



**ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ**

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Διαχείριση Δικτών (ΕΠ 17)
Εαρινό Εξάμηνο 2014 - 2015

Λήδα – Κυριακή Ζαχαροπούλου, Α.Μ. : 1115201100004
Σοφία Κυπραίου, Α.Μ. : 111520110047

ΠΕΡΙΕΧΟΜΕΝΑ

1.ΕΙΣΑΓΩΓΗ.....	3
1.1 <i>Τεχνολογίες και frameworks που χρησιμοποιήθηκαν.....</i>	<i>3</i>
1.2 <i>Οδηγίες εκτέλεσης και αρχικοποίηση.....</i>	<i>3</i>
2.ΟΠΤΙΚΟΠΟΙΗΣΗ ΔΕΔΟΜΕΝΩΝ.....	4
3.ΑΝΑΛΥΣΗ ΔΕΔΟΜΕΝΩΝ.....	5
4.ΕΞΑΓΩΓΗ ΣΥΜΠΕΡΑΣΜΑΤΩΝ.....	7

1.ΕΙΣΑΓΩΓΗ

1.1 Τεχνολογίες και frameworks που χρησιμοποιήθηκαν

Για την εργασία όπου ζητείται η ανάπτυξη ενός application, επιλέξαμε να υλοποιήσουμε ένα Web Application σε Java με την χρήση JavaServer Pages (JSP) και Servlets. Χρησιμοποιήσαμε το Bootstrap για το front-end framework.

Αναλυτικότερα, από την πλευρά του server, για την ευκολότερη αποθήκευση και επεξεργασία των δεδομένων από τα datasets, χρησιμοποιήσαμε το Hibernate. Το κύριο χαρακτηριστικό του που μας έκανε να το επιλέξουμε, ήταν το mapping που προσέφερε ανάμεσα στις κλάσεις Java και στις SQL βάσεις.

Από την πλευρά του client, χρησιμοποιήσαμε JavaServer Pages για την δημιουργία δυναμικών web pages. Εκεί γίνεται σε πολλά σημεία η παρεμβολή κώδικα Javascript και jQuery. Στην σελίδα γίνεται η χρήση google map. Εκεί, χρησιμοποιήθηκαν οι βιβλιοθήκες που παρέχονται από την google., καθώς και το API του Google Maps

Για το σχεδιασμό του web application χρησιμοποιήθηκε Bootstrap. Η δημιουργία διαγραμμάτων πραγματοποιήθηκε με τη χρήση της βιβλιοθήκης ShieldUI.

Η επικοινωνία ανάμεσα σε server και client γίνεται με την χρήση των Java servlets. Με post και get η πληροφορία μεταφέρεται ανάμεσα στις δύο οντότητες. Για να επιτευχθεί η κλήση περισσότερων από έναν servlets, χρησιμοποιήσαμε AJAX (Asynchronous JavaScript and XML). Επιπλέον, η μεταφορά της πληροφορίας έγινε με την χρήση των JSON (JavaScript Object Notation). Στις ειδικές περιπτώσεις που χρειάστηκε να επιστραφεί πληροφορία που απαιτούσε μία απεικόνιση στον χάρτη, αυτό έγινε με GeoJSON.

Όπου υπάρχει έτοιμος κώδικας, αναφέρεται στην αρχή του κάθε αρχείου ή της κάθε μεθόδου.

1.2 Οδηγίες εκτέλεσης και αρχικοποίηση

Η άσκηση εκτελείται τρέχοντας το index.jsp. Την πρώτη φορά που θα φορτώσει η σελίδα, θα κλειθεί ο αντίστοιχος server που θα κάνει το initialization.

Για το initialization απαιτούνται πληροφορίες για τα paths όπου βρίσκονται τα csv αρχεία, και για την σύνδεση με την βάση. Οι πληροφορίες αυτές εισάγονται στο property file “application.properties”.

Κατά το initialization διαβάζονται οι πληροφορίες από το αρχείο, φορτώνεται η βάση, και υπολογίζονται οι “ακριβείς” γεωγραφικές τοποθεσίες των Access Points (1. Επεξεργασία Δεδομένων). Η πληροφορία αυτή χρησιμοποιείται καθ όλη την διάρκεια της περιήγησης, επομένως είναι πιο αποδοτικό να υπολογίζεται μόνο μία φορά.

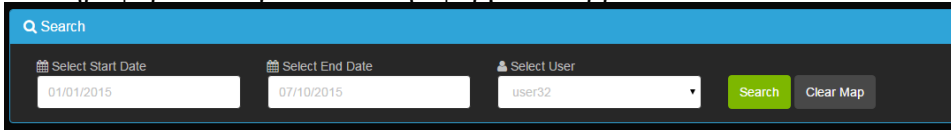
Επιπλέον, υπολογίζονται και όλοι οι χρήστες που υπάρχουν στην βάση, καθώς χρησιμοποιούνται σε άλλους υπολογισμούς σε επόμενες ενέργειες.

