Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Ανάπτυξη εφαρμογής εξόρυξη γεωγραφικών δεδομένων για την διείσδυση στη αγορά ενός νέου καταστήματος**
Ονοματεπώνυμο Φοιτητή	**Βασίλειος Κλουβάτος**
Πατρώνυμο	**Μιχαήλ**
Αριθμός Μητρώου	**ΜΠΠΛ**/ 08056
Επιβλέπων	**Δ**. **Δεσπότης**, **Καθηγητής**

Ημερομηνία Παράδοσης **Νοέμβριος** 2010

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)	(υπογραφή)	(υπογραφή)
Δ. Δεσπότης,	Δ. Αποστόλου,	Κ. Μεταξιώτης,
Καθηγητής	Επίκουρος Καθηγητής	Επίκουρος Καθηγητής.

Περιεχόμενα

Περίληψη Αγγλική

The purpose of this postgraduate paper is to combine a GIS technology with data mining techniques and visualize the results on a map. Using this application a user can decide where to build his new store depending on geography queries and data processing. For this reason the information system can be splitted in two main categories. The first one is the GIS application that is responsible for storing and managing geographical data and for the projection of data in the map. The second one is the Data Mining process that is responsible for the data processing depending on user's queries.

In the last five years a part of computer science that grows continuously is Geographical Information Systems. Big vendors like Google and Microsoft gave a depth research on GIS Systems. The conclusion of this big research is two applications that a user can have access from the internet, from Google maps or Bing Maps. The two big companies developed not only these services but also created building tools for developers' G.I.S. applications. This application is developed using Microsoft technology and Bing maps Api.

The second part of the information system is the data processing. The database contains a huge amount of data but the problem is how the system is going to know which of these data is important for data process and what kind of algorithm is going to be used. To solve this problem I built a wizard asking user question. This wizard has five steps, within these steps user is going to select data and in the end this data is going to send to the server. When the data is sent to the server the data mining process starts and the SimpleKMeans algorithm classifies the data.

The final and most important part of the application is the visualization of data. When the data process is finished the user must understand the result of data processing and make reports from this. It is very essential for the user aspect to see the results in the map. At the end the user is going to observe in the map the groups (clusters) of selected stores from the data mining process and which group belongs the new store.

Περίληψη Ελληνική

Ο σκοπός της πτυχιακής διατριβής είναι δημιουργία ολοκληρωμένου πληροφοριακού συστήματος για την εύρεση της βέλτιστης γεωγραφικής θέσης για την δημιουργία νέου καταστήματος. Ο χρήστης χρησιμοποιώντας την εφαρμογή μπορεί να αποφασίσει την γεωγραφική θέση που θα ανοίξει το νέο του κατάστημα. Η εφαρμογή χωρίζεται σε δύο βασικά μέρη. Το πρώτο μέρος είναι το γεωγραφικό σύστημα πληροφορίας (G.I.S.) το οποίο διαχειρίζεται την γεωγραφική πληροφορία του συστήματος. Στο δεύτερο μέρος χρησιμοποιούνται αλγόριθμοι εξόρυξης δεδομένων για την επεξεργασία πληροφορίας.

Τα τελευταία χρόνια έχει πραγματοποιηθεί ραγδαία ανάπτυξη στον τομέα της πληροφορικής σε συνδυασμό με γεωγραφικά συστήματα. Το αποτέλεσμα της συγκεκριμένης έρευνας είναι δύο από τις μεγαλύτερες εταιρίες πληροφορικής Microsoft® και Google να δημιουργήσουν υπηρεσίες που παρέχουν γεωγραφικές πληροφορίες μέσω του internet.Ένα ακόμη σημαντικό μέρος είναι οι βιβλιοθήκες που δημιουργήθηκαν για του προγραμματιστές ώστε να έχουν πρόσβαση σε τέτοιου είδους υπηρεσίες.

Τέλος το δεύτερο μέρος του πληροφοριακού συστήματος είναι η εξόρυξη δεδομένων και η αστικοποίηση των αποτελεσμάτων. Όταν η επεξεργασία δεδομένων έχει ολοκληρωθεί ο χρήστης πρέπει να δει το αποτέλεσμα της ανάλυσης και να παράγει αναφορές από αυτό. Είναι πάρα πολύ σημαντικό ο χρήστης να δει το αποτέλεσμα στον χάρτη. Στο τέλος ο χρήστης θα παρατηρήσει το αποτέλεσμα στον χάρτη, θα δει τις ομάδες που δημιουργήθηκαν και σε ποια ομάδα ανήκει η θέση του νέου καταστήματος που επιθυμεί να ανοίξει.

Εισαγωγή – Σύντομη Περιγραφή

Σε περίπτωση που ένας νέος επιχειρηματίας επιθυμεί να ανοίξει ένα κατάστημα έχει να λάβει μία μεγάλη απόφαση, την γεωγραφική του θέση. Το πληροφοριακό σύστημα που αναπτύχθηκε θέλει να δώσει λύση στο συγκεκριμένο πρόβλημα. Η ερώτηση αυτή χωρίζεται σε δύο μεγάλα μέρη.

Το πρώτο μέρος αφορά την ανάλυση και επεξεργασία της γεωγραφικής πληροφορίας. Η επιτυχία του νέου καταστήματος είναι άμεσα συνδεδεμένη με την γεωγραφική του θέση. Ο χρήστης επιθυμεί να έχει στην διάθεση του διάφορα εργαλεία με τα οποία θα βοηθηθεί στην ανάλυση και στην προβολή δεδομένων στον χάρτη.

Οι εταιρίες Google και Microsoft δίνουν στους προγραμματιστές χάρτες αλλά και τις βιβλιοθήκες για να τους χρησιμοποιήσουν. Μέσα από το πληροφοριακό σύστημα ο χρήστης έχει πρόσβαση σε χάρτη με δορυφορικές εικόνες ώστε να τον βοηθήσει στην καλλίτερη επιλογή σημείου. Μέσα από χάρτη προσφέρονται δύο ειδών πληροφορίες. Η πρώτη παρέχεται μέσω του provider (στο πληροφοριακό σύστημα που αναπτήχθηκε είναι η Microsoft) επάνω στον χάρτη, όπως για παράδειγμα τα ονόματα των οδών, βασικές συγκοινωνίες, πλατείες και άλλα. Το δεύτερο μέρος είναι η γεωγραφική πληροφορία που είναι συγκεντρωμένη στην Βάση δεδομένων του συστήματος. Ένα παράδειγμα είναι τα όρια των δήμων, η συγκεκριμένη πληροφορία δεν παρέχεται από τις εταιρίες Microsoft/Google αλλά είναι προσπελάσιμη μέσα από το σύστημα.

Χρησιμοποιώντας το πληροφοριακό σύστημα υπάρχει η δυνατότητα προσθήκης νέας γεωγραφικής οντότητας. Γεωγραφική οντότητα είναι ένα σημείο, μία γραμμή ένα πολύγωνο το οποίο έχει μία υπόσταση στον χάρτη. Ένα παράδειγμα χρήστη είναι η σχεδίαση ενός πολυγώνου (μέσα από τα εργαλεία του συστήματος) που θα αντιπροσωπεύει μία πλατεία. Ο χρήστη μπορεί να εισάγει ετικέτες για την συγκεκριμένη πλατεία π.χ.

Ετικέτα	Τιμή
Τύπος	Πλατεία
Σχήμα	Τετράγωνο

Το δεύτερο μέρος είναι η εξόρυξη δεδομένων. Για την δημιουργία του νέου καταστήματος δεν αρκεί μόνο η γεωγραφική πληροφορία. Ο χρήστης χρειάζεται να ξέρει για την συγκεκριμένη περιοχή τα δημογραφικά στοιχεία και να τα συγκρίνει με αυτά που αφορούν το κατάστημα του.

Στην βάση δεδομένων υπάρχουν καταστήματα που θεωρούνται επιτυχημένα (στο πληροφοριακό σύστημα επιτυχημένα καταστήματα θεωρούνται αυτά που είναι σε λειτουργία παραπάνω από 10 χρόνια). Τα καταστήματα χωρίζονται σε κατηγορίες μέσα από την εφαρμογή. Ο χρήστης επιλέγει μία από τις διαθέσιμες κατηγορίες που αντιστοιχεί το κατάστημα το οποίο επιθυμεί να ανοίξει. Στο επόμενο βήμα ο χρήστης επιλέγει τα δημογραφικά χαρακτηριστικά που είναι σημαντικά για το κατάστημα του. Στο πληροφοριακό σύστημα έχει αναπτυχθεί ένας αλγόριθμος ο οποίος συλλέγει όλη την πληροφορία που χρειάζεται για να εκπαιδεύσει τον αλγόριθμο εξόρυξης δεδομένων ώστε να δημιουργηθούν οι ομάδες των καταστημάτων.

Όταν ολοκληρωθεί η εξόρυξη δεδομένων ο χρήστης μπορεί να δει το αποτέλεσμα στον χάρτη. Θα εμφανιστούν οι ομάδες των επιτυχημένων καταστημάτων. Τέλος ο χρήστη μπορεί να δει σε πια ομάδα ανήκει το νέο κατάστημα που επέλεξε.

Στις επόμενες ενότητες που ακολουθούν θα πραγματοποιηθεί η ανάλυση του πληροφοριακού συστήματος. Αναλυτικότερα :

- Η ανάλυση της αρχιτεκτονικής του πληροφοριακού συστήματος.
  - Presentation tier
  - Business tier
  - Data Access tier
- Στην επόμενη ενότητα αναλύθηκε η σχεδίαση της βάσης δεδομένων του πληροφοριακού συστήματος.
- Στην ενότητα «Business tier» αναλύεται όλη η λογική του πληροφοριακού συστήματος στο σύνολο.
- Στην ενότητα «presentation tier» πραγματοποιείτε σε βάθος η ανάλυση των τεχνολογιών που χρησιμοποιήθηκαν για την δημιουργία των διεπαφών και την αλληλεπίδραση με τον χρήστη.
- Η επόμενη ενότητα αφορά την εκτενή ανάλυση των υποσυστημάτων του πληροφοριακού συστήματος. Αναλύει σε βάθος τις λειτουργίες των υποσυστημάτων αλλά και ο αλγόριθμος εξόρυξη δεδομένων.
- Οι δύο τελευταίες ενότητες αφορούν τα γεωγραφικά και δημογραφικά δεδομένα,
- Τέλος υπάρχει και μία ενότητα για την χρήση του Weka αλλά και του αλγόριθμου SimpleKMeans.

Αρχιτεκτονική εφαρμογής

Multitier architecture

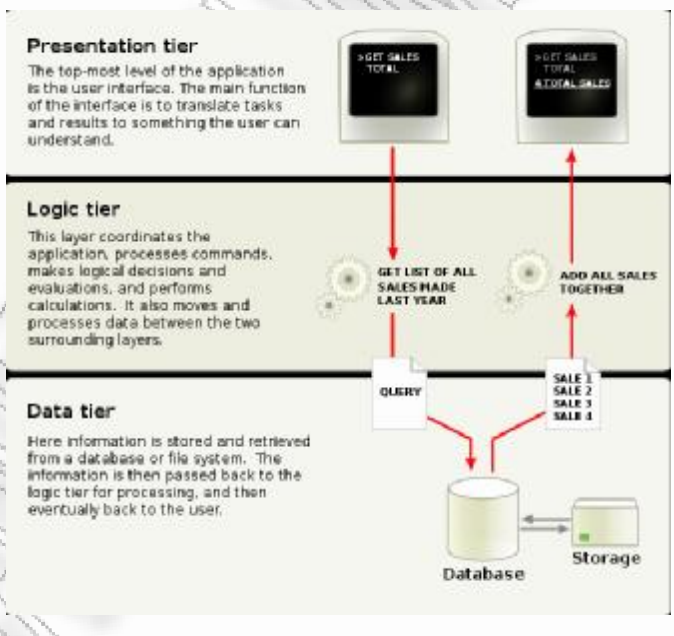
Η εφαρμογή θα αναπτυχθεί σε αρχιτεκτονική client – Server. Η αρχιτεκτονική αυτών των τύπου εφαρμογών ονομάζεται Multi tier architecture, η οποία περιλαμβάνει τα ακόλουθα επίπεδα.

- Presentation tier
- Data access tier
- Data tier

Υπάρχουν τρία επίπεδα και είναι γνωστά με τον όνομα three – tier architecture. Το πρώτο επίπεδο (Presentation tier) αφορά κυρίως το γραφικό περιβάλλον του χρήστη (User Interface), δηλαδή τον τρόπο που αλληλεπιδρά ο χρήστης με την εφαρμογή.

Το δεύτερο επίπεδο μεσολαβεί ανάμεσα στο γραφικό περιβάλλον και την βάση δεδομένων. Υποστηρίζει συναρτήσεις με τις οποίες το γραφικό περιβάλλον έχει πρόσβαση στα δεδομένα, βασικές είναι το διάβασμα, γράψιμο, τροποποίηση και διαγραφή δεδομένων.

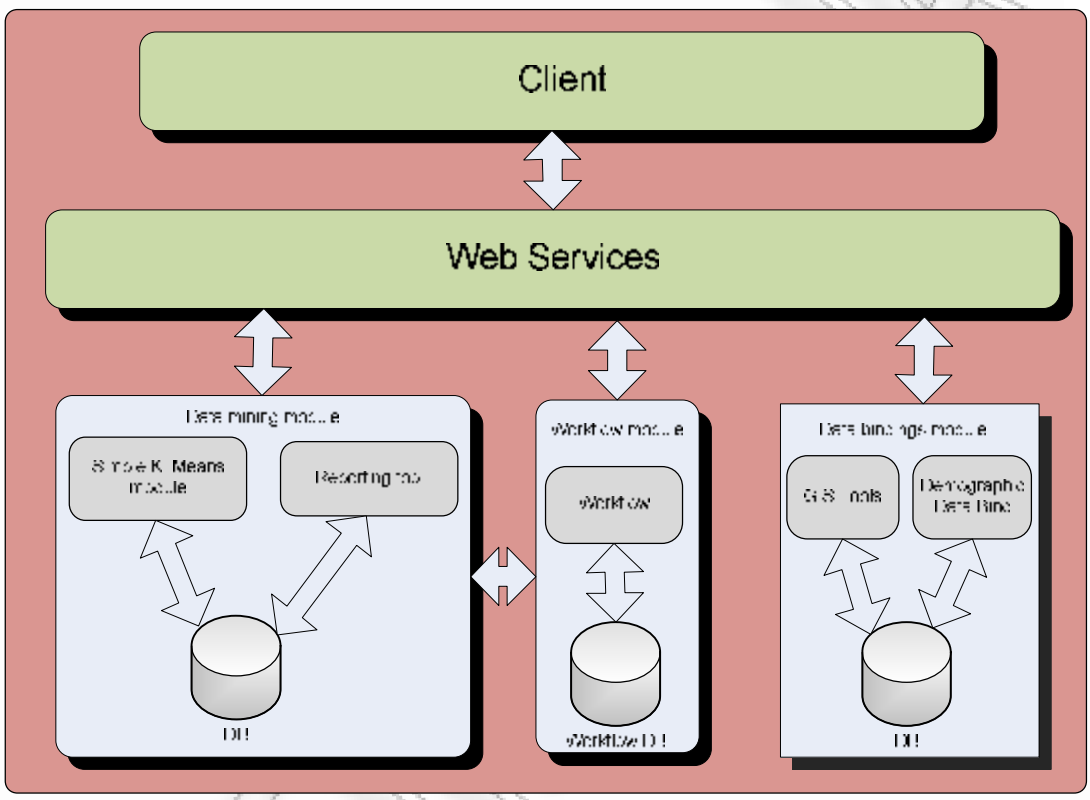
Τέλος το τρίτο επίπεδο υπάρχει η βάση δεδομένων, οι πίνακες, και τις σχέσεις μεταξύ πινάκων ώστε να αποθηκεύονται και να διαβάζονται τα δεδομένα. Στο επόμενο σχήμα περιγράφεται η συγκεκριμένη αρχιτεκτονική.



Εικόνα 1 three tier architecture

Αρχιτεκτονική

Στην αρχιτεκτονική του πληροφοριακού συστήματος θα περιγραφτεί η βασική δομή του. Βασική δομή είναι τα κομμάτια λογισμικού που θα συνθέτουν το σύστημα και οι σχέσεις που αναπτύσσουν μεταξύ τους (εικόνα 2). Στο επόμενο σχήμα πραγματοποιείται η ανάλυση της αρχιτεκτονικής.



Εικόνα 2 αρχιτεκτονική

Από το σχήμα διακρίνονται τρία βασικά module το Data mining module, Workflow module, Data bindings module.

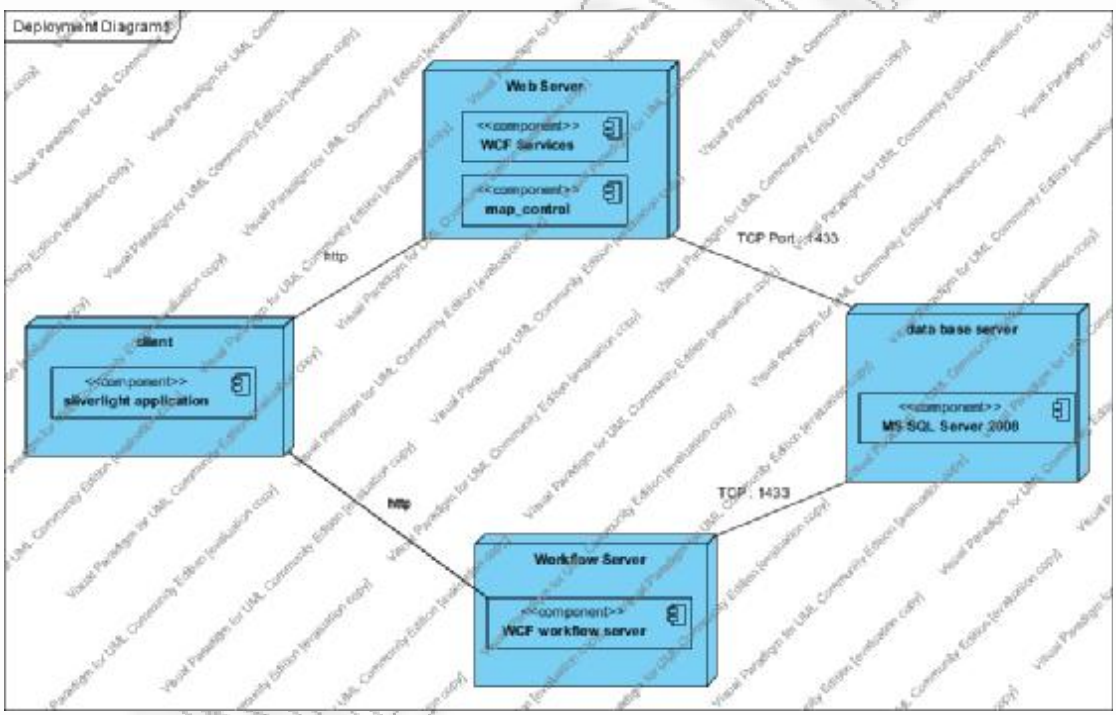
Το module data mining χωρίζεται σε δύο βασικά μέρη. Το πρώτο ονομάζεται Simple K-Means module στο οποίο εκτελείτε ο αλγόριθμος SimpleKMeans. Με την χρήση του συγκεκριμένου αλγόριθμού μπορούμε να ομαδοποιήσουμε τα δεδομένα σε ένα πολυδιάστατο χώρο. Οι διαστάσεις αυτές είναι ιδιότητες των στιγμιότυπων που έχουμε χρησιμοποιήσει για την εκπαίδευση του. Θα πραγματοποιηθεί εκτενέστερη ανάλυση όσο αφορά την λειτουργία του αλλά και πώς χρησιμοποιήθηκε μέσα από την εφαρμογή. Το δεύτερο είναι το reporting tool στο οποίο δημιουργούνται αναφορές μέσα από ερωτήματα στην Β.Δ., παράδειγμα μίας αναφοράς είναι η ανάλυση ομάδων (clusters) μέσα από τον αλγόριθμο Simple K Means.

Το επόμενο module είναι το Workflow. Η βασική του λειτουργία του είναι η αυτοματοποίηση ροών εργασίας που υπάρχουν μέσα στην εφαρμογή. Ένας δεύτερος σκοπός είναι η εκτέλεση εργασιών που χρειάζονται πολύ χρόνο για να εκτελεστούν όταν καλούνται μέσα από Web Services.

Το τρίτο module είναι Data Binding module το οποίο χρησιμοποιείται για την πρόσβαση δεδομένων στο Σύστημα Βάσης Δεδομένω.

Ένα βασικό μέρος της εφαρμογής είναι το γραφικό περιβάλλων (presentation layer) του χρήστη. Με αυτό αλληλεπιδρά ο χρήστης με τις λειτουργίες του πληροφοριακού συστήματος. Στην εικόνα 2 υπάρχει το module με όνομα «Client» το οποίο αντιπροσωπεύει το presentation layer. Η σχεδίαση του όσο και η υλοποίηση του εξαρτάτε αν θα είναι Web Based οπότε θα εξαρτάται από τον browser ή αν θα είναι παραθυρική εφαρμογή.

Τα modules που δημιουργήθηκαν στο πληροφοριακό σύστημα πρέπει να επικοινωνούν με το γραφικό περιβάλλων του χρήστη και να υπάρχει η δυνατότητα αλληλεπίδρασης. Για να επιτευχθεί η συγκεκριμένη επικοινωνία δημιουργήθηκε ένα επίπεδο ανάμεσα στο γραφικό περιβάλλων και τις λειτουργίες το οποίο ονομάζεται Web Services. Ο ρόλος του συγκεκριμένου module είναι σαν τον proxy server, δέχεται αιτήματα και τα μεταφέρει στο επόμενο επίπεδο της εφαρμογής, το Data Access Layer. Όλα τα modules θα αναλυθούν στις επόμενες ενότητες εκτενέστερα. Στην επόμενη εικόνα σχεδιάστηκε ένα Deploy διάγραμμα που περιγράφει την αρχιτεκτονική.



Εικόνα 3 Deploy diagram

Presentation tier

Το επίπεδο presentation layer περιέχει τα εργαλεία τα οποία χρησιμοποιούνται για την δημιουργία γραφικών και διαχειρίζεται την αλληλεπίδραση με τον χρήστη. Τα βασικά στοιχεία που περιλαμβάνει είναι :

1)User Interface (UI) Components . Είναι τα στοιχεία που χρησιμοποιούνται για την σχεδίαση της διεπαφής. Μέσω αυτών των στοιχείων μπορεί ο χρήστης να αλληλεπιδρά

με την εφαρμογή. Ακόμα αναλαμβάνουν να παρουσιάζουν τα δεδομένα στην σωστή μορφή και τέλος μπορούν αν κάνουν επικύρωση των δεδομένων.

2)User Process component. Τα εργαλεία αυτά χρησιμοποιούνται όταν υπάρχουν σύνθετες διεπαφές και βοηθούν στην σωστή καθοδήγηση του χρήστη.

Για την δημιουργία του presentation layer επιλέχθηκε η τεχνολογία της Microsoft το Silverlight. Βασίζεται στην RIA αρχιτεκτονική και λειτουργεί όπως μία Web Based εφαρμογή. Στην επόμενη ενότητα θα γίνει μία ανάλυση της συγκεκριμένης αρχιτεκτονικής.

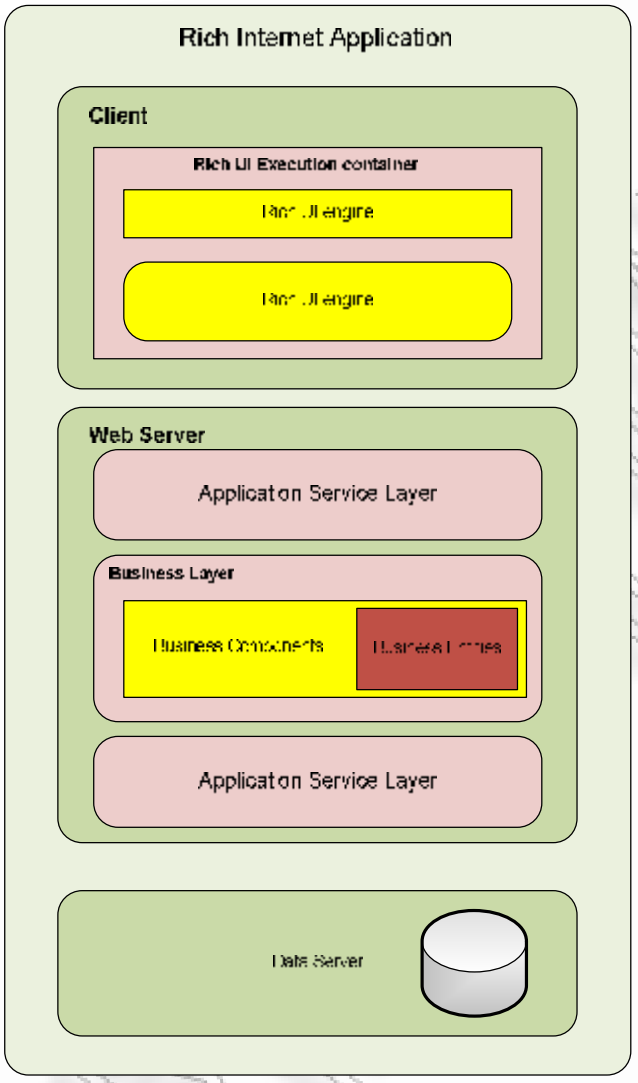
RIA (Reach Internet Application)

Στις Web based εφαρμογές ο χρήστης έχει πρόσβαση μέσω του browser στην εφαρμογή. Το βασικό χαρακτηριστικό αυτών τον εφαρμογών είναι η προσπέλαση τους μέσω του δικτύου όπως το intranet και το internet. Το πρωτόκολλο που χρησιμοποιούν για την επικοινωνία με το Data Access Layer είναι το http

Οι διεπαφές που σχεδιάζονται για web based εφαρμογές είναι πλήρως εξαρτώμενες από τον εκάστοτε browser. Για την σχεδίαση τους χρησιμοποιείται η HTML γλώσσα μαζί με την γλώσσα CSS για την μορφοποίηση του περιεχομένου . Οι αλληλεπίδραση χρήστη με την εφαρμογή χρησιμοποιώντας τις παραπάνω τεχνολογίες έχει σαν αποτέλεσμα να περιορίσει τις διεπαρφές και τις λειτουργίες σε σχέση με μία Desktop εφαρμογή. Σε μία web εφαρμογή περιορίζονται και οι λειτουργίες που μπορεί να εκτελέσει ο χρήστης σε σχέση με τις Desktop εφαρμογές.

Υλοποιήθηκαν Frameworks τα οποία βασίζονται στην τεχνολογία του Web αλλά έχουν πάρα πολλά χαρακτηριστικά Desktop εφαρμογών. Αυτή η τεχνολογία ονομάστηκε RIA (Reach Internet Application). Τα πιο γνωστά Frameworks που χρησιμοποιούνται είναι της εταιρίας Adobe το flash, της εταιρίας oracle η Java και της εταιρία Microsoft το Silverlight.

Για την αρχικοποίηση της εφαρμογής χρησιμοποιείται το λειτουργικό σύστημα του χρήστη. Αυτή είναι η βασική διαφορά με τις εφαρμογές που βασίζονται στην γλώσσα Javascript οι οποίες βασίζονται στο εκάστοτε browser. Η αρχιτεκτονική της συγκεκριμένης τεχνολογίας εμφανίζεται στην επόμενη εικόνα.



Εικόνα 4 RIA Service Αρχιτεκτονική

Στην εικόνα 4 υπάρχει η αναπαράσταση της αρχιτεκτονικής που υποστηρίζει μία RIA εφαρμογή. Η εφαρμογή εκτελείται στον υπολογιστή του χρήστη και αυτό έχει σαν αποτέλεσμα το Validation των δεδομένων αλλά και την χρήση βιβλιοθηκών από τον Η.Υ. του χρήστη αλλά και από τον Server. Ένα δεύτερο πλεονέκτημα είναι η ασύγχρονη επικοινωνία ανάμεσα στον Client και στον Server.

Silverlight – Web Service Architecture

Ο χρήστης μέσω του presentation layer μπορεί να αλληλεπιδρά με την εφαρμογή. Ο τρόπος με τον οποίο πραγματοποιείται η επικοινωνία με το Data Access layer είναι μέσω Web Services.

Το Web Service είναι μία Web βιβλιοθήκη (Application Programming Interface) και είναι προσπελάσιμη μέσω του HTTP πρωτοκόλλου. Είναι ένας τρόπος επικοινωνίας από έναν Server σε έναν άλλο. Η διεπαφή του δημιουργείται αυτόματα και έχει την μορφή ενός WSDL αρχείου. Ένας τρόπος ακόμα με τον οποίο περιγράφεται η συγκεκριμένη τεχνολογία ονομάζεται SOAP. Υπάρχουν τρείς τρόποι για την κλήση ενός Web Service.

- RPC
- SOAP
- REST

Η μέθοδος RPC είναι ένας τρόπος να καλέσει ένα πρόγραμμα μία συνάρτηση / μέθοδο που βρίσκεται κάπου απομακρυσμένα. Το πρόβλημα που δημιουργείται είναι ότι οι providers που δημιουργούν τα Web Service δεσμεύονται και με την δική τους βιβλιοθήκη. Παράδειγμα αν δημιουργηθεί ένα Web Service σε επιβάλλον Microsoftκαι η μέθοδος που εκτελείται επιστρέφει ένα αντικείμενο από το .NET Framework δεν είναι εύκολο να δημιουργηθεί ένας Client σε Java και να διαβάσει το αποτέλεσμα της συνάρτησης.

