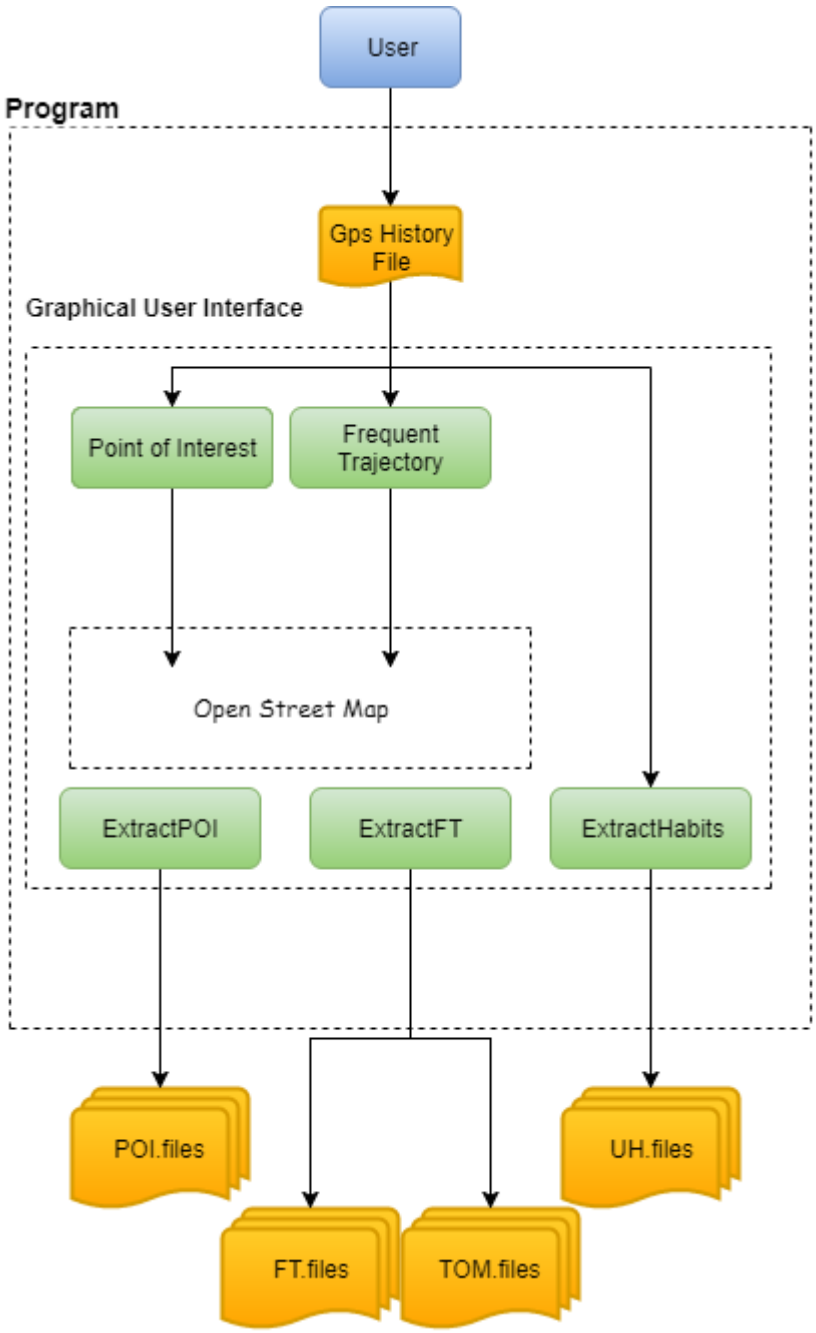
Σχεδιασμός

3.1 Αρχιτεκτονική Εφαρμογής

Η ανάπτυξη της εφαρμογής που προτείνει η παρούσα πτυχιακή εργασία, στηρίχθηκε πάνω σε μια απλή αρχιτεκτονική. Παράλληλα δημιουργήθηκε ένα βασικό γραφικό περιβάλλον για την αλληλεπίδραση μεταξύ χρήστη και προγράμματος. Μέσω αυτού ο χρήστης έχει κάποιες επιλογές, που στοχεύουν την ανάλυση των δεδομένων που έχουν οριστεί σαν είσοδος του προγράμματος. Η είσοδος του προγράμματος είναι ένα αρχείο με το ιστορικό τοποθεσιών σε μορφή JSON, το οποίο έχει τοποθετηθεί σε ξεχωριστό φάκελο (/Res/) πριν την εκτέλεση του προγράμματος.

Οι επιλογές που έχει να κάνει ο χρήστης είναι δυο ειδών. Το ένα είδος του δίνει την ευκαιρία να αναλύει τα δεδομένα με τον τρόπο που θέλει και να τα απεικονίσει πάνω στον χάρτη της OpenStreetMap που υπάρχει στο Γραφικό Περιβάλλον. Αυτό μπορεί να βοηθήσει τον χρήστη να κατανοήσει την ανάλυση καλύτερα από ότι αν το έβλεπε σε αριθμούς. Επίσης σε αυτό το σημείο δίνεται στον χρήστη η ευκαιρία να ρυθμίσει κάποιες μεταβλητές που είναι απαραίτητες για τους αλγορίθμους και έτσι να μπορεί να βγάλει πιο αποδοτικά αποτελέσματα.

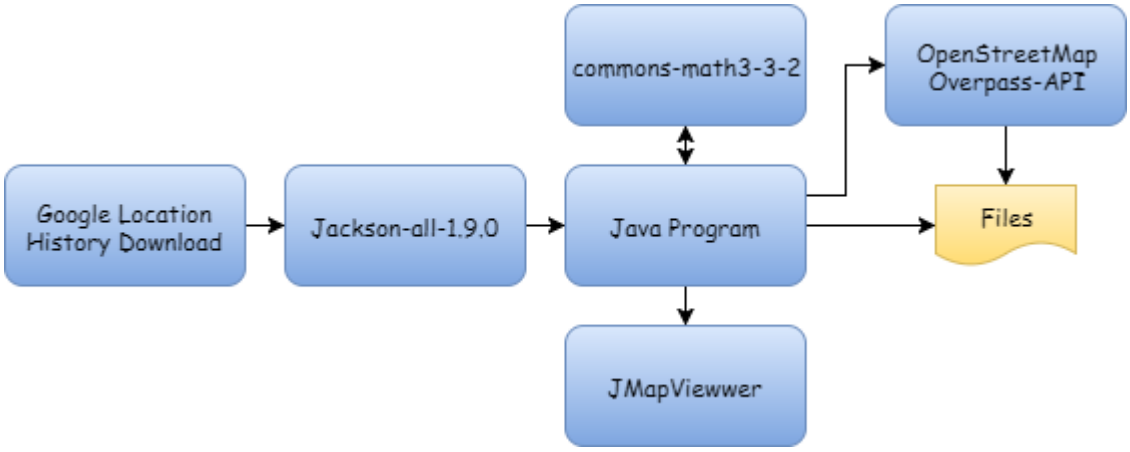
Το άλλο είδος επιλογών είναι να κάνει εξαγωγή των παραπάνω δεδομένων που προέκυψαν από την ανάλυση και να τα γράψει σε ένα αρχείο στον αντίστοιχο φάκελο. Έτσι ο χρήστης θα έχει την δυνατότητα για μελλοντική επεξεργασία των δεδομένων που απέκτησε από την ανάλυση. Στο σχήμα 3.1 απεικονίζεται η βασική αρχιτεκτονική του προγράμματος.



Σχήμα 3.1 Η βασική αρχιτεκτονική του προγράμματος

Αναλυτικότερα η εφαρμογή αποτελείται από το αρχείο του ιστορικού τοποθεσιών το οποίο το έχουμε αποκτήσει από την αντίστοιχη υπηρεσία της Google, από μερικές βοηθητικές βιβλιοθήκες, από το πρόγραμμα που δημιουργήθηκε, από την υπηρεσία της OpenStreetMap,

καθώς και από τα αρχεία που δημιουργηθήκαν από την χρήση του προγράμματος. Η Γενική αρχιτεκτονική της εφαρμογής φαίνεται στο Σχήμα 3.2.



Σχήμα 3.2 Η Γενική Αρχιτεκτονική της Εφαρμογής.

Το αρχείο με το ιστορικό είναι σε μορφή JSON. Για αυτό το λόγο χρησιμοποιείται η βιβλιοθήκη Jackson-all-1.9.0, η οποία βοηθάει να αντιστοιχηθούν τα δεδομένα του αρχείου στις αντίστοιχες κλάσεις που έχουν δημιουργηθεί στο πρόγραμμα, έτσι ώστε να είναι δυνατή η επεξεργασία δεδομένων αργότερα.

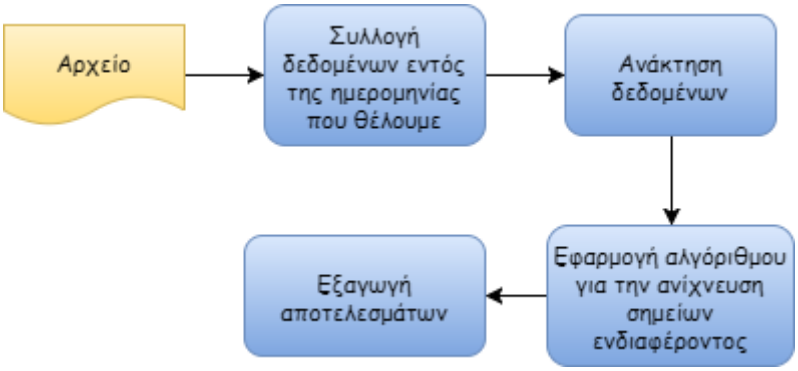
Στην συνέχεια το πρόγραμμα έχει το ήδη επεξεργασμένο αρχείο. Από αυτό βρίσκει τις συχνές τοποθεσίες του χρήστη με την βοήθεια της βοηθητικής βιβλιοθήκης commons-math-3-3-2. Για να απεικονίσει ο χρήστης αυτές τις πληροφορίες που απέκτησε χρησιμοποιείται η βιβλιοθήκη JMapViewer, η οποία παρέχει την επιλογή της ενσωμάτωσης ενός χάρτη της Open Street Map εντός του προγράμματος. Ακόμη ο χρήστης έχει την δυνατότητα αν το θέλει να εξάγει τις συνήθειες του χρήστη για τις μέρες που έχει επιλέξει, με την χρήση του Overpass-API το οποίο επιστρέφει ότι υπάρχει γύρω από μια δοσμένη γεωγραφική περιοχή.

3.2 Μεθοδολογία υλοποίησης εφαρμογής

Η εφαρμογή χωρίζεται σε 4 υποπροβλήματα. Παρακάτω θα περιγράψω με ποιον τρόπο λυθήκανε αυτά τα προβλήματα .

3.2.1 Ανίχνευση συχνών τοποθεσιών του χρήστη

Το πρώτο πρόβλημα είναι η ανίχνευση συχνών τοποθεσιών του χρήστη από το αρχείο δεδομένων. Για να φτάσουμε σε αυτό το σημείο πρώτα έγινε η διαδικασία αναγνώρισης σημείων ενδιαφέροντος του χρήστη, όπως αυτή φαίνεται στο Σχήμα 3.3.



Σχήμα 3.3 Διαδικασία ανίχνευσης Σημείων ενδιαφέροντος

Πιο συγκεκριμένα στα δεδομένα που εισάγονται εφαρμόζεται αρχικά ένας αλγόριθμος συσταδοποίησης και απομάκρυνσης θορύβου. Ουσιαστικά γίνεται αναζήτηση για τα μέρη στα οποία ο χρήστης αφιερώνει πολλές ώρες, δηλαδή τα μέρη στα οποία τα σημεία που καταγράφονται σε περιορισμένα χρονικά διαστήματα είναι πιο πυκνά. Όμως πρακτικά ο χρήστης κινείται στον χώρο και στον χρόνο. Οπότε ακόμη και αν βρίσκεται σε ένα μέρος, μπορεί να κινηθεί με διαφορετικούς τρόπους γύρω από αυτό και προς πολλές κατευθύνσεις.

Στο πλαίσιο αυτό κατανοούμε οτι χρειάζεται ένας αλγόριθμος που να δημιουργεί συστάδες από ένα σύνολο μη γραμμικών σημείων. Ο πιο αποτελεσματικός στο πρόβλημα φάνηκε να είναι ο αλγόριθμος DBSCAN (**Density-based spatial clustering of applications with noise**). Ο συγκεκριμένος αλγόριθμος εντοπίζει συστάδες από μη γραμμικά σημεία και θεωρεί σαν θόρυβο τα σημεία στις περιοχές που δεν είναι μεγάλης πυκνότητας (Kisilevich et al.,2010).