2. ΟΠΤΙΚΟΠΟΙΗΣΗ ΔΕΔΟΜΕΝΩΝ

Παραδείγματα Εκτέλεσης

Στα παρακάτω screenshots φαίνονται τα αποτελέσματα για τον χρήστη “user32” για την Οπτικοποίηση των Δεδομένων.

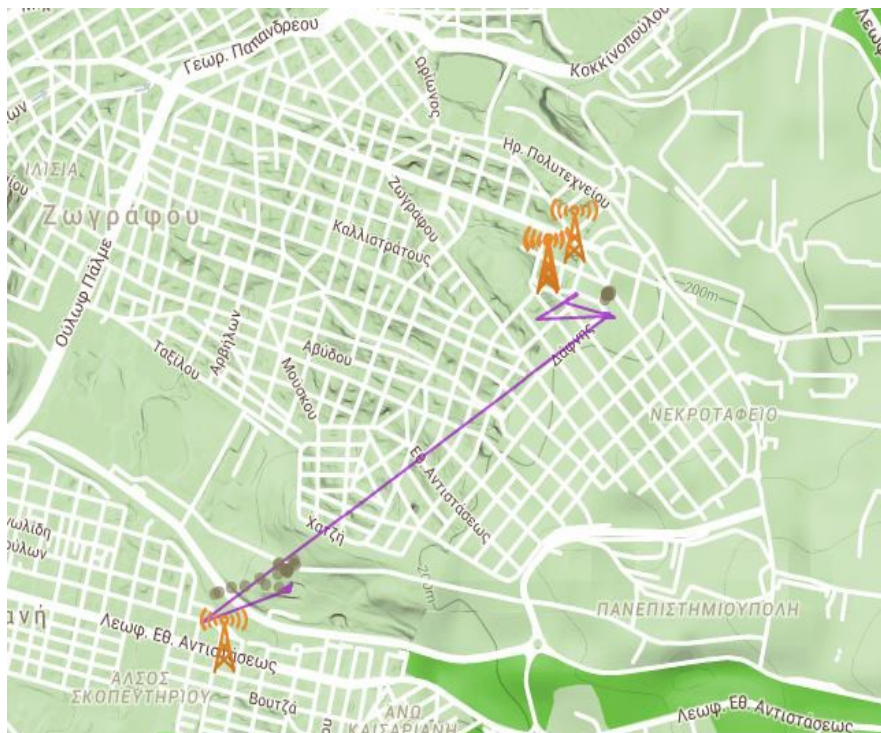
Η πληροφορία εισάγεται από την φόρμα που βρίσκεται στο Access Points και είναι ως εξής:



Με το κουμπί *search* ο ενδιαφερόμενος μπορεί να βρει όλα τα Access Points στα οποία συνδέθηκε ο χρήστης εκείνο το χρονικό περιθώριο, την διαδρομή την οποία έκανε, καθώς και τα Base Stations στα οποία συνδέθηκε.

Με το *clear map*, καθαρίζουν τα δεδομένα από τον χάρτη.

Στην εικόνα που ακολουθεί, τα Access Points απεικονίζονται με μικρές γκρι κουκίδες, όπου φαίνεται και η πυκνότητά τους. Η διαδρομή που ακολούθησε ο χρήστης είναι η μωβ γράμμη, ενώ με πορτοκαλί φαίνονται τα Base Stations (Cells) στα οποία συνδέθηκε ο χρήστης. Κάνοντας click στα Access Points και τα Base Stations, εμφανίζονται οι αντίστοιχες πληροφορίες που τα αφορούν.



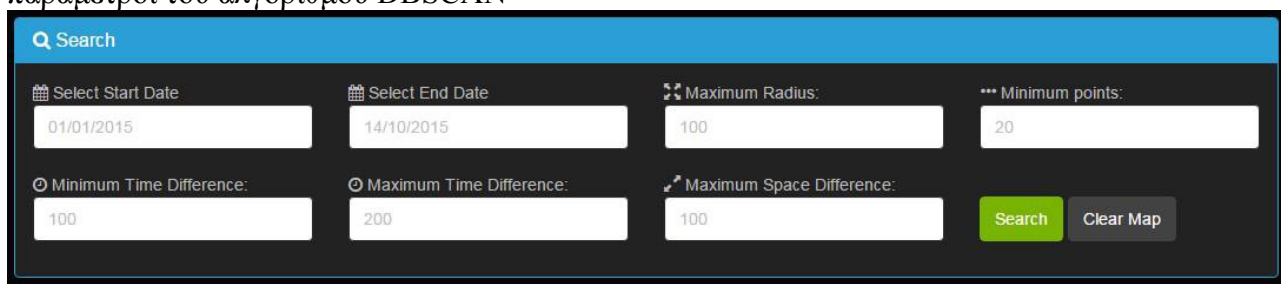
Οι συναρτήσεις που υλοποιούν τις αλλαγές στον χάρτη βρίσκονται στο `web/js/my-googlemaps-functions.js`.