Η μέθοδος SOP έχει μία σημαντική διαφορά σε σχέση με την PRC κλήση. Στην SOP μέθοδο δεν υπάρχει επικοινωνία μέσω μίας συνάρτησης αλλά μέσω μηνύματος. Αυτή η τεχνική ονομάζεται message-oriented. Χρησιμοποιώντας την συγκεκριμένη τεχνική αντιμετωπίζεται το πρόβλημα με τους διαφορετικούς providers.

Τέλος υπάρχει και η μέθοδος REST. Η συγκεκριμένη αρχιτεκτονική χρησιμοποιεί το πρωτόκολλο HTTP 1.1 και συνήθως χρησιμοποιείται για την μεταφορά hyper media δεδομένα. Τa hyper media δεδομένα το video, ήχος κλπ.

Όπως έγινε και αναφορά σε προηγούμενη ενότητα η τεχνολογία Silverlight επικοινωνεί μέσω Web Service για την μεταφορά δεδομένων. Για να πραγματοποιηθεί η επικοινωνία πρέπει να δηλωθούν κάποιοι headers. Ένα παράδειγμα είναι η δημιουργία μίας συνάρτησης για την εκτέλεση ενός Report. Στην δήλωση της κλάσης πρέπει να υπάρχουν οι αντίστοιχες δηλώσεις

```
[ServiceContract(Namespace = "")]
[AspNetCompatibilityRequirements(RequirementsMode =
AspNetCompatibilityRequirementsMode.Allowed)]
```

Για την δημιουργία της συνάρτηση ς που θα πρέπει να γραφτεί ο ακόλουθος κώδικας

```
[OperationContract]
public string view_report_user(int userId)
{
}
```

Τέλος ο κώδικας που θα πρέπει να γραφτεί στον client για να διαβάσει το Web Service είναι το ακόλουθο.

```
service_create_report.ServiceClient client = new
service_create_report.ServiceClient();
client.createReportCompleted += new
EventHandler<service_create_report.createReportCompletedEventArgs>(client_creat
eReportCompleted);
```

Η επικοινωνία που δημιουργείται ανάμεσα στον Client και στον Server είναι ασύγχρονη όποτε υπάρχει μία συνάρτηση που όταν επιστρέψει ο server τα δεδομένα τότε θα ενεργοποιηθεί. Στο παράδειγμα που περιγράφεται είναι η συνάρτηση *client_createeReportCompleted*

```
void client_createReportCompleted(object sender,
service_create_report.createReportCompletedEventArgs e)
```

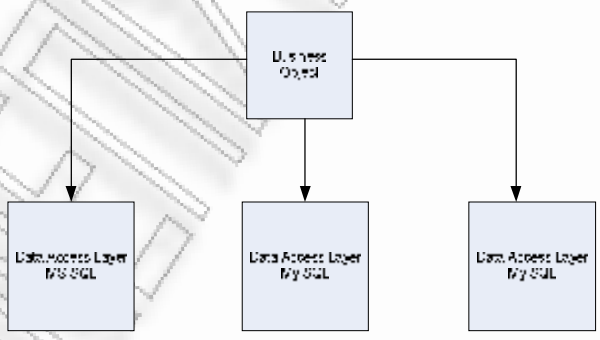
```
        {
            try
            {
            }
            catch (Exception ex)
            {

            }
        }
```

Business tier

Αυτό το επίπεδο χωρίζεται σε δύο υπό-επίπεδα. Στο πρώτο σχεδιάζονται και υλοποιούνται όλες οι κλάσεις που διατηρούν το business logic του πληροφοριακού συστήματος. Το δεύτερο υπό-επίπεδο υλοποιείται η επικοινωνία με βάση δεδομένων για να εκτελεστούν τα ερωτήματα στο Σ.Β.Δ. Ένα Business object είναι μία κλάση που διατηρεί δεδομένα και εκτελεί λειτουργίες για ένα συγκείμενο business κομμάτι λογικής. Όμως δεν μπορεί να αντιμετωπιστεί σαν ένας χώρος που μπορεί να αποθηκεύσει μόνιμα δεδομένα. Η βασική του λειτουργία είναι η αλληλεπίδραση με το σύστημα βάσης δεδομένων και η ανάκτηση δεδομένων στις κλάσεις. Ένα από τα πλεονεκτήματα που έχει η συγκεκριμένη τεχνική είναι ότι αν γίνει αλλαγή στον provider της βάσης δεδομένων αλλάζοντας μόνο το Data Access Layer μπορούν να επικοινωνήσουν οι κλάσεις με την καινούρια βάση δεδομένων.

Μία τεχνική που μπορεί να χρησιμοποιηθεί είναι ο διαχωρισμός του Business Login και του Data Access. Με αυτό τον τρόπο υπάρχει η δυνατότητα για παράδειγμα τριών Data Access Layer ένα για πρόσβαση στο MS Sql Server, ένα δεύτερο στην MySql και ένα τρίτο στην Oracle.



Εικόνα 5 Business Layer - Data Access Layer

Μια βασική αρχή που πρέπει να χρησιμοποιηθεί για την σωστή υλοποίηση του συγκεκριμένου επιπέδου είναι ότι στο Business Object δεν πρέπει να δημιουργούνται ερωτήματα στην βάση δεδομένων. Την συγκεκριμένη λειτουργία πρέπει να υποστηρίζεται από το Data Access Layer. Έχοντας δημιουργήσει για παράδειγμα ένα Data Access Layer για των MSSql Server μπορεί με εύκολο τρόπο να δημιουργηθεί ένα Data Access Layer για των MySql Server το οποίο θα εκτελεί τις ίδιες συναρτήσεις. Αυτό έχει σαν αποτέλεσμα να μην επηρεαστεί το Business logic και να μην χρειαστεί να σχεδιαστεί – δημιουργηθεί από την αρχή.

Η τεχνολογία που επιλέχθηκε για την δημιουργία του Business logic και του Data Access είναι της Microsoft το .NET Framework 4.0 Με βάση το Framework .NET 4.0 δημιουργήθηκαν όλες οι κλάσεις για το Business logic αλλά και για το DAL.

Data tier

Στο Data tier υπάρχει το Σύστημα Βάσης Δεδομένων. Έχουν σχεδιαστεί και αναπτυχθεί πίνακες που διατηρούν τα δεδομένα αλλά και σχέσεις ανάμεσα σε αυτούς.

Το Σ.Β.Δ. θα πρέπει να διαχειρίζεται και να αναλύει τα δεδομένα, αυτό μπορεί να επιτευχθεί μόνο με μία Βάση Δεδομένων που υποστηρίζει σχέσεις (Relation Database Model). Για την σωστή λειτουργία του Πληροφοριακού συστήματος θα πρέπει να ληφθεί και μία ακόμα σημαντική παράμετρος που θα πρέπει να υποστηρίζει το Σ.Β.Δ. να διαχειρίζεται πληροφορία που είναι γεωγραφική ή γεωμετρική.

Συγκρίνοντας ΣΒΔ για την διαχείριση γεωγραφικών δεδομένων

Για την σχεδίαση και υλοποίηση του συστήματος θα πρέπει να γίνει επιλογή της βάσης δεδομένων. Η επιλογή της θα γίνει βάση τις δυνατότητες που θα προσφέρει για την προσθήκη – τροποποίηση – διαγραφή και εύρεση γεωγραφικής πληροφορίας.

Στο εμπόριο υπάρχουν τρείς κατασκευαστές βάσεων δεδομένων οι οποίοι υποστηρίζουν τέτοιες λειτουργίες. Η Microsoft η PostgreSQL και η Mysql είναι οι πιο συνηθισμένες που χρησιμοποιούνται για την υλοποίηση τέτοιων συστημάτων.

Για την επιλογή του ΣΒΔ που θα υποστηρίξει το πληροφοριακό σύστημα δημιουργήθηκε ένα πίνακας στον οποίο υπάρχει μία βασική σύγκριση μεταξύ των τριών κατασκευαστών.

Χαρακτηριστικό	SQL Server 2008 (RC0)	MySQL 5.1/6	PostgreSQL 8.3/PostGIS 1.3/1.4
Λειτουργικό σύστημα	Windows XP, Windows Vista, Windows 2003, Windows 2008	Windows XP, Windows Vista, (haven't tested on 2008), Linux, Unix, Mac	Windows XP, Windows Vista, (haven't tested on 2008), Linux, Unix, Mac
Άδεια χρήσης	Υπάρχει άδεια χρήσης εμπορική. Η έκδοση ονομάζεται Express edition έχει αρκετές δυνατότητες αλλά έχει δύο περιορισμούς ο πρώτος είναι ότι μπορεί να χρησιμοποιήσει έναν επεξεργαστή και η δεύτερη είναι ότι χρησιμοποιεί 1Gb μνήμη.	Υπάρχει άδεια χρήσης η οποία είναι εμπορική (GPL) κάποια κομμάτια του κώδικα είναι ανοικτά βάση της άδειας (GPL)	Υπάρχει άδεια χρήσης η οποία μπορεί να χρησιμοποιηθεί με βάση την GPL. Αν υπάρχει κάποια αλλαγή στον κώδικα πρέπει να δοθεί στην κοινότητα.
Free GIS Data Loaders	Υπάρχει ένα εργαλείο (shp	OGR2OGR, shp2mysql.pl script	Περιλαμβάνονται τα shp2pgsql,

	dataloader) που έχει ανατηχθεί από τον Morten Nielsen για τον SQL server 2008 και είναι στην κατάσταση RC0		OGR2OGR, QuantumGIS SPIT, SHP loader για PostGIS
Commercial GIS Data Loaders	Υπάρχουν τα ακόλουθα λογισμικά Manifold, Safe FME Objects, ESRI ArcGIS 9.3 (με το τελευταίο πακέτο)	Safe FME Objects	SharpMap.Net, JDBC postgis.jar περιλαμβάνει το postgis, JTS etc. tons for Java, GDAL C++, AutoCad
Εργαλεία που διατίθενται.	Δεν υπάρχουν ακόμα κάποια πακέτα λογισμικού επίσημα. Σε διαδικασία ανάπτυξης είναι το SharpMap.NET eventually και υπάρχει σε έκδοση ADO.NET 3.5+	GDAL C++, SharpMap via OGR, AutoCAD FDO	SharpMap.Net, JDBC postgis.jar included with postgis, JTS etc. tons for Java, GDAL C++, AutoCad FDO beta support
Ελεύθερο λογισμικό για δημιουργία σχεσιακών μοντέλων για χάρτες	Υπάρχει ένα .Net project το οποίο ονομάζεται NHibernateSpatial και διατίθεται σε beta έκδοση.	Ένα java project είναι το Hibernate Spatial	NHibernateSpatial και το HibernateSpatial
Ελεύθερο λογισμικό για προβολή και επεξεργασία χαρτών	Στο μέλλον θα υπάρξει κάποιο λογισμικό το οποίο να προβάλει και να τροποποιεί χάρτες. Στην τωρινή έκδοση υπάρχει μόνο η προβολή χαρτών	GvSig	OpenJump, QuantumGIS, GvSig, uDig
Εμπορικές εφαρμογές για προβολή και τροποποίηση χαρτών για την	ESRI ArcGIS 9.3 Server SDE, Manifold, FME	FME	ESRI ArcGIS 9.3 Server, ZigGIS for desktop, Manifold, FME
Web Mapping ToolKits	Manifold, MapDotNet, ArcGIS 9.3, UMN MapServer, MapGuide Open Source (using beta FDO driver)	UMN Mapserver, GeoServer, MapGuide Open Source	UMN Mapserver, GeoServer, MapGuide Open Source
Spatial Functions	Υποστηρίζει OGC SFSQL MM and	OGC mostly only MBR (bounding box	OGC mostly only MBR (bounding box

	Geodetic custom (πάνω από 70 συναρτήσεις)	functions). Περιορίζεται σε 2D γραφικά	functions) few true spatial relation functions, 2D only
Spatial Indexes	Ναι - 4 επίπεδα Multi-Level grid	R-Tree quadratic splitting - indexes only exist for MyISAM	GIST - a variant of R-Tree
Geodetic support -	Ναι	Όχι	Όχι
Shared Hosting	Πολλούς	Πολλούς	Πολύ λίγους

Στο επόμενο πίνακα θα συγκρίνουμε τους τρεις προμηθευτές για τις συναρτήσεις που παρέχουν για την διαχείριση γεωγραφικών δεδομένων.

Χαρακτηριστικά	Sql Server 2008 (RC0) Geometry	Sql Server 2008 (RC0) Geography	Mysql 5.1/6	PostgradeSql 8.3 /Post GIS 1.3/1.4
Γεωμετρικοί τύποι που υποστηρίζονται	Polygon, Point, LineString, MultiLineString, MultiPoint, MultiPolygon, GeometryCollection. Υποστηρήζει 2D,3D,4D (M,Z) αποθήκευση	Polygon, Point, LineString, MultiLineString, MultiPoint, MultiPolygon, GeometryCollection, Υποστηρήζει 2D,3D,4D (M,Z) αποθήκευση	2D, can store 3D 4D(e.g. M and Z) but no functions do anything with those) - Polygon, Point, LineString, MultiPoint, MultiPolygon, MultiLineString, GeometryCollection	2D, some 3D, 4D (support for storing Z,M but most spatial functions ignore the higher dimensions) and some curve support - Polygon, Point, LineString, MultiPoint, MultiPolygon, MultiLineString, GeometryCollection, CircularString, CompoundCurve, CurvePolygon, MultiCurve, MultiSurface
Δηματότητα μεταφορά από μία χωροταξική αναφορά σε μία άλλη.	Οχι χρειάζονται εργαλεία από τρίτους για την υλοποίηση μεταφοράς.	Οχι	Οχι	ST_Transform – Μεατροπή από 2D και από 3D, αλλά όχι για Circular Curve Types
Εξαγωγή γεωμετρικών δεδομένων.	STAsBinary(), STAsText(), AsGML(), AsTextZM(),	Ιδια με τον SQL Server2008 Geometry	AsBinary(), AsText()	ST_AsBinary(), ST_AsText(), ST_AsSVG(), ST_AsGML(), ST_AsKML(), ST_AsGeoJson() -- new in 1.3.4, ST_AsEWKT(), ST_AsHexEWKB()
Εισαγωγή γεωμετρικών δεδομένων	STGeomFromText(), STGeomFromWKB(), GeomFromGML()	Ιδια με τον SQL Server2008 Geometry	GeomFromText(), GeomFromWKB()	ST_GeomFromText(), ST_GeomFromWKB()
Intersects and	STIntersects(),	Ιδια με τον SQL	MBRIntersects	ST_Intersects(),

Intersection	STIntersection( ) (spatial ref of both geoms must be the same)	Server2008 Geometry με την μόνη διαφορά όταν πραγματοποιείται εισαγωγή πολυγώνων τα οποία επικαλύπτονται	()*, μόνο για περιγράμματα	ST_Intersection( )
Άλλες συσχετίσεις	STContains(), STDifference( ), STDisjoint(), STEquals(), STOverlaps(), STRelate(), STSymDifference( ), STTouches(), STWithin()	STDisjoint()	MBRContains( ), MBREqual(), MBROverlaps, MBRTouches() , MBRWithin(), bounding	ST_Contains(), ST_Disjoint(), ST_Difference( ), ST_Equals(), ST_Overlaps(), ST_Relate(), ST_SymDifference( ), ST_Touches(), ST_Within() (spatial ref must be the same)
Accessors and Editors	BufferWithTolerance() ,MakeValid(), ST_Simplify(), STBoundary(), STBuffer(),STCentroid(), STConvexHull(), STDimension(), STEndPoint(), STExteriorRing(), STGeometryN(), STGeometryType(), STInteriorRingN(), STIsClosed(), STIsEmpty(), STIsSimple(), STNumGeometries( ), STNumPoints() (only applies to LINESTRINGS), STPointN((), STStartPoint(), STSRID(), STUnion()	NumRings(), RingN(), STArea(), STBuffer(), STDimension(), STEndpoint(),STGeometryN(), STGeometryType( ), STIsClosed(), STIsEmpty(), STLength(), STNumGeometries(), STNumPoints(), STPointN(), STSrid(), STStartPoint(), STUnion()	Centroid(), Dimension(), EndPoint(), Envelope(), ExteriorRing(), GeomtryN(), GeometryType(), InteriorRingN( ), IsClosed(), IsRing(), NumPoints(), PointN(), SRID(), StartPoint()	ST_Affine( ) , ST_Boundary(), ST_Buffer() (2 variants similar to STBuffer() and BufferWithTolerance( )), ST_Centroid(), ST_ConvexHull(), ST_Dimension(), ST_EndPoint(), ST_ExteriorRing(), ST_GeometryN(), ST_GeometryType(), ST_InteriorRingN(), ST_IsClosed(), ST_IsEmpty(),ST_IsRing( ), ST_ISSimple(), ST_NumGeometries(),ST_NumPoints() (only applies to linestrings), ST_NPoints() - returns num vertexes regardless of geometry type, ST_PointN(), ST_Simplify(), ST_StartPoint(), ST_SRID(),ST_Translate( ), ST_Union(), various MakeLine,MakePolygon , buildarea etc
Μετρήσεις	STArea(), STLength(), STDistance() (measurements in unit of spatial ref)	Same as geometry - except measurements can be in meters, sq meters, or units of spatial ref	Area(), GLength(), Distance()	ST_Area(), ST_Length(), ST_Distance(), ST_Distance_Spheroid() , ST_Length_Spheriod (non-sphere, non-spheriod functions units are in spatial ref ,

				sphere and spheroid are in meters but only work for point geometries) and equivalents for 3D geometries.
Γραμμικές αναφορές	Δεν υπάρχει κάποια που να υποστηρίζεται από τον sql ,Αλλά μπορεί να υποστηριχτεί αοπτον Isaac Knunen	Δεν υπάρχει κάποια που να υποστηρίζεται από τον sql . Υπάρχουν εργαλεία για τον SQL Server τα οποία μπορούν να βρεθούν στο code plex project	none	ST_Line_Interpolate_point(), ST_Line_Substring(), ST_line_locate_point(), ST_locate_along_measure(), ST_locate_between_measures()
Χωρικά σύνολα	Κανένα ενσωματωμένο πρέπει να δημιουργηθεί το αντήστηχο CLR για το οποίο θα υποστηρίζει σύνολα. Υπάρχουν αρκετά παραδείγματα στο internet.	όχι	όχι	ST_Extent(), ST_Collect(), ST_Union(), ST_Accum(), ST_MakeLine(), ST_Polygonize()

Με βάση την σύγκριση που έγινε θα πρέπει να επιλεγεί το κατάλληλο ΣΒΔ το οποίο θα υποστηρίξει την εφαρμογή.

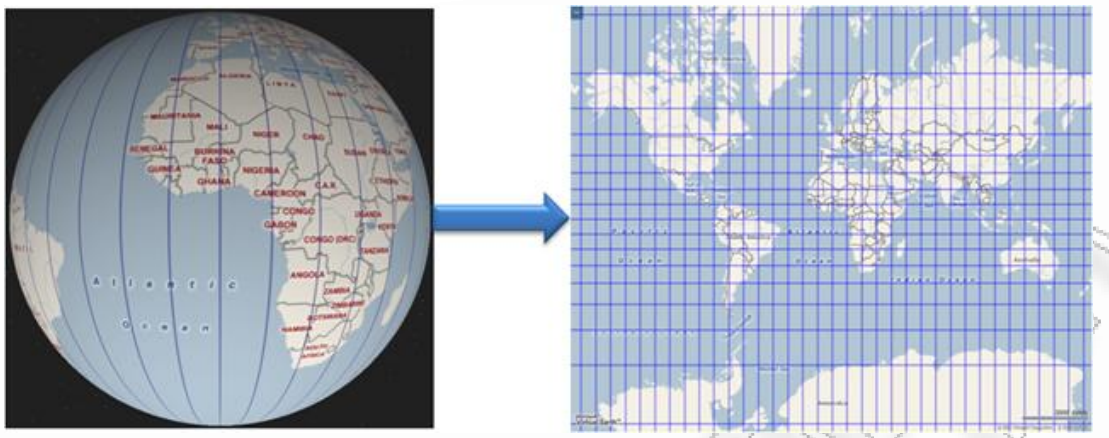
Διαχείριση γεωγραφικών δεδομένων στον MSSQL Server 2008

Επιλέγοντας σαν καλλίτερη λύση των SQL Server 2008 για την διαχείριση των γεωγραφικών δεδομένων θα πραγματοποιηθεί μία σύντομη εισαγωγή για το πώς μπορούμε να διαχειριστούμε γεωγραφικά δεδομένα.

Το σημαντικό κομμάτι που πρέπει να υλοποιηθεί για να λειτουργήσει το πληροφοριακό σύστημα είναι η σύνδεση του Bing Map με την βάση δεδομένων SQL Server 2008.

Εισαγωγική ενότητα

Σ' αυτή την ενότητα θα πραγματοποιηθεί μία σύντομη εισαγωγή για το τι είναι τα γεωγραφικά δεδομένα και πως τα χειριζόμαστε. Ο κόσμος ο οποίος ζούμε δεν είναι επίπεδος όπως βλέπουμε σε ένα χάρτη (εικόνα 6). Αυτό έχει σαν συνέπια όταν απευθυνόμαστε σε διαστάσεις που είναι δύο σημείων να είναι τελικά τρείς. Για να πραγματοποιηθεί αυτή η μετατροπή σωστά παρέχονται οι κατάλληλες βιβλιοθήκες από την Microsoft ©

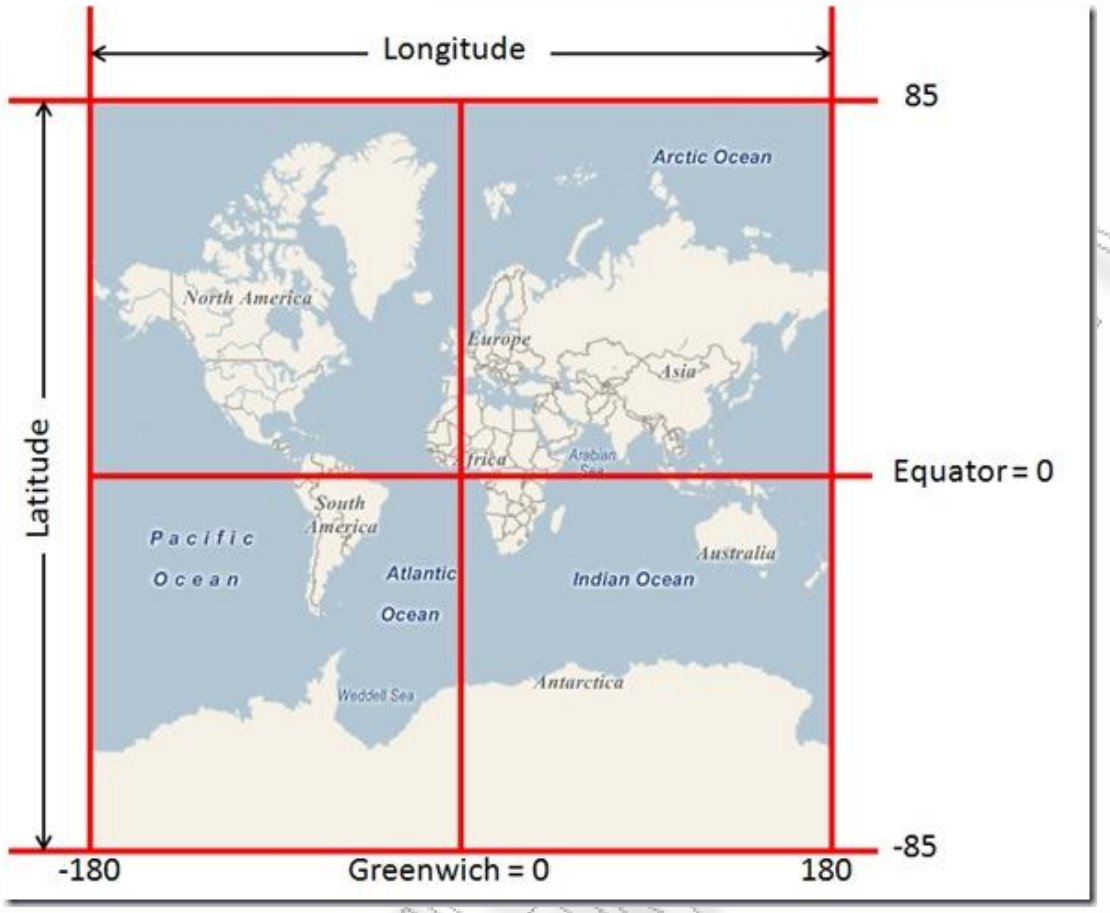


Εικόνα 6 Προβολή της Γής

Υπάρχουν και παρά πολλά συστήματα αναφορών για συντεταγμένες. Ένα παλιό σύστημα αναφοράς είναι το OSGB36 (Ordnance Survey Great Britain 1936). Αυτό που χρησιμοποιούμε σήμερα είναι το WGS84 (World Geodetic System 1984). Το οποίο χρησιμοποιήθηκε στο GPS (Global Positioning System).

Μια βασική οντότητα σε ένα χάρτη είναι το στοιχείο. Σαν ορισμό μπορούμε να πούμε ότι «είναι μία σταθερή επιφάνεια η οποία περιγράφει μία γεωγραφική θέση ή ένα φαινόμενο στην Γή». Η συντεταγμένες που χρησιμοποιούνται για την αναφορά μίας γεωγραφικής θέσης είναι βασισμένες στο στοιχείο (damus).

Στο σύστημα αναφοράς (WGS84) έχουμε δύο τρόπους με τους οποίους μπορούμε να αναφερθούμε σε ένα σημείο. Ο πρώτος είναι με το καρτεσιανό σύστημα και ο δεύτερος είναι με τις τιμές latitude και longitud. Το Latitudes παίρνει τιμές από -90^{O} μέχρι +90^{O} και η τιμή 0 αναφέρεται στον ισημερινό ενώ ο Longitud μπορεί να πάρει τιμές από -180^{O} μέχρι 180^{O} και 0 είναι η θέση στο βασιλικό παρατηρητήριο στο Γκρήνουιτς στο Ηνωμένο Βασίλειο. Ένα πολύ σημαντικό σημείο είναι ότι όταν εργαζόμαστε σε δύο διαστάσεις τότε η μέγιστη τιμή που μπορούμε να έχουμε για το latitude είναι από -85^{O} μέχρι +85^{O} (Εικόνα 7).



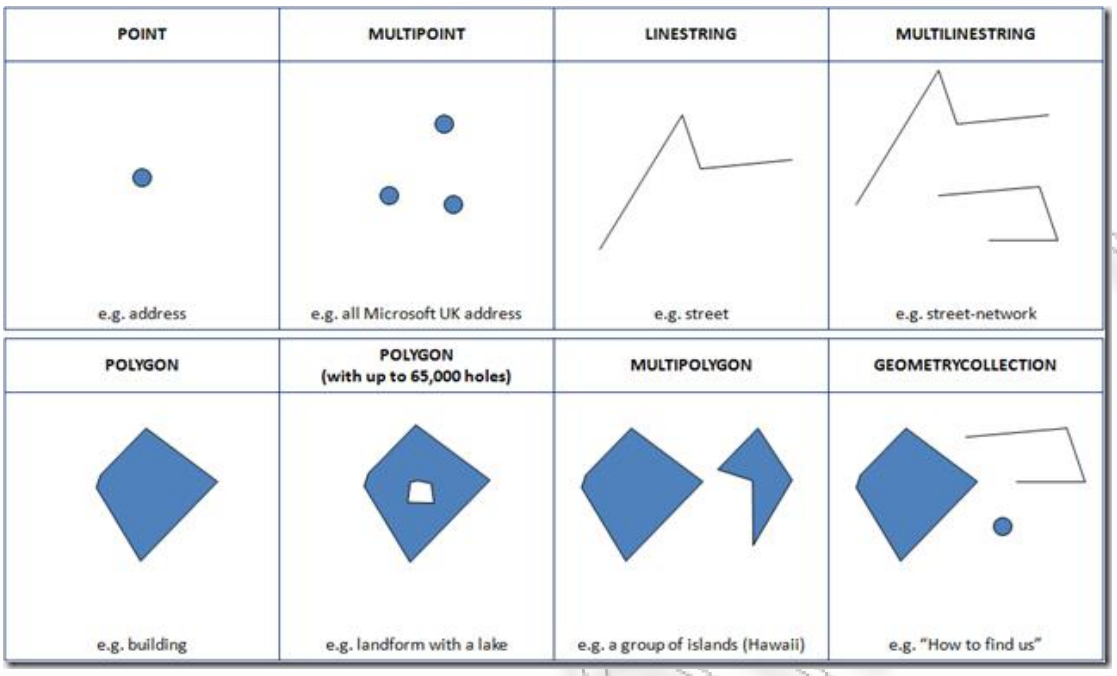
Εικόνα 7 Προβολή σε 2D

MS SQL Server 2008 για διαχείριση γεωγραφικών δεδομένων

Η έκδοση του MS Sql Server 2008 υποστηρίζει δύο τύπους γεωγραφικών δεδομένων. Ο πρώτος είναι ο γεωγραφικός και ο δεύτερος είναι ο γεωμετρικός. Και οι δύο τύποι υποστηρίζονται στο framework της Microsoft.NET. Ο γεωμετρικός τύπος έχει απλά χαρακτηριστικά και υποστηρίζεται όταν έχουμε συντεταγμένες σε επίπεδο. Ο γεωγραφικός τύπος δεδομένων μπορεί να αποθηκεύσει ελλειψοειδή στοιχεία όπως συντεταγμένες γεωγραφικού πλάτους και μήκους. Ο δεύτερος τρόπος είναι και ο προηγούμενος που μπορεί να χρησιμοποιηθεί με το Bing Maps.

Υποστηριζόμενες Γεωμετρίες

Οι υποστηριζόμενες γεωμετρίες εμφανίζονται στο παρακάτω σχήμα.



Εικόνα 8 υποστηριζόμενες γεωμετρίες από τον SQL server 2008

Το API της Microsoftυποστηρίζει σημεία γραμμές πολύγονα. Για την αποθήκευση εύρεση και διαχείριση τέτοιων στοιχείων υπάρχουν στο SQL Server οι αντίστοιχες συναρτήσεις.

Point

Το Σημείο(Point) είναι ο βασικός γεωμετρικός τύπος και τον χρησιμοποιούμε για να καθορίσουμε ένα σημείο στον χώρο. Ένα σημείο δεν έχει μέγεθος στον χώρο. Στην εφαρμογή αλλά και γενικά στα συστήματα που ασχολούνται με γεωγραφικά δεδομένα το σημείο χρησιμοποιείται για να αναπαραστήσει μίας τοποθεσίας επάνω στην γη. Ένα συνηθισμένο παράδειγμα είναι το σημείο να αναπαριστά μία διεύθυνση σε μία χώρα.

LineStrings

Αν έχουμε καθορίσει μία σειρά από σημεία στον χάρτη μπορούμε να σχεδιάσουμε γραμμές οι οποίες θα συνδέουν τα σημεία μεταξύ τους. Το LineString είναι διαφορετικά σημεία στον χάρτη συνδεδεμένα μεταξύ τους. Είναι χωροταξικό αντικείμενο (spatial object) το οποίο αναπαριστάστε σε μία διεύθυνση και δεν έχει ορισμένο μέγεθος αλλά δεν ορίζει μία περιοχή. Οι τύποι γραμμών που μπορούμε να έχουμε είναι οι ακόλουθοι.

- Simple LineString. Είναι μία απλή γραμμή η οποία συνδέει σημεία. Η γραμμή αυτή δεν τέμνεται.
- Close LineString. Η διαφορά αυτού του τύπου είναι ότι εκεί που τελειώνει είναι το ίδιο σημείο που άρχισε.
- LineString Simple and Close. Το ίδιος αυτής της γραμμής περιλαμβάνει και τις δύο ιδιότητες.

Πολύγωνα

Ένα πολύγωνο ορίζεται στον χώρο από συνδεδεμένα σημεία τα οποία είναι τύπου Close LineString και σχηματίζουν έναν εξωτερικό δακτύλιο. Το πολύγωνο περιέχει εκείνα τα σημεία που είναι μέσα στον δακτύλιο και θα πρέπει να έχει μόνο ένα εξωτερικό δακτυλίδι το καθορίζει και τα όρια του. Μπορεί να περιέχει ένα ή περισσότερα πολύγωνα εσωτερικά.

Αυτός ο τύπος γεωμετρίας έχει δύο διαστάσεις, έχει συγκεκριμένο μέγεθος και ορίζει μία περιοχή. Το μέγεθος του πολύγονου ορίζεται ως το άθροισμα των αποστάσεων γύρο από την περίμετρο, δεν συμπεριλαμβάνει τα τυχών εσωτερικά πολύγωνα που μπορεί να έχει.

Γεωγραφικό σύστημα αναφοράς στον SQL Server 2008

Ο SQL Server 2008 υποστηρίζει όλα τα συστήματα αναφοράς για χωροταξικά δεδομένα. Εκτελώντας το ερώτημα SELECT * FROM sys.spatial_reference_system . Με την εκτέλεση του ερωτήματος επιστρέφει ένα πίνακα, στην πρώτη γραμμή με spatial_reference_id 4120 περιγράφει το σύστημα αναφοράς για την Ελλάδα.

Spatial Indexing

Τα Spatial Indexing βοηθούν στην καλλίτερη απόδοση των ερωτημάτων στην βάση δεδομένων που αφορά χωρικά δεδομένα. Χρησιμοποιώντας αυτή την λειτουργία δεν αλλοιώνονται τα δεδομένα αλλά αν επιλεγούν σοφά τότε μπορούν να δημιουργηθούν ευρετήρια. Για την κατηγορία γεωγραφικών δεδομένων έχουμε τους ακόλουθους τύπους δεδομένων.

- geography1.STIntersects(geography2)
- geography1.STEquals(geography2)
- geography1.STDistance(geography2)
- geography1.STDistance(geography2)

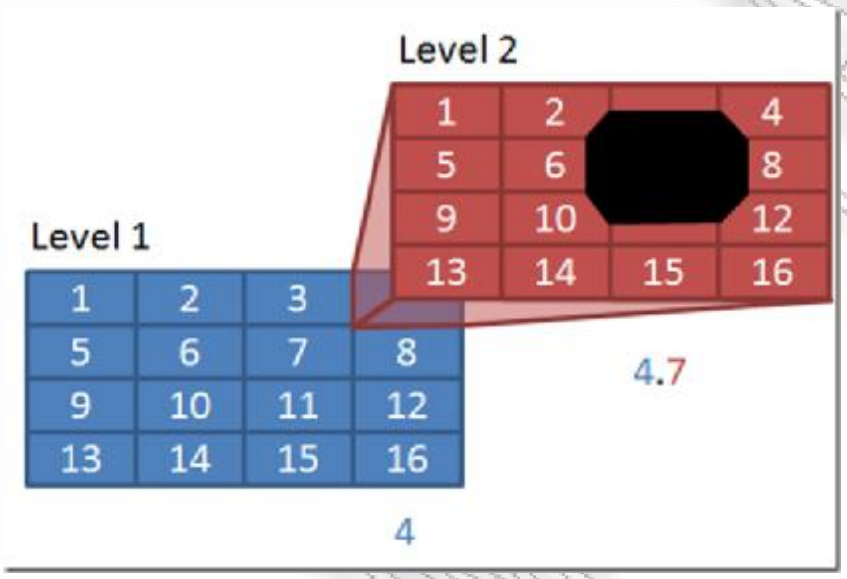
Για τα γεωμετρικά δεδομένα έχουμε τους ακόλουθους τύπους.

- geometry1.STContains(geometry2)
- geometry1.STDistance(geometry2)
- geometry1.STDistance(geometry2)
- geometry1.STEquals(geometry2)
- geometry1.STIntersects(geometry2)
- geometry1. STOverlaps (geometry2)
- geometry1.STTouches(geometry2)
- geometry1.STWithin(geometry2)

Στους Spatial Index τα δεδομένα οργώνονται σε B-δένδρα. Ο αριθμός των κυττάρων μπορεί να είναι σε μορφή 4X4 (χαμηλή πυκνότητα πλέγματος) 8X8 (μέση πυκνότητα πλέγματος) και 16X16 (μεγάλη πυκνότητα πλέγματος). Ο SQL Server χρησιμοποιεί τον αλγόριθμο tessellation για να δημιουργήσει τα κύτταρα. Εκτελώντας την διαδικασία αυτή χρησιμοποιεί τους ακόλουθους κανόνες.

1)Μέσα σε ένα επίπεδο τα κύτταρα αριθμούνται από αριστερά προς τα δεξιά και από πάνω προς τα κάτω.

2)Όταν το κύτταρο καλυφτεί από ένα αντικείμενο τότε ο αλγόριθμος θα το αγνοήσει το συγκεκριμένο. Αυτό το βλέπουμε στο επόμενο παράδειγμα που το κύτταρο 4 στο επίπεδο 1 έχει καλυφθεί. Παρατηρούμε ότι με βάση το αλγόριθμο στο επίπεδο 2 έχει καλυφθεί το πολύγωνο 7.

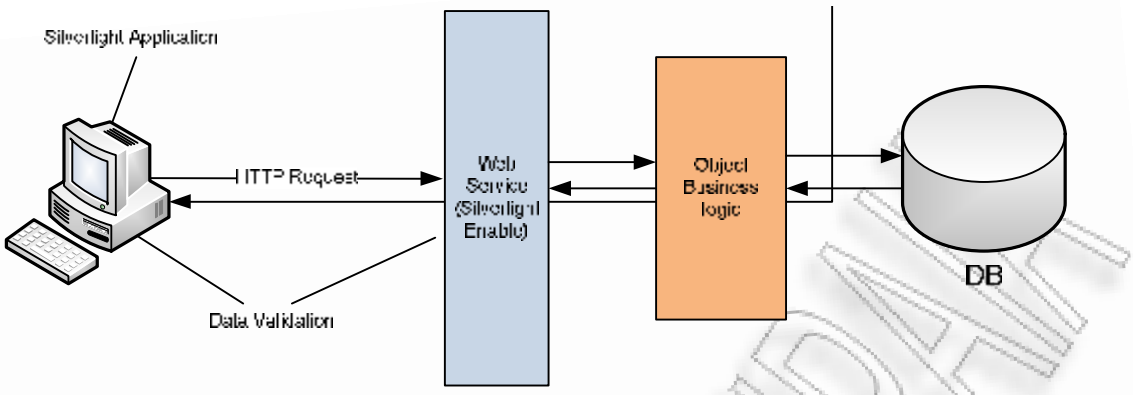


Εικόνα 9 tessellation algorithm

Όταν δημιουργούνται κατάλογοι τύπου Spatial index πόσα κύτταρα θα υπάρχουν ανά αντικείμενο. Όταν ο αλγόριθμος φτάσει το όριο τότε σταματά.

Ανάλυση ερωτήματος στην Βάση Δεδομένων.

Για την ολοκλήρωση της ανάλυσης αρχιτεκτονικής του πληροφοριακού συστήματος παρουσιάζεται στην επόμενη εικόνα (10) η διαδικασία εκτέλεσης ενός ερωτήματος που δημιουργείτε από το γραφικό περιβάλλον το οποίο προσφέρεται στον χρήστη. Το ερώτημα θα αποσταλεί σε ένα Web Server ο οποίος έχει εγκατεστημένα Web Service. Τα Web Service μπορούν να καλέσουν τα αντικείμενα που υλοποιούν την Business λογική. Με την σειρά τους μπορούν να καλέσουν τις συναρτήσεις στο Data Access Layer για την εκτέλεση του ερωτήματος.



Εικόνα 10 Εκτέλεση ερωτήματος στο Σ.Β.Δ.

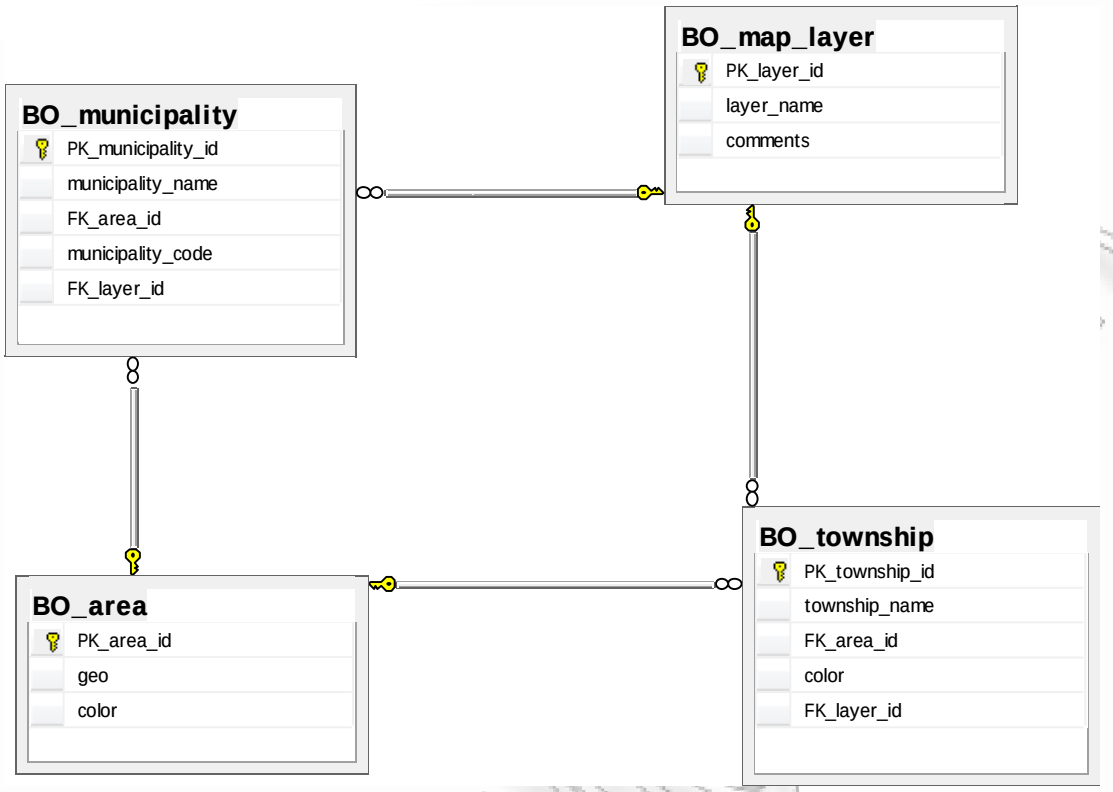
Σχεδίαση Βάσης Δεδομένων

Διαχείριση Νομών – Δήμων

Στο παρακάτω διάγραμμα αναλύονται οι οντότητες δήμος, κοινότητα και οι σχέσεις που αναπτύσσονται μεταξύ τους.

Ανάλυση διαγράμματος:

- Δήμος
  - Ο κάθε δήμος έχει ένα όνομα
  - Ο κάθε δήμος έχει μία περιγραφή
  - Ο κάθε δήμος μπορεί να ανήκει σε ένα επίπεδο
  - Τα όρια του δήμου ορίζονται μέσα από ένα πολύγωνο.

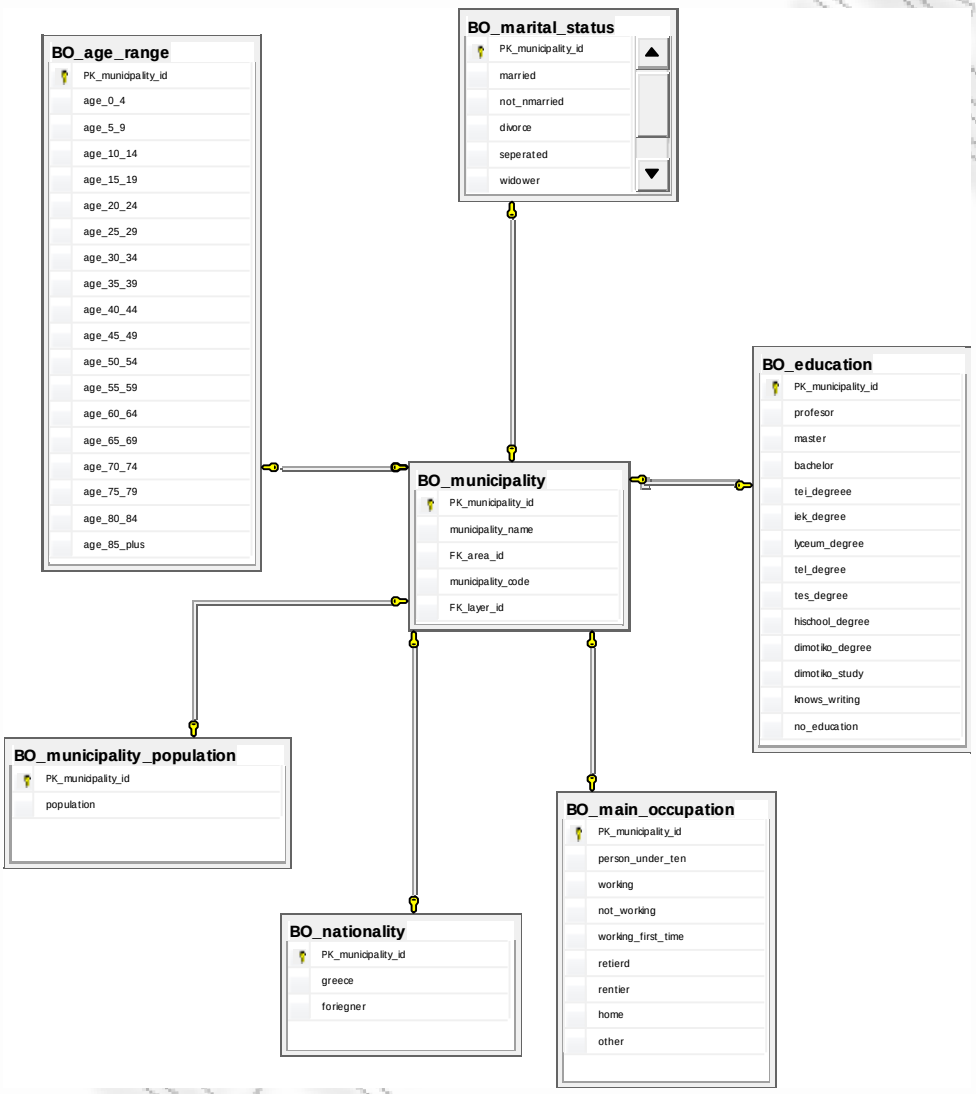


Εικόνα 11 Σχεσιακό μοντέλο Νομών – Δήμων

Στο παραπάνω διάγραμμα έχουμε τις βασικές οντότητες δήμος και κοινότητα αντίστοιχα *BO_municipality* και *BO_township* . Τα όρια του δήμου ορίζονται από ένα πολύγωνο στον πίνακα *BO_area* το πεδίο geo αποθηκεύει την αντίστοιχη πληροφορία γεωγραφικά.

Στον πίνακα *BO_area* δεν αποθηκεύεται γεωγραφικές πληροφορίες μόνο για τους δήμους αλλά μπορούν να αποθηκευτούν και περιοχές (πολύγωνα) ανεξάρτητα, για τον λόγο αυτό δημιουργήθηκε η σχέση μεταξύ του *BO_area* και του *BO_municipality* αντίστοιχα το ίδιο ισχύει και για την οντότητα *BO_township*.

Ανάλυση δημογραφικών στοιχείων δήμου



Εικόνα 12 Δημογραφικά στοιχεία

Ο κάθε δήμος έχει δημογραφικά στοιχεία. Η βασική οντότητα του διαγράμματος είναι το *BO_municipality* η οποία περιγράφει των δήμο. Σ' αυτών τον πίνακα αναπτύσσονται σχέσεις με τους ακόλουθους πίνακες. *BO_age_range, BO_municipality_population, BO_nationality, BO_main_occupation,BO_education,BO_marital_status.* Η αντιστοίχηση των πινάκων είναι η ακόλουθη.

BO_age_range	**Εύρος ηλικιών για κάθε δήμο χωρισμένο ανά τετράδες ξεκάνοντας από το 0 και φτάνοντας μέχρι το 85+**
BO_municipality_population,	Ο πίνακας αυτός περιέχει πληροφορίες

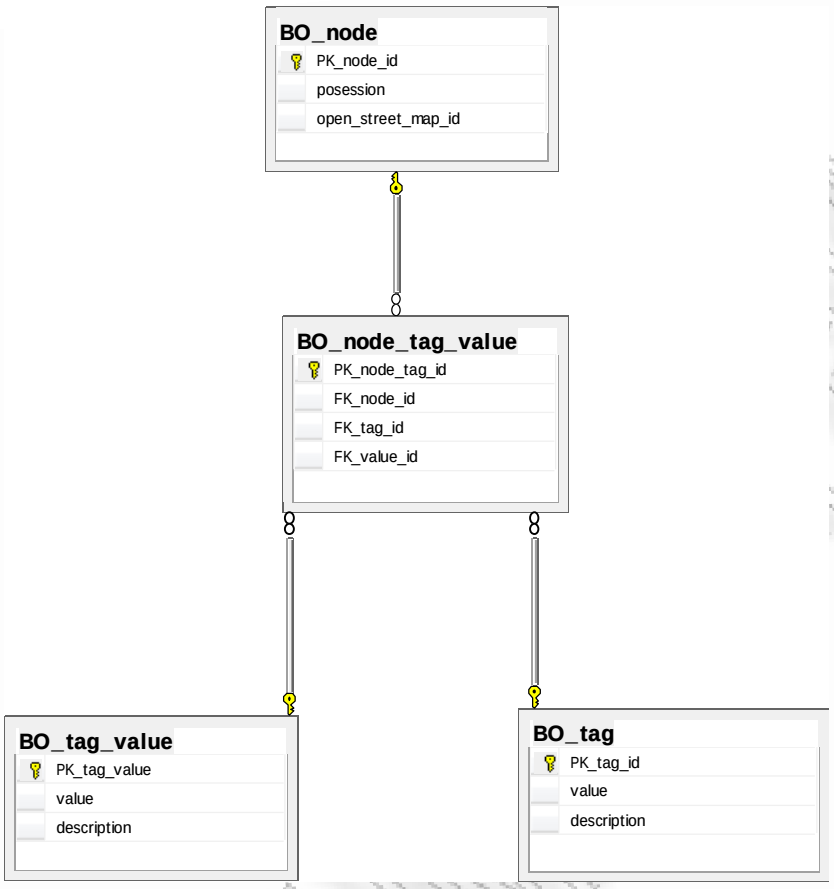
	για τον πληθησμό κάθε δήμου.
BO_nationality	Ο πίνακας αυτός έχει τα στοιχεία που αφορούν την εθνικότητα έχει δύο τιμές • Έλληνας • Αλλοδαπός
BO_main_occupation,	Ο συγκεκριμένος πίνακας έχει αποθηκευμένες τιμές για την απασχόληση των πολιτών στους δήμους. Οι τιμές που μπορεί να πάρει είναι ακέραιες και περιγράφουν τα ακόλουθα πεδία : • Κάτων από 18 • Εργάζεται • Ζητά εργασία • Ζητά εργασία για πρώτη φορά • Συνταξιούχος • Άνεργος • Εισοδηματίας
BO_education	Ο συγκεκριμένος πίνακας έχει αποθηκευμένες τιμές για την εκπαίδευση των πολιτών στους δήμους. Οι τιμές που μπορεί να πάρει είναι ακέραιες και περιγράφουν τα ακόλουθα πεδία : • Διδακτορικό • Μεταπτυχιακό • Προπτυχιακό ΑΕΙ • Προπτυχιακό ΤΕΙ

	• Πτυχίο ΙΕΚ • Πτυχίο Λυκείου • Πτυχίο ΤΕΣ • Πτυχίο Λυκείου • Πτυχίο Γυμνασίου • Πτυχίο Δημοτικού • Γνωρίζει να γράφει • Αγράμματος
BO_marital_status	Ο συγκεκριμένος πίνακας έχει αποθηκευμένες τιμές για την εκπαίδευση των πολιτών στους δήμους. Οι τιμές που μπορεί να πάρει είναι ακέραιες και περιγράφουν τα ακόλουθα πεδία : • Έγγαμος • Άγαμος • Διαζευγμένος • Σε διάσταση

Οι σχέσεις που αναπτύσσουν οι πίνακες που περιγράφτηκαν με τον πίνακα *BO_municipality*είναι ένα προς ένα γιατί ένα δήμος έχει μοναδική τιμή σε σχέση με τους πίνακες με τα δημογραφικά στοιχεία.

Geography tagging

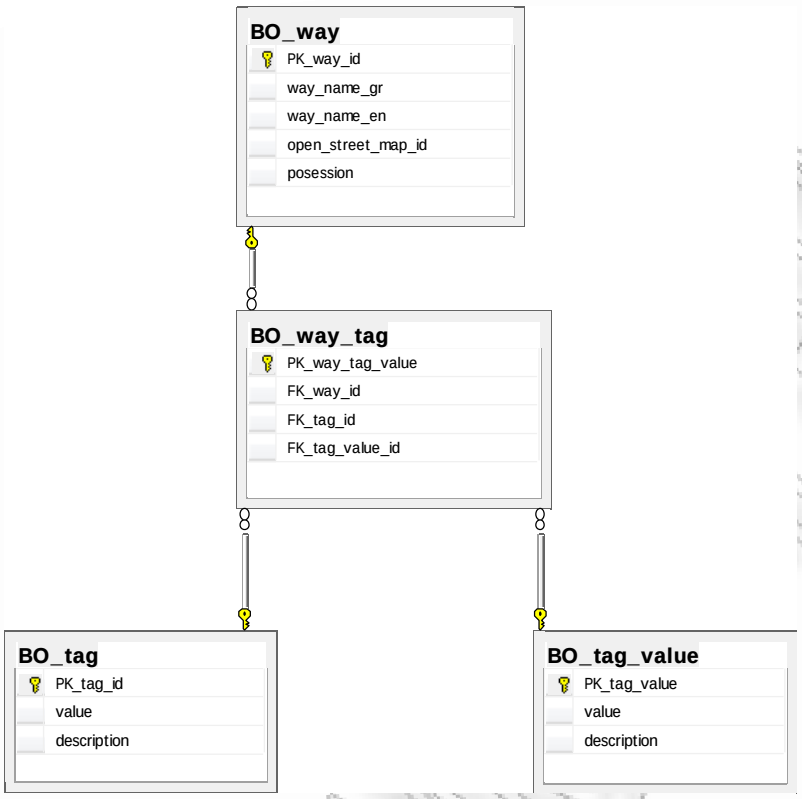
Το πληροφοριακό σύστημα πρέπει να υποστηρίζει λειτουργίες όπως tagging γεωγραφικών περιοχών (που μπορεί να είναι και δήμοι), ή tagging δρόμων ή και για ένα σημείου. Ο όρος tagging περιγράφει την δυνατότητα καταγραφή πληροφοριών (μπορεί να είναι 1 ή πολλά) για ένα στοιχείο. Στην βάση δεδομένων χωρίζεται σε τρείς βασικές οντότητες, η πρώτη είναι για ένα γεωγραφικό σημείο.



Εικόνα 13 geography tag node

Στο παραπάνω διάγραμμα περιγράφονται οι σχέσεις που δημιουργούνται. Ο πίνακας *BO_node* αποθηκεύει τα γεωγραφικά σημεία. Στον πίνακα *OB_node_value* αναπτύσσονται τρεις σχέσεις η οποίες περιγράφουν ότι το συγκεκριμένο γεωγραφικό σημείο έχει μία ένα ετικέτα (tag) και μία τιμή. Ο πίνακας *BO_tag_value* διατηρεί τις τιμές και ο πίνακας *BO_tag* διατηρεί τις ετικέτες. Με την δημιουργία μίας τέτοιας σχέσης ένα γεωγραφικό σημείο μπορεί να έχει μία ή παραπάνω ετικέτες και αυτές να έχουν μία τιμή. Παράδειγμα ότι το σημείο είναι ένα μαγαζί τηλεπικοινωνιών της Vodafone, τότε θα υπάρχει ένα γεωγραφικό σημείο με γεωγραφικό πλάτος και μήκος που θα υποδεικνύει την θέση του καταστήματος και μπορεί να έχει ένα tag με κείμενο «Τηλεπικοινωνίες» και μια τιμή «Vodafone».

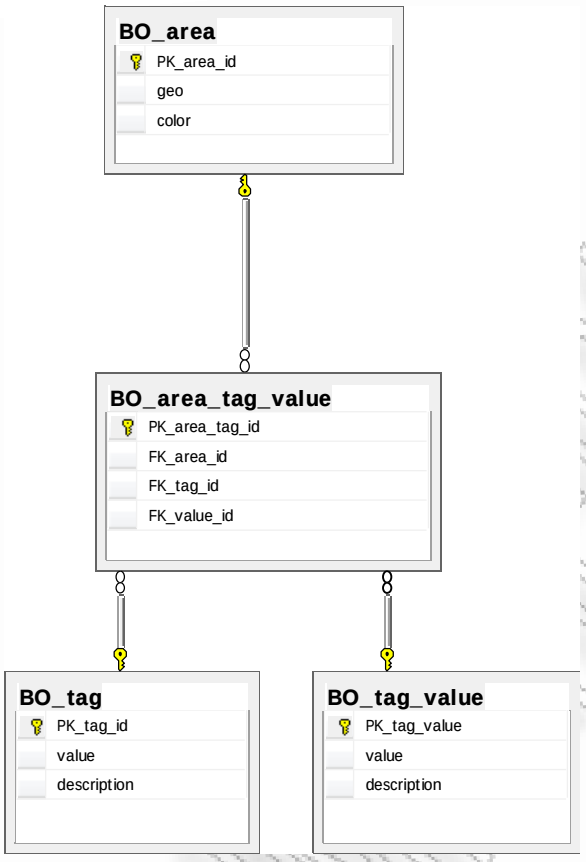
Το ίδιο ισχύει και για δρόμους όπως εμφανίζεται στο επόμενο διάγραμμα.



Εικόνα 14 geography tag way

Το σημαντικό σημείο που πρέπει να επισημανθεί βλέποντας το συγκεκριμένο διάγραμμα είναι ότι οι πίνακες *BO_tag* και *BO_tag_value* είναι αυτοί που χρησιμοποιήθηκαν και για τα γεωγραφικά σημεία και το αυτό που αλλάζει είναι ο πίνακας *BO_way* που διατηρεί γεωγραφικές οντότητες τύπου γραμμών και ο πίνακας *BO_way_tag* ο οποίος αποθηκεύει την σχέση δρόμου με ετικέτα και τιμή.