Ο αλγόριθμος DBSCAN δε λαμβάνει υπόψη ότι η γη δεν είναι τελείως σφαιρική και άρα από μόνος του ο αλγόριθμος θα οδηγούσε σε σφάλμα. Επίσης έγινε η παραμετροποίηση του αλγορίθμου ώστε να λαμβάνει ως παράμετρο και το χρονικό διάστημα στο οποίο θα αναζητά μεγάλη πυκνότητα σημείων. Αυτό έγινε, καθώς ο χρήστης μπορεί να επισκεφτεί ένα σημείο σε διαφορετικές χρονικές στιγμές και συνεπώς το να ψάχνουμε για πυκνά στίγματα στο ίδιο σημείο χωρίς να λαμβάνουμε υπόψη μας το χρόνο, θα οδηγούσε σε σφάλματα. Για αυτό χρειαστήκαν μερικές παραμετροποιήσεις και ο αλγόριθμος υλοποιήθηκε με βάση τον αλγόριθμο που παρουσιάζεται σε προηγούμενη διπλωματική εργασία (Τραγοπούλου,2016).

Αφού βρεθούν λοιπόν τα σημεία ενδιαφέροντος του χρήστη υπάρχει η δυνατότητα να βρεθεί η συχνότητα κάθε σημείου ενδιαφέροντος και αυτό με την μεγαλύτερη θα είναι και το μέρος που συχνάζει πιο πολύ ο χρήστης. Ακόμη δίνεται και ποια μέρα προτιμάει ο χρήστης το σημείο αυτό.

3.2.2 Ανίχνευση τύπου κινήσεων του χρήστη

Σε αυτό το πρόβλημα χρησιμοποιήθηκαν κατ' αρχήν το αρχείο του ιστορικού και κατά δεύτερο η υπηρεσία Overpass-API.

Το αρχείο έχει δομή JSON, μέσα σε αυτό υπάρχουν πολλές πληροφορίες για κάθε ίχνος του χρήστη. Μέσα σε αυτές τις πληροφορίες δίνεται και αυτή που λέει με τι τρόπο κινείται ο χρήστης την χρονική στιγμή που πέρασε από το σημείο που καταγράφτηκε, δηλαδή για παράδειγμα αν περπάταγε μια χρονική στιγμή Χ, τότε στο αρχείο JSON θα έχει καταγραφτεί ότι ο χρήστης Α την χρονική στιγμή Χ περπάταγε με βεβαιότητα 90%. Αυτή είναι μια πληροφορία χρήσιμη για τους στόχους της πτυχιακής εργασίας. Για αυτό λοιπόν αυτή η πληροφορία χρησιμοποιήθηκε για να εξαχθούν στατιστικά σχετικά με τον τύπο κίνησης που προτιμά ο χρήστης. Έτσι από ένα σύνολο δεδομένων του ιστορικού γίνεται να φανεί σε τι ποσοστό ο χρήστης χρησιμοποίησε τις παρακάτω κινήσεις:

- Ακίνητος
- Περπάτημα
- Ποδήλατο
- Αυτοκίνητο
- Άλλο τύπο κίνησης.

Πέρα από την παραπάνω γνώση χρησιμοποιείται και μια επιπλέον διαδικασία για να βρεθεί επιπλέον πληροφορία για τις συνήθειες του χρήστη. Στο σημείο αυτό χρησιμοποιείται ένας αλγόριθμος για να βρεθούν σημεία που ο χρήστης στάθηκε σε ένα χώρο, όπως για παράδειγμα ένα στάδιο στο οποίο έτρεχε για να αθληθεί. Ο αλγόριθμος που χρησιμοποιήθηκε είναι ο Stay_points_detection, που προτείνεται από τους Li et al. (2008). Ο αλγόριθμος αυτός επιλέχτηκε για τις ελάχιστες υπολογιστικές απαιτήσεις καθώς και για την απόδοση του. Αφού λοιπόν εντοπιστούν τα σημεία αυτά, χρησιμοποιείται το Overpass-API, το οποίο επιστρέφει σαν απάντηση αν υπάρχει κάποιο μέρος από τα παρακάτω:

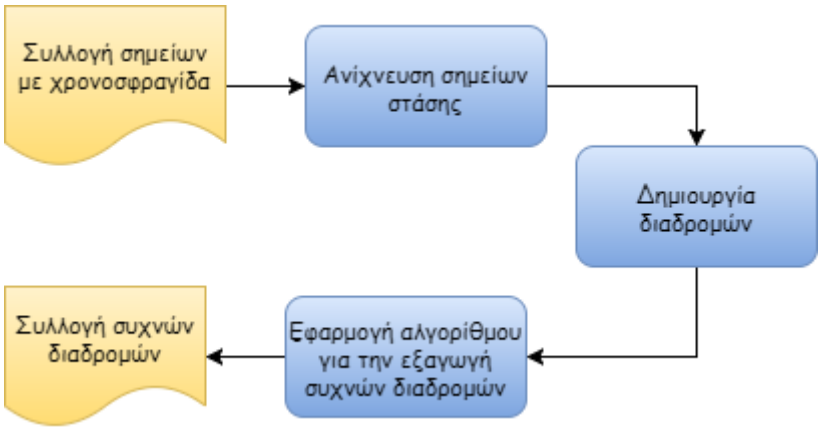
- Πάρκο

- Στάδιο
- Γήπεδο αναψυχής
- Κολυμβητήριο
- Γυμναστήριο
- Αθλητικό κέντρο
- Παραλία
- Άλλο

Στην περίπτωση αυτή προκύπτει σε τι ποσοστό χρησιμοποιεί τα παραπάνω μέρη και έτσι στο τέλος υπάρχει η πληροφορία σχετικά με τον τύπο κινήσεων που προτιμάει ο χρήστης.

3.2.3 Ανίχνευση συχνών διαδρομών του χρήστη

Σε αυτό το σημείο θα γίνει περιγραφή της διαδικασίας εξαγωγής των συχνών διαδρομών. Όπως φαίνεται και στο Σχήμα 3.4 η διαδικασία για την εξαγωγή διαδρομών είναι η παρακάτω. Αρχικά σαν είσοδο το πρόγραμμα έχει την συλλογή από τα σημεία με τις χρονοσφραγίδες τους. Μέσω αυτών γίνεται η ανίχνευση των μερών στα οποία ο χρήστης έκανε στάση ή κάθισε για αρκετή ώρα γύρω από αυτά. Ο τρόπος που πραγματοποιείται αυτό είναι μέσω του αλγόριθμου ανίχνευσης σημείων στάσεων όπως αυτός προτείνεται από τους Li et al.,(2008). Έτσι θεωρείται ότι ανάμεσα από τα διαστήματα μεταξύ των σημείων στάσης υπάρχει κάποια διαδρομή. Στην συνέχεια αυτή η συλλογή διαδρομών δίνεται ως είσοδος στον αλγόριθμο, ο οποίος εντοπίζει και επιστρέφει τις διαδρομές που εμφανίζονται πιο συχνά. Ο αλγόριθμος που χρησιμοποιήθηκε για τον σκοπό αυτό είναι ο clustering based sequential mining (CBM) που προτάθηκε από τους Shaw & Gopalan, (2014).



Σχήμα 3.4 Διαδικασία ανίχνευσης συχνών διαδρομών

3.2.4 Εξαγωγή των συνηθειών του χρήστη

Αφού λοιπόν υλοποιήθηκαν όλα τα παραπάνω υπάρχει η δυνατότητα να βρεθούν πολλές πληροφορίες σχετικά με τις συνήθειες του χρήστη.

Από τα σημεία ενδιαφέροντος του χρήστη μπορεί κανείς να πει οτι αυτό με τη μεγαλύτερη συχνότητα εμφάνισης είναι και το πιο συχνό. Αυτό που γίνεται ουσιαστικά είναι να ανιχνεύεται από μια ταξινομημένη λίστα με συχνές τοποθεσίες, όχι μόνο η συχνότητα εμφάνισης τους αλλά και άλλες πληροφορίες. Όπως για παράδειγμα ποια μέρα προτιμάει ο χρήστης να βρίσκεται στην συγκεκριμένη τοποθεσία. Επίσης για την συγκεκριμένη τοποθεσία με την χρήση του Overpass–API ανακαλύπτεται τι υπάρχει γύρω της σε μια εμβέλεια m μέτρων. Συγκεκριμένα γίνεται αναζήτηση για το αν υπάρχει κάποιο κατάστημα (amenity) όπως καφετέρια, εστιατόριο και άλλα παρόμοια μέρη. Όταν βρεθεί και αυτό γίνεται η εξαγωγή του αρχείου με τις παρακάτω πληροφορίες:

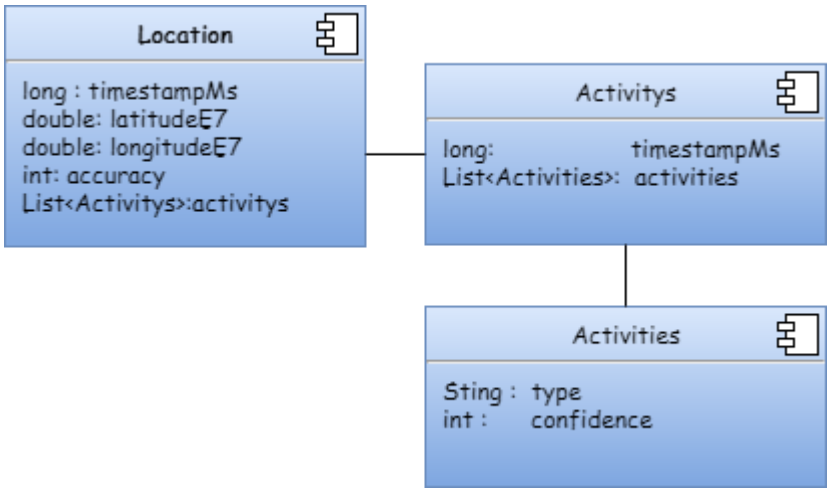
- Γεωγραφικό πλάτος τοποθεσίας
- Γεωγραφικό μήκος τοποθεσίας
- Χρόνος
- Τη πιο συχνή μέρα που γίνεται επίσκεψη της τοποθεσίας
- Την συχνότητα που εμφανίζεται η τοποθεσία
- Σε περίπτωση που υπάρχει κάποιο amenity,το όνομα του.
- Σε περίπτωση που υπάρχει κάποιο amenity, τον τύπο του (καφετέρια, εστιατόριο κτλ)

3.3 Εργαλεία Υλοποίησης της Εφαρμογής

3.3.1 Δομή του αρχείου σε μορφή JSON

Παρακάτω αναλύεται η δομή του αρχείου .json που περιέχει το ιστορικό τοποθεσιών. Η δομή του αποτελείται από το αντικείμενο Location που περιέχει την χρονοσφραγίδα που καταγράφτηκε, το γεωγραφικό μήκος, το γεωγραφικό πλάτος, καθώς και τις ενέργειες του χρήστη Activitys (Τα πιο σύγχρονα αρχεία περιέχουν αυτό το χαρακτηριστικό ως Activitys και κάποια παλιότερα ως Activity), τον χρόνο που καταγράφτηκε αυτή η ενέργεια, τον τύπο της, δηλαδή αν ο χρήστης περπάταγε, ήταν ακίνητος, ήταν σε ποδήλατο και άλλα πολλά. Φυσικά μαζί με τον τύπο υπάρχει ένας ακέραιος αριθμός, με εύρος τιμών από το μηδέν μέχρι το εκατό,

που υποδηλώνει με τι βεβαιότητα έγινε αυτή η πράξη. Στο Σχήμα 3.5 απεικονίζεται η παραπάνω δομή.



Σχήμα 3.5 Δομή του αρχείου ιστορικού δεδομένων

3.3.2 Βιβλιοθήκη Jackson

Η βιβλιοθήκη Jackson είναι μια βιβλιοθήκη για την Java. Σκοπός της είναι η επεξεργασία αρχείων σε μορφή JSON. Στην παρούσα εργασία η έκδοση που χρησιμοποιήθηκε είναι η Jackson-all-1.9.0. Ο λόγος που χρησιμοποιήθηκε είναι για την αντιστοίχιση του αρχείου ιστορικού τοποθεσιών με τις αντίστοιχες κλάσεις του προγράμματος.

3.3.3 Βιβλιοθήκη Commons Math

Το Apache Commons είναι ένα έργο του Apache Software Foundation, που έχει ως σκοπό να παρέχει επαναχρησιμοποιούμενο λογισμικό ανοιχτού κώδικα Java. Συγκεκριμένα η βιβλιοθήκη Commons Math είναι μια βιβλιοθήκη ελαφρών, αυτοτελών μαθηματικών και στατιστικών στοιχείων που καλύπτουν τα πιο συνηθισμένα προβλήματα που δεν είναι διαθέσιμα στη γλώσσα προγραμματισμού Java. Η έκδοση που χρησιμοποιήθηκε στην παρούσα πτυχιακή εργασία είναι η commons-math-3.3.2.