3. ΑΝΑΛΥΣΗ ΔΕΔΟΜΕΝΩΝ

Για τον υπολογισμό των Stay Points χρησιμοποιήθηκε ο αλγόριθμος που υπάρχει στην εκφώνηση, καθώς και όσες αλλαγές/ διορθώσεις υπήρξαν αργότερα στο eclass.

Για τον αλγόριθμο DBSCAN ακολουθήσαμε την υπολοποίηση που βρίσκεται εδώ (<https://github.com/KanwarBhajneek/DBSCAN>).

Ακολουθούν μετρήσεις με διάφορες από τος παραμέτρους των αλγορίθμων για Tmin, Tmax (ελάχιστος και μέγιστος χρόνος παραμονής ενός χρήστη σε ένα σημείο), και Dmax (η μέγιστη απόσταση ανάμεσα στα σημεία της συστάδας). Οι παράμετροι Radius και Minimum Points είναι οι παράμετροι του αλγορίθμου DBSCAN



The screenshot shows a web interface for the DBSCAN algorithm. It has a blue header with a search icon and the word 'Search'. Below the header, there are several input fields arranged in a grid. The first row contains 'Select Start Date' with a calendar icon and a date input field showing '01/01/2015', 'Select End Date' with a calendar icon and a date input field showing '14/10/2015', 'Maximum Radius:' with a range icon and a numeric input field showing '100', and 'Minimum points:' with a range icon and a numeric input field showing '20'. The second row contains 'Minimum Time Difference:' with a range icon and a numeric input field showing '100', 'Maximum Time Difference:' with a range icon and a numeric input field showing '200', 'Maximum Space Difference:' with a range icon and a numeric input field showing '100', and two buttons: 'Search' (green) and 'Clear Map' (grey).

- Είσοδος:
Tmin: 30
Tmax: 180
Dmax: 300
Radius: 300
Minimum Points: 20

Αποτέλεσμα:
Points of interest: 9

Με τους αλγορίθμους βρίσκουμε πως προκύπτουν 9 Points Of Interest.

- Είσοδος:
Tmin: 30
Tmax: 180
Dmax: 300
Radius: 300
Minimum Points: 15

Αποτέλεσμα:
Points of interest: 13

Παρατηρούμε πως όταν μειώσαμε τα ελάχιστα απαιτούμενα στοιχεία για μία συστάδα, ο αριθμός των Points of interest αυξήθηκε, όπως περιμέναμε

- Είσοδος:
Tmin: 100
Tmax: 200
Dmax: 300
Radius: 100
Minimum Points:10

Αποτέλεσμα:
Points of interest:15

Αυξάνοντας τον ελάχιστο χρόνο παραμονής σε ένα σημείο (μειώνοντας συνολικά τον χρόνο παραμονής), και μειώνοντας ελάχιστα τα minimum points, παρατηρούμε πως συνεχίζουν να αυξάνονται τα Points of interest, παρόλο που μειώσαμε την ακτίνα.

- Είσοδος:
Tmin: 30
Tmax: 200
Dmax: 300
Radius: 300
Minimum Points:2

Αποτέλεσμα:
Points of interest:63

Σε αυτή τη δοκιμή φαίνεται ξεκάθαρα η άμεση συσχέτιση που υπάρχει ανάμεσα στο πλήθος των minimum points με τα Points of interest, αφού είναι μεγάλη η αύξησή των δευτέρων σε σχέση με τα πρώτα.

- Είσοδος:
Tmin: 100
Tmax: 200
Dmax: 100
Radius: 100
Minimum Points:20

Αποτέλεσμα:
Points of interest:7

Μειώνοντας μόνο το Dmax σε σύγκριση με προηγούμενη δοκιμή, παρατηρούμε πως ο αριθμός των points of interest μειώθηκε δραματικά.

Παραδοχές

Δεν καταλάβαμε πώς από τις συντεταγμένες του κεντροειδούς σημείου και τις ελάχιστες και μέγιστες των σημείων της συστάδας του προκύπτει το παραλληλόγραμμο που ζητείται,γι' αυτό τα Points of Interest παρουσιάζονται με τη μορφή marker,ενώ σε σχόλια υπάρχει και η υλοποίηση τους ως polygons (τρίγωνα),με κορυφές το κεντροειδές,το πιο κοντινό της συστάδας και το πιο μακρίνο.

4. ΕΞΑΓΩΓΗ ΣΥΜΠΕΡΑΣΜΑΤΩΝ

Οι υλοποιήσεις βρίσκονται στο "Statistics", εκτός από το 4.2 που βρίσκεται στο "Optimal Access Points"

4.1) Έχουμε προσθέσει ένα επιπλέον διάγραμμα, μία επέκταση του διαγράμματος που ζητείται στο ερώτημα αυτό. Περιέχει για κάθε μέρα, και για κάθε ώρα της ημέρας αναλυτικά τους χρήστες όπου βρέθηκαν με χαμηλή μπαταρία. Πάνω στο διάγραμμα ο ενδιαφερόμενος μπορεί να κάνει zoom στα δεδομένα που τον ενδιαφέρουν για καλύτερη οπτικοποίηση.

4.2) Παράδειγμα εκτέλεσης:

