

Γαβάνας Χρήστος - 1115201100003

Γρηγόρης Δημόπουλος - 1115201100198

Διάβασμα από αρχεία CSV

Με ότι έχει να κάνει με το διάβασμα των δεδομένων από τα αρχεία **csv** με αυτό ασχολούνται οι κλάσεις του πακέτου **records**(ειδικότερα η κλάσεις **BaseStationRecords**,**BatteryRecords**,**GpsRecords** και **WifiRecords**).

Οι κλάσεις αυτές διαβάζουν τα δεδομένα από τα αντίστοιχα αρχεία η κάθε μια και τα αποθηκεύουν σε μια λίστα κατάλληλων οντοτήτων (κάθε μια οντότητα αντιστοιχεί σε μια έγγραφη από κάποιο αρχείο **csv**)

Οι οντότητες αυτές είναι διαφορετικές για κάθε ένα τύπο έγγραφης διαφορετικού αρχείου **csv**. Το πακέτο λοιπόν που περιγράφονται και υλοποιούνται αυτές οι οντότητες έγγραφων των αρχείων **csv** είναι το **Interfaces** όπου περιέχει τις κλάσεις για τις έγγραφες των BaseStation (**BaseStationWrapper**), των Battery (**BatteryWrapper**), των Gps(**GpsWrapper**), των AccessPoints/Wifi(**WifiWrapper**).

Οι κλάσεις αυτές αποθηκεύουν πλήρως την πληροφορία μιας έγγραφης από το αντίστοιχο **csv** αρχείο που αντιχίζονται, και επίσης υλοποιούν χρήσιμες συναρτήσεις για τον χειρισμό των δεδομένων και την πιο εύκολη χρήση τους.

Όσο αναφορά την συνολική πληροφορία αποθηκεύεται όπως προείπαμε στις κλάσεις του πακέτου **records** μέσα σε λίστες οντοτήτων τύπου Wrapper. Επιπλέον υπάρχει η δυνατότητα δημιουργίας ευρετήριου επάνω στα δεδομένα ώστε να γίνεται πιο εύκολη και γρήγορη η αναζήτηση πάνω σε συγκεκριμένα πεδία.

Το πακέτο που φιλοξενεί τα ευρετήρια είναι το **indexes**. Αυτό περιέχει κλάσεις με ευρετήρια για όλα τα πιθανά δεδομένα. Τα ευρετήρια αυτά μπορούν να δημιουργηθούν πάνω σε κάποιο πεδίο της έγγραφης για παράδειγμα τον **user** η το **timestamp**. Η κλάση ευρετηρίων περιέχει μέσα HashMaps για γρήγορη αναζήτηση και τα δεδομένα σε λίστα επίσης όπως και βοηθητικές συναρτήσεις για καλύτερο χειρισμό των δεδομένων. Το path των αρχείων **csv** δίνεται από ένα **property file** έτσι ώστε να υπάρχει η δυνατότητα αλλαγής πηγής δεδομένων.

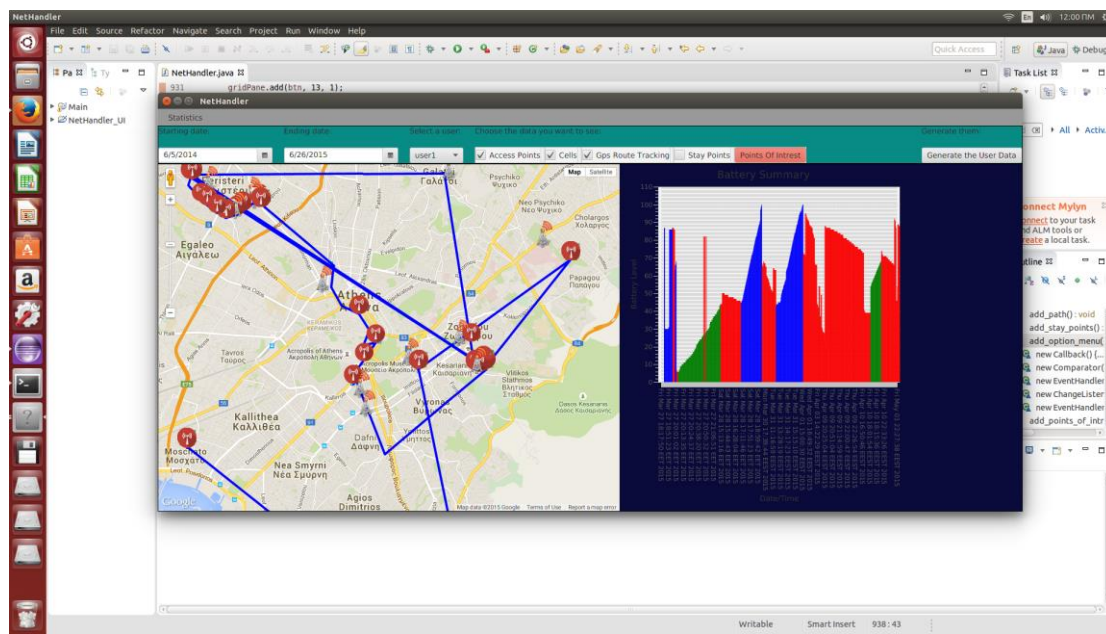
Σημείωση: Όσο αναφορά τα δεδομένα **Base Station** ορισμένες έγγραφες παρατηρήθηκε ότι δεν έχουν δεδομένα στο γεωγραφικό μήκος και πλάτος. Αυτές οι έγγραφες εκ παραδοχής περνάνε στο σύστημα με τον κωδικό 666 ως ένδειξη της μη ύπαρξης δεδομένων (ούτε το γεωγραφικό πλάτος μπορεί να είναι 666 ούτε το γεωγραφικό μήκος).

Σχετικά με το γραφικό(GUI)

Σε ότι έχει να κάνει με το γραφικό χρήσης (Gui) υλοποιήθηκε σε **java fx** και χωρίζεται σε ορισμένες συναρτήσεις στο κεντρικό πακέτο της εφαρμογής **application** στην κλάση **Main**.