3.3.4 JMapViewer

Το JMapViewer είναι ένα στοιχείο της Java, που σου επιτρέπει να ενσωματώσεις στο πρόγραμμα μια απεικόνιση ενός χάρτη της OpenStreetMap. Σου παρέχει επίσης την δυνατότητα να προσθέσεις στον χάρτη σημάδια. Αυτό μας βοηθάει να απεικονίσουμε τα ίχνη του χρήστη από το ιστορικό.

3.3.5 OpenStreetMap

Το OpenStreetMap (OSM) είναι ένας χάρτης με ελεύθερη άδεια ο οποίος αναπτύσσεται από μια κοινότητα εθελοντών που συνεισφέρουν και διατηρούν δεδομένα σχετικά με δρόμους, μονοπάτια, καφετέριες, σιδηροδρομικούς σταθμούς, και πολλά περισσότερα, σε όλον τον κόσμο. Οι συνεισφέροντες χρησιμοποιούν αεροφωτογραφίες, συσκευές GPS, και τοπικούς χάρτες χαμηλής τεχνολογίας για να σιγουρευτούν πως το OSM είναι ακριβές και ενημερωμένο στο μικρότερο δυνατό επίπεδο. Το OpenStreetMap έχει ελεύθερα δεδομένα που διατίθενται με άδεια Open Data Commons Open Database License(ODbL), τα οποία μπορεί να χρησιμοποιήσει κανείς για οποιονδήποτε σκοπό, εφόσον μνημονευθεί το OpenStreetMap και οι συνεισφέροντες του.

3.3.6 Overpass-API

Η διεπαφή προγραμματισμού εφαρμογών Overpass (ή αλλιώς Overpass- API), είναι ένα API μόνο για ανάγνωση που εξυπηρετεί προσαρμοσμένα επιλεγμένα τμήματα των δεδομένων χαρτών OSM. Λειτουργεί ως βάση δεδομένων μέσω του ιστού, ο πελάτης στέλνει ένα ερώτημα στο API και επαναφέρει το σύνολο δεδομένων που αντιστοιχεί στο ερώτημα. Ανάλογα με το ερώτημα επιστρέφει και τις αντίστοιχες πληροφορίες. Στην εργασία το χρησιμοποιούμε για να δούμε τι υπάρχει γύρω από την ζητούμενη περιοχή που θέλουμε να ερευνήσουμε. Συγκεκριμένα γίνεται ερώτημα για το αν υπάρχει κάποιο κατάστημα γύρω από την ζητούμενη περιοχή.

3.4 Εργαλεία Ανάπτυξης της Εφαρμογής

3.4.1 Netbeans-IDE

Το NetBeans IDE είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης, ανοιχτού κώδικα. Υποστηρίζει την ανάπτυξη όλων των τύπων εφαρμογών Java: Java SE (συμπεριλαμβανομένων των JavaFX), Java ME, web, EJB και εφαρμογές για κινητά. Επίσης το NetBeans περιέχει όλες τις λειτουργικές μονάδες που απαιτούνται για την ανάπτυξη προγραμμάτων Java σε μία και μοναδική λήψη, επιτρέποντας στον χρήστη να αρχίσει να εργάζεται αμέσως. Για την ανάπτυξη της εφαρμογής της εργασίας χρησιμοποιήθηκε αυτό το περιβάλλον.

Κεφάλαιο 4

Υλοποίηση

Στο κεφάλαιο αυτό θα αναλυθεί η υλοποίηση της παρούσας πτυχιακής εργασίας. Το πρώτο πράγμα που θα αναλυθεί είναι η επεξεργασία του αρχείου με το ιστορικό τοποθεσιών, καθώς υπάρχει η ανάγκη για μετατροπή των δεδομένων σε αντικείμενα για να είναι εφικτό αργότερα να γίνει η επεξεργασία τους από το πρόγραμμα. Ύστερα θα αναλυθεί η υλοποίηση της ανίχνευσης συχνών τοποθεσιών, των τύπων κινήσεων που προτιμά ο χρήστης, της ανίχνευσης των συχνών διαδρομών, όπως και η υλοποίηση της διαδικασίας για την εξαγωγή ενός αρχείου για τις συνήθειες του χρήστη. Τέλος θα γίνει η ανάλυση για την υλοποίηση του γραφικού περιβάλλοντος χρήστη.

4.1 Επεξεργασία δεδομένων του ιστορικού τοποθεσιών

4.1.1 Περιγραφή του κώδικα της επεξεργασίας

Στην ενότητα αυτή θα γίνει η περιγραφή για την επεξεργασία των δεδομένων που βρίσκονται στο αρχείο με το ιστορικό τοποθεσιών της Google. Συγκεκριμένα ο στόχος είναι η μετατροπή από τα δεδομένα που βρίσκονται σε μορφή JSON σε αντικείμενα.

Αρχικά δημιουργηθήκανε τα αντικείμενα Location, Activity, Activities (Σχήμα 3.5) για να γίνει η αντιστοίχιση από το αρχείο σε αυτές. Αυτό πραγματοποιήθηκε με την χρήση της βοηθητικής βιβλιοθήκης Jackson και την ενσωμάτωση της κλάσης ObjectMapper.

Η κλάση περιέχει την μέθοδο readValue(File,Class), η οποία αυτό που κάνει είναι η αντιστοίχιση των δεδομένων του File στην Class. Ως είσοδος της μεθόδου αυτής δόθηκε το αρχείο JSON και η κλάση LocationHistory. Η κλάση αυτή περιέχει μια λίστα με Location αντικείμενα. Αφού λοιπόν ολοκληρωθεί η αντιστοίχιση έχει δημιουργηθεί μια λίστα με Location αντικείμενα, δηλαδή τα ίχνη που καταγράφτηκαν από τον χρήστη. Η λίστα αυτή χρησιμοποιείται μέσα στο πρόγραμμα για να την πραγματοποίηση των στόχων της εργασίας.

4.1.2 Ο κώδικας για την μετατροπή JSON σε Αντικείμενο

Στην συνέχεια φαίνεται το κομμάτι του κώδικα που κάνει την αντιστοίχιση. Η μέθοδος getLocations() επιστρέφει όλες τις τοποθεσίες μέσα σε μια λίστα.

```
public  ArrayList<Location> getLocations() throws IOException
{
    ObjectMapper        mapper    = new ObjectMapper();
    ArrayList<Location> locations = new ArrayList<Location>();

    //JSON  to Object
    LocationHistory lh            = mapper.readValue(new File(FILEPATH),
                                    LocationHistory.class);

    for(int i=0  ; i<lh.getLocations().size();i++)
    {
       String jsonInString  =  mapper.writeValueAsString(lh.getLocations().get(i));
       Location location    =  mapper.readValue(jsonInString,Location.class);
       locations.add(location);
    }

    return locations;
}
```

Εικόνα 4.1 Η μέθοδος για την αντιστοίχιση του αρχείου JSON στις κλάσεις

4.2 Υλοποίηση ανίχνευσης συχνών τοποθεσιών

Σε αυτήν την ενότητα θα γίνει η περιγραφή για την ανίχνευση συχνών τοποθεσιών του χρήστη. Για την διαδικασία αυτή χρειάστηκε πρώτα η εύρεση των σημείων ενδιαφέροντος του χρήστη και στην συνέχεια η εύρεσης συχνότητας εμφάνισης των τοποθεσιών. Οι τοποθεσίες στη συνέχεια συλλέγονται σε μια λίστα στην οποία γίνεται ταξινόμηση με βάση τις συχνότητες εμφάνισης. Παρακάτω αναλύεται η διαδικασία ανίχνευσης σημείων ενδιαφέροντος και στη συνέχεια η διαδικασία εύρεσης συχνοτήτων εμφάνισης.

4.2.1 Περιγραφή υλοποίησης ανίχνευσης σημείων ενδιαφέροντος

Για την ανίχνευση των σημείων ενδιαφέροντος υλοποιήθηκε η μέθοδος FindPointsOfInterest(List<Location>,time) η οποία δέχεται σαν είσοδο μια λίστα με τα ίχνη του χρήστη και το χρόνο για τον οποίο επιθυμείται να βρεθούν τα σημεία ενδιαφέροντος. Το εύρος του χρόνου για το οποίο θα γίνει η ανίχνευση είναι σε μέρες.

Ο χρήστης της εφαρμογής επιλέγει μια μέρα που θέλει να γίνει η ανίχνευση και το πρόγραμμα έχει επιλογή να βρίσκει τα σημεία ενδιαφέροντος από την μέρα που διάλεξε ο χρήστη μέχρι N μέρες αργότερα στο ιστορικό. Η τιμή N είναι προκαθορισμένη στον κώδικα να

είναι επτά. Η τιμή αυτή αντιστοιχεί στην μεταβλητή DAYSRANGE που βρίσκεται στην κλάση GoogleLocations.

Για παράδειγμα αν ο χρήστης επιλέξει στον Γραφικό περιβάλλον,στο ημερολόγιο να του εμφανίσει τα σημεία ενδιαφέροντος της 1-1-2017, η ανίχνευση θα γίνει για τις μέρες του ιστορικού που ανήκουν στο σύνολο (1-1-2017 μέχρι 8-1-2017). Αυτό γίνεται επειδή το ιστορικό περιέχει πολλά δεδομένα και θα χρειαζόταν τεράστια υπολογιστική ισχύς. Έτσι δημιουργείται η ανάγκη του περιορισμού του πλήθους των δεδομένων για ένα πιο λειτουργικό σύστημα.

Αυτό γίνεται μέσω της μεθόδου getPoints(List<locations>,time). Ως είσοδος της μεθόδους αυτής είναι όλο το ιστορικό με τα ίχνη του χρήστη και ως έξοδος είναι η λίστα με τα ίχνη που είναι εντός του ζητούμενου συνόλου. Η μεταβλητή time είναι ο χρόνος σε millisecond.

Ύστερα γίνεται χρήση της βοηθητικής βιβλιοθήκης commons-math για να χρησιμοποιηθεί ο αλγόριθμος DBSCAN, ο οποίος είναι ένας αλγόριθμος ομαδοποίησης δεδομένων. Ο αλγόριθμος έχει παραμετροποιηθεί όπως προτείνεται από την διπλωματική εργασία της Τραγοπούλου (2016). Έτσι δίνονται στον αλγόριθμο σαν είσοδο τα ζητούμενα δεδομένα και επιστρέφονται οι συστάδες. Ουσιαστικά επιστρέφει τα σημεία ενδιαφέροντος,τα οποία προσθέτονται μέσα σε μια λίστα, η οποία προκύπτει από την μέθοδο FindPointsOfInterest().

4.2.2 Ο Κώδικας για την ανίχνευση σημείων ενδιαφέροντος

Παρακάτω δίνεται ο κώδικας υλοποίησης της μεθόδου που σκοπός της είναι η εύρεση σημείων ενδιαφέροντος.

```
//=============================================================================================
// Method: Find points of Interest
//=============================================================================================
public  ArrayList<DoublePoint> FindPointsOfInterest(ArrayList<Location> locations,long time)
{
    ArrayList<DoublePoint> poi = new ArrayList<DoublePoint>();
    DBSCANClusterer dbscan ;

    dbscan = new DBSCANClusterer(0.1, 10, (DistanceMeasure) new POIDistanceMeasure());

    List<DoublePoint>           points   =  getPoints(locations,time); //Return points we want
    List<Cluster<DoublePoint>>  clusters =  dbscan.cluster(points);    //Use DBSCAN

    for(Cluster<DoublePoint> c: clusters)
    {
        for(int i=0 ; i<c.getPoints().size();i++)
        {
            poi.add(c.getPoints().get(i));//Collect points of interest
        }
    }
    return poi;
}
```

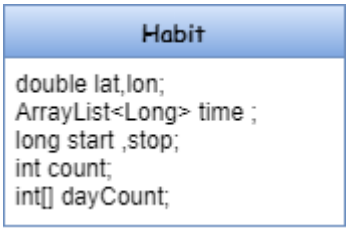
Εικόνα 4.2 Η μέθοδος που επιστρέφει την λίστα με τα σημεία ενδιαφέροντος

Με βάση τη λίστα αυτή γίνεται η εύρεση συχνών τοποθεσιών.

4.2.3 Περιγραφή υλοποίησης ανίχνευσης συχνών τοποθεσιών

Αφού λοιπόν υπάρχει η λίστα με τα σημεία ενδιαφέροντος του χρήστη υπάρχει και η δυνατότητα εξαγωγής των πιο συχνών. Αυτό γίνεται με επεξεργασία της λίστας αυτής.

Πρακτικά εντοπίζονται στη λίστα όλες οι φορές που εμφανίζεται κάθε σημείο. Για την ανάγκη τον συχνών τοποθεσιών δημιουργήθηκε η κλάση Habit. Κάθε αντικείμενο Habit έχει τα παρακάτω χαρακτηριστικά.