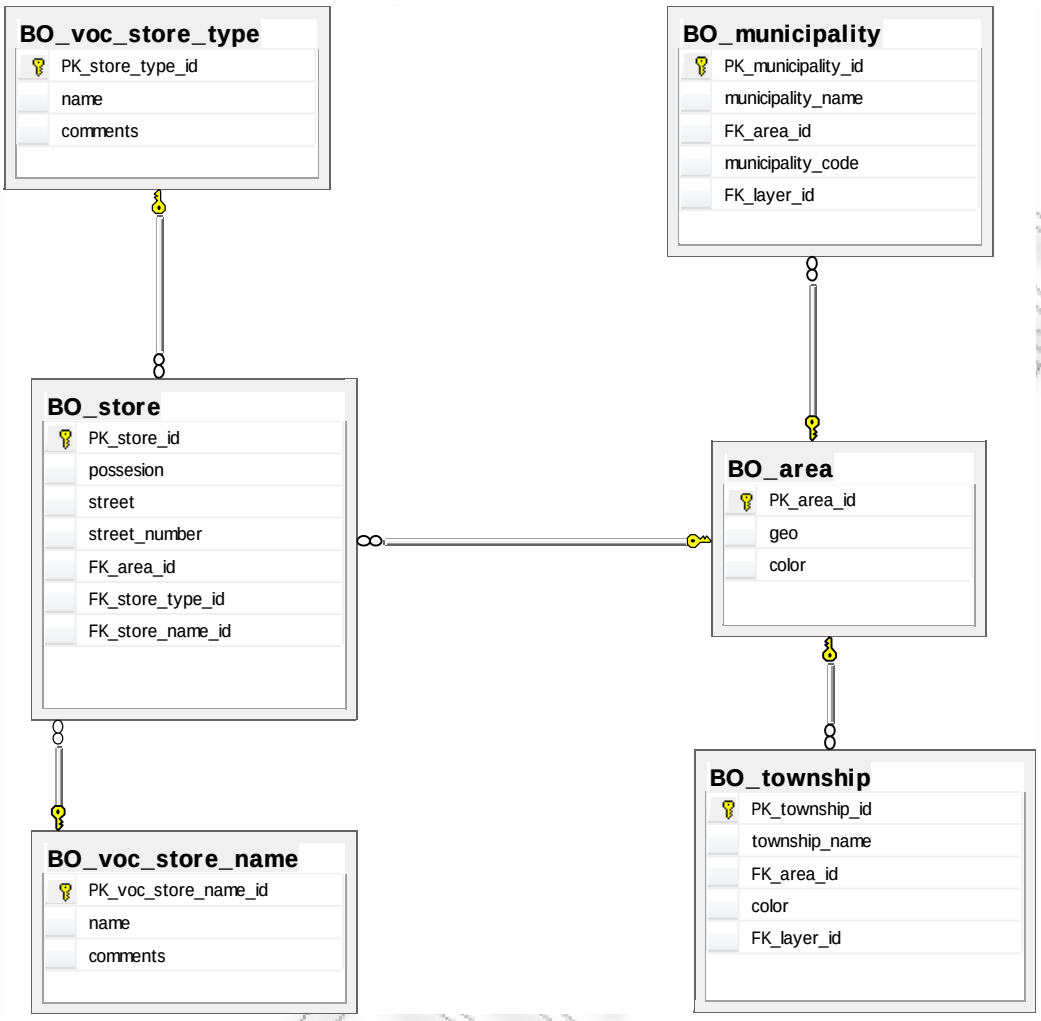
Τέλος υπάρχει και το διάγραμμα για περιοχές στην επόμενη εικόνα και έχει την ίδια λογική με τα προηγούμενα δύο με την μόνη διαφορά ότι αποθηκεύει πολύγωνα.



Εικόνα 15 geography tag area

Ανάλυση διαγράμματος καταστήματος

Στην επόμενη εικόνα περιγράφεται η ανάλυση της οντότητας κατάστημα στο πληροφοριακό σύστημα.



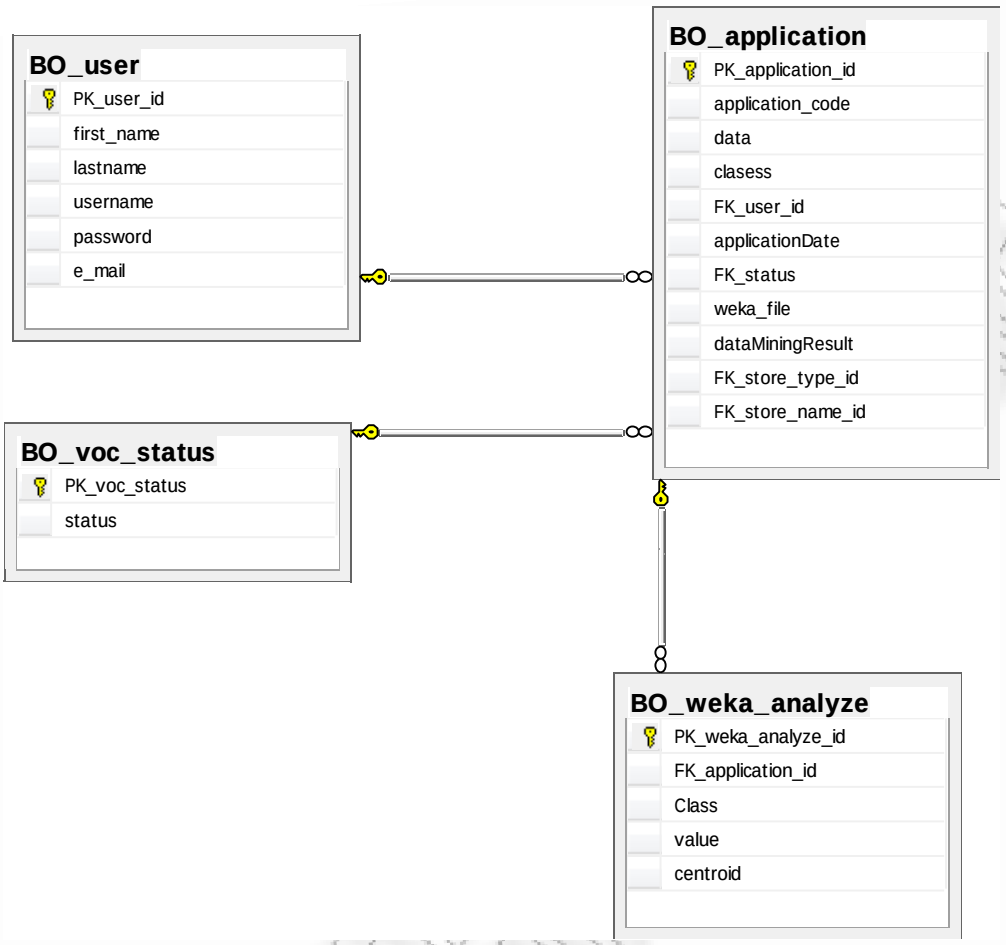
Εικόνα 16 Διάγραμμα καταστήματος

Η βασική οντότητα είναι το BO_store στην οποία διατηρείται η διεύθυνση του και η γεωγραφική του θέση. Σ' αυτόν τον πίνακα αναπτύσσονται σχέσεις για να περιγράψουν σε ποια κατηγορία ανήκει και ποιο είναι το όνομά του. Ένα κατάστημα μπορεί να ανήκει σε μία κατηγορία και αυτή η σχέση είναι ανάμεσα στον πίνακα *BO_store* και *BO_voc_store_type*. Το κατάστημα μπορεί να έχει μία ονομασία η οποία αποθηκεύεται στον πίνακα *BO_VOC_store_name*.

Το μαγαζί έχει μία συγκεκριμένη γεωγραφική θέση (αποθηκεύεται στον πεδίο possession) αυτή η θέση ανήκει σε ένα δήμο της Αθήνας. Για την διατήρηση την σχέσης αυτής χρησιμοποιήθηκε ο πίνακας *BO_area*.

Ανάλυση διαγράμματος «Αίτημα από χρήστη»

Ο τελικός χρήστης που θα χρησιμοποιήσει την εφαρμογή αυτό που θέλει να κάνει είναι να στείλει μία αίτηση στο πληροφοριακό σύστημα και αυτό να του απαντήσει με τις ομάδες που δημιούργησε εκτελώντας το Data Mining Tool. Οι αιτήσεις αυτές θα πρέπει να αποθηκεύονται στην Βάση Δεδομένων. Στο επόμενο διάγραμμα περιγράφονται όλες οι σχέσεις που αναπτύσσονται στο Σ.Β.Δ. για να διαχειριστεί μία τέτοια πληροφορία.



Εικόνα 17 Διάγραμμα Application

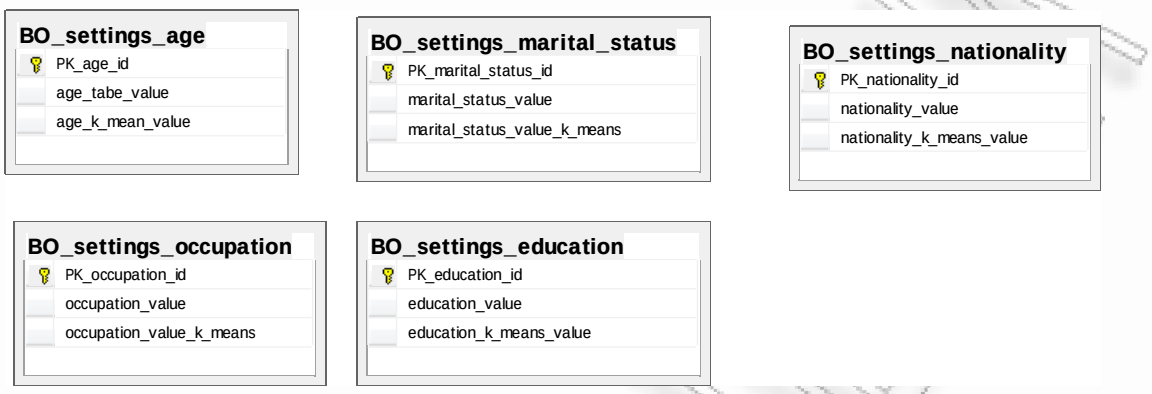
Ο βασικό πίνακας είναι ο *BO_application* στον οποίο αποθηκεύεται η αίτηση που έχει κάνει ένας χρήστης του συστήματος. Στο πεδίο *data* αποθηκεύεται το xml που στέλνει ο χρήστης μέσω της εφαρμογής. Στο πεδίο *application_code* αποθηκεύεται ένας μοναδικός κωδικός ο οποίος αφορά το συγκεκριμένο αίτημα. Στο πεδίο *classes* αποθηκεύεται ο αριθμός κλάσεων (ομάδων) που ζήτησε ο χρήστης. Στο πεδίο *Weka_file* αποθηκεύεται το αρχείο arff που δημιουργήθηκε από την εφαρμογή και τέλος στο πεδίο *dataminingResult* αποθηκεύεται το αποτέλεσμα την το Weka.

Το αίτημα που μπορεί να κάνει κάποιος στο πληροφοριακό σύστημα είναι προσωπικό. Οπότε ο πίνακας *BO_application* έχει μία σχέση ένα προς πολλά με τον πίνακα *BO_user*. Το αίτημα μπορεί να περνάει από πολλές καταστάσεις όπως για παράδειγμα δημιουργία header arff αρχείου, δημιουργία body arff αρχείου κτλπ. Αυτή η πληροφορία αποθηκεύεται στον πίνακα *BO_voc_status*.

Τέλος δημιουργήθηκε ο πίνακας *BO_weka_analyze* ο οποίος διατηρεί τα αποτελέσματα από αναφορές που δημιούργησε ο χρήστης. Στο πεδίο *class* αποθηκεύονται οι κλάσεις που ζήτησε ο χρήστης, Στο πεδίο *centroid* αποθηκεύονται τα κέντρα που δημιουργήθηκαν από την εκτέλεση του Data Mining

Διάγραμμα Settings

Στο διάγραμμα settings έχουμε τους πίνακες που αποθηκεύουν την αντιστοίχηση μεταξύ πεδίων του Weka για δημογραφικά στοιχεία και των στηλών που υπάρχουν στην βάση.



Εικόνα 18 Διάγραμμα Settings

Ανάλυση stored procedures

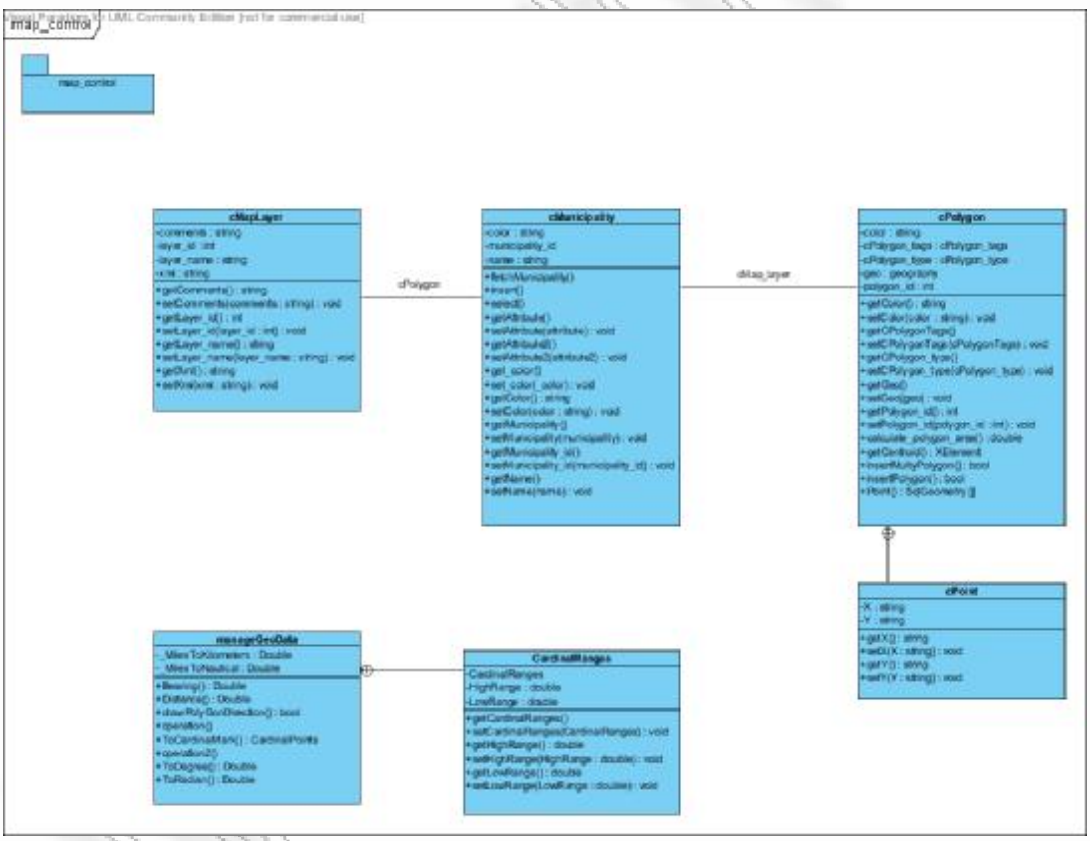
Στο πληροφοριακό σύστημα δημιουργήθηκαν κάποιες Stored procedures. Ο λόγος που δημιουργήθηκαν είναι ότι χρειάστηκα να εκτελεστούν κάποια ερωτήματα στην βάση δεδομένων που ήταν αρκετά σύνθετα.

- Η συνάρτηση *SP_associate_area_tag* συσχετίζει ένα πολύγωνο που θέλουμε να είσάγουμε πληροφορία με μίε ετικέτα και με μία τιμή.
- Η συνάρτηση *SP_convert_point_to_geometry* έχει δύο παραμέτρους, το γεωγραφικό πλάτος και το γεωγραφικό μήκος και επιστρεφεί την γεωγραφική τους αναπαράσταση.
- Η συνάρτηση *SP_convert_polygon_to_geometry* επιστρέφει πληροφορίες για ένα συγκεκριμένο δήμο. Δέχεται μία παράμετρο, το id του δήμου και επιστρέφει το ονομάτου και το πολύγωνο σε γεωμετρική αναπαράσταση.
- Η συνάρτηση *SP_fetch_polygon_tags* δέχεται σαν παράμετρο το id του πολυγώνου και επιστρέφει όλες τις ετικέτες για αυτό.
- Η συνάρτηση *SP_fetch_polygon_values* επιστρέφει όλες τις τημές που έχουν οι ετικέτες ένως πολυγώνου.
- Η συνάρτηση *SP_fetch_store_posession* επιστρέφει την γεωμετρική θέση του καταστήματος.
- Η συνάρτηση *SP_find_attribute_in_db* επιστρέφει για ένα στοιχείο των δημογραφικών δεδομένων των πίνακα και στην στήλη που βρίσκονται τα δεδομένα του.
- Η συνάρτηση *SP_find_municipality_point* δέχεται σαν παράμετρο ένα γεωγραφικό σημείο και επιστρέφει τον δήμο που ανοίκει αυτό.
- Η συνάρτηση *SP_find_polygon_centroids* δέχετεαι σαν παράμετρο ένα id ενώς πολυγώνου που είναι αποθυκευμένο στον πίνακα *B_area* και το αποτελσμά του είναι το κέντο του πολυγώνου σε γεωμετρική αναπαράσταση.

- Η συνάρτηση SP_find_what_is_near επιστρέφει τι βρίσκεται κοντά σε ένα σημείο που το δέχεται σαν παράμετρο. Επιστρέφει τα γεωγραφικά σημεία ή γραμμές που έχουν κάποια ετικέτα αλλά και τιμή σε αυτή.
- Η συνάρτηση *SP_insert_municipality* εισάγει ένα νέο πολύγωνο στην βάση δεδομένων τύπου δήμου.
- Η συνάρτηση *SP_insert_township* εισάγει ένα νέο πολύγωνο στην βάση δεδομένων τύπου κοινώτητα.

Σχεδίαση Business tier

Η σχεδίαση του επιπέδου Business tier όπως αναφέρθηκε και σε προηγούμενο κεφάλαιο διατηρεί το Business logic της εφαρμογής και υπάρχει και ένα ξεχωριστό επίπεδο το Data Access Layer.Σ' αυτή την ενότητα θα αναλυθούν οι κλάσεις που υπάρχουν στο συγκεκριμένο επίπεδο.



Εικόνα 19 class diagram municipality

Στο Class diagram της εικόνας 19 περιγράφονται κλάσεις που έχουν σχέση με την οντότητα δήμος (municipality). Σε όλο το project οι κλάσεις έχουν το πρόθεμα στην αρχή c και τα Interface το **I**. Περιγραφή κλάσεων :

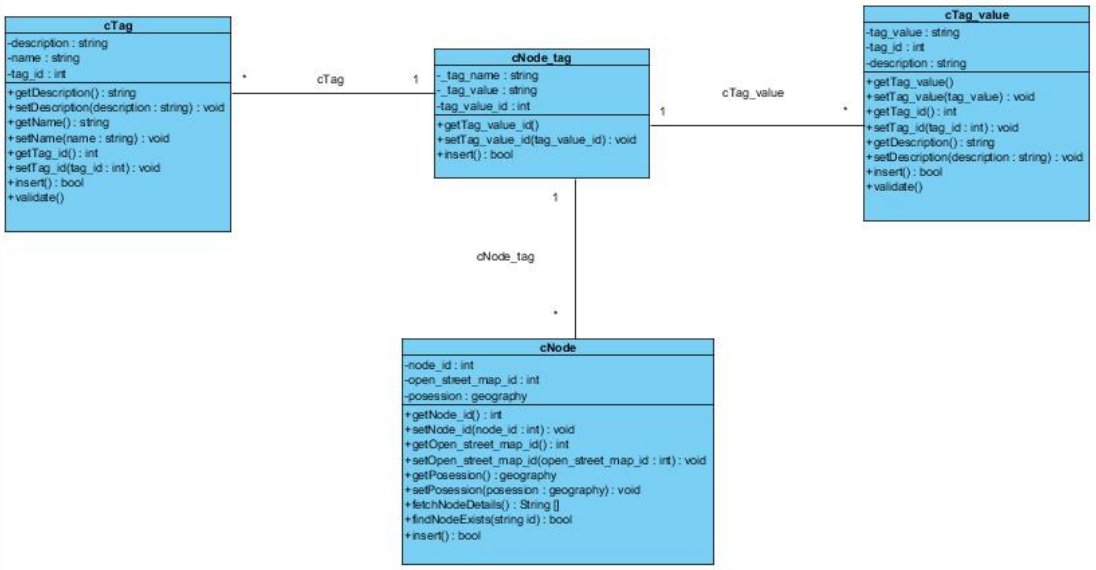
cMunicipality : Η κλάση αυτή περιγράφει την οντότητα δήμος.

cMapLayer : Με την κλάση αυτή υπάρχει πρόσβαση στην οντότητα Map Layer. Με αυτή την οντότητα μπορεί το σύστημα να διαχωρίσει τα γεωγραφικά δεδομένα σε επίπεδα και τα επίπεδα αυτά δημιουργούνται από τον χρήστη.

cPolygon : Η οντότητα αυτή είναι πολύ βασική στο πληροφοριακό σύστημα και διαχειρίζεται τα γεωγραφικά πολύγωνα που είναι αποθηκευμένα στην εφαρμογή. Μπορεί να δημιουργήσει ένα νέο πολύγωνο ή να διαβάσει ένα πολύγωνο από την βάση δεδομένων.

CardinalRanges : Η κλάση αυτή βοηθάει στην διαχείριση γεωγραφικών δεδομένων.

Το επόμενο class diagram που θα αναλυθεί αφορά τα γεωγραφικά σημεία που διαχειρίζεται η εφαρμογή.



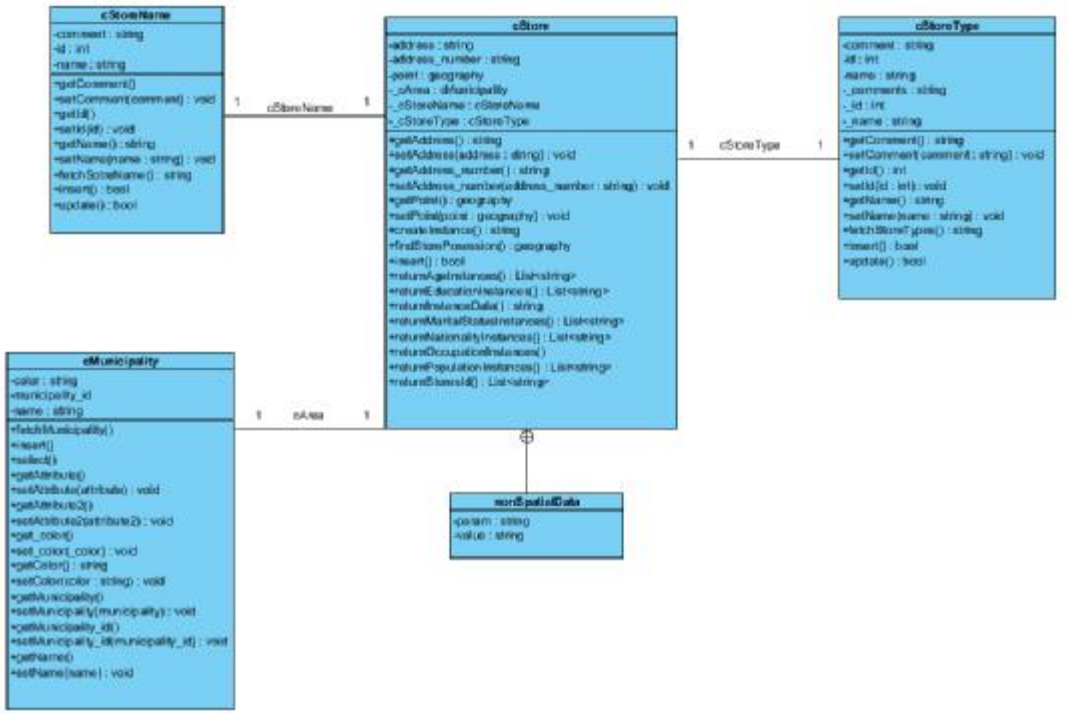
Εικόνα 20 class diagram node

Ανάλυση κλάσεων

cNode : Η κλάση αυτή είναι πολύ σημαντική και διαχειρίζεται τα γεωγραφικά σημεία που έχει αποθηκευμένα η εφαρμογή στην βάση δεδομένων. Με την κλάση αυτή μπορεί να εισαχθεί ένα νέο γεωγραφικό σημείο (γεωγραφική πλάτος και γεωγραφικό μήκος) .

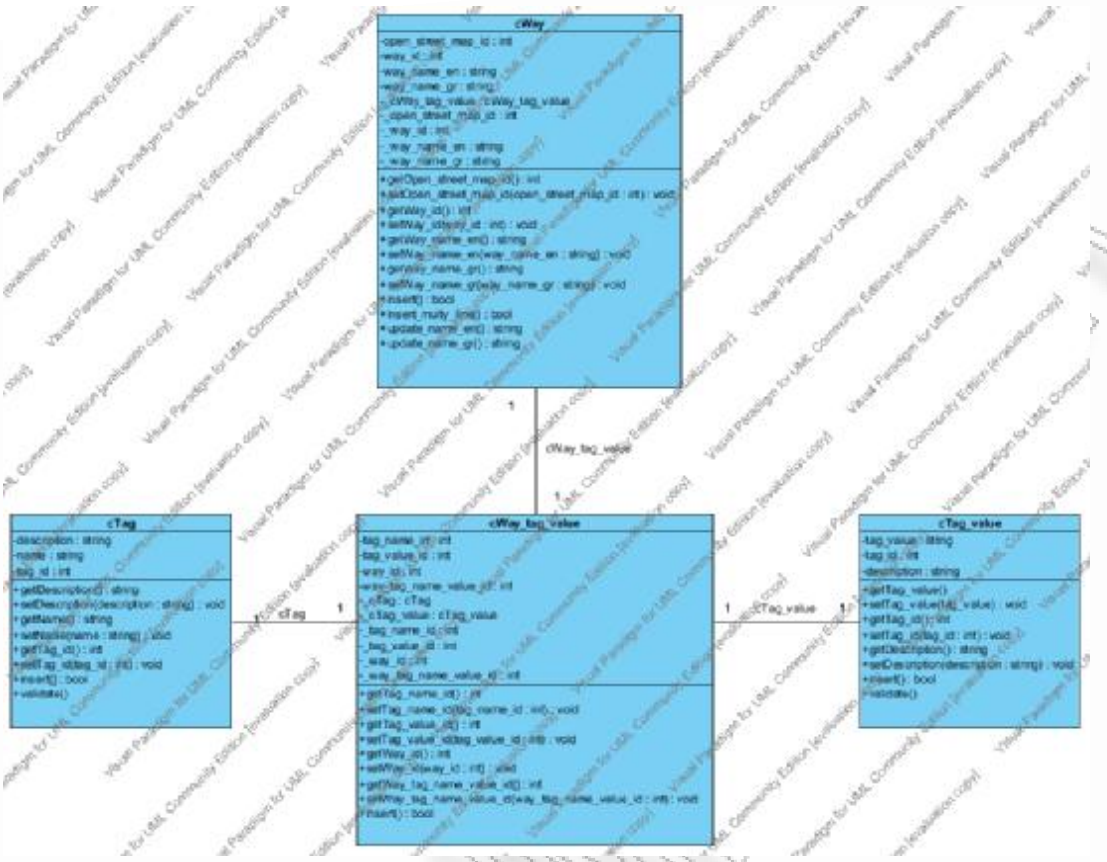
cNode_tag : Η κλάση αυτή διαχειρίζεται την σχέση ανάμεσα στα γεωγραφικά σημεία και στις ετικέτες που έχουν αποθηκευθεί για το συγκεκριμένο σημείο.

Στην εικόνα 20 πραγματοποιείται η ανάλυση των κλάσεων για την οντότητα κατάστημα.



Εικόνα 21 class Diagram store

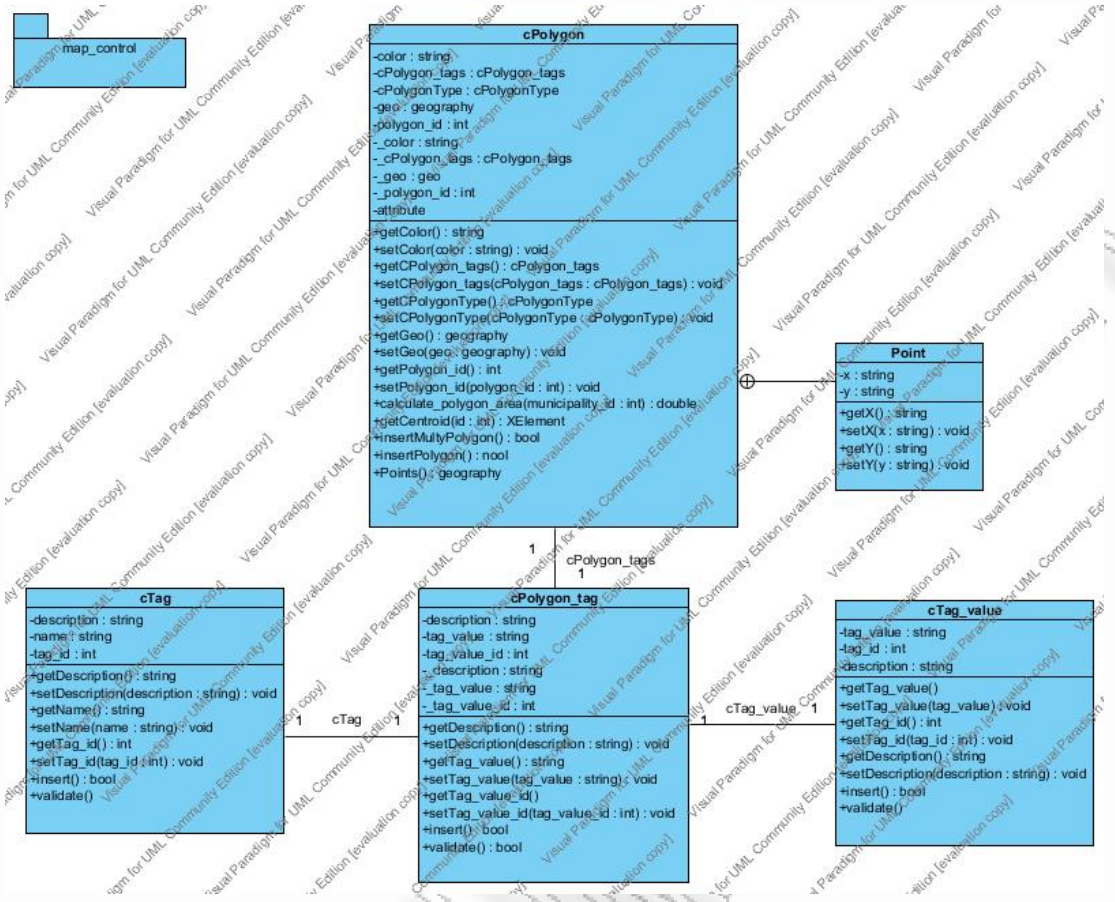
Η βασική οντότητα στο class διάγραμμα είναι η cStore με την οποία περιγράφεται ένα κατάστημα στην βάση δεδομένων. Τα βασικά στοιχεία του είναι η διεύθυνση ο αριθμός διεύθυνσης και η γεωγραφική του θέση. Η κλάση cStoreName διαχειρίζεται το όνομα του καταστήματος και η κλάση cStoreType τον τύπο του καταστήματος. Το κάθε κατάστημα ανήκει σε ένα δήμο για τον λόγο αυτό υπάρχει η σχέση ένα προς ένα με την κλάση cMunicipality.



Εικόνα 22 class diagram way

Στο συγκεκριμένο class diagram αναλύεται η οντότητα δρόμος (Way) στο πληροφοριακό συστήμα.

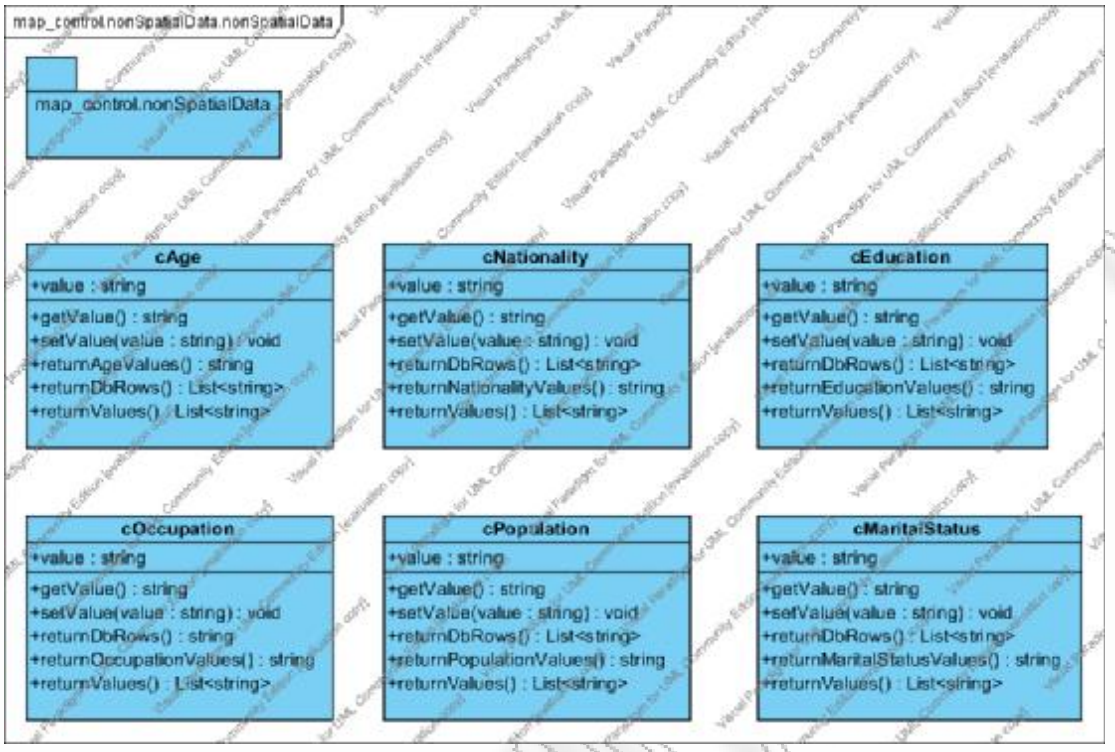
Η βάση κλάση είναι η cWay με την οποία μπορεί να εισάγει μία γραμμή στο στην βάση δεδομένων. Η κλάση δίνει την δυνατότητα εισαγωγής και πολλαπλών γραμμών με την συνάρτηση *insert_munti_line()*.



Εικόνα 23 class Diagram polygon

Στο διάγραμμα της εικόνας 23 αναλύεται η οντότητα πολύγωνο. Στην βάση δεδομένων μπορεί να αποθηκευτεί ένα πολύγωνο ή πολλαπλά. Το πολύγωνο μπορεί να έχει ένα ή περισσότερες ετικέτες και η κάθε ετικέτα να έχει μία τιμή.

Στα προηγούμενα class Diagram περιγράφτηκαν όλες οι βασικές οντότητες με τις οποίες μπορεί να διαχειριστεί ένα γεωγραφικό αντικείμενο. Στα επόμενα διαγράμματα θα αναλυθούν οι κλάσεις που διαχειρίζονται τα δημογραφικά στοιχεία.



Εικόνα 24 class Diagram Δημογραφική στοιχεία

Στο διάγραμμα της εικόνας 24 υπάρχουν οι κλάσεις που διαχειρίζονται τα δημογραφικά στοιχεία. Η κάθε κλάση αντιπροσωπεύει ένα δημογραφικό στοιχείο.

cPopoulation	Η συγκεκριμένη κλάση αντιπροσωπεύει τον πληθυσμό ανά δήμο.
cAge	Η κλάση cAge αντιπροσωπεύει τον αριθμό ανθρώπων για κάθε κλάση ηλικίας.
cNationality	Η κλάση cNationality αντιπροσωπεύει την εθνικότητα για κάθε δήμο.
cEducation	Η κλάση cEducation αντιπροσωπεύει την μόρφωση που έχει κάθε άνθρωπος σε ένα δήμο.
cOcupation	Η κλάση cOcupation αντιπροσωπεύει την απασχόληση που έχει ο κάθε άνθρωπος στον δήμο.
cMaritalStatus	Η κλάση cMaritalStatus αντιπροσοπεύει την οικογενειακή κατάσταση που έχει κάθε άνθρωπος στον δήμο.

Οι συγκεκριμένες έξη κλάσεις έχουν κάποιες συναρτήσεις που είναι κοινές. Η συνάρτηση ***returnValues()*** επιστρέφει τα δεδομένα από την βάση δεδομένων για κάθε δήμο. Η χρήση της συγκεκριμένης συνάρτησης είναι για την εξαγωγή συγκεκριμένων δεδομένων μέσα από τις επιλογές του χρήστη.

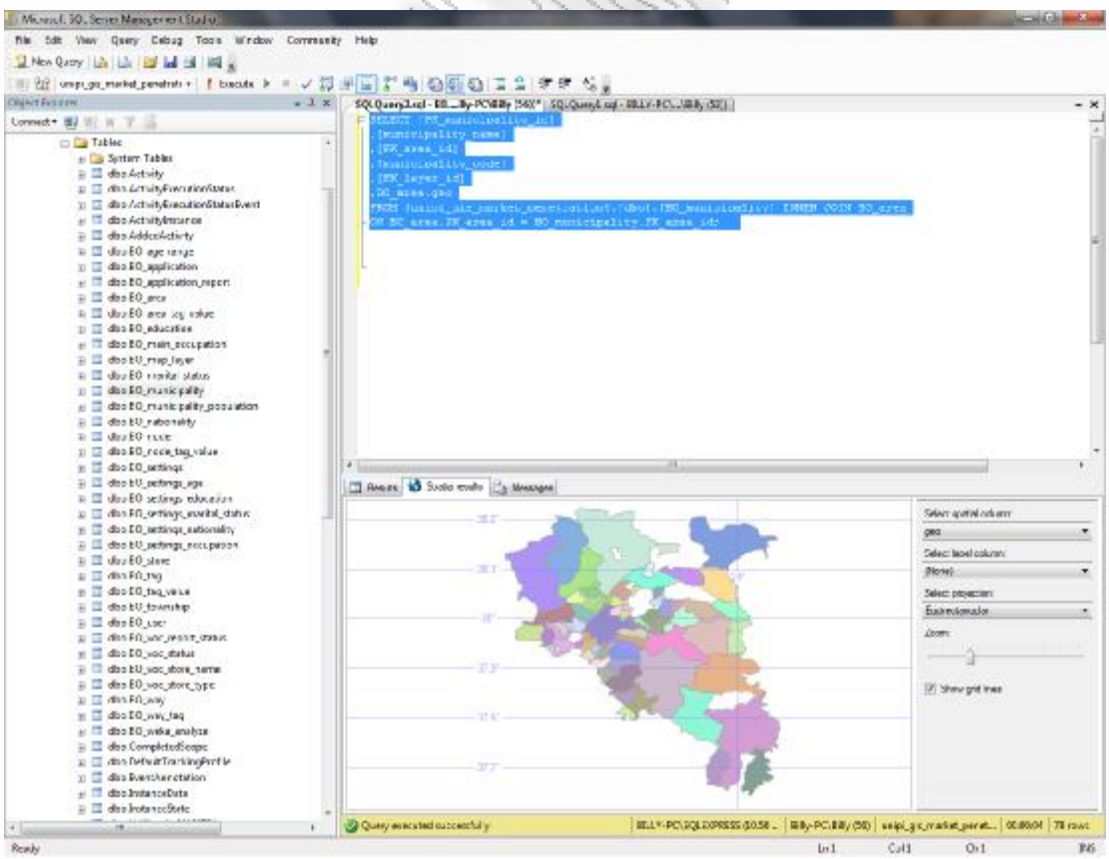
Στο σχεσιακό μοντέλο του πληροφοριακού συστήματος υπάρχουν κάποιοι πίνακες που κάνουν την αντιστοίχηση των δημογραφικών παραμέτρων που έχει επιλέξει ο χρήστης και των πεδίων που έχει ο πίνακας στον SQL Server. Η συνάρτηση *returnDBrows()* επιστρέφει τα πεδία του συγκεκριμένου πίνακα για τις επιλογές που έκανε ο χρήστης.

Η κάθε κλάση έχει μία συνάρτηση που χρησιμοποιείται για την εξαγωγή στοιχείων για κάθε δημογραφικό δεδομένο. Η συγκεκριμένη συνάρτηση θα αναλυθεί σε βάθος για την κατανόηση της.

Το πρώτο πρόβλημα που δημιουργήθηκε είναι ότι στην βάση δεδομένων υπήρχαν δημογραφικά στοιχεία ανά δήμο. Αν επιλέξουμε πέντε μαγαζιά μέσα στον ίδιο δήμο τότε θα έχουν τα ίδια δημογραφικά στοιχεία. Από τον Sql Server εκτελέσουμε το ακόλουθο query τότε θα αναδείξει τους δήμους που είναι αποθηκευμένοι στην βάση δεδομένων.

```
SELECT [PK_municipality_id]
,[municipality_name]
,[FK_area_id]
,[municipality_code]
,[FK_layer_id]
,BO_area.geo
FROM [unipi_gis_market_penetration].[dbo].[BO_municipality] INNER
JOIN BO_area
ON BO_area.PK_area_id = BO_municipality.FK_area_id;
```

Το αποτέλεσμα του ερωτήματος εμφανίζεται στην επόμενη εικόνα.



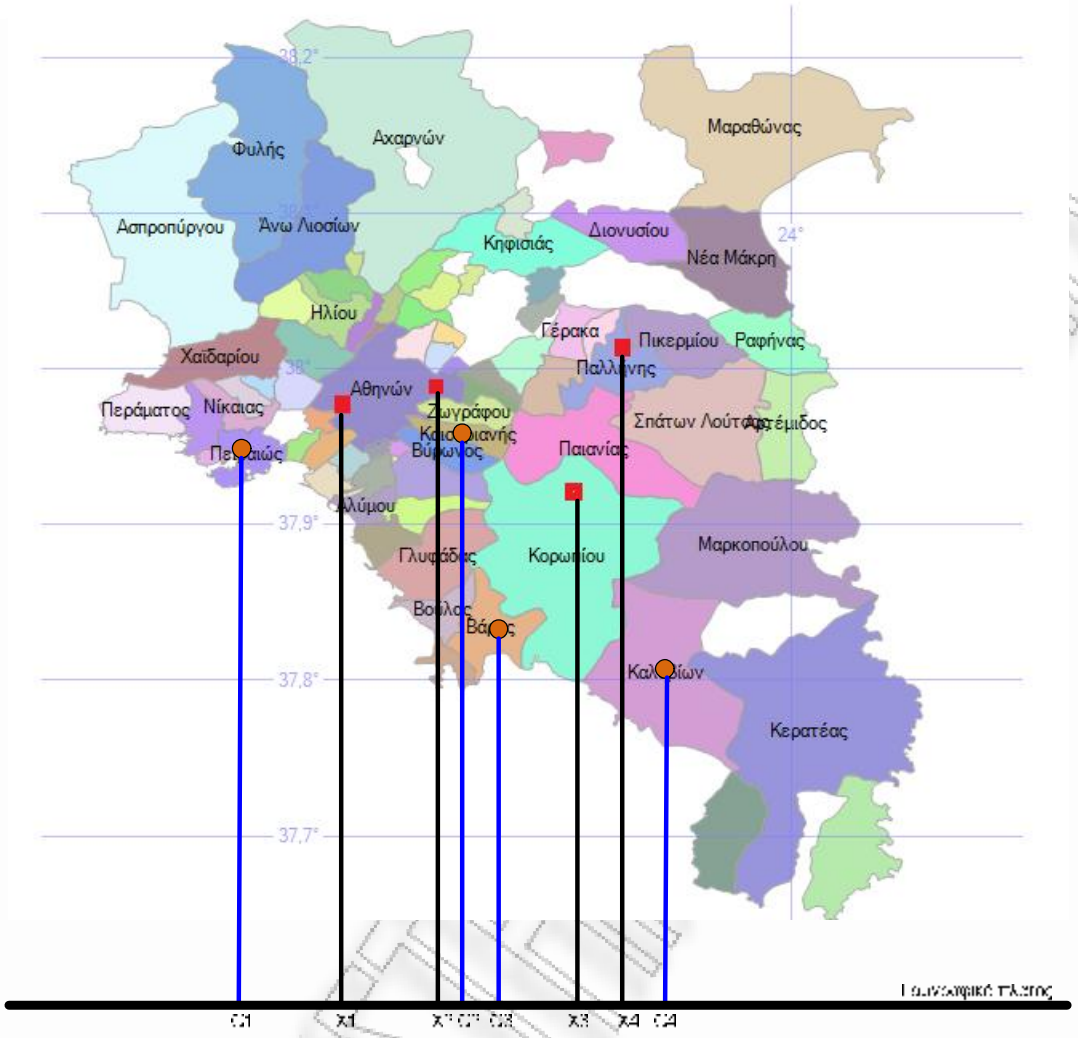
Εικόνα 25 Sql ερώτημα για τους δήμους

Για κάθε δήμο που είναι αποθηκευμένος στην βάση δεδομένων υπάρχει και οι αντίστοιχες τιμές στα δημογραφικά στοιχεία. Η αντιμετώπιση στο πρόβλημα που περιγράφτηκε πριν αντιμετωπίζεται με το Interpolation

Ανάλυση αριθμών – θεώρημα παρεμβολής

Η ανάλυση αριθμών είναι μία μέθοδος η οποία ανήκει στην μαθηματική ανάλυση. Η μέθοδος αυτή μπορεί να δημιουργεί νέα σημεία μέσα σε ένα εύρος διακριτών γνωστών τιμών. Κατά την συλλογή δεδομένων που γίνονται σε ένα project συλλέγονται χιλιάδες σημεία από τα οποία πρέπει να δημιουργηθεί μία συνάρτηση η οποία να είναι κοντά στα σημεία. Αυτή η τεχνική ονομάζεται curve fitting και το θεώρημα παρεμβολής είναι ένα μέρος αυτής της τεχνικής. Η ιδιαιτερότητα της συγκεκριμένης τεχνικής είναι ότι η συνάρτηση που θα δημιουργηθεί πρέπει να διασχίζει όλα τα σημεία.

Στην εφαρμογή υπάρχουν αποθηκευμένοι οι δήμοι τις αττικής. Στους δήμους υπάρχουν τα μαγαζιά. Αν τεθεί στο ερώτημα μία ιδιότητα από τα δημογραφικά στοιχεία π.χ. τι τιμή θα έχει στο συγκεκριμένο κατάστημα κατά προσέγγιση; Τα δημογραφικά στοιχεία που υπάρχουν αποθηκευμένα στην βάση δεδομένων είναι για ολόκληρο τον δήμο και όχι για ένα σημείο. Οπότε το πρόβλημα που αντιμετωπίζεται είναι τι τιμή θα έχει το συγκεκριμένο γεωγραφικό σημείο. Η αναπαράσταση του προβλήματος εμφανίζεται και στην επόμενη εικόνα.



Εικόνα 26 GIS Interpolation

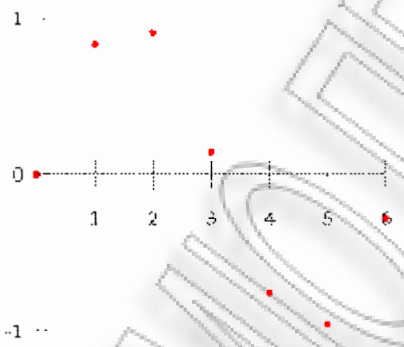
Στην εικόνα 26 τα *X1, X2, X3, X4* είναι τα γεωγραφικά σημεία των καταστημάτων που υπάρχουν στην βάση δεδομένων. Τα *C1, C2, C3, C4* είναι τα κέντρα των δήμων. Ο κάθε δήμος είναι ένα πολύγωνο που έχει ένα κέντρο. Για τον υπολογισμό του πολυγώνου χρησιμοποιείται η συνάρτηση `[dbo].[SP_find_polygon_centroids]` η οποία παίρνει μία παράμετρο, το id του δήμου. Πάρδειγμα εκτέλεσης της συνάρτησης εμφανίζεται στον επόμενο πίνακα.

```
EXEC  @return_value = [dbo].[SP_find_polygon_centroids] @id = 60;
```

Γνωρίζοντας τα κέντρα του κάθε δήμου στο σύστημα παίρνει μία παραδοχή. Όπως αναφέρθηκε και πρίν για παράδειγμα στον δήμο με id=50 το ποσό των ανθρώπων που ανοίκει στο εύρος ηλικίας 5-9 είναι 200. Αυτό θεωρήται ότι βρίσκεται στο κέντρο του δήμου και όσο απομακρινεται από αυτό τότε μικρένει η τιμή αυτή. Οπότε η τιμή του C1 για το εύρος ηλικίας 5-9 είναι 200 πόσο θα είναι στο X1; Το θετικό είναι ότι γνωρίζουμε για όλους τους δήμους την αττικής τα κέντρα του και για κάθε κέντρο την τιμή του. Άρα γνωρίζοντας αυτές τις τιμές (οι οποίες έιναι διακριτές) θα πρέπει να βρεθεί η συνάρτηση για να δωθεί η τιμή που έχει στο σημείο X1. Η λύση στο πρόβλημα αυτό είναι χρησιμοποιώντας από την αριθμιτική ανάλυση το θεώρημα της παρεμβολής.

Ένα παράδειγμα του θεωρήματος εμφανίζεται στον επόμενο πίνακα.

x	F(x)
0	0
1	0.8415
2	0.9093
3	0.1411
4	-0.7685
5	-0.9589
6	-0.2794

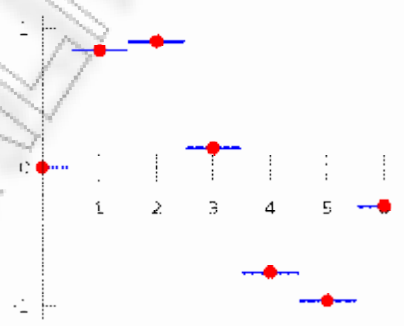


Εικόνα 27 Παράδειγμα interpolation

Στην εικόνα 26 εμφανίζεται η αναπαράσταση των τιμών του παραπάνω πίνακα. Θα πρέπει να φρεθεί η τιμή *F(2.5)* δηλαδή για X=2,5 τι τιμή έχει το F(x); Για την επίλυση του προβλήματος υπάρχουν τρείς τρόποι.

Πρώτος τρόπος επίλυσης

Ο πρώτος τρόπος επίλυσης είναι το σημείο που επιθυμούμε να μάθουμε την τιμή του θα εξαρτηθεί από πίο σημείο είναι κοντά σ' αυτό. Σ' αυτό το σημείο που θα βρεθεί πίο κοντά θα πάρει και την τιμή που έχει. Είναι ο πίο εύκολος τρόπος επίλυσεις του προβλήματος.



Εικόνα 28 Παράδειγμα interpolation Πρώτος τρόπος προσέγγισης επίλυσης

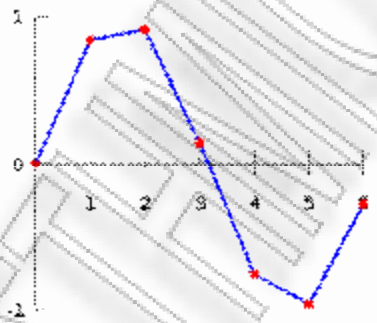
Δεύτερος τρόπος επίλυσης

Ο δεύτερος τρόπος επίλυσης προβλήματος είναι η γραμμική προσέγγιση. Και αυτός ο τρόπος είναι απλός. Στο παράδειγμα του προβλήματος πρέπει να βρεθεί η τιμή του F(x) για x=2.5. Το 2.5 είναι ανάμεσα στο 2 και στο 3 αν το F(2) έχει την τιμή 0.9093 και το F(3) έχει την τιμή 0.1411 τοτε το F(2.5) θα πάρει τιμή ανάμεσα στο F(2) και F(3). Ο γενικός τύπος για την επίλυση εμφανίζεται στην επομενη εικόνα.

$$y = y_a + (y_b - y_a)\frac{(x - x_a)}{(x_b - x_a)}$$

Εικόνα 29 Παράδειγμα interpolation Δεύτερος τρόπος προσέγγισης επίλυσης - τύπος γραμμικής προσέγγισης

Η χρήση της γραμμικής προσέγγισης είναι γρήγορη αλλά δεν είναι ακριβής. Η γραφική αναπαράσταση της γραμμικής προσέγγισης εμφανίζεται στην επόμενη εικόνα.



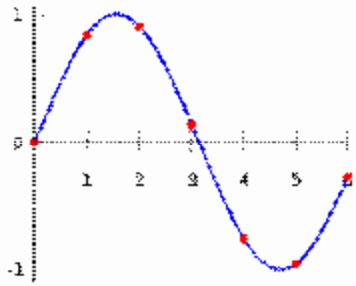
Εικόνα 30 Γραφική παράσταση γραμμικής προσέγγισης

Τρίτος τρόπος επίλυσης

Ο τρίτος τρόπος επίλυσης του προβλήματος είναι μέσω πολυωνύμου. Είναι μία γενίκευση σε σχέση με την γραμμική προσέγγιση. Κάθε στοιχείο του πίνακα το αντικαθιστούμε με ένα υψηλότερο βαθμό στο πολυώνυμο.

f(x) = − 0.0001521x6 − 0.003130x5 + 0.07321x4 − 0.3577x3 + 0.2255x2 + 0.9038x.

Γενικά αν υπάρχουν *n* δεδομένα τότε το πολυώνυμο που θα διασχίζει όλα σημεία θα είναι βαθμού *n-1*. Η συγκεκριμένη προσέγγιση είναι καλύτερη από την γραμμική. Η γραφική αναπαράσταση εμφανίζεται στην εικόνα 31.



Εικόνα 31 Γραφική παράσταση πολυώνυμου

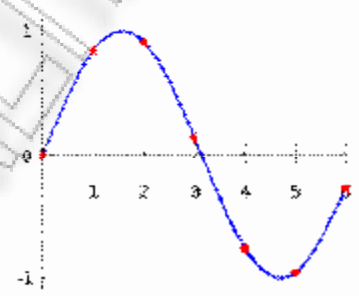
Τέταρτος τρόπος επίλυσης

Ο τέταρτος τρόπος επίλυσης είναι με την μέθοδο SPLine. Για την επίλυση χρησιμοποιούνται χαμηλών βαθμών πολυώνυμα. Τα πολυώνυμα δημιουργούνται ανάμεσα στα σημεία. Με βάση το παράδειγμα θα δημιουργηθούν τα αντίστοιχα πολυώνυμα.

$$f(x) = \begin{cases} -0.1522x^3 + 0.9937x, & \text{if } x \in [0,1], \\ -0.01258x^3 - 0.4189x^2 + 1.4126x - 0.1396, & \text{if } x \in [1,2], \\ 0.1403x^3 - 1.3359x^2 + 3.2467x - 1.3623, & \text{if } x \in [2,3], \\ 0.1579x^3 - 1.4945x^2 + 3.7225x - 1.8381, & \text{if } x \in [3,4], \\ 0.05375x^3 - 0.2450x^2 - 1.2756x + 4.8259, & \text{if } x \in [4,5], \\ -0.1871x^3 + 3.3673x^2 - 19.3370x + 34.9282, & \text{if } x \in [5,6]. \end{cases}$$

Εικόνα 32 SPLine πολυώνυμα

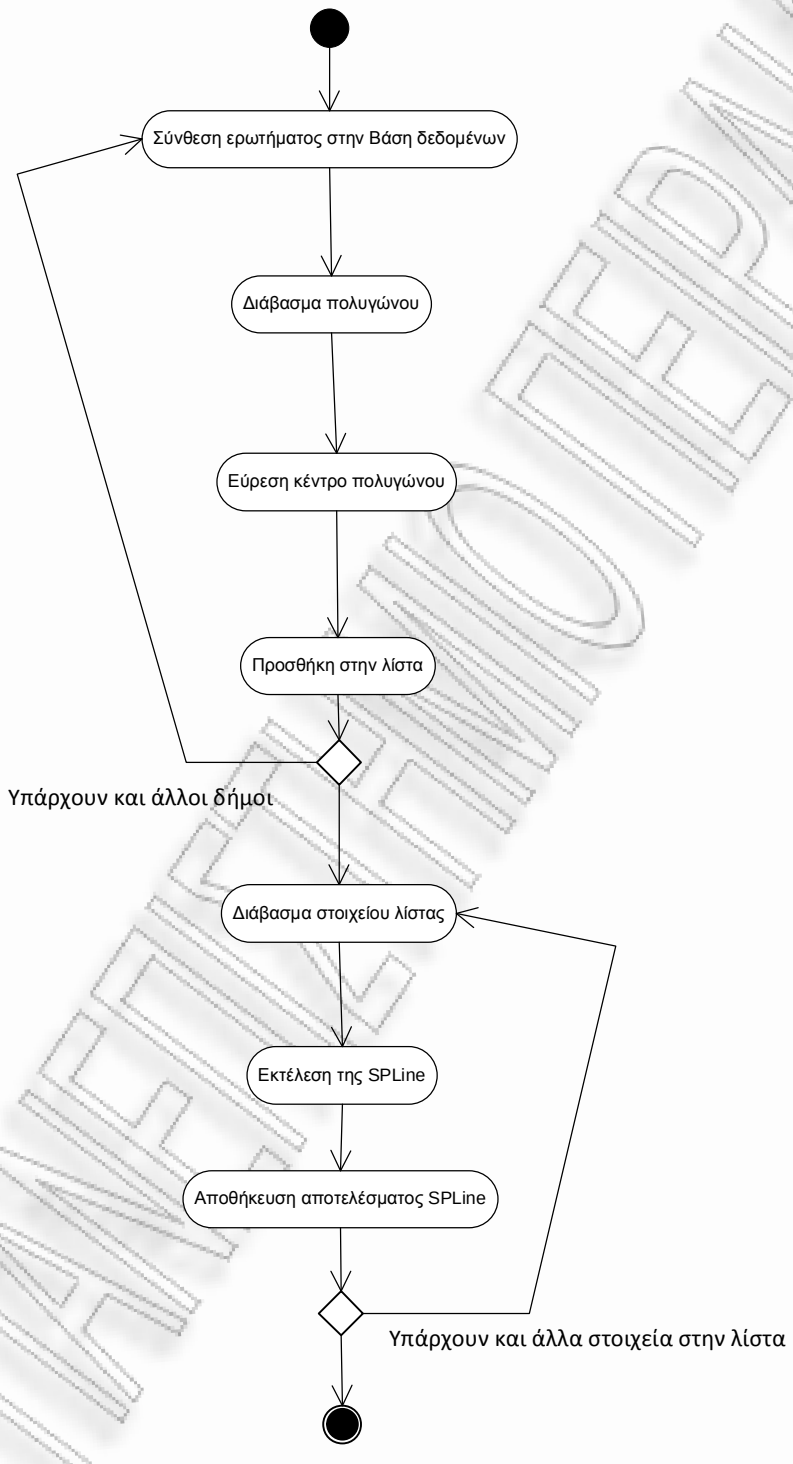
Με την χρήση της SPLine συνάρτηση υπάρχει το μικρότερο σφάλμα σε σχέση με τις άλλες προσεγγίσεις. Η γραφική αναπαράσταση εμφανίζεται στην επόμενη εικόνα 32.



Εικόνα 33Γραφική παράσταση πολυώνυμου SPLine

Η συγκεκριμένη προσέγγιση χρησιμοποιήθηκε και από την εφαρμογή για την εύρεση των σημείων X2,X2 κτλ.

Στο επόμενο activity διάγραμμα αναπαριστάτε το πώς αναζητούνται στην βάση δεδομένων τα δημογραφικά στοιχεία και πώς με την χρήστη της συνάρτησης SPLine βρίσκεται η τιμή που έχει το συγκεκριμένο σημείο στον δήμο.



Εικόνα 34 Activity diagram SPLine

Διαχείριση Δημογραφικών δεδομένων

Ο χρήστης εφόσον επιλέξει τα στοιχεία που θέλει να χρησιμοποιήσει για ομαδοποίηση των δεδομένων θα τα στείλει στον server για εξεργασία. Η κλάση που χειρίζεται το συγκεκριμένο μήνυμα εμφανίζεται στην επόμενη εικόνα.

Εικόνα 35 κλάση διαχείρισης Xml μηνύματος

Στην εικόνα 35 αναπαριστάτε η κλάση που διαχειρίζεται το αίτημα του χρήστη. Δηλαδή η συγκεκριμένη κλάση δημιουργεί το Arff αρχείο για να αποθηκεύσει στο WEKA. Η συγκεκριμένη κλάση είναι στατική και έχει τρείς συναρτήσεις με τις οποίες δημιουργεί το αρχείο στην μνήμη του Server.