Το γραφικό αποτελείται απο το menu επιλογών (**option menu**)οπού πάνω στο menu αυτό γίνονται οι επιλογές μας σχετικά με τα δεδομένα που θέλουμε να δούμε πάνω στον χάρτη και τα δεδομένα που θέλουμε να εξετάσουμε στα γραφήματα. Επιπλέον υπάρχει ο χάρτης που τον εξετάζουμε σε άλλο σημείο πιο αναλυτικά,πανω σε αυτόν φαίνονται τα **Access Points** ,**BaseStation**,**Path** ,**Stay Points**,**Points of Intersts** και **νέα Access Points** του ερωτήματος 4.4.

Επιπλέον υπάρχουν γραφήματα για την κατανάλωση της μπαταρίας στο χρονικό περιθώριο που έχει ζητήσει ο χρήστης.Τα στοιχεία στο γράφημα είναι color-coded όταν ο χρήστης είναι συνδεδεμένος στο φορτιστή το γράφημα στην περιοχή εκείνη είναι πρασινο,οταν είναι συνδεδεμένο στον υπολογιστή είναι μπλε ενώ όταν δεν φορτίζει με κοπούν τρόπο είναι κοκκινο.Επιπλέον αν επιλέξεις μπορείς να δεις το γράφημα με τους χρήστες που είχαν λιγότερο από 15% μπαταριά σε νέο παράθυρο για μια συγκεκριμένη μέρα



Σχετικά με τον χάρτη

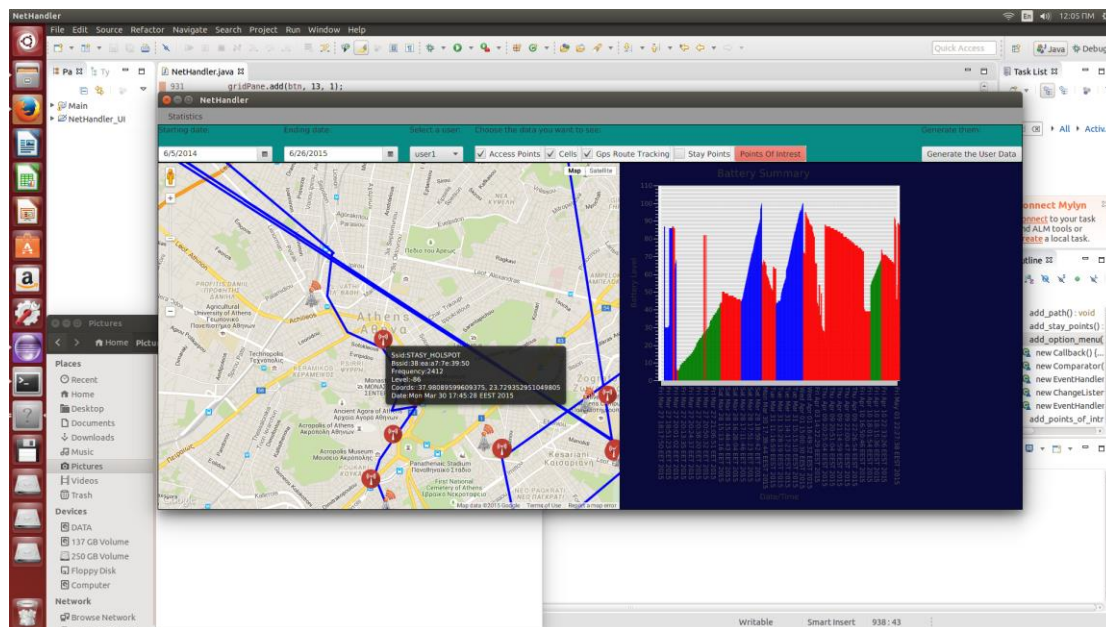
Για την δημιουργία του χάρτη χρησιμοποιήθηκε η βιβλιοθήκη για τους χάρτες της **google** για desktop app σε συνδυασμό με **java fx** . Η βιβλιοθήκη είναι η παρακάτω : <http://rterp.github.io/GMapsFX/> και τα javadocs της είναι αυτα : <http://rterp.github.io/GMapsFX/apidocs/>
Η βιβλιοθήκη είναι opensource και γίνεται import στο project από jar.

Χρησιμοποιώντας λοιπόν την Javafx και την παραπάνω βιβλιοθήκη παριστάνουμε τα δεδομένα στο χαρτη.Τα Access points και τα Base Stations γίνονται εισαγωγή με

μορφή marker αλλά με διαφορετικό εικονίδιο το κάθε ένα για να μπορούμε να τα ξεχωρίζουμε το οποίο επιλέγεται από το **property file**.Κανώντας hover(αιώρηση) πάνω από κάποιο **Access Point** ή **Base Station** στον χάρτη περνούμε πληροφορίες σχετικά με το **Access Point** ή το **BaseStation**.Όσο αναφορά το path που έχει διανύσει κάποιος χρήστης αυτό μπορεί να φανεί με ένα μπλέ polyline επάνω στο χάρτη.

Κάθε φορά που αλλάζουμε στοιχεία /επιλογές από option menu και πάμε να κάνουμε generate τα στοιχεία στο χάρτη ,ο χάρτης ξανά καλεί την συνάρτηση που τον ξανά αρχικοποιεί και γεμίζει τον χάρτη με τα στοιχεία που του έχουμε πει στις επιλογές μας.

Σημείωση: Ο χάρτης δεν μπορεί να λειτουργήσει αν δεν υπάρχει σύνδεση στο διαδίκτυο.Έτσι όταν δεν υπάρχει σύνδεση η εφαρμογή τερματίζει αυτόματα και εμφανίζεται κατάλληλο μήνυμα.