Εικόνα 4.3 Χαρακτηριστικά κλάσης Habit

Αναλυτικά κάθε αντικείμενο Habit αποτελείται από τις γεωγραφικές συντεταγμένες (latitude,longitude), από μια λίστα με τις χρονικές στιγμές που εμφανίστηκε αυτή η τοποθεσία μέσα στην λίστα των σημείων ενδιαφέροντος, από δυο χρόνους start και stop που είναι πότε ξεκίνησε και πότε σταμάτησε η εμφάνιση αυτής της τοποθεσίας, από την μεταβλητή count η οποία μετράει την συχνότητα εμφάνισης της τοποθεσίας, και τον πίνακα dayCount που αποτελείται από επτά θέσεις και μετράει την συχνότητα εμφάνισης της κάθε μέρας. Δηλαδή το στοιχείο dayCount[0] είναι πόσες φορές εμφανίστηκε η Κυριακή, το dayCount[1] πόσες φορές εμφανίστηκε η Δευτέρα κτλ.

Η υλοποίηση για την εύρεση συχνών τοποθεσιών πραγματοποιείται στην μέθοδο getHabits(List<DoublePoint> points), η οποία βρίσκει της συνήθειες του χρήστη. Συγκεκριμένα δέχεται σαν είσοδο μια λίστα με τα σημεία ενδιαφέροντος και την επεξεργάζεται για να επιστρέψει μια λίστα με αντικείμενα Habit αφού την έχει ταξινομήσει με βάση την συχνότητα εμφάνισης κάθε σημείου.

Για να γίνει αυτό πρώτα γίνεται η επεξεργασία της λίστας με όλα τα σημεία ενδιαφέροντος. Η επεξεργασία ακολουθεί την παρακάτω διαδικασία.

Αρχικά για κάθε σημείο της λίστας γίνεται έλεγχος αν υπάρχει ήδη κάποιο αντικείμενο Habit που να αντιστοιχεί στο σημείο αυτό. Αν δεν υπάρχει τότε δημιουργεί ένα αντικείμενο Habit με συντεταγμένες ίδιες με του σημείου και προσθέτει το αντικείμενο αυτό στην λίστα exist που περιέχει τα αντικείμενα Habit που έχουν δημιουργηθεί. Στη συνέχεια για το αντικείμενο Habit βρίσκει πόσες φορές εμφανίζονται σημεία με τις ίδιες συντεταγμένες εντός της λίστας των σημείων ενδιαφέροντος. Ακόμη βρίσκει την συχνότητα που εμφανίζεται κάθε μέρα της εβδομάδας. Επίσης κάθε αντικείμενο Habit έχει μια λίστα με χρόνους. Αυτή η λίστα περιέχει κάθε χρονική στιγμή που εμφανίστηκε το σημείο. Τέλος γίνεται η ταξινόμηση της λίστας με τα αντικείμενα Habit.

4.2.4 Ο Κώδικας για την ανίχνευση συχνών τοποθεσιών

Παρακάτω δίνεται το μέρος του κώδικα που επιστρέφει την λίστα με τα αντικείμενα Habit.

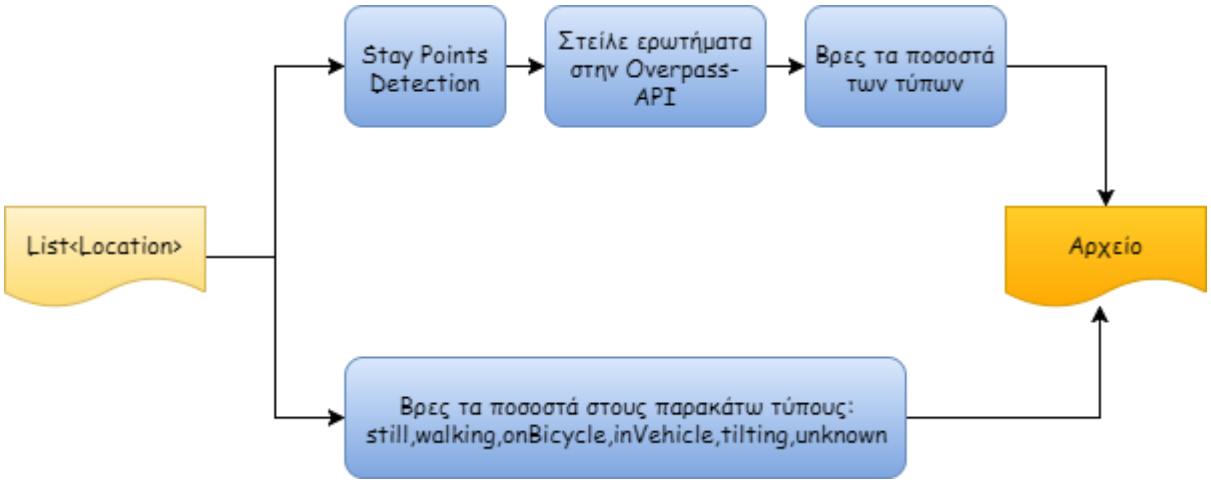
```
//================================================================================================
// Method : Return user habits
//================================================================================================
public ArrayList<Habit> getHabits(ArrayList<DoublePoint> p){
    ArrayList<Habit>      habits = new ArrayList<>();   //List with habits
    ArrayList<DoublePoint> exist = new ArrayList<>();   //Auxiliary list
    Date d ;
    Habit habit;                                        //Habit Object
    boolean pointExist = false;

    for(int i=0;i<p.size();i++){
        pointExist = CheckIfPointExist(p.get(i),exist); //Method:Return true if point already exist

        //If point does not exist create new Habit and find frequence of this habit
        if(!pointExist){
            habit = new Habit(p.get(i).getPoint()[1]/10000000,p.get(i).getPoint()[0]/10000000);//create new Habit
            DoublePoint newPoint = new DoublePoint(p.get(i).getPoint());
            exist.add(newPoint);
            //Find frequence of newPoint
            for(int x=0;x<p.size();x++){
                if(p.get(x).getPoint()[1]/10000000==habit.getLat()
                   &&
                   p.get(x).getPoint()[0]/10000000==habit.getLon())
                {
                    d = new Date((long)p.get(x).getPoint()[2]);
                    habit.AddDayCount(d.getDay());                //Find for each day frequence
                    habit.setCount(habit.getCount()+1);           //Find frequence of this point
                    habit.addTime((long)p.get(x).getPoint()[2]);  //Add new time in the time List
                }
            }
            habits.add(habit);    // Add new habit
        }
    }
    habits=sortHabits(habits);    //Sort Habits list
    return habits;
}
```

Εικόνα 4.4 Η μέθοδος που επιστρέφει την λίστα με τις συνήθειες του χρήστη

Ο λόγος που γίνεται διαίρεση τα lat, lon του σημείου με το 10000000 είναι ότι πρέπει η υποδιαστολή να βρίσκεται στο δεύτερο ψηφίο του αριθμού, διαφορετικά οι υπολογισμοί θα οδηγήσουν σε λάθος συμπεράσματα.

4.3 Υλοποίηση ανίχνευση τύπων κινήσεων

Για την ανίχνευση τύπων κινήσεων χρειάστηκαν δυο διαδικασίες όπως φαίνεται στο Σχήμα 4.2. Η μια είναι για την ανίχνευση κινήσεων του χρήστη με βάση την λίστα με τις τοποθεσίες που δημιουργήθηκε από το αρχείο με το ιστορικό τοποθεσιών. Η άλλη είναι με την χρήση της υπηρεσίας Overpass-API της OpenStreetMap.



Σχήμα 4.2 Διαδικασία ανίχνευσης τύπου κινήσεων

Σχετικά με την πρώτη διαδικασία έχει δημιουργηθεί μια μέθοδος που δέχεται μια λίστα με αντικείμενα Location. Μέσα υπάρχει η δυνατότητα να βρεθεί αν στο συγκεκριμένο αντικείμενο υπάρχουν Activities και αν ναι τι τύπου είναι. Στην συνέχεια υπολογίζονται τα ποσοστά για κάθε τύπο, και έτσι στο τέλος η μέθοδος επιστρέφει μια μεταβλητή τύπου String με τα ποσοστά αυτά. Η μέθοδος αυτή χρησιμοποιείται για την εξαγωγή του αρχείου.

Όσον αφορά την δεύτερη διαδικασία. αυτό που γίνεται είναι να βρίσκεται από το σύνολο των σημείων που υπάρχει κάποιες τοποθεσίες που ο χρήστης φάνηκε να κάνει στάση. Στην συνέχεια στέλνονται ερωτήματα στη Overpass-API αν υπάρχει κάτι γύρω από το σημείο αυτό. Όπως για παράδειγμα κάποιο πάρκο, ή κάποια πισίνα.

Ο τρόπος που γίνεται ο εντοπισμός για τα σημεία που έκανε στάση ο χρήστης είναι με βάση τον αλγόριθμο που προτείνεται από τους Li et al. (2008). Όταν βρεθούν και τα τελευταία ποσοστά τότε όλα αυτά τα δεδομένα που συλλέχθηκαν αποθηκεύονται σε ένα αρχείο.

Παρακάτω φαίνεται ο αλγόριθμος StayPoint_Detection για τα σημεία.

```
// Algorithm StayPoint_Detection(P,distThreh,timeThreh)

Input : A GPS log P, a distance threshold distThreh and time span threshold timeThreh
Output: A set of stay points SP={S}

1. i=0, pointNum = |P|;                //The number of GPS points
2. while i < pointNum do,
3.      j:=i+1; Token:=0;
4.      while j < pointNum do,
5.          dist:=Distance(pi, pj);     //Calculate the distance between points
6.          if dist > distThreh then
7.              ΔT:=pj.T-pi.T;          //Calculate the time span between two points
8.              if ΔT>timeThreh then
9.                  S.coord:=ComputMeanCoord({pk | i<=k<=j})
10                  S.arvT:= pi.T; S.levT=pj.T ;
11.                 SP.insert(S);
12.                 i:=j; Token:=1;
13.             break;
14.         j:=j+1;
15.     if Token!=1 then i:=i+1;
16. return SP.
```

Εικόνα 4.5 Ο ψευδοκώδικας για την εύρεση σημείων στάσης του χρήστη.

Ο αλγόριθμος δέχεται ως είσοδο τρεις παραμέτρους. Η πρώτη είναι η συλλογή P που περιέχει την λίστα με τα σημεία. Η δεύτερη είναι η distance Threshold. Η τρίτη είναι η time threshold. Οι προκαθορισμένες τιμές για τις δυο παραμέτρους είναι distThreh = 200m και για το χρονικό όριο είναι 30 λεπτά. Αυτό που κάνει ο αλγόριθμος είναι να βρίσκει αν ο χρήστης ήταν γύρω από ένα σημείο σε εμβέλεια 200m για πάνω από 30 λεπτά. Αν είναι αληθής αυτό τότε ορίζεται το σημείο S. Το S έχει συντεταγμένες που προκύπτουν από την μέθοδο ComputMeanCoord(), S.arvT τη χρονική στιγμή που ο χρήστης έφτασε στο σημείο αυτό και S.levT τη χρονική στιγμή που ο χρήστης έφυγε από αυτό το σημείο. Τέλος προσθέτει το σημείο S στο σύνολο SP={} το οποίο στο τέλος του αλγορίθμου περιέχει όσα σημεία στάσης βρεθήκανε.

Παρακάτω φαίνεται η υλοποίηση του παραπάνω αλγορίθμου σε γλώσσα Java,όπως αυτός υλοποιήθηκε για τις ανάγκες τις παρούσας εργασίας.

```
public ArrayList<StayPoint> FindStayPoints(ArrayList<TrajectoryPoint> P)
 {
        //Variables
        int pointNum = P.size();
        StayPoint  S = new StayPoint();
        int        i = 0;
        int        j = 0;
        int    token = 0;

        ArrayList<StayPoint> SP = new ArrayList<StayPoint>();

        while(i<pointNum-1){
            j=i+1;
            token=0;
            while(j<pointNum){
                dist = Distance(P.get(i),P.get(j));
                if(dist>distThreh){
                    timeDelta = (long)Math.abs(P.get(j).getTimestampMs()-(long)P.get(i).getTimestampMs());
                    if(timeDelta>timeThreh){
                        pk = new ArrayList<TrajectoryPoint>();
                        for(int k=i;k<=j;k++){
                            pk.add(P.get(k));
                        }
                        S = computMeanCoord(pk);
                        S.setArrivTime(P.get(j).getTimestampMs());
                        S.setLeaveTime(P.get(i).getTimestampMs());
                        SP.add(S);
                    }
                    i=j;
                    token=1;
                    break;
                }else{
                  j=j+1;
                }

            }
            if(token!=1)
                i+=1;

        }
        return SP;
}
```

Εικόνα 4.6 Υλοποίηση του αλγόριθμου για τα σημεία στάσης.