Η συνάρτηση setXml() αποθηκεύει σε μία μεταβλητή το αίτημα του χρήστη. Τα αιτήματα που στέλνει ο χρήστης μέσω της εφαρμογής είναι τύπου XML θα αναλυθεί η δημιουργία του στο presentation layer όπως και η ανάλυση του σε ξεχωριστό κεφάλαιο.

Το αρχείο που πρέπει να δημιουργηθεί για το Weka είναι τύπου arff Η ανάλυση του θα γίνει σε ξεχωριστό κεφάλαιο. Στο συγκεκριμένο κεφάλαιο θα δοθεί σημασία στον αλγόριθμο δημιουργίας του αρχείου. Η μορφή του χωρίζεται σε δύο μεγάλα μέρη, το πρώτο είναι η περιγραφή ιδιοτήτων των στιγμιότυπων που έχει το αρχείο και το δεύτερο μέρος είναι οι τιμές αυτών. Για τον λόγο αυτό κατασκευάστηκαν δύο διαφορετικές συναρτήσεις η πρώτη έχει το όνομα *createAttributes* και η δεύτερη έχει το όνομα *createArffFile()*.

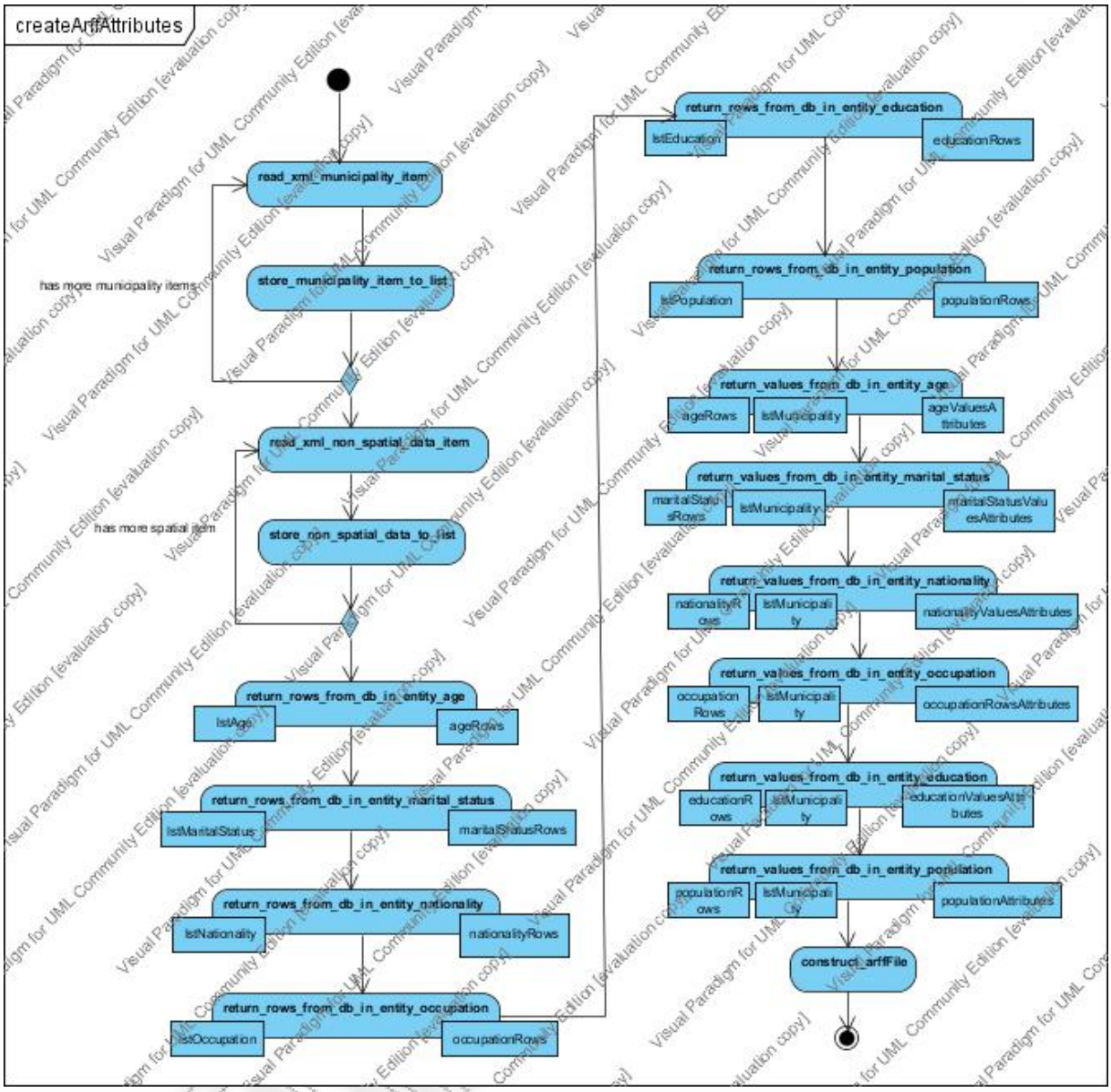
Όπως αναφέρθηκε και πριν η συνάρτηση *createAttribute()* δημιουργεί τις ιδιότητες των instances. Παράδειγμα ενός instance είναι το ακόλουθο

@attribute [*όνομα ιδιότητας*] [*τύπος ιδιότητας*]

Οπότε η συνάρτηση θα πρέπει να διαβάζει το xml αρχείο, να διαβάζει τις ιδιότητες που έχει επιλέξει ο χρήστης και να τις τοποθετεί στο αρχείο. Η ανάλυση για την δημιουργία του αλγόριθμου το περιγράφει το επόμενο Activity Diagram.

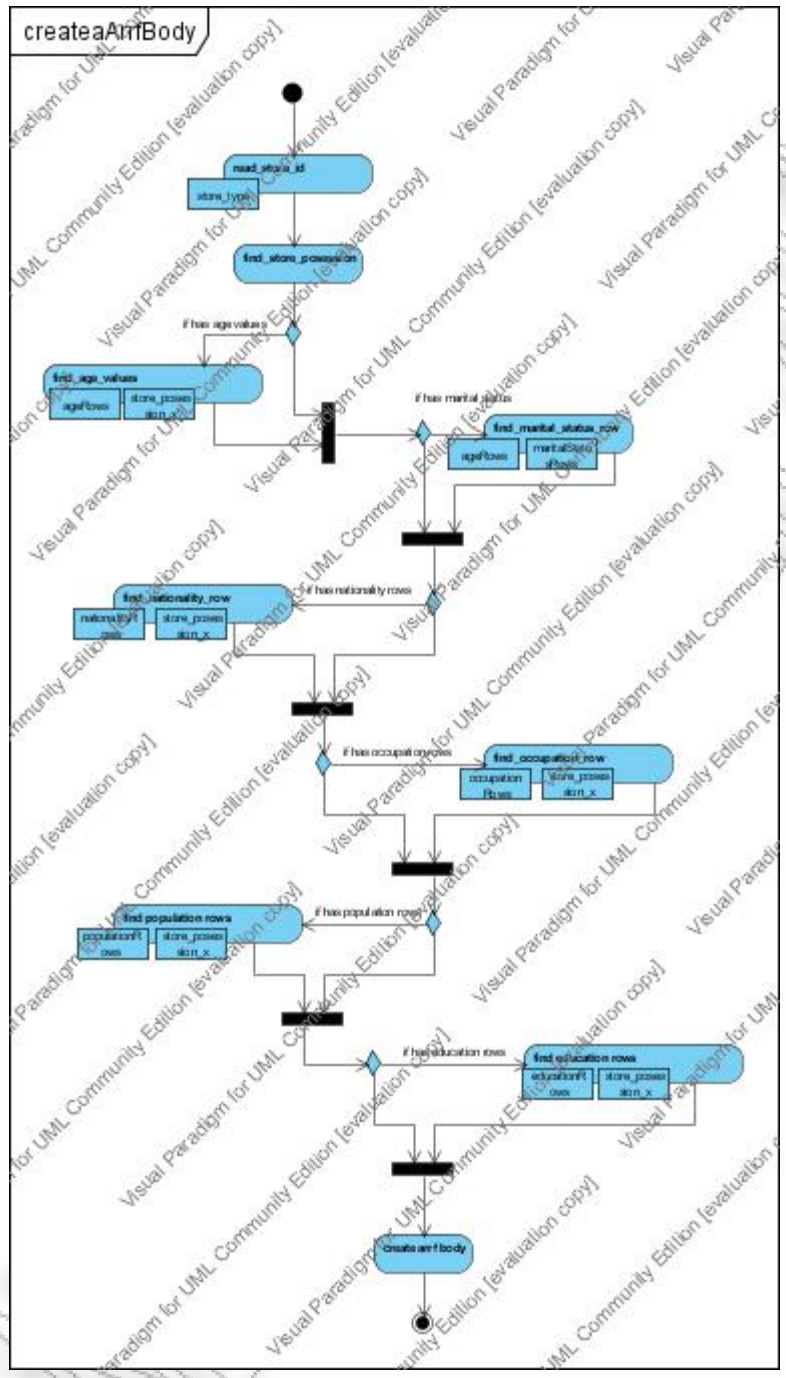
Στο παρακάτω διάγραμμα υπάρχει η ανάλυση του αλγόριθμου. Μπορεί να χωριστεί σε τρία βασικά μέρη. Το πρώτο μέρος είναι η συλλογή δεδομένων από το αίτημα του χρήστη. Συλλέγονται οι δήμοι που έχει επιλέξει ο χρήστης και τα δημογραφικά στοιχεία. Αυτά αποθηκεύονται σε λίστες στην μνήμη. Το επόμενο βήμα είναι να βρεθούν τα πεδία των πινάκων από τα δημογραφικά στοιχεία για να δημιουργηθούν οι αντίστοιχες ιδιότητες των στιγμιότυπων. Τέλος το τελευταίο βήμα είναι να βρεθούν οι τιμές που έχουν οι ιδιότητες αυτές για τους συγκεκριμένους δήμους.

Υπάρχει ένα ξεχωριστό Activity diagram το οποίο δημιουργεί στο αρχείο arff τις ιδιότητες.



Εικόνα 36 Activity diagram construct arffFile

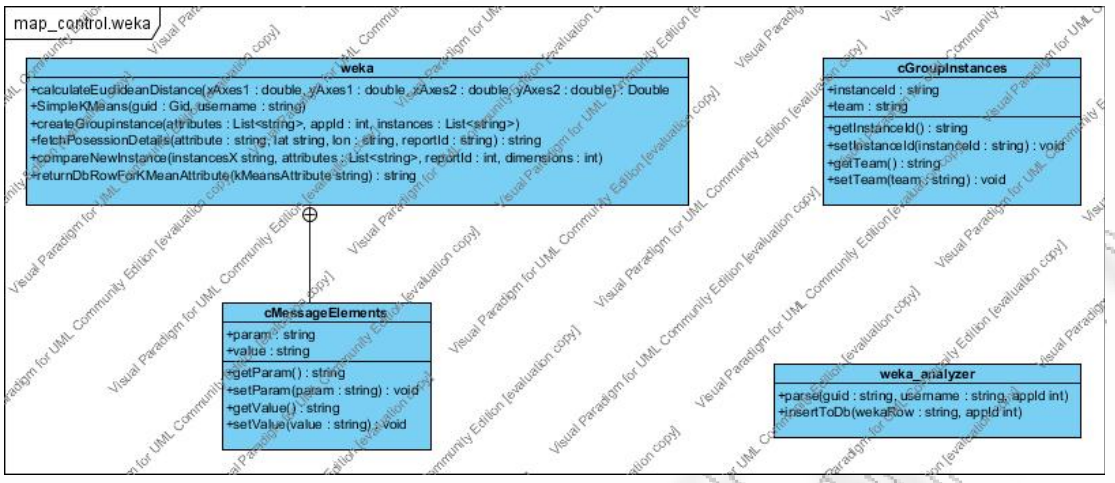
Στην επόμενη συνάρτηση δημιουργείται το σώμα του arff αρχείου. Αυτό αποτελείται από όλα τα στιγμιότυπα της εφαρμογής. Τα στιγμιότυπα είναι καταστήματα που έχει η εφαρμογή. Η εύρεση του καταστήματος γίνεται από τους δήμους που έχει επιλέξει ο χρήστης. Για να βρεθούν τα καταστήματα πρέπει να δοθούν δύο παράμετροι. Η πρώτη παράμετρος είναι οι δήμοι που έχει επιλέξει ο χρήστης. Η δεύτερη παράμετρος είναι ο τύπος του καταστήματος. Στο επόμενο Activity diagram περιγράφεται η ανάλυση του αλγόριθμου.



Εικόνα 37 Activity diagram create arffbody

Η συγκεκριμένη συνάρτηση πρέπει να εκτελεστεί μετά από την δημιουργία του *createAttribute()*. Από την προηγούμενη συνάρτηση έχει αποθηκευτεί στην μνήμη για κάθε δημογραφικό στοιχείο που έχουμε στην εφαρμογή η τιμή του. Με την εκτέλεση της συνάρτησης συλλέγονται τα αντίστοιχα δεδομένα από την βάση για να δημιουργηθεί το συγκεκριμένο στιγμιότυπο.

Το επόμενο Class diagram που θα αναλυθεί είναι το διάγραμμα που αφορά το Weka. Την εκτέλεση του Data Mining αλλά και την εκτέλεση reports.



Εικόνα 38 Weka Class Diagram

Το συγκεκριμένο διάγραμμα περιγράφει κλάσεις οι οποίες είναι ο βασικός πυρήνας για την εκτέλεση του Data Mining. Η κλάση Weka έχει κάποιες συναρτήσεις, η βασική της λειτουργία είναι να δέχεται ένα αρχείο arff και να παράγει τις ομάδες «clusters».

Ανάλυση συναρτήσεων :

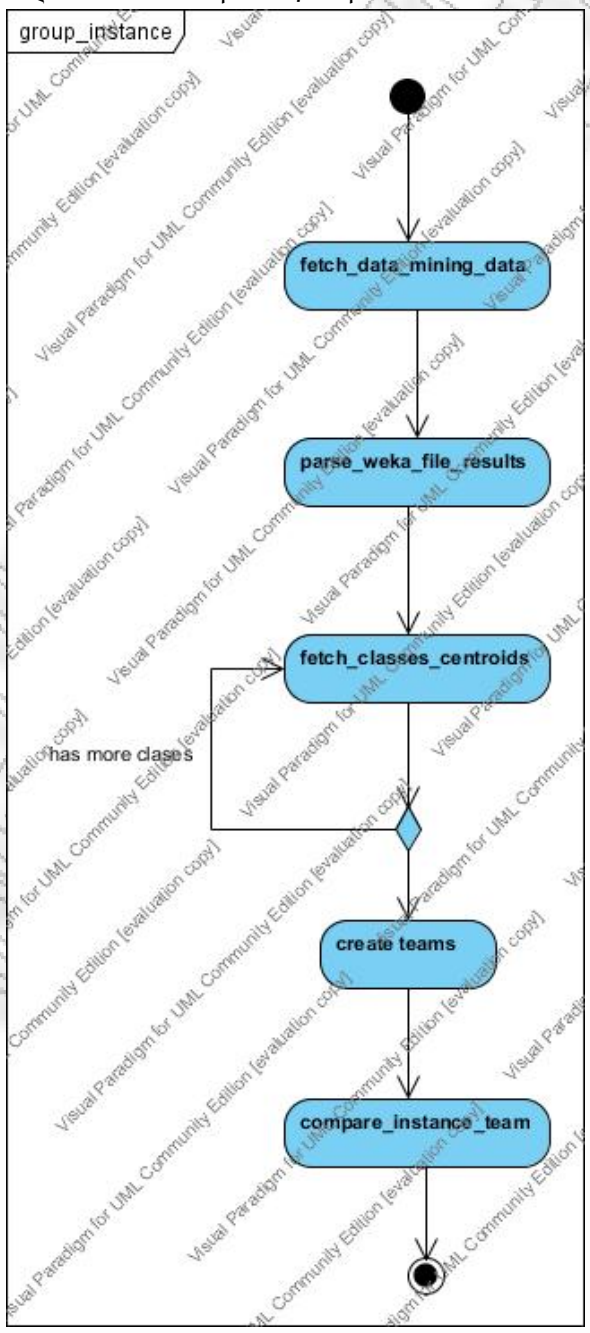
- Η Συνάρτηση SimpleKMeans είναι η πιο σημαντική. Η βασική της λειτουργία είναι να συλλέξει από την βάση δεδομένων τα στοιχεία που χρειάζεται για να εκτελέσει το αλγόριθμο K-Means. Τα στοιχεία αποθηκεύονται στο arff αρχείο που αποτελείτε από τα στιγμιότυπα. Συλλέγοντας όλα τα δεδομένα το Weka θα δημιουργήσει ένα αρχείο το οποίο θα έχει τις ομάδες και τα centroid κάθε ιδιότητας.

- Η συνάρτηση *calculateEuclideanDistance* μπορεί να δεχτεί δύο σημεία και να βρει την ευκλείδεια απόσταση.

- Η συνάρτηση *returnDbRowForKMeanAttribute* επιστρέφει τις τιμές που έχει μία τιμή δημογραφικού στοιχείου στου πίνακες της βάσης.

- Η συνάρτηση *createGroupinstance* επιστρέφει μία λίστα τύπου string με τις ιδιότητες των στιγμιότυπων που έχει η αίτηση του χρήστη. Για την εκτέλεση της πρέπει να δοθούν τρείς παράμετροι. Η πρώτη παράμετρος είναι τα δημογραφικά στοιχεία που έχει επιλέξει ο χρήστης για την ανάλυση δεδομένων. Η μορφή της συγκεκριμένης παραμέτρου είναι λίστα τύπου αλφαριθμητικού. Η συγκεκριμένη λίστα χρειάζεται για να βρεθούν από την βάση δεδομένων οι τιμές των δημογραφικών στοιχείων που επέλεξε ο χρήστης. Η δεύτερη παράμετρος είναι ένας ακέραιος αριθμός. Ο αριθμός αυτός αντιπροσωπεύει την συγκεκριμένη αίτηση του χρήστη. Τρίτη παράμετρος είναι τα στιγμιότυπα που έχει επιλέξει ο χρήστης για την ανάλυση του. Το αποτέλεσμα της συνάρτηση θα είναι μία λίστα αλφαριθμητικών. Οι τιμές τις λίστας θα είναι οι τιμές που έχουν τα δημογραφικά στοιχεία για την συγκριμένη αίτηση που έχει τα συγκεκριμένα στιγμιότυπα.

- Η συνάρτηση *compareInstance* μπορεί να συγκρίνει ένα στιγμιότυπο σε σχέση με τα κέντα των κλάσεων που έχουν δημιουργηθεί από την εκτέλεση του αλγόριθμου SimpleKMeans. Το αποτέλεσμα της συνάρτησης θα είναι μία λίστα αλφαριθμητικών που θα περιέχει τις ομάδες των στιγμιότυπων που έχουν δημιουργηθεί και καθ' ένα από αυτά σε ποια ομάδα ανήκει. Για την εκτέλεση της πρέπει να δοθούν τρείς παράμετροι. Ο πρώτος είναι τα στιγμιότυπα τα οποία θα ομαδοποιηθούν, δεύτερη είναι τα δημογραφικά στοιχεία, δηλαδή οι διαστάσεις που έχουν επιλεχτεί από την εκτέλεση του Data Mining, τέταρτη παράμετρος είναι ο αριθμός της αίτησης που έχει κάνει ο χρήστης. Με δεδομένα τις παραμέτρους μπορεί η συνάρτηση να εξάγει μία λίστα με τις ομάδες που έχουν δημιουργηθεί αλλά και κάθε στιγμιότυπο σε ποια ομάδα ανήκει. η εκτέλεση του αλγόριθμου αναπαριστάτε και στην επόμενη εικόνα, ένα Activity Diagram.



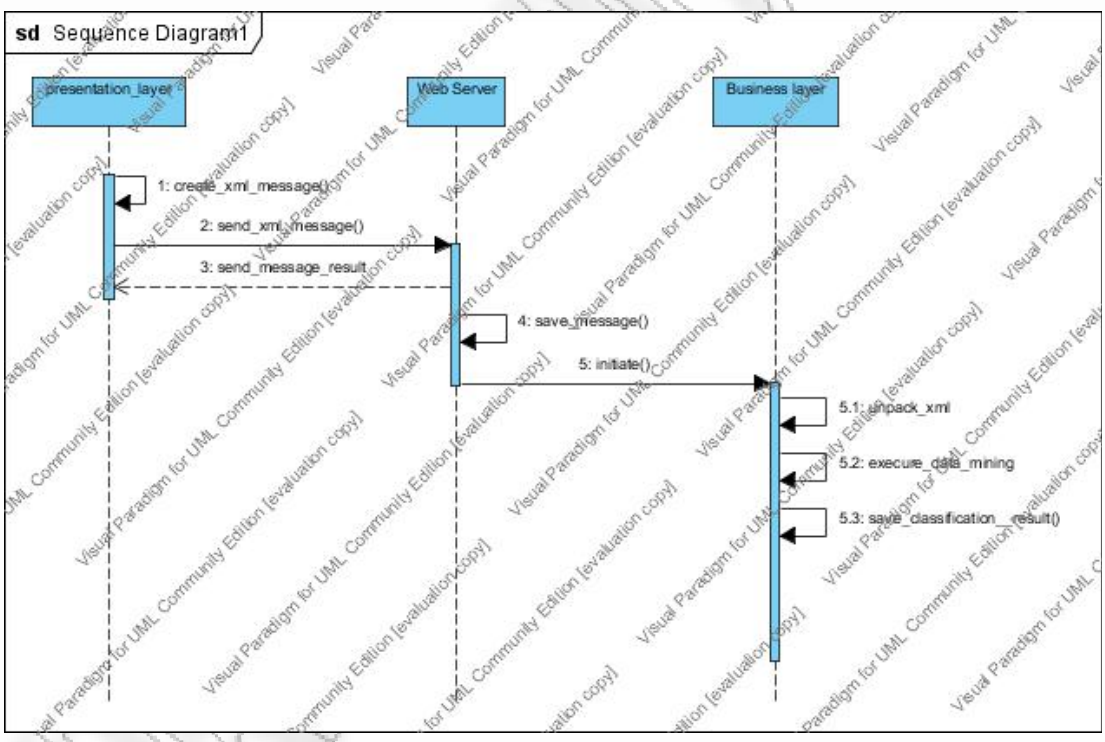
Εικόνα 39 Activity Diagram Group Instance συνάρτηση

- Η συνάρτηση *compareNewInstance* συγκρίνει ένα νέο στιγμιότυπο με το τις υπάρχουσες ομάδες από την εκτέλεση του Data Mining μέσω του αλγόριθμου SimpleKMeans. Το αποτέλεσμα της συνάρτησης είναι η κατάταξη του συγκεκριμένου στιγμιότυπου σε μία από τις ομάδες.

Εκτέλεση Data Mining - SimpleKMeans αλγόριθμου

Σε όλοι την προηγούμενη ενότητα έχει γίνει ανάλυση όλων των κλάσεων που έχει το Business logic και σε αυτή την ενότητα θα αναλυθεί πως αλληλεπιδρούν τα στιγμιότυπα των κλάσεων για να εκτελεστεί το Data Mining. Ο κεντρικός εξυπηρετητής έχει ένα Web Service ο οποίος δέχεται τα αιτήματα από των client. Στο Back end υπάρχει ένας αλγόριθμος ο οποίος επεξεργάζεται το αίτημα και εκτελεί το Data Mining.

Στην επόμενη εικόνα πραγματοποιείται μία ανάλυση της διαδικασίας εκτέλεσης για το διάβασμα του αιτήματος και την εκτέλεση του Data Mining.



Εικόνα 40 Sequence diagram xml message

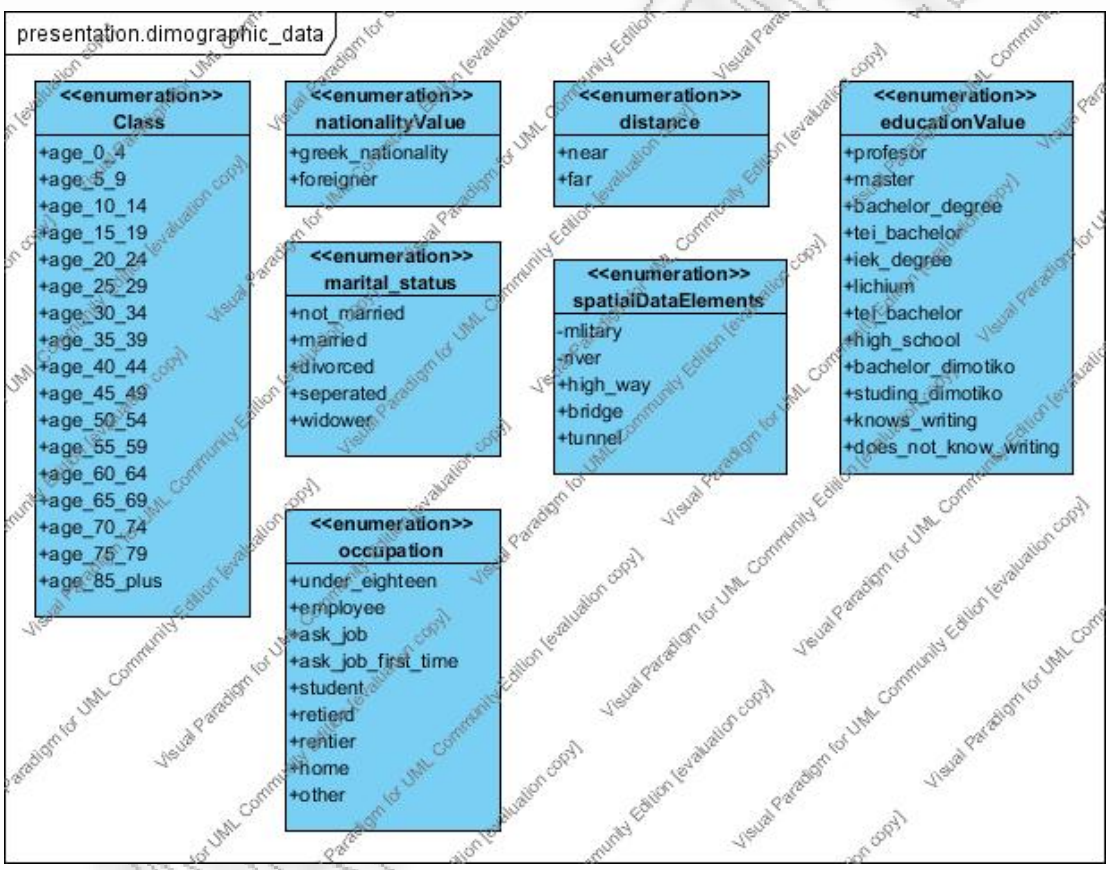
Το συγκεκριμένο μήνυμα έχει την μορφή XML αρχείου για να είναι σωστό και εύκολα αναγνωρίσιμο τόσο από άνθρωπο αλλά και από μηχανή.

Σχεδίαση Presentation tier

Business object on presentation layer

Όπως έχει αναφερθεί ένα από τα πλεονεκτήματα που έχει η χρήση του Silverlight είναι ένα μέρος του Business logic μπορεί να μεταφερθεί στον client. Δημιουργήθηκαν κάποιες κλάσεις οι οποίες βοηθούν στην συλλογή δεδομένων από τον χρήστη. Ο χρήστης μπορεί να επιλέξει κάποια στοιχεία από τα δημογραφικά δεδομένα. Ο κλάσεις αυτές βοηθούν στην συλλογή των δεδομένων.

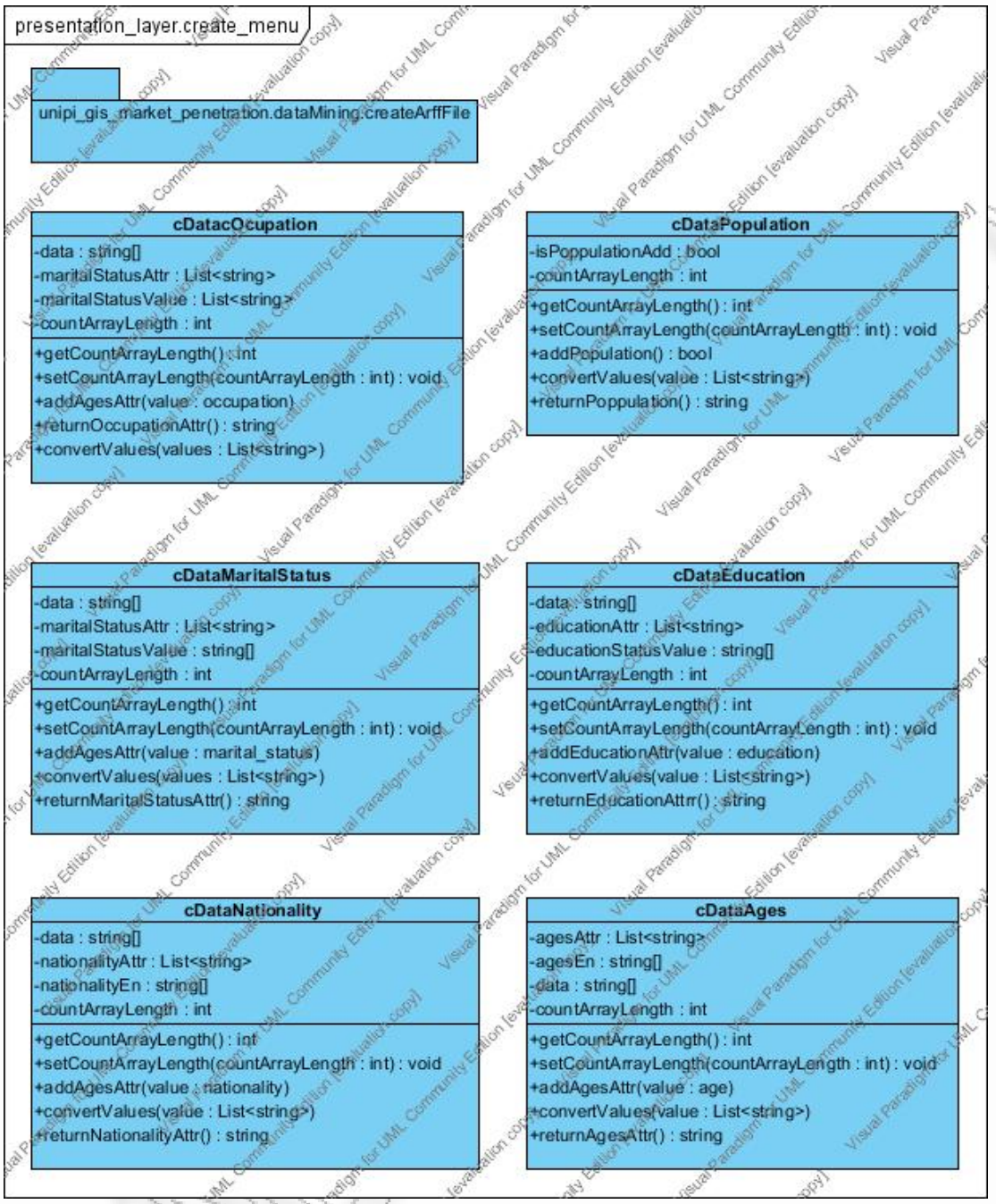
Τα δημογραφικά στοιχεία έχουν κάποιες προκαθορισμένες τιμές. Παράδειγμα το στοιχείο «Ηλικία» έχει τιμές από 0 έως 85 και πάνω και αυτά χωρίζονται ανά τετράδες. Για την αναπαράσταση των δεδομένων στο presentation layer δημιουργήθηκαν κάποια enumeration. Αυτά περιγράφονται στο επόμενο class diagram.



Εικόνα 41 Activity Diagram enumeration - presentation layer

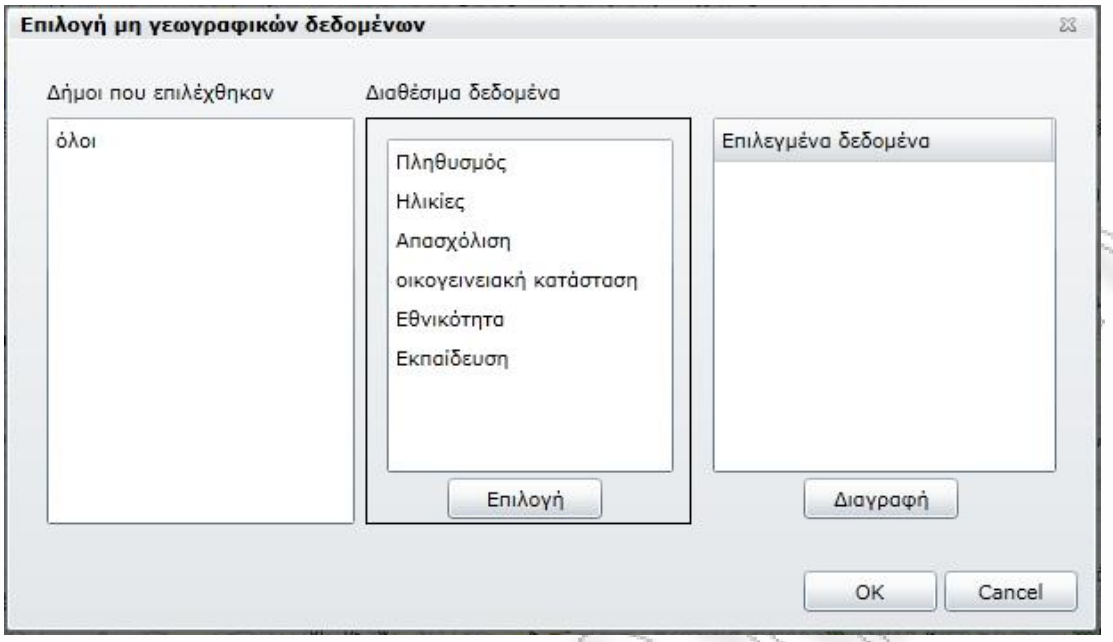
Τα συγκεκριμένα enumeration δημιουργήθηκαν για να διατηρήσουν σωστή την δομή δεδομένων. Είναι πολύ σημαντικό η αντιστοίχηση των δεδομένων γιατί πρέπει να διαβαστούν από το πληροφοριακό σύστημα για να πραγματοποιηθεί η ανάλυση τους. Οπότε πρέπει η δομή και η ορθογραφία των στοιχείων να είναι απόλυτη μεταξύ του presentation layer και του Business Layer.

Οι κλάσεις που χρησιμοποιούνται για την συλλογή πληροφοριών από τον χρήστη ώστε να εκτελεστεί η ανάλυση δεδομένων από το πληροφοριακό σύστημα εμφανίζονται στην επόμενη εικόνα.



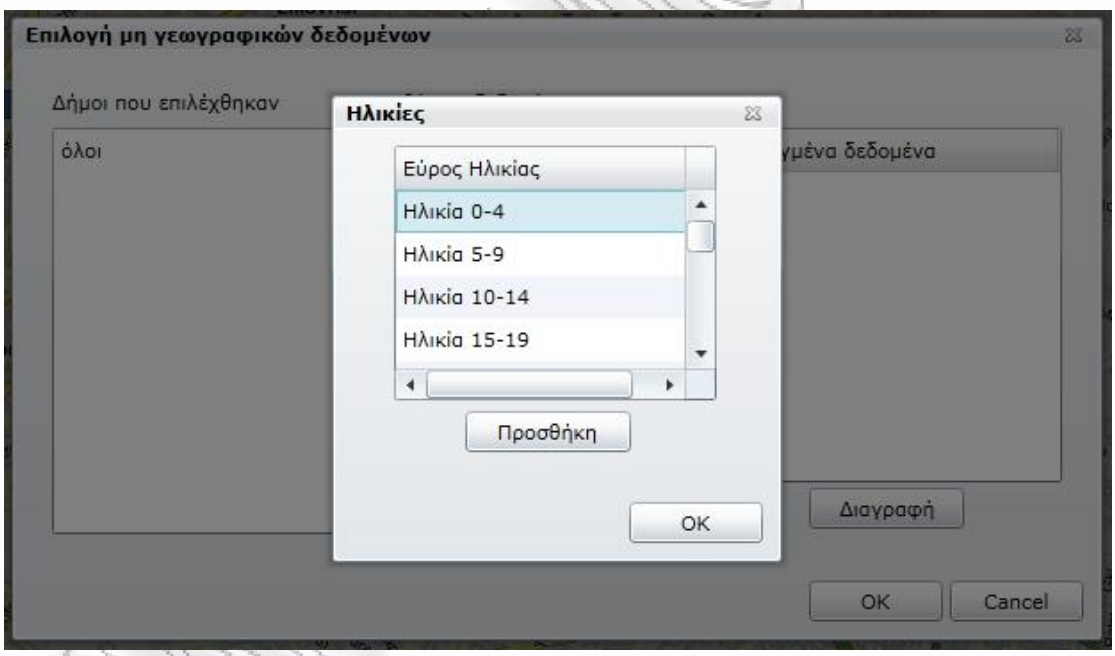
Εικόνα 42 Class Diagram presentation layer δημογραφικά στοιχεία

Χρησιμοποιώντας τις κλάσεις δημιουργούνται τα αντίστοιχα λεξιλόγια για τα δημογραφικά στοιχεία και αποθηκεύονται στην μνήμη τα επιλεγμένα από τον χρήστη. Η αλληλεπίδραση με τον χρήστη εμφανίζεται στην επόμενη εικόνα. Ένα παράδειγμα χρήσης των διεπαφών της εφαρμογή με τις λειτουργίες του presentation layer είναι στην επόμενη εικόνα. Η συγκεκριμένη διεπαφή εκτελείται στο 3ο βήμα. Ο χρήστης πρέπει να επιλέξει από τα διαθέσιμα δημογραφικά στοιχεία αυτά που χρειάζεται για να εκτέλεση την ανάλυση δεδομένων.



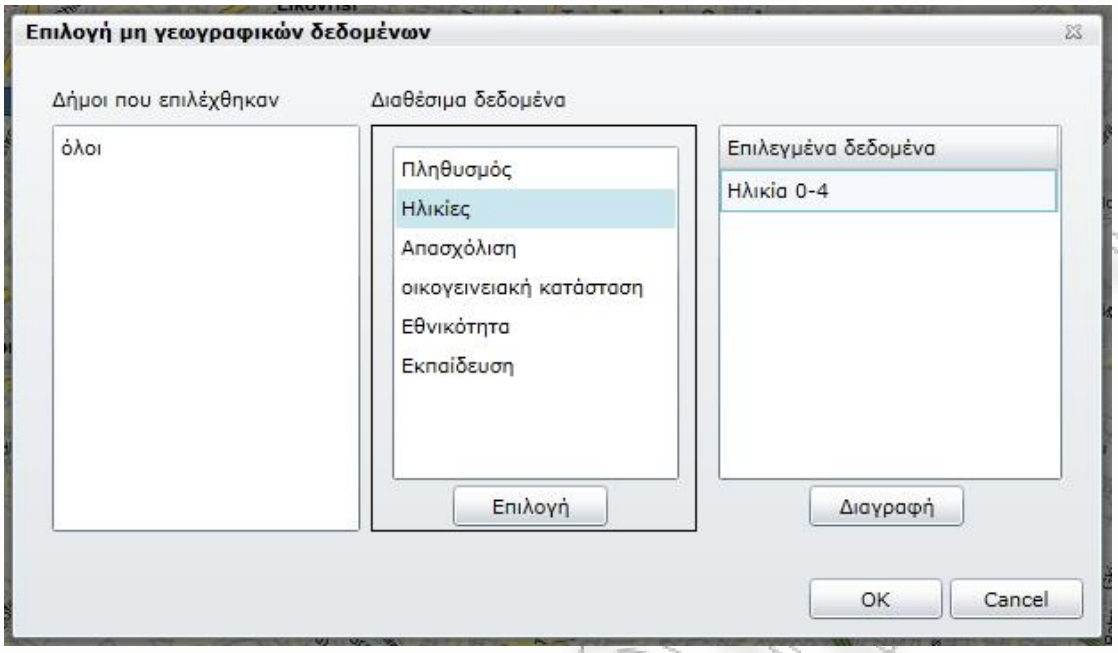
Εικόνα 43 Επιλογή δημογραφικών στοιχείων βήμα 1ο

Στο επόμενο βήμα ο χρήστης έχει επιλέξει το να προβάλει τις ηλικίες που υπάρχουν στην εφαρμογή (εικόνα 44).



Εικόνα 44 Επιλογή δημογραφικών στοιχείων βήμα 2ο

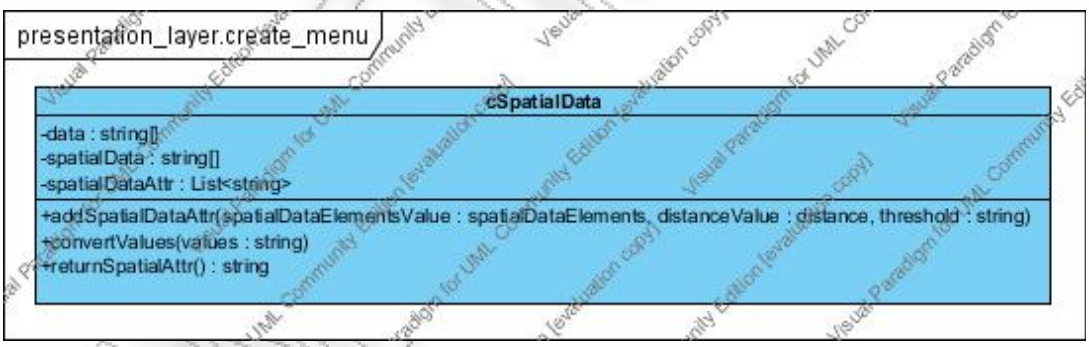
Η λίστα των ηλικιών που υπάρχουν διαθέσιμές στο σύστημα εμφανίζονται. Αν ο χρήστης επιλέξει ένα εύρος ηλικίας τότε πατάει το κουμπί προσθήκη και εμφανίζεται η επόμενη εικόνα.



Εικόνα 45 Επιλογή δημογραφικών στοιχείων βήμα 3ο

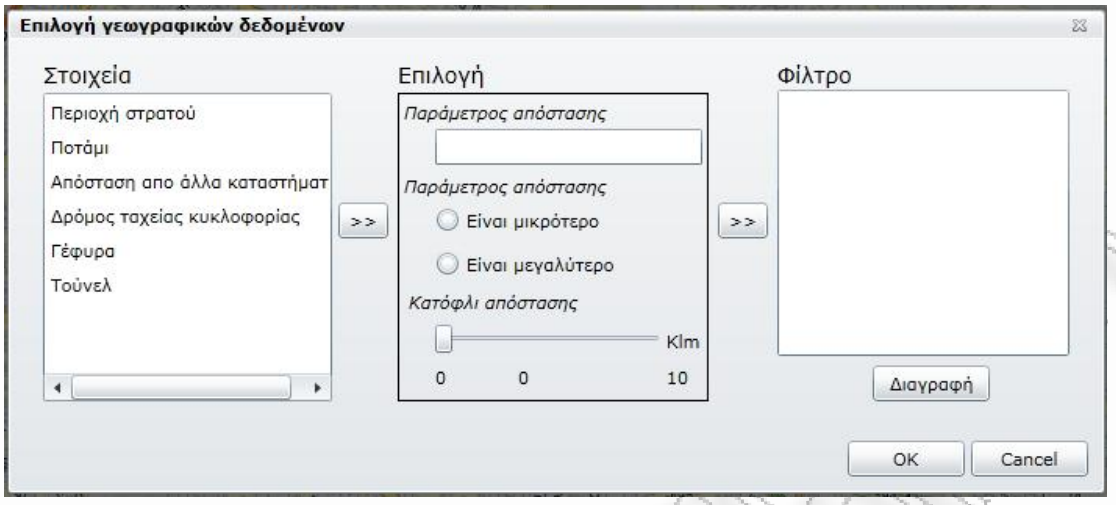
Με την χρήση των enumerations και με τις κλάσεις διατηρούνται όλα τα δεδομένα που έχει επιλέξει ο χρήστης. Στο παράδειγμα η συνάρτηση *adaAge* από την κλάση *cDataAges* θα αποθηκεύσει στην μνήμη ότι επιλέχθηκε από τα δημογραφικά στοιχεία τύπου ηλικία, το εύρος ηλικιών 0-4. Η συνάρτηση *returnAgeAttr()* επιστρέφει όλα τα στοιχεία που δεν έχει επιλέξει ο χρήστης.

Μία ξεχωριστή κλάση από τα δημογραφικά στοιχεία είναι η *cSpatialData*



Εικόνα 46 Class Diagram cSpatialData

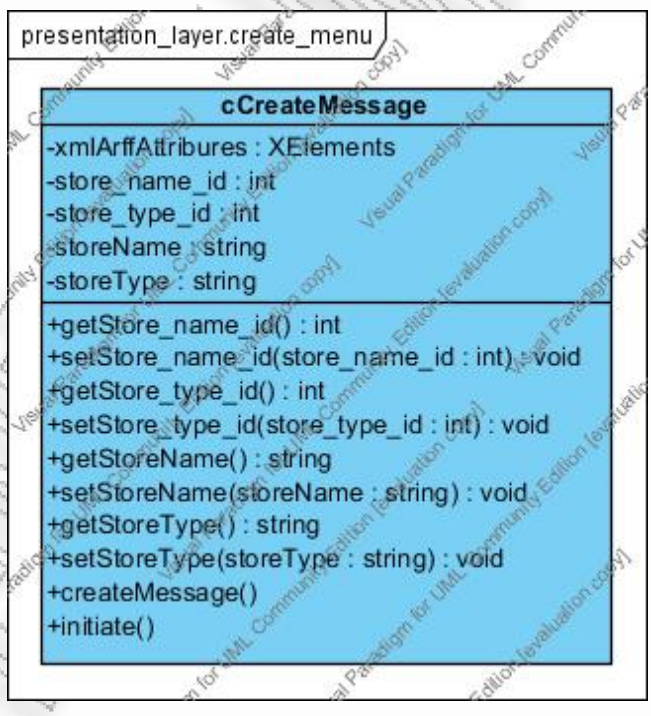
Η αλληλεπίδραση με τον χρήστη εμφανίζεται στην επόμενη εικόνα.



Εικόνα 47 Διεπαφή επιλογής δημογραφικών δεδομένων

Ο χρήστης επιλέγει το γεωγραφικό στοιχεί που θέλει να συμπεριλάβει στην ανάλυση του, το όριο που θέλει και αν επιθυμεί να είναι κοντά σε αυτό ή μακριά και τέλος πατάει το κουμπί ok. Όταν ο χρήστης πατήσει το κουμπί ok εκτελείται η συνάρτηση *addSpatialDataAttr* και αποθηκεύονται προσωρινά στην μνήμη.

Για την δημιουργία του μηνύματος εκτελείται η συνάρτηση *createMessage()* της κλάσης *cCreateMessage*. Η κλάση εμφανίζεται στην επόμενη εικόνα.



Εικόνα 48 Κλάση cCreateMessage

Στην κλάση υπάρχουν δύο συναρτήσεις η *createMessage* και η *initate()* που χρησιμοποιούνται για την δημιουργία του μηνύματος. Η δομή του μηνύματος που δημιουργεί ο χρήστης μέσα από τις επιλογές τους είναι η ακόλουθη.

```
<message>
```

```
<user>
   <user_id></user_id>
   <user_name></user_name>
</user>
<data>
   <municipalitys>
       <municipality id=""/>
   <municipalitys>
   <nonSpatialData>

   </nonSpatialData>
   <spatialData>

   </spatialData>
   <selected_items>

   </selected_items>
</data>
</message>
```

Το μήνυμα χωρίζεται σε δύο βασικά μέρη. Το πρώτο έχει τα προσωπικά στοιχεία του χρήστη που στέλνει (user_id και user_name) και το δεύτερο μέρος έχει τα δεδομένα που επέλεξε ο χρήστης. Στο tag *municipality* έχουν αποθηκευτεί όλοι οι δήμοι που επέλεξε ο χρήστης. Στο attribute id έχει τιμή τον κωδικό του δήμου. Το tag *nonSpatialData* έχει τα δημογραφικά στοιχεία που επέλεξε ο χρήστης για ανάλυση και στο tag spatialData έχει τα γεωγραφικά δεδομένα που έχει επιλέξει ο χρήστης για την ανάλυσή του. Τέλος το tag selected_items έχει τα γεωγραφικά σημεία που έχει επιλέξει ο χρήστης για την ανάδειξη τους σε ποια ομάδα ανήκουν. Ένα παράδειγμα μίας αίτησης εμφανίζεται στον επόμενο πίνακα.

```
<message>
  <user>
    <user_id>2</user_id>
    <user_name>Βασίλης</user_name>
  </user>
  <data>
    <municipalitys>
      <municipality id="0">όλοι</municipality>
    </municipalitys>
    <nonSpatialData>
      <attribute param="age">@age_0_4</attribute>
      <attribute param="nationality">@foreigner</attribute>
      <attribute param="population">@population</attribute>
      <attribute param="education">@bachelor_degree</attribute>
      <attribute param="education">@iek_degree</attribute>
      <attribute param="education">@tel_bachelor</attribute>
      <attribute param="education">@lichium</attribute>
    </nonSpatialData>
    <spatialData>
      <attribute param="spatial_data" filer_distance="near"
threshold="5,64klm">@high_way</attribute>
    </spatialData>
    <selected_items>
      <item lat="37,9712492894805" lon="23,6898328095101"/>
      <item lat="38,0837610801942" lon="23,8136212033396"/>
    </selected_items>
```

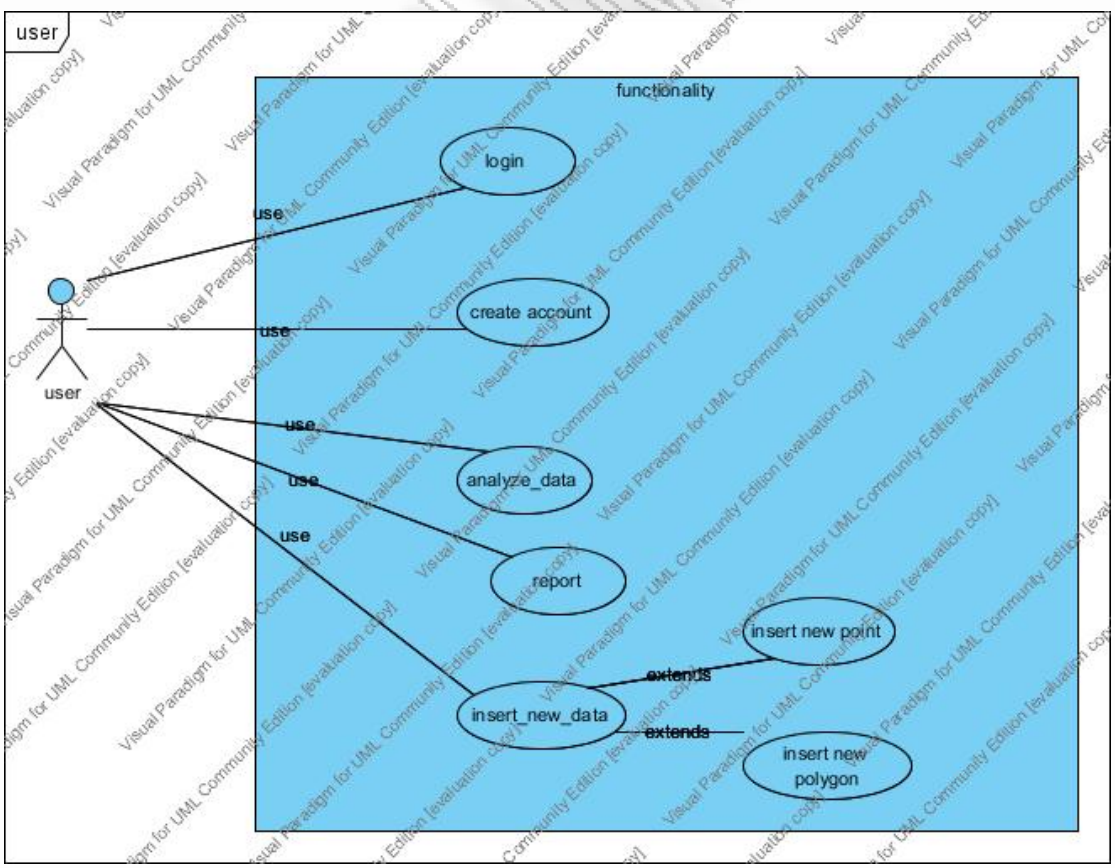
```
  </data>
</message>
```

Στην συγκεκριμένη αίτηση ο χρήστης με userId =2 και όνομα Βασίλης έχει κάνει μία αίτηση για ανάλυση δεδομένων. Έχει επιλέξει όλους τους δήμους της Αττικής για συλλογή δεδομένων. Ο χρήστης έχει επιλέξει να συλλέξει στοιχεία που αφορούν ηλικία, εθνικότητα, πληθυσμό, εκπαίδευση. Έχει δημιουργήσει ένα φίλτρο χρησιμοποιώντας γεωγραφικά δεδομένα. Τέλος έχει επιλέξει δύο σημεία που επιθυμεί να εγκαταστήσει νέο κατάστημα. Το συγκριμένο αίτημα θα σταλεί μέσω ενός Web Service στον κεντρικό Server για περεταίρω επεξεργασία.

Η συνάρτηση *initiate()* ελέγχει για όλες τις κλάσεις των δημογραφικών στοιχείων και συλλέγει εκείνα τα στοιχεία που έχει επιλέξει ο χρήστης για να κάνει την ανάλυση δεδομένων. Η συνάρτηση *createMessage()* δημιουργεί το μήνυμα που πρέπει να αποσταλεί στον κεντρικό εξυπηρετητή.

Ανάλυση λειτουργιών πληροφοριακού συστήματος

Οι λειτουργίες που υπάρχουν στο πληροφοριακό σύστημα αναλύονται στο παρακάτω Activity diagram.



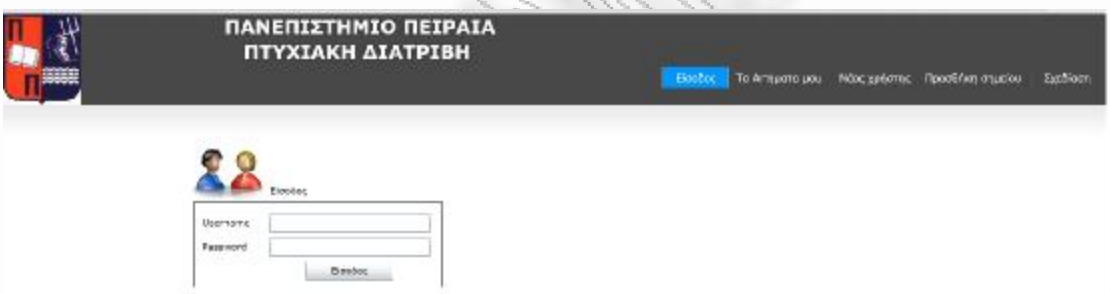
Εικόνα 49 Activity diagram

Στο παραπάνω Activity διάγραμμα υπάρχει ένας Actor που είναι ο χρήστης και μπορεί να κάνει login στην εφαρμογή εφόσον έχει κάποιο λογαριασμό. Μπορεί να δημιουργήσει ένα νέο λογαριασμό. Έχοντας κάνει login μπορεί να αιτηθεί μία νέα ανάλυση, να δει τα αποτελέσματα αναλύσεων και να κάνει εισαγωγή νέων δεδομένων στην εφαρμογή. Στις επόμενες ενότητες θα πραγματοποιηθεί μία λεπτομερής ανάλυση των συγκεκριμένων λειτουργιών. Οι λειτουργίες θα χωριστούν σε υποενότητες οι οποίες θα είναι :

- Υποσύστημα Είσοδου
- Υποσύστημα Δημιουργίας νέου λογαριασμού
- Υποσύστημα Ανάλυση δεδομένων
- Υποσύστημα Αιτήματα χρήστη
- Υποσύστημα Σχεδίασης

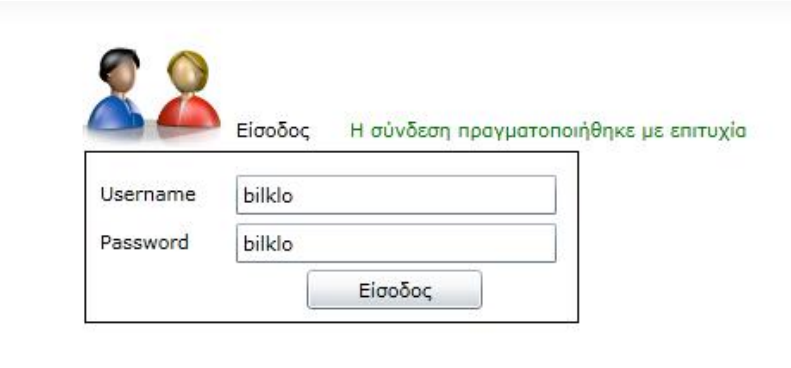
Υποσύστημα εισόδου

Ο χρήστης για έχει πρόσβαση σε οποιαδήποτε λειτουργία του συστήματος θα πρέπει να εισέλθει στην εφαρμογή. Για την εισαγωγή του χρήστη στο σύστημα εμφανίζεται η επόμενη οθόνη.



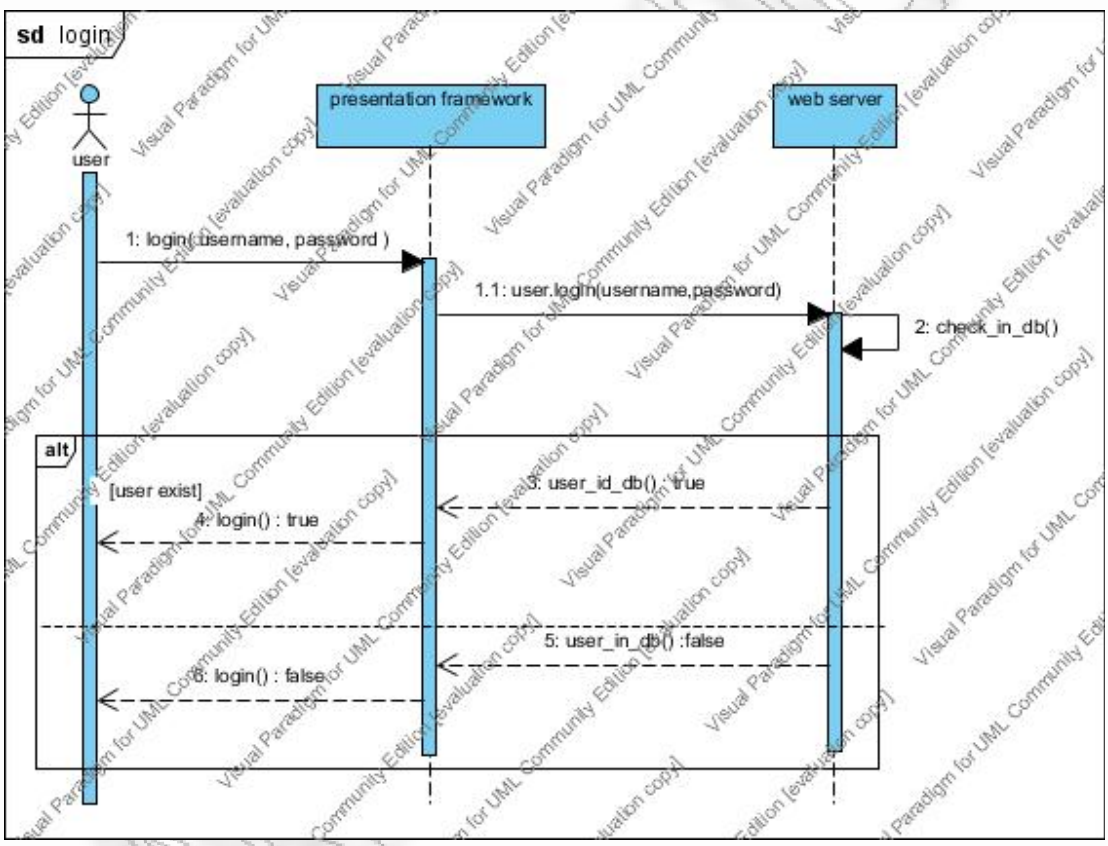
Εικόνα 50 Login form

Ο χρήστης πληκτρολογώντας το Username και το password και πατώντας το κουμπί είσοδος μπορεί να κάνει login. Αν ο χρήστη πληκτρολογήσει σωστά τα στοιχεία του τότε θα του εμφανιστεί η επόμενη εικόνα.