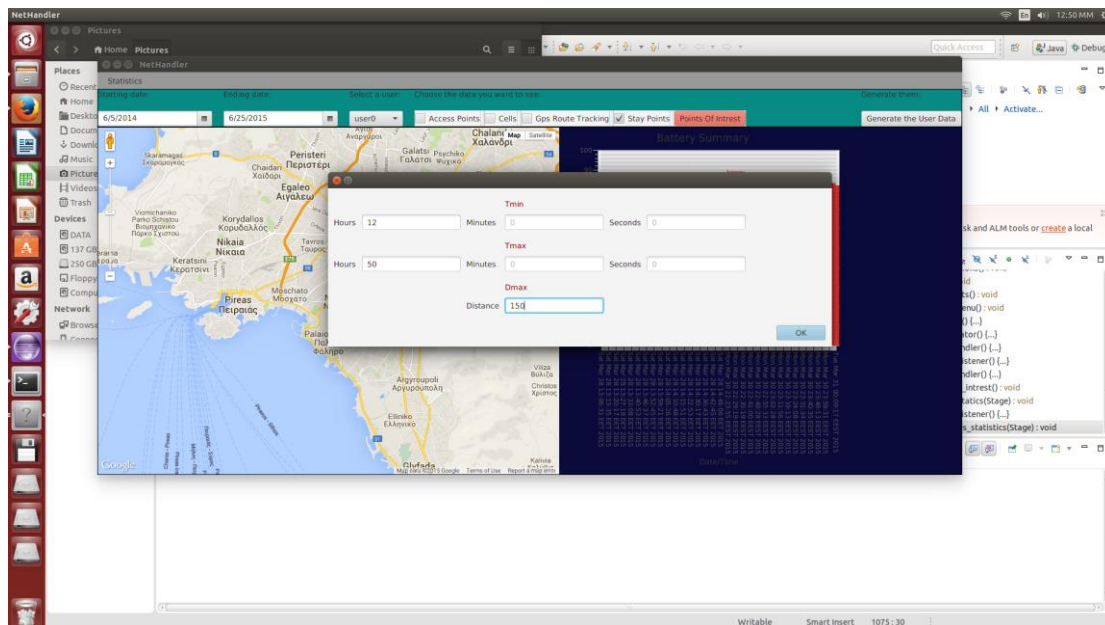
Σχετικά με την θέση των Access Points(ερώτημα 1)

Όσο αναφορά των υπολογισμό της ακριβής θέσης κάθε **Access Point** αυτό πραγματοποιείται στο πακέτο **ap_elaboration** στην κλάση **ApLocator**.Στην κλάση αυτή με την χρήση της συνάρτησης **averageLoc()** βρίσκουμε την ακριβή γεωγραφική τοποθεσία κάθε **Access Point** χρησιμοποιώντας των αλγόριθμο που μας δίνεται στο ερώτημα 1 βάση το **Bssid** κάθε **Access Point**.Τα αλλαγμένα δεδομένα στην συνέχεια αποθηκεύονται σε όλο το εύρος των δεδομένων της εφαρμογής όπως και μέσα στα ευρετήρια.

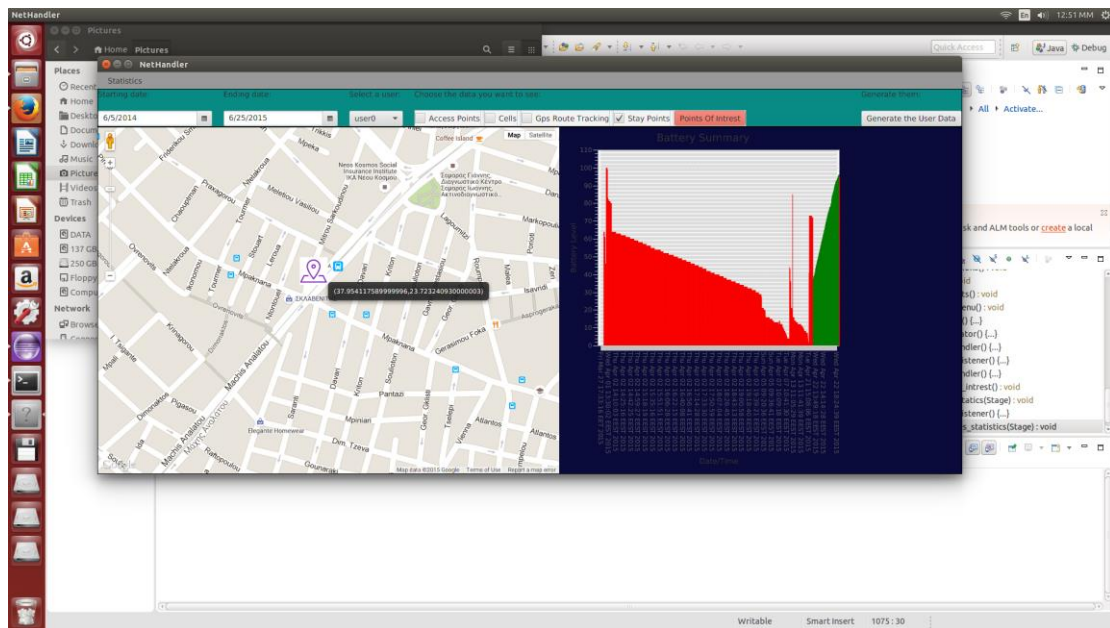
Μέρος 3^ο

Σε αυτό το κομμάτι της εργασίας ο αλγόριθμος για την εύρεση των stay points έχει υλοποιηθεί όπως ακριβώς προτείνεται στο e-class στην περιοχή συζητήσεων. Οι παράμετροι του αλγορίθμου δίνονται ως είσοδο από τον χρήστη όταν αυτός κλικάρει την επιλογή Stay Points. Τότε του εμφανίζεται μια μικρή φόρμα και πρέπει να δώσει τιμές για το Tmin, Tmax, Dmax σε hours/minutes/seconds, hours/minutes/seconds και μέτρα αντίστοιχα. Παρατίθεται ένα screenshot με την συμπλήρωση των παραμέτρων και ένα με την εμφάνιση των Stay Point του user που φαίνεται:

1. Εισαγωγή Παραμέτρων

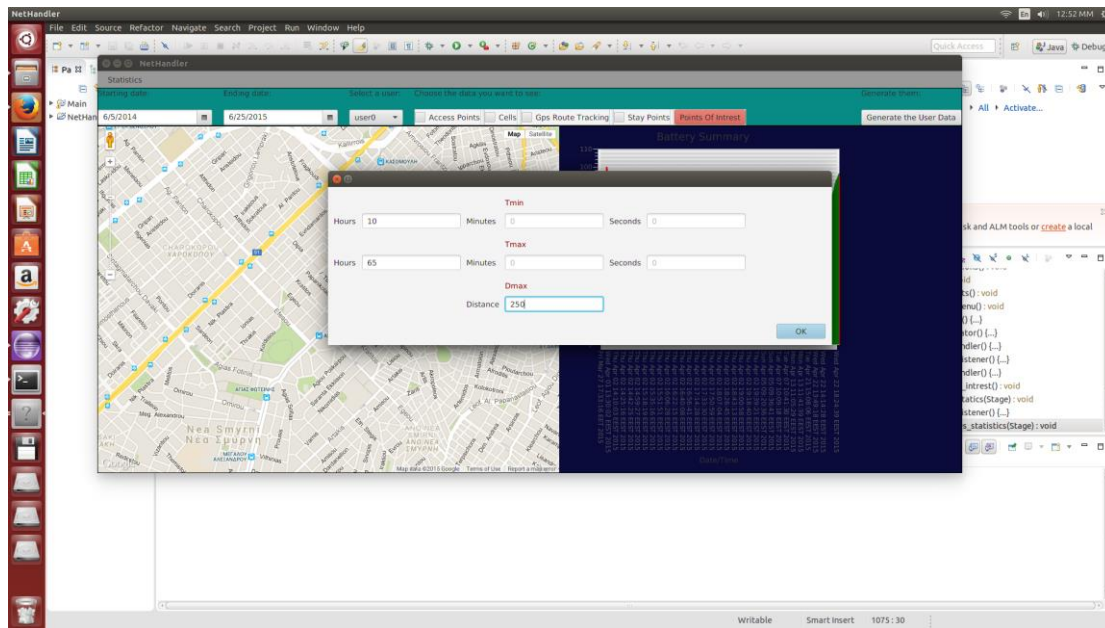


2. Εκτέλεση αλγορίθμου και αποτελέσματα

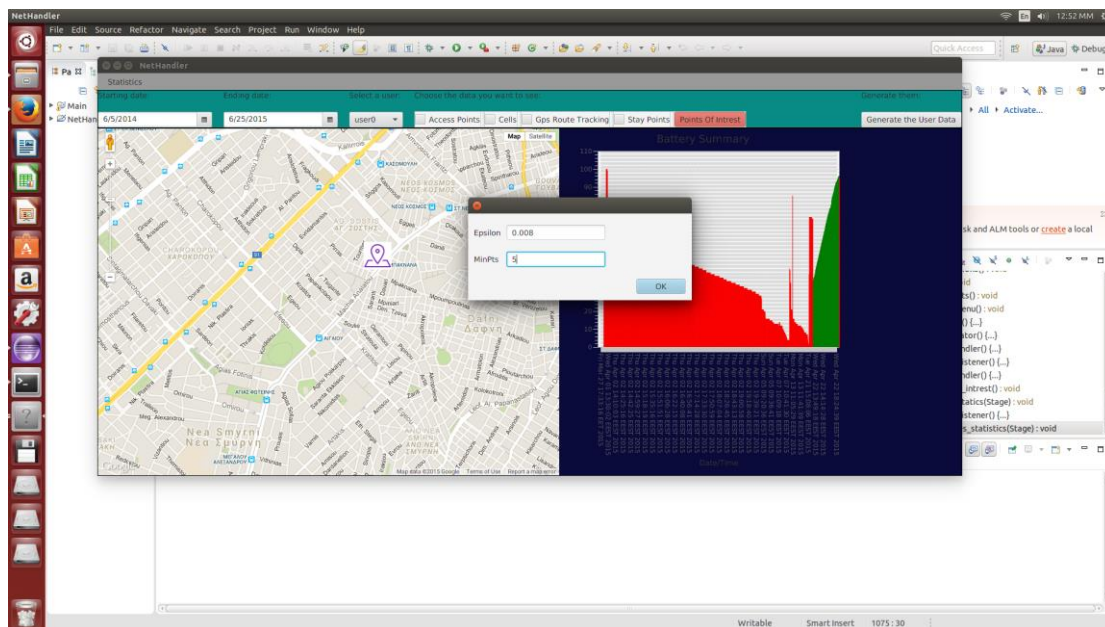


Για το δεύτερο ερώτημα που μας ζητήθηκε να χρησιμοποιήσουμε μια έτοιμη ή δικιά μας υλοποίηση του αλγορίθμου DBSCAN, βρήκα μία έτοιμη υλοποίηση στο net και την έχω ενσωματώσει σε ένα jar αρχείο στο project. Συνεπώς το μόνο που χρειάστηκε να κάνω είναι να ορίσω την κλάση που έχω για τα Stay Points ως generic, να κάνω implement μια Clusterable κλάση και να τρέξω τον αλγόριθμο. Ομοίως οι παράμετροι για τον αλγόριθμο δίνονται από τον χρήστη όπως και στο προηγούμενο ερώτημα αφού πρέπει να υπολογιστούν πρώτα τα Stay Points κάθε χρήστη, και τα 2 νέα ορίσματα που χρειάζεται ο DBSCAN δίνονται από τον χρήστη με παρόμοιο τρόπο. Το 1^ο είναι η μέγιστη απόσταση που μπορούν να έχουν 2 σημεία για να ανήκουν στο ίδιο cluster της οποίας μετρική είναι η ευκλείδεια και το 2^ο όρισμα είναι ο αριθμός των σημείων που χρειάζεται για να δημιουργηθεί ένα cluster. Τα cluster στον χάρτη αναπαρίστανται όπως τα ζητήσατε, δηλαδή σε μορφή παραλληλογράμμων όπου οι κορυφές τους είναι οι ελάχιστες και μέγιστες συντεταγμένες των σημείων που τα απαρτίζουν. Παρακάτω παρατίθενται screenshots με την εισαγωγή των παραμέτρων αλλά και την εκτέλεση του αλγορίθμου:

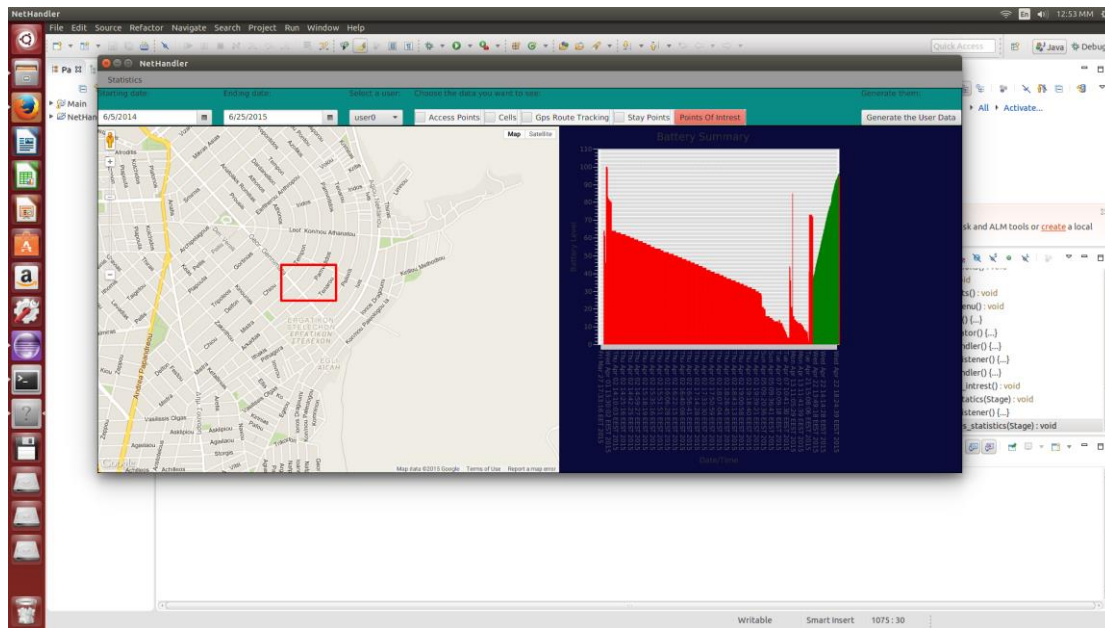
1. Εισαγωγή Παραμέτρων



2. Εισαγωγή Παραμέτρων DBSCAN

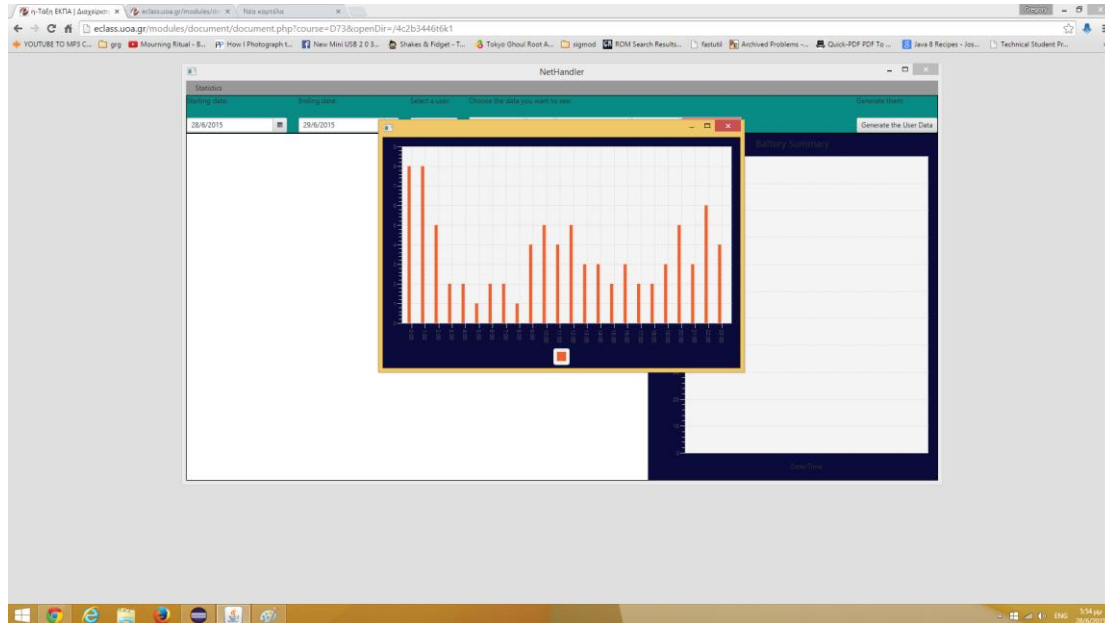


3. Εκτέλεση αλγορίθμου και αποτελέσματα



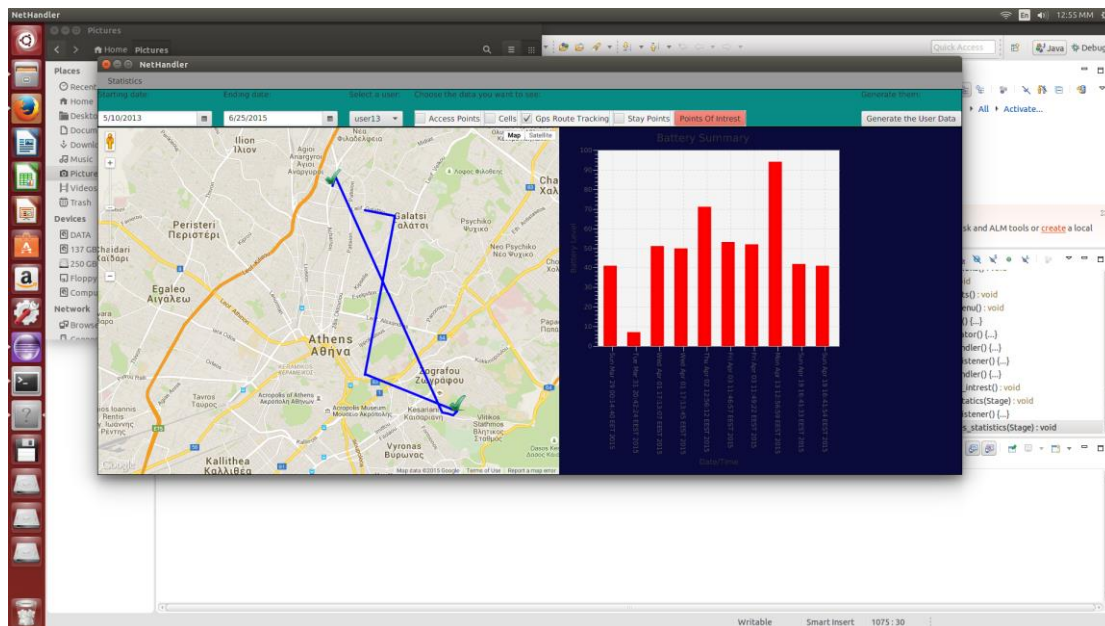
Μέρος 4^ο

Για τα διαγράμματα αυτού του μέρους έχω προσθέσει μία μπάρα που δίνει την δυνατότητα να επιλέξεις ποια από τα τρία διαγράμματα θες να δεις. Στο διάγραμμα του α υποερωτήματος έχω βάλει στον άξονα X όλες τις ώρες της ημέρας και στον άξονα Y τον αριθμό των χρηστών που βρέθηκαν με μπαταρία κάτω από 15%. Παρατίθεται screenshot:

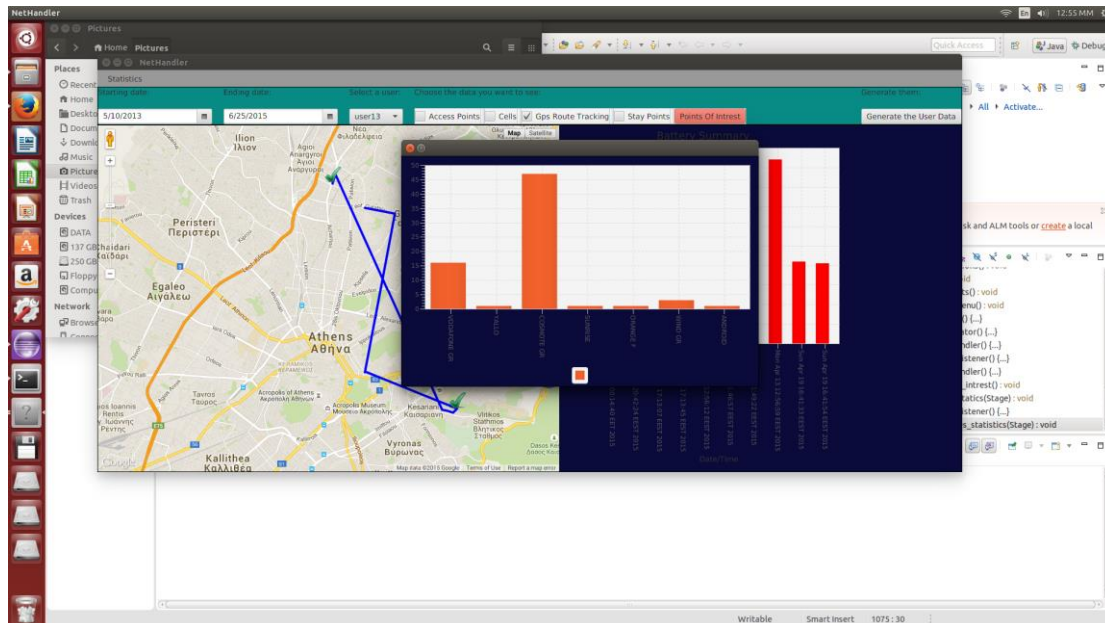


Να αναφέρω πως υπάρχει ένα ακόμη διάγραμμα που μπορείς να δεις το οποίο δείχνει το ίδιο πράγμα με αυτό το ερώτημα όμως διαλέγεις εσύ την ημέρα. Αυτό το είχα φτιάξει στην αρχή γιατί νόμιζα ότι αυτό ζητάει το α υποερώτημα και είπα να μην το σβήσω μιας και το έχω φτιάξει.

Στο επόμενο ερώτημα που ζητάει να βρούμε με χρονική σειρά τα access point που συμφέρει τον χρήστη να συνδεθεί για να κάνει εξοικονόμηση ενέργειας, πιστεύω ότι δεν μπορεί να υλοποιηθεί με 100% ακρίβεια γιατί υπάρχει πρόβλημα στα δεδομένα. Πιο συγκεκριμένα θα εξηγήσω πως το έχω υλοποιήσει και μετά θα αναφέρω τι προβλήματα αντιμετώπισα. Αυτό που κάνω είναι για κάθε θέση που έχουμε καταγεγραμμένη για τον εκάστοτε χρήστη, κοιτάω την ώρα της καταχώρησης που βρισκόταν σε αυτήν την θέση και μετά ψάχνω αν με απόκλιση 5 δευτερολέπτων το πολύ, ανιχνεύτηκε κάποιο access point εκείνη την χρονική στιγμή. Αν ναι, διαλέγω αυτό με το καλύτερο σήμα (δηλαδή RSSI όσο πιο κοντά στο 0 γίνεται). Εδώ εμφανίζεται το πρόβλημα ότι 2 access points τα οποία βρίσκονται αρκετά μακριά μεταξύ τους (συγκεκριμένο παράδειγμα στον user10 με απόσταση 750 μέτρα), έχουν καταγραφεί στην βάση δεδομένων ακριβώς την ίδια χρονική στιγμή (με 1 sec το πολύ απόκλιση). Συνεπώς πρόσθεσα άλλο ένα κριτήριο επιλογής του καλύτερου access point, ο χρήστης και το access point να μην απέχουν πάνω από 50 μέτρα αν και αυτό δεν είναι καθόλου αξιόπιστο γιατί ακόμα και 5 μέτρα να απέχουν που θεωρητικά θα πρέπει ο χρήστης να έχει εξαιρετικό σήμα μπορεί να κάτι να μπαίνει ανάμεσά τους και να παρεμβάλει. Τέλος ένα ακόμη πρόβλημα που αντιμετώπισα στον user1 αν θυμάμαι καλά, είναι ότι όπως παραπάνω τα timestamps δεν είναι πάντα σωστά, ενώ στο χάρτη φαίνεται ότι ο χρήστης πέρασε από κάπου και υπάρχει διαθέσιμο access point να συνδεθεί, στην πραγματικότητα λόγω του τελείως διαφορετικού timestamp της καταγραφής του access point και της καταγραφής της θέσης του χρήστη δεν θα επιλεγεί ποτέ. Να σημειώσω πως η σειρά με την οποία τα access point επιλέχθηκαν φαίνεται πηγαίνοντας το ποντίκι πάνω από το marker. Παρατίθεται screenshot:



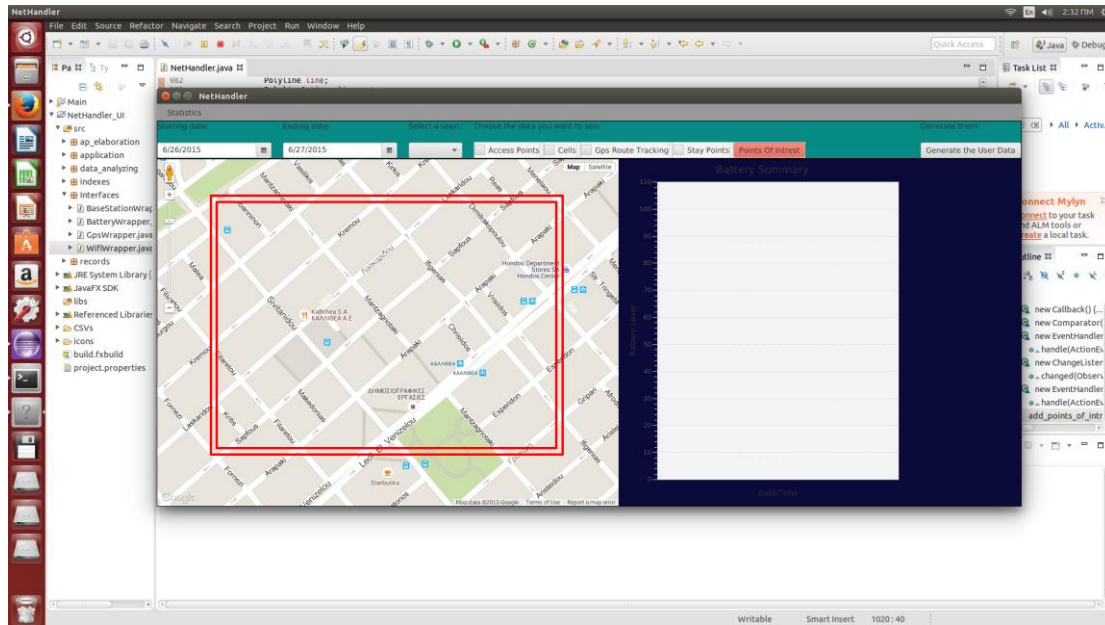
Στο 2^ο διάγραμμα αυτού του μέρους έχω βάλει στον άξονα X το όνομα κάθε εταιρείας και στον άξονα Y τον αριθμό των χρηστών που της αντιστοιχούν. Παρατίθεται screenshot:



Το τελευταίο ερώτημα αυτού του μέρους δεν το υλοποίησα σε κώδικα γιατί μέσα σε κάθε point of interest βρίσκονται πάρα πολλά logs από access points και αυτό δεν βοηθάει στο να φανεί μια καλή επιλογή των σημείων των 2 νέων access point. Συνεπώς θα εξηγήσω πως θα έκανα αυτή την επιλογή. Πρώτα απ' όλα η επιλογή αυτών των access point γίνεται βάση 2 παραγόντων:

1. Ύπαρξη νεκρού σημείου (δηλαδή χωρίς κανένα access point κοντά)
2. Έλεγχος των συχνοτήτων εκπομπής των γειτονικών access point

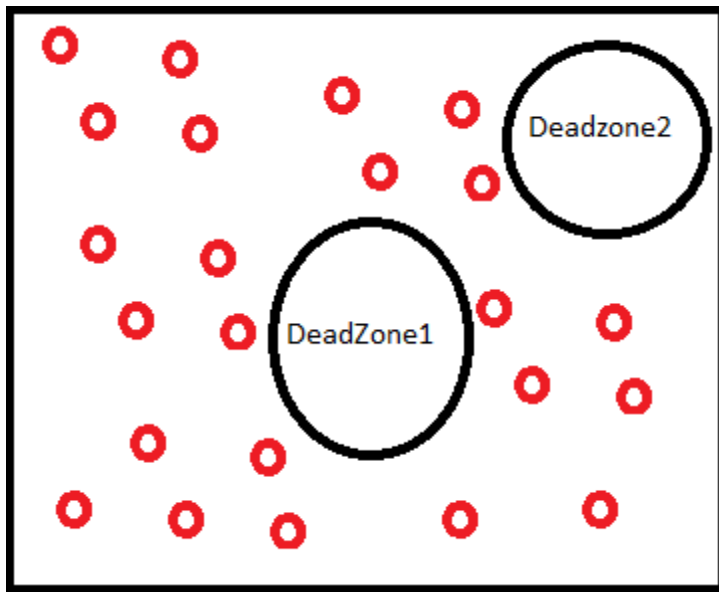
Πρώτα απ' όλα σκέφτηκα να μην ελέγξω μόνο τα access point που βρίσκονται αποκλειστικά μέσα στο point of interest αλλά να δώσω ένα περιθώριο 10 μέτρων ώστε να λάβω υπόψη και τα access point που μπορούν να συνδεθούν όσοι βρίσκονται εντός του παραλληλογράμμου ακόμα κι αν αυτά είναι έξω από αυτό. Σχηματικά αυτό θα φαινόταν κάπως έτσι:



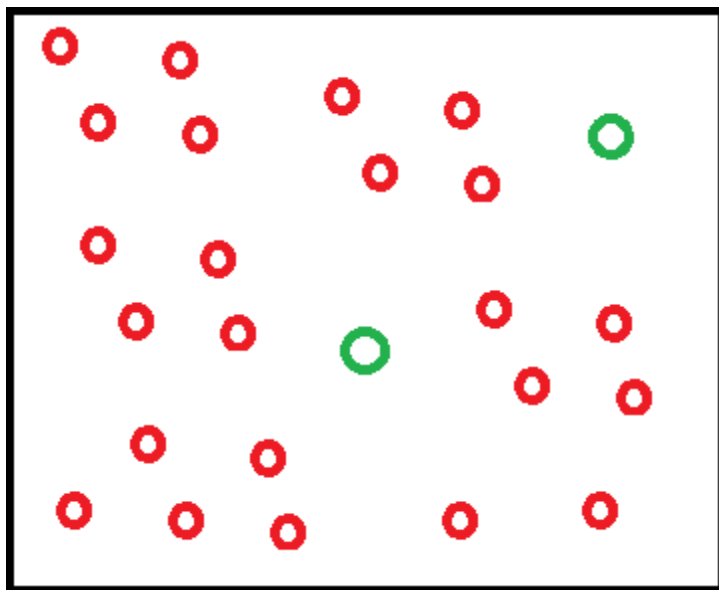
Έπειτα πρέπει να βρούμε για όλα αυτά τα access point που μαζεύτηκαν, που υπάρχουν τα μεγαλύτερα νεκρά σημεία που δεν μπορούν να συνδεθούν οι χρήστες που βρίσκονται μέσα σε αυτά. Έστω λοιπόν πως βρήκαμε 2 τέτοια νεκρά σημεία, το επόμενο βήμα είναι να δούμε σε πιο σημείο ακριβώς μέσα σ' αυτά θα προσθέσουμε το κάθε access point. Το ιδανικό θα ήταν να το βάλουμε ακριβώς στο κέντρο της κάθε περιοχής. Πρέπει όμως να λάβουμε υπόψη και τη συχνότητα που κάνουν broadcast τα γειτονικά access point έτσι ώστε να ελαχιστοποιήσουμε τις παρεμβολές μεταξύ τους και εν συνεχεία και το packet loss. Συνεπώς βάζοντας το access point απλά στη μέση της περιοχής αυτής, αν τύχει τα γύρω γύρω access points να χρησιμοποιούν όλο το εύρος των συχνοτήτων τότε καλύτερο θα ήταν να το μετακινήσουμε προς μια κατεύθυνση όπου θα ελευθερωνόντουσαν κάποιες συχνότητες. Αυτό θα είχε ως αποτέλεσμα κάποιοι χρήστες μέσα σε αυτή τη περιοχή να έχουν πιο αδύναμο σήμα αλλά δεν θα υπήρχε η παρεμβολή που ανέφερα παραπάνω. Με άλλα λόγια το αν αξίζει να γίνει αυτή η μετακίνηση του access point από το κέντρο της νεκρής περιοχής εξαρτάται από 2 παράγοντες, αν αξίζει να χάσει λίγο σήμα ένα κομμάτι αυτής της περιοχής για να γλιτώσουν κάποιες παρεμβολές.

Θα προσπαθήσω να εξηγήσω με κάποια απλά σχήματα αυτό που ανέφερα.

Έστω πως για ένα point of interest ισχύει κάτι τέτοιο:

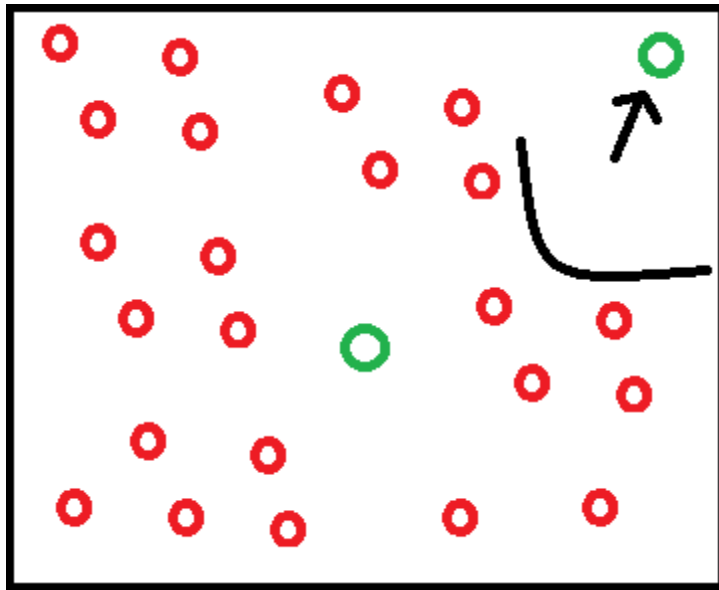


Προφανώς με κόκκινο είναι όλα τα access point. Προφανώς η ιδανική επιλογή για να βάλουμε τα 2 νέα access point θα ήταν η παρακάτω:



Δηλαδή στο κέντρο κάθε νεκρής περιοχής. Όμως υπάρχει η πιθανότητα όλα τα γειτονικά access point να έχουν καταλάβει όλες τις συχνότητες που είναι διαθέσιμες

στα 2.4 Ghz. Συνεπώς ίσως θα ήταν προτιμότερο να μετακινήσουμε λίγο τα νέα access point προς μια κατεύθυνση για να μειώσουμε την παρεμβολή:



Τώρα το νέο access point σίγουρα θα έχει μικρότερη εμβέλεια για τους χρήστες που βρίσκονται στα όρια της διαχωριστικής γραμμής αλλά δεν θα υπάρχει παρεμβολή μεταξύ των υπολοίπων access point. Βέβαια αυτό είναι ένα μόνο σενάριο αλλά αυτή είναι η λογική για την επιλογή του κατάλληλου σημείου. Τέλος να αναφέρω πως διάβασα το link που μας δώσατε και θα λάμβανα υπόψη ότι το κάθε access point πρέπει για να έχει μεταδίδει σε απόσταση 3 με 4 καναλιών όπου το κάθε κανάλι έχει εύρος 5 Mhz εκτός του 14^{ου} καναλιού που πρέπει να έχει απόσταση 12 Mhz.