Αφού λοιπόν έχουμε τα σημεία αυτά για κάθε ένα από αυτά στέλνεται ένα ερώτημα στην OverPass-API. Το ερώτημα που γίνεται έχει να κάνει με το αν υπάρχει εντός 100 μέτρων γύρω από αυτό το σημείο κάποιο μέρος όπως πάρκο, πισίνα, γυμναστήριο, παραλία, κτλ. Στην συνέχεια υπολογίζεται σε τι ποσοστό βρέθηκε κάτι από όλα αυτά. Όταν τελειώσει και αυτό τα αποτελέσματα αποθηκεύονται σε ένα αρχείο.

4.4 Υλοποίηση ανίχνευσης συχνών διαδρομών

Σε αυτήν την ενότητα περιγράφεται η υλοποίηση της διαδικασίας εύρεσης συχνών διαδρομών. Όπως φαίνεται και στο Σχήμα 3.4 η ροή της διαδικασίας περιλαμβάνει τα παρακάτω βήματα. Πρώτα από το σύνολο των σημείων που υπάρχουν εντοπίζεται αν υπάρχουν στάσεις που έκανε ο χρήστης. Από αυτές τις στάσεις δημιουργούνται οι διαδρομές. Οι διαδρομές που

δημιουργηθήκαν τοποθετούνται σαν είσοδος στον αλγόριθμο ανίχνευσης συχνών διαδρομών. Παρακάτω θα αναλυθούν περισσότερο τα βήματα αυτά.

4.4.1 Δημιουργία διαδρομών

Το πρώτο πράγμα που έγινε είναι η εύρεση σημείων που ο χρήστης έκανε κάποια στάση και με βάση αυτά τα σημεία δημιουργήθηκαν οι διαδρομές. Για παράδειγμα αν από ένα σύνολο Ν σημείων εντοπιστούν δυο μέρη που έκανε στάση ο χρήστης και αυτό είναι τα S1 και S2,τότε θα δημιουργηθεί μια διαδρομή που θα αποτελείται από τα σημεία που έχουν χρονοσφραγίδα μεταξύ του S1.levT και S2.arvT. Ουσιαστικά θεωρείται ότι ο χρήστης έκανε μια διαδρομή μεταξύ των δυο σημείων στάσης.

4.4.2 Περιγραφή αλγορίθμου για την Ανίχνευση συχνών διαδρομών

Σε αυτό το σημείο υπάρχει μια συλλογή από διαδρομές. Ο αλγόριθμος που υλοποιήθηκε για την ανάγκη αυτή είναι αυτός που προτείνεται από τους Shaw και Gopalan (2014). Ο αλγόριθμος αυτός είναι ο «clustering based sequential mining». Ο σκοπός του είναι μέσα από μια συλλογή με διαδρομές να εξάγει τις πιο συχνές. Για να το κάνει αυτό χρησιμοποιεί μια μέθοδο συσταδοποίησης σημείων.

Αναλυτικότερα σαν είσοδο δέχεται ένα σύνολο από σημεία διαδρομών, μια μεταβλητή s, και μια μεταβλητή ξ. Η μεταβλητή s ορίζει το εμβαδόν τετραγώνου που θα καταλαμβάνει στο χάρτη κάθε συστάδα. Για παράδειγμα αν το s είναι 10000 αυτό σημαίνει ότι η συστάδα θα είναι 100x100 μέτρα. Η μεταβλητή ξ είναι το ελάχιστο πλήθος σημείων που πρέπει να έχει μια συστάδα για να θεωρηθεί ενεργή (δηλαδή συχνά επισκεπτόμενη από τις διαδρομές του χρήστη).

Μετά την είσοδο το πρώτο βήμα είναι να βρεθεί η ελάχιστη και η μέγιστη περιοχή m x n όπου οι διαδρομές καλύπτουν τον άξονα x και y. Αφού βρεθεί η περιοχή το επόμενο βήμα είναι να υπολογιστεί το k, το οποίο συμβολίζει το πλήθος των συστάδων (υποπεριοχών) στις οποίες υποδιαιρείται η περιοχή. Το k υπολογίζεται από τον παρακάτω τύπο:

$$k = \frac{m*n}{s}$$

Μόλις υπολογιστεί το k δημιουργούνται μέσα στην περιοχή m x n , k τετραγωνικές συστάδες με εμβαδόν ίσο με το s. Στην συνέχεια γίνεται αντιστοίχιση κάθε συντεταγμένης που υπάρχει εντός της περιοχής στην συστάδα που έχει το κέντρο της πιο κοντά σε αυτή. Ύστερα υπολογίζεται το νέο centroid που βγαίνει από την μέση τιμή των σημείων που ανήκουν στην συστάδα. Μετά γίνεται ο υπολογισμός του αριθμού E. Πρακτικά υπολογίζεται για κάθε συστάδα η απόσταση του παλιού centroid με το καινούργιο. Γίνεται ο υπολογισμός αυτός για όλες τις συστάδες και στην συνέχεια υπολογίζεται η μέση τιμή. Η τιμή που θα προκύψει είναι το E. Αφού λοιπόν έχει υπολογιστεί το E τότε γίνεται έλεγχος στην τιμή αυτή. Αν η τιμή είναι πολύ μεγάλη τότε μειώνουμε το k σε k- k/4 και επαναλαμβάνουμε την διαδικασία αυτή. Με την τελευταία ενέργεια επιτυγχάνουμε να διαιρέσουμε περιοχές τις οποίες διατρέχει συχνά ο χρήστης σε μικρότερες περιοχές και έτσι να βελτιώσουμε την ακρίβεια των τροχιών μας.

Παρακάτω φαίνεται το μέρος των βημάτων του αλγορίθμου που περιγράφει την διαδικασία αυτή.

```
3.2. CBM algorithm

Input: A trajectory coordinates database D, cluster size as s and a minimum support
be ξ.
Output: Set of coordinates where the FT of the moving objects pass.

The method is described as follows.
Step 1:  Start;/* Creation of clusters.
                 Each cluster has some properties and
                 their respective initial values set
                 like count = 0, previous and next cluster links,
                 active as false and processed as false. */
Step 2:  Declare an array A [ ]; // To store active set of clusters.
                                 // Scan the trajectory dataset D and
                                 // increment the corresponding cluster.
Step 3:  Read a trajectory dataset D;
Step 4:  Find the minimum and maximum area where the trajectories are spanning across
x and y axis;
// To find number of square clusters, of size k × k in the trajectory span area m ×
n.
Step 5:  k = m × n/s;
Step 6:  Create k uniform square clusters of size s with indices Ci,j;
Step 7:  Repeat;

Step 8:  Assign or reassign each coordinate of the cluster
         to which the coordinate is the most similar,
         based on the mean value of the coordinates in the cluster;

Step 9:  Update the cluster means mi,j
         by calculating the mean value of
         the objects for each cluster Ci,j;
Step 10: Calculate E;
Step 11: Reduce cluster size k = k – k/4;
Step 12: Until square error, E converges;
```

Εικόνα 4.7 Τα βήματα 1-12 του αλγόριθμου CBM

Όταν το Ε γίνει πολύ μικρό τότε συνεχίζει ο αλγόριθμος και βρίσκει τις συστάδες που είναι ενεργές, τις οποίες τις εκχωρεί σε μια λίστα Α. Για να είναι μια συστάδα ενεργή πρέπει το πλήθος των σημείων που ανήκουν σε αυτήν να είναι μεγαλύτερο του ξ. Αφού βρεθούν οι ενεργές συστάδες τότε γίνεται ταξινόμηση της λίστας. Μόλις γίνει η ταξινόμηση με βάση την μεταβλητή count τότε το πρώτο στοιχείο του πίνακα γίνεται HeaderCluster και CurrentCluster.

Τα βήματα του αλγορίθμου που περιγράφτηκαν φαίνονται παρακάτω.

```
3.2. CBM algorithm
// Select the active set of clusters.
Step 13: Set c as 0;
Step 14: For (i = mmin; i ≤ mmax; i + k);
Step 15: For (j = nmin; j ≤ nmax; j + k);
Step 16: If Ci,j count ≥ ξ then;
Step 17: Set Ci,j active as true;
Step 18: A[c] = Ci,j count;
Step 19: c + +;
Step 20: End if;
Step 21: Next j;
Step 22: Next i;
// Sort the active clusters in array A.
Step 23: For (i = 0; i < c – 1; i + +);
Step 24: For (j = i + 1; j < c; j + +);
Step 25: If A[i] < A[j], then;
Step 26: A[i] = A[j];
Step 27: End if:
Step 28: Next j;
Step 29: Next i;
Step 30: Header cluster A[0];
Step 31: Current cluster A[0];
```

Εικόνα 4.8 Τα βήματα 13-31 του αλγόριθμου CBM

Στην συνέχεια γίνεται επανάληψη μέχρι όλες οι ενεργές συστάδες υποστούν την ίδια επεξεργασία. Αυτό που γίνεται είναι για την CurrentCluster βρίσκει τις γειτονικές της συστάδες. Από αυτές ψάχνει αυτές που δεν έχουν επεξεργαστεί και αναζητεί αν υπάρχει κάποια που έχει count μεγαλύτερο ή ίσο από της CurrentCluster. Αν βρεθούν παραπάνω από μια, διαλέγει μια αυθαίρετα και με τις υπόλοιπες συνεχίζει αργότερα. Στην συνέχεια ενώνει τις δυο αυτές συστάδες και ορίζει ως CurrentCluster την συστάδα που βρέθηκε. Για την προηγούμενη συστάδα που ήταν CurrentCluster απλά δηλώνει ότι την επεξεργάστηκε. Ύστερα ελέγχει αν υπάρχει κάποια ενεργή συστάδα που δεν επεξεργάστηκε. Αν υπάρχει τότε εκτελεί τα βήματα 32-48. Τέλος συλλέγει όλες αυτές τις συστάδες που έχουν ενωθεί. Αυτές οι ενώσεις αποτελούν

και τις συχνές διαδρομές. Παρακάτω φαίνονται τα βήματα του αλγορίθμου που περιγράφουν αυτή την διαδικάσία.

```
// Loop to link the active set of unlinked clusters from the sorted array A.

Step 32: Repeat;

Step 33: From current cluster find the unprocessed and active clusters

         surrounding the nearest eight neighborhood clusters
          having count ≥ current   cluster's count;
Step 34: If found then;
Step 35: If more than one clusters having same count found then;
Step 36: Choose any cluster arbitrarily adjacent to current cluster;
Step 37: End if;
Step 38: Set current cluster to currently found cluster;
Step 39: Link the current cluster with previous current cluster;
Step 40: set current cluster, processed as true;
Step 41: End if;
Step 42: Check any unprocessed clusters exists in array A[ ];
Step 43: If found then;
Step 44: Repeat Steps 32-48;
Step 45: Else;
Step 46: Go to Step 49;
Step 47: End if;
Step 48: Until all unprocessed, active set of clusters are processed;
Step 49: From each header cluster to current cluster collect all the linked clusters
in sequence;
Step 50: End of algorithm.
```

Εικόνα 4.9 Τα βήματα 32-50 του αλγορίθμου CBM

4.5 Εξαγωγή αρχείου με τις συνήθειες του χρήστη

Στην ενότητα αυτή θα αναλυθεί ο τρόπος με τον οποίο έγινε η εξαγωγή του αρχείου με τις συνήθειες του χρήστη.

Αρχικά δημιουργήθηκε για τον σκοπό αυτό η αντίστοιχη μέθοδος extractHabits(List<DoublePoint> p). Πρώτα από όλα στην μέθοδο χρησιμοποιείται ως είσοδος μια λίστα με τα σημεία ενδιαφέροντος, μέσα από εντοπίζονται οι συχνές τοποθεσίες του χρήστη, και στην συνέχεια για κάθε μια από αυτές γίνεται χρήση της υπηρεσίας Overpass-API για να γίνει αναζήτηση σχετικά με το τι υπάρχει γύρω από αυτή την τοποθεσία. Στο πρόγραμμα αυτή την δουλειά την κάνουν οι κλάσεις OverpassAPI και η API. Αφού γίνει ένα ερώτημα στην Overpass-API επιστρέφεται η απάντηση σε μορφή JSON. Με παρόμοια διαδικασία που

περιγράφτηκε στην ενότητα 4.1.2 έγινε αντιστοίχιση της απάντησης που είναι σε μορφή JSON στις αντίστοιχες κλάσεις API, Osm3s, Element, και Tags.

Έτσι ένα αντικείμενο API έχει όλες τις πληροφορίες που αποκτήθηκαν για το συγκεκριμένο σημείο που έγινε το ερώτημα. Μέσω αυτού του αντικειμένου λοιπόν βρίσκεται τι amenity υπάρχει σε αυτό το σημείο. Οπότε υπάρχει μια συλλογή με αρκετές πληροφορίες για κάθε συχνή τοποθεσία. Σαν τελευταίο βήμα όλες αυτές οι πληροφορίες αποθηκεύονται σε ένα αρχείο. Το αρχείο αυτό περιέχει τις συχνές τοποθεσίες με φθίνουσα σειρά. Η πρώτη τοποθεσία είναι και η πιο συχνή. Μαζί με τις τοποθεσίες αναγράφονται και πληροφορίες για αυτές όπως η συχνότητα εμφάνισης της τοποθεσίας και την συχνότερη μέρα που προτιμάει ο χρήστης να πηγαίνει στην συγκεκριμένη τοποθεσία.

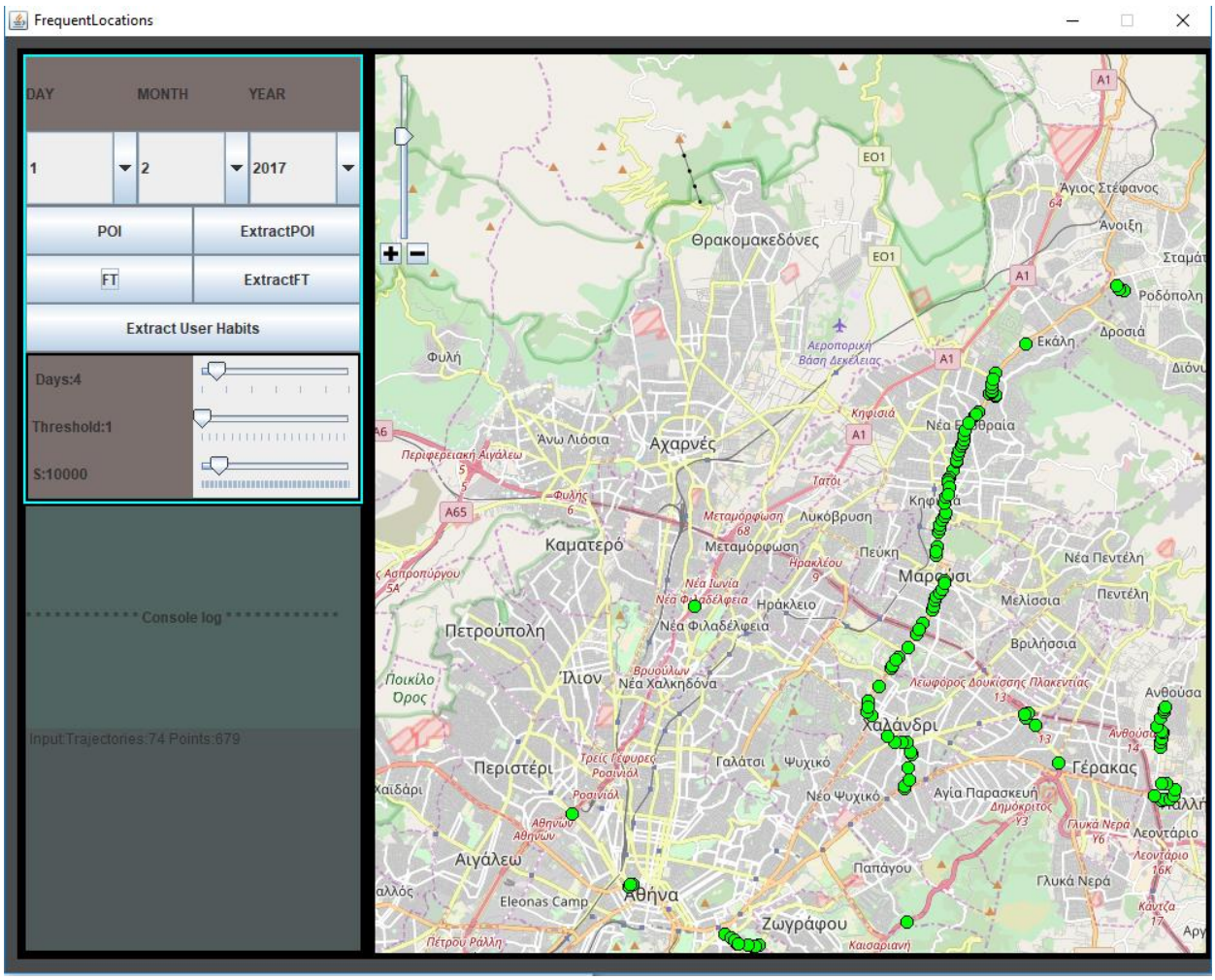
4.6 Δημιουργία Γραφικού Περιβάλλοντος Χρήστη

Αφού έγινε η περιγραφή για την υλοποίηση όλων των παραπάνω προβλημάτων, σε αυτήν την ενότητα θα γίνει η περιγραφή της ανάπτυξης του γραφικού περιβάλλοντος χρήστη της συγκεκριμένης εφαρμογής.

Το γραφικό περιβάλλον αυτό υλοποιήθηκε με την βοήθεια της βιβλιοθήκης swing που περιέχεται στο πακέτο javax. Αυτή περιέχει εξαρτήματα όπως πλαίσια, κουμπιά και άλλα που είναι χρήσιμα για την ανάπτυξη ενός γραφικού περιβάλλοντος.

Συγκεκριμένα για την εφαρμογή αυτή φτιάχτηκε ένα πλαίσιο μέσα στο οποίο περιέχονται κουμπιά που είναι χρήσιμα για την αλληλεπίδραση του χρήστη με τους αλγόριθμους, ένας χάρτης για την απεικόνιση των αποτελεσμάτων, μερικά JSlider εξαρτήματα που είναι χρήσιμα για μερικές ρυθμίσεις σχετικά με την εφαρμογή, ένα μικρό πλαίσιο που εμφανίζει κάποια κείμενα σχετικά με το πρόγραμμα, και ένα ημερολόγιο βασικής χρήσης που φτιάχτηκε για τις ανάγκες τις εφαρμογής. Η κλάση που περιέχει όλο τον κώδικα που είναι για την δημιουργία του γραφικού είναι η GUI.

Το γραφικό περιβάλλον που περιγράφτηκε φαίνεται στην εικόνα 4.10.



Εικόνα 4.10 Απεικόνιση του γραφικού περιβάλλοντος χρήστη

Όπως φαίνεται και στην εικόνα 4.10 ο χρήστης της εφαρμογής έχει πολλαπλές επιλογές να κάνει. Παρακάτω θα περιγραφτεί η χρήση της εφαρμογής αυτής. Αρχικά ο χρήστης επιλέγει μια μέρα για την οποία θέλει να επεξεργαστεί τα δεδομένα. Αφού το κάνει έχει της επιλογές POI, FT οι οποίες είναι για την ανίχνευση σημείων ενδιαφέροντος και την ανίχνευση συχνών διαδρομών. Αφού επιλέξει μια από τις δυο επιλογές στην συνέχεια εμφανίζονται στον χάρτη της OpenStreetMap τα δεδομένα από την εφαρμογή των αντίστοιχων αλγόριθμων. Η ανίχνευση τύπων κινήσεων γίνεται αυτόματα μόλις πατηθεί το κουμπί FT. Πριν πατήσει το κουμπί FT ο χρήστης πρέπει να είναι σίγουρος και για τις ρυθμίσεις. Οι ρυθμίσεις είναι για τον αλγόριθμο της ανίχνευσης συχνών διαδρομών.

Η πρώτη ρύθμιση είναι για τον αριθμό των ημερών που θέλει ο χρήστης να εφαρμόσει τον αλγόριθμο. Συγκεκριμένα αν επιλέξει 4 μέρες και έχει επιλέξει στο ημερολόγιο 1-2-2017 τότε ο αλγόριθμός σαν είσοδο θα δεχτεί όλα τα δεδομένα από το ιστορικό τοποθεσιών που είναι από τις 1-2-2017 μέχρι και 4 μέρες μετά. Η επόμενη ρύθμιση είναι ο ελάχιστος αριθμός που

πρέπει να έχει μια συστάδα στον αλγόριθμο για να θεωρηθεί ενεργή. Και η τελευταία ρύθμιση είναι το εμβαδόν του τετραγώνου σε μέτρα που καταλαμβάνει κάθε συστάδα στον αλγόριθμο.

Το επόμενο πράγμα που μπορεί να κάνει ο χρήστης είναι οι επιλογές να πατήσει τα κουμπιά, extractPOI, extractFT, Extract User Habits. Η πρώτη αυτό που κάνει είναι να εξάγει τα δεδομένα της ανίχνευσης σημείων ενδιαφέροντος σε ένα αρχείο. Το αρχείο αυτό μέσα θα έχει συντεταγμένες, χρονοσφραγίδα κάθε σημείου, την ζώνη ώρας, και αν είναι εργάσιμη ή όχι μέρα. Ο λόγος που υπάρχει η επιλογή αυτή είναι για μελλοντική χρήση των δεδομένων. Ομοίως η extractFT εξάγει τις διαδρομές που βρεθήκαν από τον χρήστη του αλγορίθμου σε ένα αρχείο. Η επιλογή «extract User Habits» είναι για την εξαγωγή του αρχείου που περιέχει συχνές τοποθεσίες, συχνότητα εμφάνισης των σημείων αυτών, συχνότερη μέρα εμφάνισης υπάρχει κάποιο amenity γύρω από την συγκεκριμένη τοποθεσία γράφεται ο τύπος του και το όνομα του.

Στο γραφικό περιβάλλον χρήστη υπάρχουν ακόμη ένας χάρτης της OpenStreetMap και ένα πλαίσιο (κάτω αριστερά στην Εικόνα 4.10), το οποίο εκτυπώνει κάποια μηνύματα σχετικά με την πορεία του προγράμματος.

Κεφάλαιο 5

Αποτελέσματα

Στο κεφάλαιο αυτό θα αναλυθούν τα αποτελέσματα που προκύπτουν από τις παραπάνω υλοποιήσεις. Πρώτα θα αναλυθεί η ανίχνευση των συχνών τοποθεσιών. Στη συνέχεια η ανίχνευση τύπων διαδρομών και η ανίχνευση των συχνών διαδρομών, και τέλος η εξαγωγή συνηθειών του χρήστη στο αρχείο. Η ανάλυση θα γίνει σε δύο κατηγορίες. Η μία είναι αυτή των αποτελεσμάτων που παράγει η εφαρμογή, και η άλλη είναι για τον χρόνο ανταπόκρισης του προγράμματος σε διαφορετικές εισόδους.

Τα αποτελέσματα της εφαρμογής στηριχτήκαν στον υπολογιστή με τα παρακάτω χαρακτηριστικά:

Πίνακας 5.1 Χαρακτηριστικά υπολογιστή που στηρίχτηκαν τα αποτελέσματα

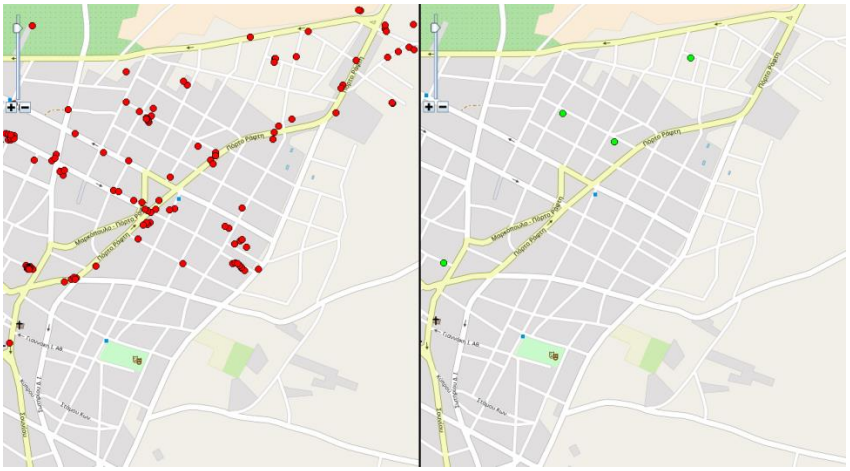
Επεξεργαστής :	Intel® Core™ 2 Duo E8500 3.16 GHz
Κάρτα μνήμης (RAM) :	4 GB
Λειτουργικό σύστημα :	Windows 10 64-bit
Έκδοση Java :	8.0

5.1 Ανάλυση αποτελεσμάτων συχνών τοποθεσιών

Μετά την εφαρμογή του αλγόριθμου στη συλλογή με τα σημεία, αυτό που θα πρέπει να φανεί πρώτα είναι η απομάκρυνση θορύβου. Στη συνέχεια, πρέπει να φανεί η εμφάνιση ενός σημείου ενδιαφέροντος στην λίστα από μια φορά και πάνω. Αυτό πρέπει να φανεί γιατί ο χρήστης μπορεί να βρέθηκε στο ίδιο σημείο διαφορετικές χρονικές στιγμές.

Παρακάτω παρουσιάζεται η εφαρμογή του αλγόριθμου για μια τυχαία εβδομάδα. Στην Εικόνα 5.1 φαίνονται τα αποτελέσματα του αλγορίθμου ανίχνευσης σημείων ενδιαφέροντος. Στην αριστερή εικόνα είναι όλα τα σημεία του ιστορικού τοποθεσιών για τις μέρες που έγινε η εφαρμογή του αλγορίθμου. Στην δεξιά εικόνα εμφανίζονται τα σημεία ενδιαφέροντος, όπως αυτά προκύπτουν από την εφαρμογή του αλγόριθμου που περιγράφτηκε.

Είναι φανερό από το σχήμα ότι έχει απομακρυνθεί ο θόρυβος και έχουν μείνει σημεία στα οποία ο χρήστης είχε έντονη δραστηριότητα. Αυτό που δεν φαίνεται στο σχήμα είναι ότι τα σημεία ενδιαφέροντος που βρίσκονται στην λίστα μπορεί να εμφανίζονται από μία και πάνω φορές. Αυτό γίνεται καθώς ο χρήστης μπορεί να έχει βρεθεί στο ίδιο σημείο ενδιαφέροντος διαφορετικές χρονικές στιγμές. Αυτό είναι χαρακτηριστικό στην περίπτωση των σημείων ακριβώς κάτω από το zoom slider της εικόνας 5.1 όπου ο χρήστης βρέθηκε πολλές φορές αλλά σε διαφορετικά χρονικά σημεία, πιθανά γιατί πέρασε από εκεί αρκετές φορές μέσα στη μέρα, αλλά δε στάθηκε ποτέ για μεγάλο διάστημα. Για το λόγο αυτό στο δεξί μέρος της εικόνας δεν έχουν εντοπιστεί σχετικά σημεία στάσης.



Εικόνα 5.1 Στιγμιότυπα πριν και μετά από την ανίχνευση σημείων ενδιαφέροντος.

Ο αλγόριθμος για την ανίχνευση σημείων ενδιαφέροντος λειτουργεί και έχουμε αποδοτικά αποτελέσματα. Στην συνέχεια από την λίστα μπορεί να βρεθεί η συχνότητα εμφάνισης κάθε σημείου. Η συχνότητα αυτή δίνει τα μέρη στα οποία ο χρήστης έχει βρεθεί περισσότερες φορές.

Παρακάτω στην Εικόνα 5.2 δίνεται ένα μέρος από το αρχείο που έχει αποθηκευτεί αυτή η πληροφορία. Όπως φαίνεται σε αυτό αναγράφονται η συντεταγμένες του σημείου, η συχνότητα εμφάνισης του και η συχνότερη μέρα που γίνεται επίσκεψη.

```
Latitude        Longitude
37.883818       23.9331544

Location frequence:39
Frequent Day: Sunday
```

Εικόνα 5.2 Στιγμιότυπο από το αρχείο με τις συνήθειες του χρήστη

Στην συνέχεια θα αναλυθούν οι χρόνοι εκτέλεσης της ανίχνευσης σημείων ενδιαφέροντος για διαφορετικές εισόδους. Στον συνολικό χρόνο θα πρέπει να προστεθεί και ο χρόνος εκτέλεσης της διαδικασίας ανίχνευσης συχνότητας κάθε τοποθεσίας. Ο χρόνος αυτός για όλες τις τιμές εισόδου που φαίνονται στον πίνακα 5.2 ήταν μηδενικός. Άρα εξετάζεται μόνο ο χρόνος εκτέλεσης του αλγορίθμου για την ανίχνευση σημείων ενδιαφέροντος.

Στον πίνακα 5.2 υπάρχουν τρεις στήλες. Η πρώτη στήλη είναι το πλήθος των σημείων που έχει η λίστα πριν την εφαρμογή του αλγορίθμου. Η δεύτερη στήλη είναι το πλήθος των σημείων μετά την εφαρμογή του αλγορίθμου. Η τελευταία στήλη είναι ο χρόνος που χρειάστηκε για να ολοκληρωθεί η διαδικασία. Από τον πίνακα 5.2 φαίνεται ότι από τα σημεία εισόδου έγινε η απομάκρυνση θορύβου. Για παράδειγμα στην τελευταία γραμμή του πίνακα σαν είσοδος δόθηκε ένα σύνολο από 6945 σημεία και σαν έξοδο 5854 σημεία. Αυτό πρακτικά σημαίνει ότι απομακρύνθηκε ο θόρυβος. Υπενθυμίζεται ότι μέσα στην λίστα με τα 5854 σημεία υπάρχει η πιθανότητα ένα σημείο να εμφανίζεται περισσότερο από μια φορά αλλά σε διαφορετικό χρόνο.

Τα αποτελέσματα για τον χρόνο εκτέλεσης που βρεθήκαν δείχνουν ότι όσο μεγαλύτερη είναι η είσοδος, ο χρόνος εκτέλεσης θα αυξάνεται. Έτσι φαίνεται ότι για αρκετά μεγάλες εισόδους η εφαρμογή θα απαιτεί αρκετά μεγάλο χρόνο εκτέλεσης.

Πίνακας 5.2 Χρόνοι εκτέλεσης σημείων ενδιαφέροντος

Πλήθος σημείων εισόδου	Πλήθος σημείων εξόδου	Χρόνος εκτέλεσης (seconds)
628	105	0
757	148	1
3419	2781	18
4432	4068	30
4688	3930	36
5068	4102	41
6945	5854	82

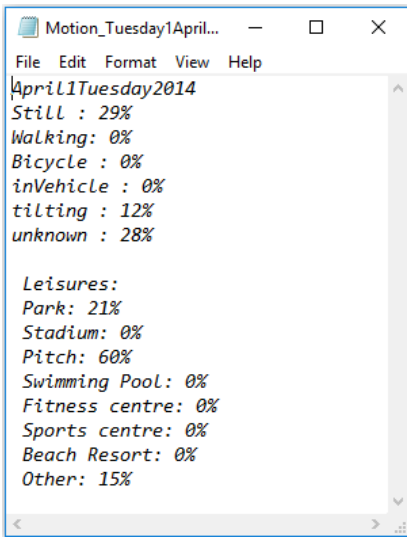
5.2 Ανάλυση αποτελεσμάτων ανίχνευσης τύπων κινήσεων

Σε αυτή την ενότητα θα αναλυθούν τα αποτελέσματα για τους τύπους κινήσεων του χρήστη. Για την ανάγκη της ανάλυσης επιλέχτηκε μια τυχαία μέρα από το ιστορικό ενός

χρήστη. Η ανίχνευση γίνεται αυτόματα μόλις ο χρήστης της εφαρμογής πατήσει το κουμπί για να εξάγει τις συχνές διαδρομές. Τα αποτελέσματα αυτά αποθηκεύονται σε ένα αρχείο.

Στην Εικόνα 5.3 φαίνεται το αρχείο που αποθηκεύτηκε η πληροφορία αυτή. Όπως φαίνεται στα αποτελέσματα η προτίμηση του χρήστη εκείνη την μέρα ήταν να κάθεται ακίνητος. Από την άλλη πολύ κοντά σε αυτή την προτίμηση είναι και η μεταβλητή unknown που δηλώνει άλλους τύπους κινήσεων που δεν υπάρχουν στο πρόγραμμα. Οπότε πολύ πιθανό εκείνη την ημέρα να κινήθηκε με άλλους τρόπους που δεν αναφέρονται στο πρόγραμμα. Ακόμη από το αρχείο φαίνεται ότι ο χρήστης εκείνη την ημέρα πολύ πιθανό να μην χρησιμοποίησε κάποιο όχημα.

Μαζί με τις παραπάνω πληροφορίες περιέχονται και αυτές που δηλώνουν σε τι είδη περιοχών προτίμησε ο χρήστης να βρίσκεται την συγκεκριμένη μέρα. Στα αποτελέσματα φαίνεται ότι από όλο το ιστορικό το 21% καταγράφτηκε κοντά σε κάποιο πάρκο, το 60% καταγράφτηκε σε κάποιο γήπεδο. Ένα ποσοστό 15% του ιστορικού καταγράφτηκε σε άλλα είδη τοποθεσιών που δεν ανήκουν στο πρόγραμμα.

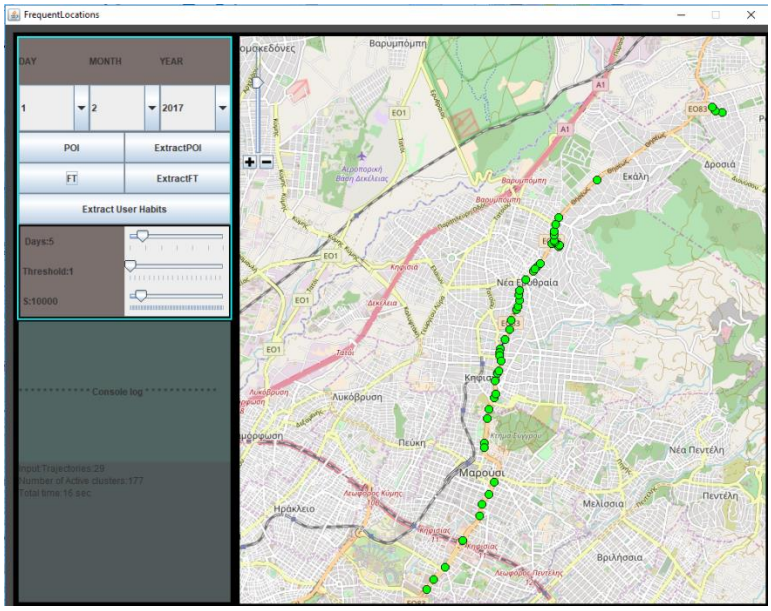


Εικόνα 5.3 Αρχείο που περιέχει τις πληροφορίες για τους τύπους κινήσεων

Υπάρχουν χρονικές στιγμές που τα αποτελέσματα δεν είναι ικανοποιητικά. Αυτό είναι πιθανό να συμβαίνει επειδή δεν έχουν καταγραφεί στο αρχείο JSON, με τα ίχνη του χρήστη, οι πληροφορίες για τον πιθανό τύπο κίνησης. Όσον αφορά στην περίπτωση που δεν είναι ικανοποιητικά τα αποτελέσματα τον τύπο των περιοχών εκεί είναι πολύ πιθανό να μην υπάρχουν αρκετές πληροφορίες για τη συγκεκριμένη περιοχή που δραστηριοποιήθηκε ο χρήστης.

5.3 Ανάλυση αποτελεσμάτων συχνών διαδρομών

Στην ενότητα αυτή θα γίνει η ανάλυση των αποτελεσμάτων της ανίχνευσης συχνών διαδρομών. Στην Εικόνα 5.4 εμφανίζονται τα αποτελέσματα για την ανίχνευση συχνών διαδρομών όπως αυτά φαίνονται στην εφαρμογή.



Εικόνα 5.4 Στιγμιότυπο μετά την ανίχνευση συχνών διαδρομών

Για τις ανάγκες της ανάλυσης δοκιμάστηκε η διαδικασία αυτή για διαφορετικές εισόδους και με τις μεταβλητές s=10.000 , Threshold=6. Στον πίνακα 5.3 φαίνονται τα αποτελέσματα.

Πίνακας 5.3 Χρόνοι εκτέλεσης της διαδικασίας των συχνών διαδρομών

Πλήθος διαδρομών	Ενεργές συστάδες	Πλήθος συχνών διαδρομών	Μέγιστο μήκος διαδρομών	Χρόνος εκτέλεσης(second)
38	49	40	4	2
72	49	40	4	4
107	84	68	4	12
136	89	71	4	17
194	95	92	6	74
493	82	80	8	282
947	148	109	15	556

Όπως φαίνεται από τα παραπάνω αποτελέσματα όσα περισσότερα δεδομένα έχει η εφαρμογή τόσες περισσότερες συχνές τοποθεσίες βρίσκονται. Επίσης όσο περισσότερα δεδομένα έχει η είσοδος αυξάνεται και το μέγιστο μήκος διαδρομών.

Από την άλλη μεριά είναι φανερό ότι όσο μεγαλύτερη είναι η είσοδος τόσο αυξάνεται ο χρόνος εκτέλεσης. Αυτό σημαίνει ότι για ένα τεράστιο σύνολο διαδρομών η εφαρμογή θα πάψει να είναι λειτουργική καθώς θα χρειάζεται αρκετή ώρα για να εκτελέσει την διαδικασία. Επίσης φαίνεται ότι όσες περισσότερες διαδρομές έχει η είσοδος τόσες περισσότερες θα είναι και η ενεργές συστάδες στον αλγόριθμο. Στην περίπτωση που οι ενεργές συστάδες είναι αρκετά κοντά η μία στην άλλη είναι ένα θετικό αποτέλεσμα. Όσο πιο κοντά είναι οι ενεργές συστάδες και το πλήθος τους αυξάνει τόσο πιο κοντά στην πραγματικότητα θα είναι τα αποτελέσματα.

Στην συνέχεια φαίνονται τα αποτελέσματα για σταθερό αριθμό διαδρομών και διαφορετικών τιμών για την μεταβλητή S.

Πίνακας 5.4 Χρόνοι εκτέλεσης συχνών διαδρομών με διαφορετικό S

Πλήθος διαδρομών	**S**	**Πλήθος συχνών διαδρομών**	**Μέγιστο μήκος διαδρομών**	**Χρόνος εκτέλεσης (sec)**
38	90.000	18	10	0
38	40.000	26	5	0
38	10.000	40	4	2
38	5.000	44	2	5
38	2.500	50	2	12

Όπως φαίνεται στον πίνακα για μεγάλο S οι χρόνοι εκτέλεσης της διαδικασίας είναι μηδενικοί. Από την άλλη όσο μικραίνει το S τότε οι χρόνοι αυτοί αυξάνονται. Αυτό δίνει την πληροφορία ότι αν χρειάζεται να μειωθεί ο χρόνος εκτέλεσης για ένα μεγάλο όγκο δεδομένων, μια λύση είναι να αυξηθεί η μεταβλητή S. Αυτό συμφέρει κυρίως όταν η καταγραφή σημείων έχει γίνει με τέτοιο τρόπο ώστε τα σημεία να είναι αραιά. Από τα αποτελέσματα φαίνεται επίσης ότι όσο πιο μεγάλο αριθμό έχει το S τόσες λιγότερες συχνές διαδρομές επιστρέφονται από τον αλγόριθμο όμως αυξάνεται και το μέγιστο μήκος αυτών.

Τελευταία ανάλυση για την διαδικασία αυτή θα γίνει για να μελετηθεί αν η μεταβλητή Threshold επηρεάζει τα αποτελέσματα. Για την δοκιμή αυτή θα γίνει ένα πείραμα με σταθερό

αριθμό διαδρομών ως είσοδο, σταθερό το S = 10.000 και διαφορετικές τιμές για το threshold. Στον πίνακα 5.5 φαίνονται τα αποτελέσματα

Πίνακας 5.5 Μελέτη της μεταβλητής threshold

Πλήθος διαδρομών	Threshold	Ενεργές συστάδες	Πλήθος συχνών διαδρομών	Μέγιστο μήκος διαδρομών
38	1	49	40	4
38	3	24	23	2
38	6	18	18	1
38	10	15	15	1
38	20	12	12	1
38	40	9	9	1
38	80	7	7	1

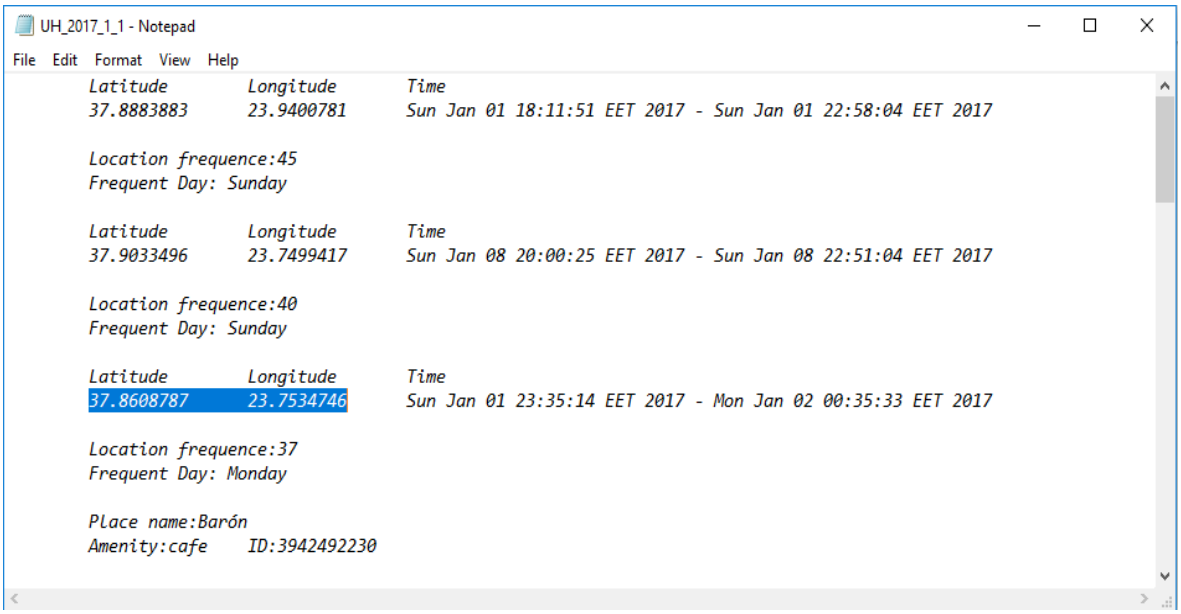
Τα αποτελέσματα έδειξαν όπως είναι το λογικό ότι όσο αυξάνεται η τιμή threshold του αλγορίθμου τόσες λιγότερες ενεργές συστάδες θα δημιουργηθούν. Αυτό συμβαίνει γιατί για να θεωρηθεί μια συστάδα ενεργή θα πρέπει να έχει μεγαλύτερο αριθμό σημείων από την τιμή του threshold. Όποτε αν η τιμή αυτή είναι αρκετά μεγάλη η πιθανότητες να δημιουργηθεί μια ενεργή συστάδα μειώνεται. Ακόμη φαίνεται από τα αποτελέσματα ότι όσο αυξάνεται η τιμή της μεταβλητής threshold τόσο μειώνεται και το πλήθος των συχνών διαδρομών καθώς και το μέγιστο μήκος διαδρομών. Το γεγονός ότι μειώνεται και το μέγιστο μήκος διαδρομής δεν είναι αρνητικό γεγονός. Και αυτό διότι όσο αυξάνεται η τιμή threshold αυτό σημαίνει ότι η ενεργές συστάδες αποτελούνται από μεγαλύτερο πλήθος σημείων. Άρα πρακτικά στην περίπτωση που στα δεδομένα υπάρχει τουλάχιστον μια συχνή διαδρομή αυτό θα μας οδηγήσει στην καλύτερη ανάλυση της.

5.4 Ανάλυση αποτελεσμάτων εξαγωγής συνηθειών του χρήστη

Η τελευταία διαδικασία που θα αναλυθεί είναι αυτή της εξαγωγής συνηθειών του χρήστη σε ένα αρχείο. Αυτό που γίνεται ουσιαστικά είναι η εύρεση των συχνών τοποθεσιών του χρήστη και στην συνέχεια η αναζήτηση για το αν υπάρχει κάποιο amenity γύρω από την συγκεκριμένη τοποθεσία.

Συγκεντρωτικά μέσα στο αρχείο αυτό υπάρχουν πληροφορίες όπως συχνές τοποθεσίες, συχνές μέρες που τις επισκέπτεται ο χρήστης. Μια άλλη πληροφορία που δίνεται είναι η μέρα και η ώρα που βρέθηκε ο χρήστης στην συγκεκριμένη τοποθεσία, ομοίως δίνεται η μέρα και η ώρα για την στιγμή που επισκέφτηκε ο χρήστης από την τοποθεσία για τελευταία φορά. Ακόμη δίνεται πληροφορία για το όνομα της τοποθεσίας σε περίπτωση που υπάρχει κάποιο amenity, καθώς και ο τύπος του.

Στην Εικόνα 5.5 φαίνεται ένα αρχείο που δημιουργήθηκε από ένα ιστορικό.



Εικόνα 5.5 Στιγμιότυπο από το αρχείο που δημιουργήθηκε με τις συνήθειες του χρήστη

Κεφάλαιο 6

Συμπεράσματα

6.1 Συμπεράσματα εφαρμογής

Όπως φαίνεται και από τα αποτελέσματα οι εφαρμογή είναι αποδοτική ως προς τα δεδομένα που προκύπτουν από την χρήση των αλγορίθμων, όμως ο χρόνος εκτέλεσης των διαδικασιών μπορεί να επηρεάσει την αποτελεσματικότητα της εφαρμογής.

Για τεράστιο αριθμό δεδομένων η εφαρμογή δεν είναι αρκετά αποδοτική ως προς τον χρόνο. Για να λυθεί αυτό το πρόβλημα μια λύση είναι να γραφτεί μια εφαρμογή με βάση τον παράλληλο προγραμματισμό. Αυτό θα χωρίσει τον φόρτο των υπολογισμών στους αντίστοιχους επεξεργαστές και έτσι θα υπάρχουν καλύτερα αποτελέσματα για τους χρόνους εκτέλεσης των υπολογισμών αυτών.

Η εφαρμογή αυτή χρησιμοποιεί ένα αρχείο που περιέχει το ιστορικό τοποθεσιών του χρήστη, του οποίου η απόκτηση δεν γίνεται αυτόματα. Οπότε ίσως θα ήταν μια καλή ιδέα να φτιαχτεί μια εφαρμογή η οποία θα έκανε την διαδικασία αυτή αυτόματη. Μια άλλη λύση σε αυτό θα ήταν να φτιαχτεί μια εφαρμογή για έξυπνα τηλέφωνα που να κάνει καταγραφή των τοποθεσιών του χρήστη να της αποθηκεύει σε μια βάση δεδομένων και στην συνέχεια να συνδέεται με την παρούσα εφαρμογή για να δημιουργηθεί έτσι ένα πιο αυτόματο σύστημα.

Τέλος τα δεδομένα που αποκτήθηκαν από την εφαρμογή θα μπορούσαν να αποθηκεύονται σε μια βάση δεδομένων με μια δομή, έτσι ώστε μελλοντικές εφαρμογές να μπορούν να εκμεταλλευτούν τις πληροφορίες αυτές για τους σκοπούς τους.

Βιβλιογραφία

Chen, G., Chen, B., & Yu, Y. (2010, December). Mining frequent trajectory patterns from GPS tracks. In Computational Intelligence and Software Engineering (CiSE), 2010 International Conference on (pp. 1-6). IEEE.

Feng, Z., & Zhu, Y. (2016). A survey on trajectory data mining: Techniques and applications. IEEE Access, 4, 2056-2067.

Kisilevich, S., Mansmann, F., & Keim, D. (2010, June). P-DBSCAN: a density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos. In Proceedings of the 1st international conference and exhibition on computing for geospatial research & application (p. 38). ACM.

Li, Q., Zheng, Y., Xie, X., Chen, Y., Liu, W., & Ma, W. Y. (2008, November). Mining user similarity based on location history. In Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems (p. 34). ACM.

Noulas, A., Scellato, S., Mascolo, C., & Pontil, M. (2011). An empirical study of geographic user activity patterns in foursquare. ICwSM, 11, 70-573.

Shaw, A. A., & Gopalan, N. P. (2011). Frequent pattern mining of trajectory coordinates using apriori algorithm. International Journal of Computer Applications, 22(9).

Shaw, A. A., & Gopalan, N. P. (2014). Finding frequent trajectories by clustering and sequential pattern mining. Journal of Traffic and Transportation Engineering (English Edition), 1(6), 393-403.

Ye, J., Zhu, Z., & Cheng, H. (2013, May). What's your next move: User activity prediction in location-based social networks. In Proceedings of the 2013 SIAM International Conference on Data Mining (pp. 171-179). Society for Industrial and Applied Mathematics.

Ying, J. J. C., Lu, E. H. C., Kuo, W. N., & Tseng, V. S. (2012, August). Urban point-of-interest recommendation by mining user check-in behaviors. In *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*(pp. 63-70). ACM.

Yuan, Q., Cong, G., Ma, Z., Sun, A., & Thalmann, N. M. (2013, July). Time-aware point-of-interest recommendation. In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval (pp. 363-372). ACM.

Zhang, Z., Huang, K., & Tan, T. (2006, August). Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In Pattern Recognition, 2006. ICPR 2006. 18th International Conference on (Vol. 3, pp. 1135-1138). IEEE.

Τραγοπούλου Σ. (2016), Ανίχνευση συνηθειών χρήστη με χρήση κινητών τηλεφώνων, Χαροκόπειο Πανεπιστήμιο.

Άλλες πηγές

OpenStreetMap Wiki, JMapViewer, http://wiki.openstreetmap.org/wiki/JMapViewer. [Online·τελευταία προσπέλαση 15-Σεπτεμβρίου-2017].

OpenStreetMap Wiki, Overpass-API, http://wiki.openstreetmap.org/wiki/Overpass_API. [Online·τελευταία προσπέλαση 15-Σεπτεμβρίου-2017].

Wikipedia The Free Encyclopedia, Apache Commons, https://en.wikipedia.org/wiki/Apache_Commons. [Online τελευταία προσπέλαση 15-Σεπτεμβρίου-2017].

Wikipedia The Free Encyclopedia, Netbeans, https://en.wikipedia.org/wiki/NetBeans. [Online τελευταία προσπέλαση 15-Σεπτεμβρίου-2017].

Wikipedia The Free Encyclopedia, Jackson_(API), https://en.wikipedia.org/wiki/Jackson_(API). [Online·τελευταία προσπέλαση 15-Σεπτεμβρίου-2017].

Βικιπαίδεια, OpenStreetMap, https://el.wikipedia.org/wiki/OpenStreetMap. [Online·τελευταία προσπέλαση 15-Σεπτεμβρίου-2017].