Εικόνα 51 login screen

Στο επόμενο Sequence διάγραμμα αναπαριστάτε ο τρόπος με τον οποίο πραγματοποιείται η λειτουργία login.

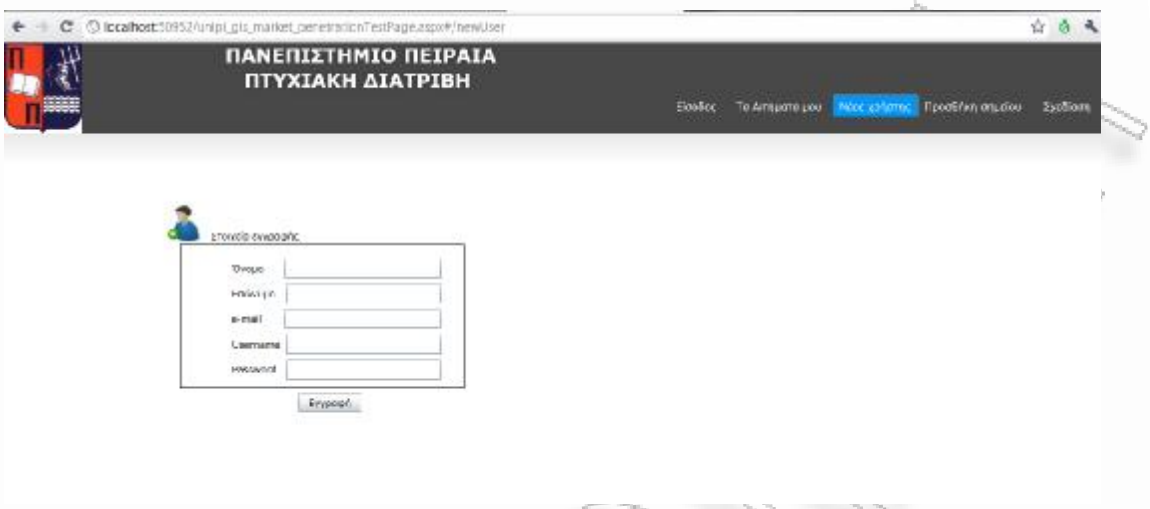


Εικόνα 52 sequence diagram login

Το συγκεκριμένο διάγραμμα αναλύει πως ο χρήστης εισέρχεται στο σύστημα. Πληκτρολογεί το username και το Password και πατώντας το κουμπί Login ελέγχεται αν ο λογαριασμός υπάρχει. Όταν ο χρήστης πατάει το κουμπί «Είσοδος» τότε καλείται ασύγχρονα το Web Service *user.login(username , password)*. Το συγκεκριμένο Web Service ελέγχει αν ο λογαριασμός υπάρχει και ενημερώνει το presentation layer.

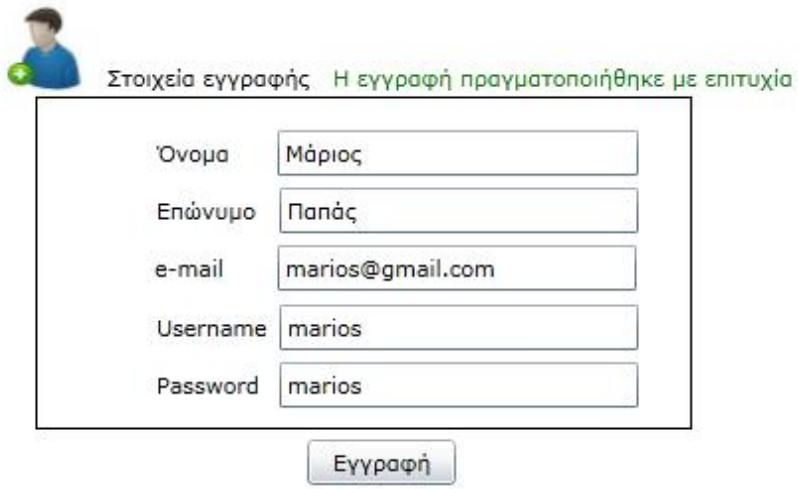
Υποσύστημα δημιουργίας Νέου λογαριασμού

Για να δημιουργήσει ένας νέος χρήστης έναν καινούριο λογαριασμό μπορεί μέσα από τον menu να πλοηγηθεί στην επιλογή «Νέος χρήστης». Τότε θα του εμφανιστεί η επόμενη εικόνα.



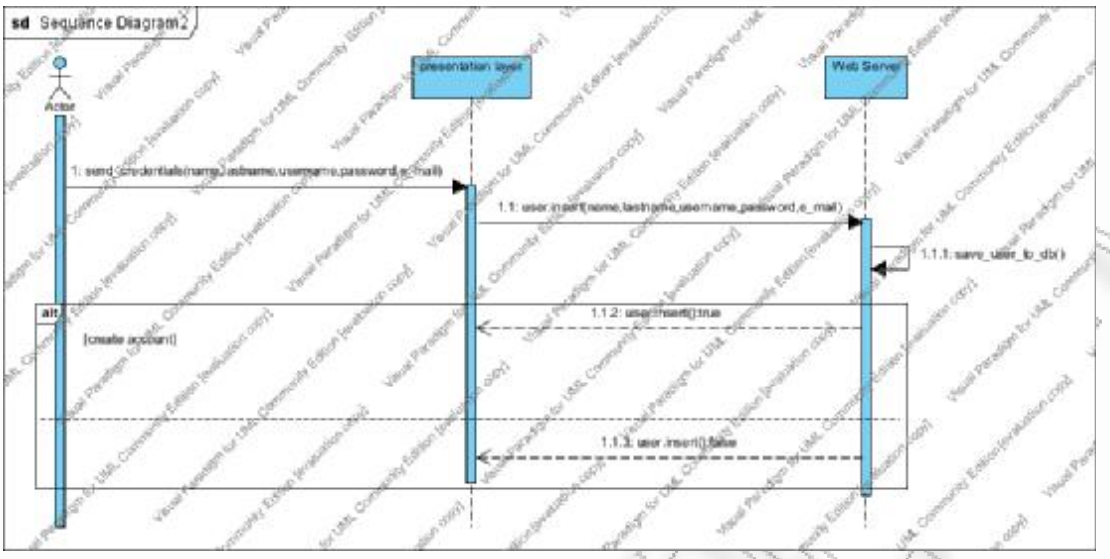
Εικόνα 53 create new account

Ο χρήστης όταν πληκτρολογήσει όλα τα στοιχεία του και πατήσει το κουμπί εγγραφή θα του εμφανιστεί η επόμενη εικόνα.



Εικόνα 54 create new account βήμα 2ο

Στο επόμενο Sequence διάγραμμα περιγράφεται η διαδικασία εγγραφή στον πληροφοριακό σύστημα.



Εικόνα 55 Sequnce diagram create new user

Στο διάγραμμα της εικόνας 55 περιγράφεται η διαδικασίας δημιουργίας νέου λογαριασμού στο πληροφοριακό σύστημα. Ο χρήστης όταν πληκτρολογήσει τα στοιχεία του και πατήσει το κουμπί «Εγγραφή» θα σταλούν τα δεδομένα στο Web Service *user.insert*.

Υποσύστημα - Ανάλυση δεδομένων

Εισαγωγή

Η ανάλυση δεδομένων είναι η διαδικασία με την οποία ο χρήστης μπορεί να δημιουργήσει ένα νέο αίτημα στο πληροφοριακό σύστημα. Το βασικό σενάριο χρήσης του πληροφοριακού συστήματος είναι η εύρεση της καλλίτερης γεωγραφικής θέσης για την δημιουργία νέου καταστήματος. Ο χρήστης θέλει να δημιουργήσει ένα νέο κατάστημα και θέτει διάφορες γεωγραφικές θέσεις που επιθυμεί να δημιουργήσει το νέο κατάστημα, επιλέγει τους δήμους που επιθυμεί να εκτελεστεί η ανάλυση δεδομένων και τέλος σημειώνει το είδος του καταστήματος, και την ονομασία. Σημειώνει τα δημογραφικά στοιχεία που θα των ενδιέφεραν για το κατάστημα του (όπως για παράδειγμα ότι το μαγαζί θα αφορά νέου ηλικίας 19-16 και αφορά αλλοδαπούς). Ο χρήστης μπορεί να εισάγει και γεωγραφικές παραμέτρους στο αίτημα του, όπως για παράδειγμα να είναι κοντά (σε απόσταση 5km) από αυτοκινητόδρομο). Το πληροφοριακό σύστημα θα αναλύσει τα δεδομένα συλλέγοντας στοιχεία από την βάση. Μια σημαντική σημείωση είναι ότι ο χρήστης για να εκτελέσει οποιοιδήποτε σενάριο χρήσης πρέπει να έχει κάνει login στο πληροφοριακό σύστημα

Το συγκεκριμένο υποσύστημα χωρίζεται σε δύο βασικά μέρη. Το Πρώτο είναι η δημιουργία του μηνύματος από τον χρήστη. Το δεύτερο μέρος είναι η επεξεργασία του μηνύματος και η εκτέλεση του αλγόριθμου SimpleKMeans. Στο πρώτο μέρος συμμετέχουν ο χρήστης και το σύστημα, στο δεύτερο μέρος η επεξεργασία γίνεται αυτόματα με την εντολή του χρήστη. Στην επόμενη ενότητα θα πραγματοποιηθεί μία εισαγωγή στο Data Mining χρησιμοποιώντας γεωγραφικά δεδομένα.

Εξόρυξη δεδομένων χρησιμοποιώντας γεωγραφικά στοιχεία

Χρησιμοποιώντας GIS συστήματα αποθηκεύονται δεδομένα που έχουν σχέση με γεωγραφική πληροφορία. Το αποτέλεσμα είναι να υπάρχει ένας μεγάλος όγκος δεδομένων που μπορεί να αφορά διάφορος τομείς όπως, μετακινήσεις, τηλεπικοινωνίες, marketing. Όλα αυτά τα δεδομένα αποθηκεύονται σε βάσεις δεδομένων που χειρίζονται γεωγραφικά δεδομένα. Υπάρχουν τεχνικές που μπορεί να εξαχθεί σημαντική πληροφορία.
Υπάρχουν πέντε βασικά βήματα τα οποία μπορούν να εκτελεστούν για την εύρεση πληροφορίας από μεγάλο όγκο δεδομένων.

- Επιλογή
- Προετοιμασία δεδομένων
- Μορφοποίηση
- Εξόρυξη δεδομένων
- Αξιολόγηση

Το πρόβλημα που εμφανίζεται στα GIS συστήματα λόγο των δεδομένων που χειρίζονται είναι ότι θέλουν πολύ περισσότερο χρόνο για να εκτελέσει τις διαδικασίες.
Ο συσχετίσεις που μπορούν να δημιουργηθούν σε μία βάση που έχει γεωγραφική πληροφορία είναι τρεις :

- Απόσταση. Η απόσταση δύο σημείων μπορεί να δοθεί με την Ευκλείδεια απόσταση.
- Οριοθέτηση. Οι σχέσεις που δημιουργούνται μεταξύ αντικειμένων ανάλογα με την οριοθέτηση που έχουν στο χάρτη.
- Τοπολογιών. Οι σχέσεις αυτές αναπτύσσονται μεταξύ γεωγραφικών αντικειμένων και κατατάσσονται (Cluster). Η κατηγοριοποίηση μπορεί να γίνε με τις ακόλουθες παραμέτρους.
    - Equal
    - Disjoint
    - Touches
    - Within
    - Crosses
    - CoverdBy

Για την υλοποίηση του Data mining πρέπει να γίνει μία προετοιμασία στην βάσης δεδομένων. Ένα από τα βασικά βήματα που πρέπει να εκτελεστούν είναι να δοθούν τα σωστά χαρακτηριστικά, είναι ένα από τα βασικότερα βήματα γιατί με βάση την επιλογή αυτή θα έχει επηρεαστεί το αποτέλεσμα. Το τελικό αποτέλεσμα θα είναι να δημιουργηθεί ένας μεγάλο αρχείο με δεδομένα που κάθε γραμμή θα είναι ένα στιγμιότυπο. Ένα χαρακτηριστικό για τον δήμο της Καλλιθέας είναι ότι είναι δήμος. Οι στήλες που θα έχει κάθε στιγμιότυπο είναι προκαθορισμένες. Οι στήλες μπορούν να έχουν δεδομένα που δεν είναι γεωγραφικής φύσης όπως πληθυσμός αλλά και γεωγραφικές σχέσεις π.χ. contains_river. Οι γεωγραφικές συσχετίσεις που δημιουργούνται είναι αποτέλεσμα SQL ερωτημάτων έχοντας σαν βάση το συγκεκριμένο στιγμιότυπο t (Παράδειγμα δήμος Καλλιθέας) του χαρακτηριστικού **T**(Δήμος) και όλων των χαρακτηριστικών **o**(Λεωφόρο Συγγρού) όλων των σχετικών χαρακτηριστικών **O** (Λεωφόρους) σε ένα σύνολο στοιχείων S (δρόμους, ποτάμια..) που έχον γεωγραφική συσχέτιση στο T. Σαν αποτέλεσμα υπάρχει ένα σύνολο από στιγμιότυπα T={t1,t2,t3,...,tn} S={O1,O2,O3,...,On} και
Oi={o1,o2,o3,...,on} το αποτέλεσμα είναι εξαγωγή γεωγραφικών συσχετίσεων πραγματοποιείται μέσα από κάθε στιγμιότυπο T κάθε στιγμιότυπου O για όλα τα O,S.

Οι τυπολογικές σχέσεις είναι μοναδικές, για παράδειγμα υπάρχει μία τυπολογική σχέση μεταξύ της της Καλλιθέας και της Λεωφόρο Συγγρού. Ένα παράδειγμα τοπολογικής σχέσης ειναι το ακόλουθο.

TargetF (city)	Relevant F instance	Relationship
1	River_1	Contains
1	River_2	Crossess
2	River_3	Contains
2	River_4	Crosses
3	River_2	Crosses
1	Slum_1	Contains
2	Slum_2	Contains

Η μορφή που θα δημιουργηθεί για να εισαχθεί στο WEKA εμφανίζεται στον επόμενο πίνακα.

Target F(city)	River_1	River_2	River_3	River_4	Slum_1
1	Contains	Crosses	?	?	Contains
2	?	?	Contains	Crosses	?
3	?	Crosses	?	?	?

Όπως αναφέρθηκε και πριν η επιλογή χαρακτηριστικών έχει μεγάλη σημασία. Παράδειγμα για την πόλη 1 μπορεί να μην έχει τοπολογική σχέση (contains, crosses) με κάποιο ποτάμι. Έτσι μπορούμε να έχουμε τον ακόλουθο πίνακα.

Target_F	Contains_River	Crossess_River	Contains_Slum
1	YES	YES	YES
2	YES	YES	YES
3	?	YES	?

Σχέσεις αποστάσεων

Οι σχέσεις αποστάσεων δημιουργούνται ανάμεσα στα χαρακτηριστικά που έχει επιλέξει ο χρήστης. Ένα παράδειγμα εμφανίζεται στον επόμενο πίνακα.

Target_1	instance	Releshionship
1	River_1	veryClose
1	River_2	Close
2	River_3	Close
2	River_4	Close
3	River_2	VeryClose
1	River_1	Close
2	River_2	VeryClose

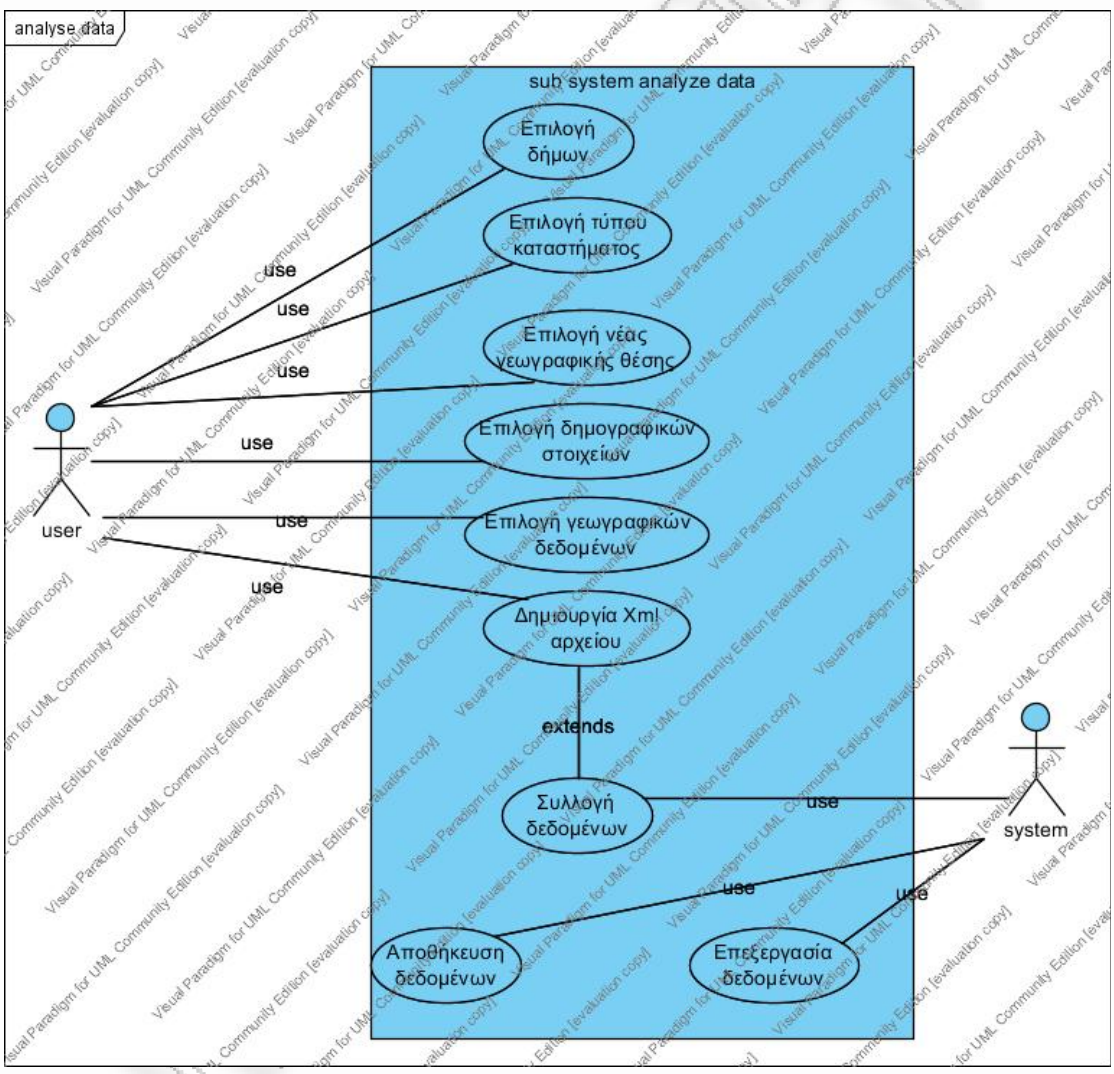
Ο πίνακας που θα δημιουργηθεί για το WEKA είναι ο ακόλουθος

Targer F	River_1	River_2	River_3	River_4
1	veryClose	Close	?	?
2	?	?	Close	Close
3	?	VeryClose	?	?

Εάν μία παράμετρος δοθεί (απόσταση) τότε οι γείτονες που είναι κοντά στο σημείο έχουν την τιμή veryClose. Αν έχουμε δύο αποστάσεις (δύο παραμέτρους) τότε τα σημεία που είναι κοντά στην παράμετρο dist_1 θα έχουν την τιμή veryClose και ανάμεσα στην τιμή dist_2 και dist_1 τότε θα πάρουν την τιμή close. Την τιμή far δεν την χρησιμοποιούμε γιατί αν ένα γείτονας δεν είναι κοντά (δεν έχει την τιμή veryColse) τότε είναι μακριά.

Ανάλυση υποσυστήματος - Δημιουργία Xml μηνύματος

Στην εικόνα 56 αναπαριστάτε ένα διάγραμμα use case στο υπάρχουν δύο ρόλοι. Ο πρώτος ρόλος είναι ο χρήστης που έχει κάνει login στο σύστημα και θέλει να κάνει μία νέα αίτηση. Ο δεύτερος ρόλος είναι το σύστημα το οποίο συλλέγει τα συγκεκριμένα στοιχεία από τον χρήστη και τα αποθηκεύει για ανάλυση τους.



Εικόνα 56 Use case diagram ρόλοι user – system

Με βάση το διάγραμμα της εικόνας 56 ο χρήστης έχει την δυνατότητα μέσα από κάποια Activities να δημιουργήσει ένα ερώτημα προς το σύστημα. Το ερώτημα θα αποθηκευτεί και θα αναλυθεί (Επεξεργασία δεδομένων) για να εξεργαστούν τα δεδομένα του χρήστη.

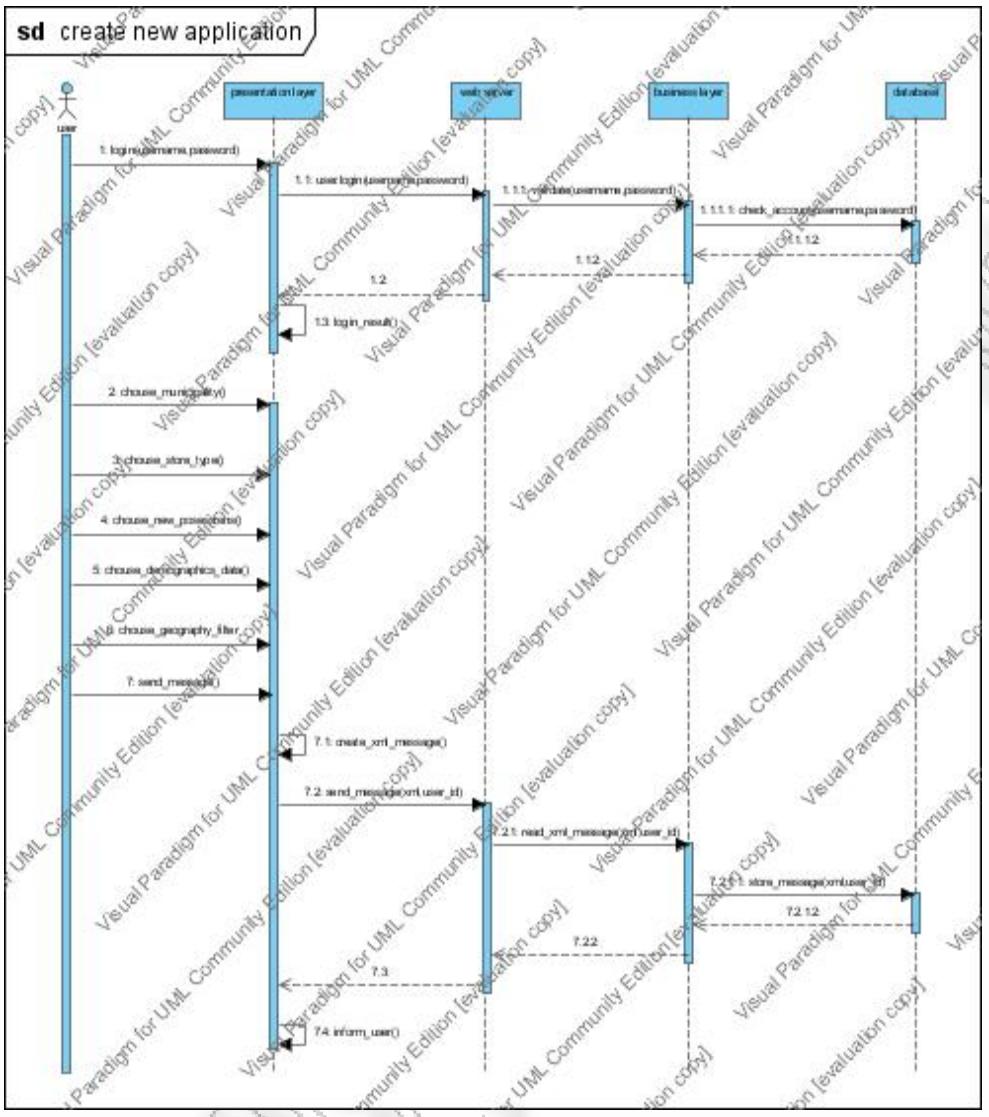
Στο επόμενο διάγραμμα (Use Case) αναλύεται ο τρόπος με τον οποίο συλλέγονται δεδομένα από τον χρήστη. Ο τρόπος που επιλέχτηκε είναι μέσω ενός Wizard.

Τα βήματα του wizard είναι 6

- Επιλογή δήμων
- Επιλογή τύπου καταστήματος
- Επιλογή νέας γεωγραφικής θέσης
- Επιλογή δημογραφικών στοιχείων
- Επιλογή γεωγραφικών φίλτρων.
- Δημιουργία XML μηνύματος.

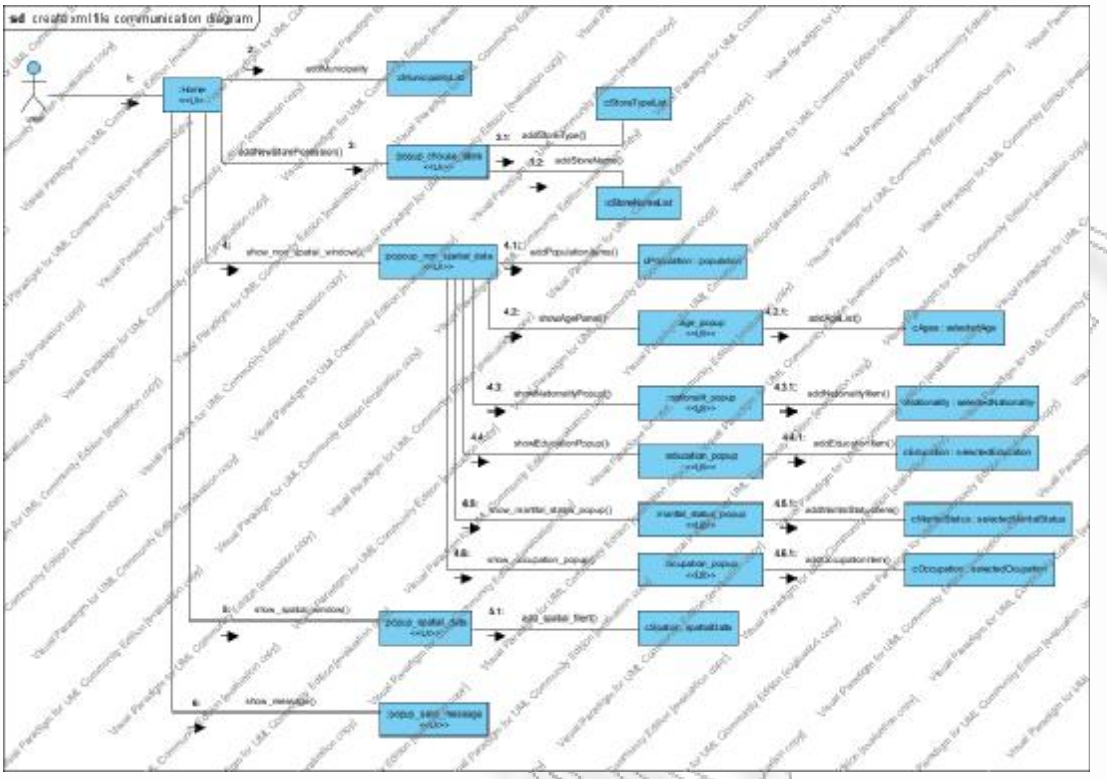
Ο συγκεκριμένος Wizard έχει μεγάλη σημασία στην ανάλυση δεδομένων. Ο λόγος που δημιουργήθηκε είναι να περιοριστούν το μεγαλύτερο σύνολο δεδομένων (ιδιοτήτων) που υπάρχουν στην Βάση Δεδομένων.

Στην ανάλυση δεδομένων έπρεπε να αντιμετωπιστεί ένα πρόβλημα το οποίο είναι γνωστό σε Data Mining προβλήματα ως «η κατάρα της πολυδιάστατης». Στην βάση δεδομένων υπάρχουν ομαδοποιημένα δημογραφικά στοιχεία. Η κάθε ομάδα έχει n στήλες. Το σύνολο των στηλών για όλες τις ομάδες είναι πάνω από 50. Αν για παράδειγμα εκτελεστεί ο αλγόριθμος SimpleKMeans για ένα μεγάλο αριθμό ιδιοτήτων, που δεν είναι σίγουρο αν θα έχουν μεγάλο συντελεστή βαρύτητας στο αποτέλεσμα του Data Mining υπάρχει μεγάλη περίπτωση να μην είναι σωστή η ομαδοποίηση των δεδομένων. Για τον συγκεκριμένο λόγο δημιουργήθηκε ο Wizard με την προϋπόθεση να συλλέξει εκείνα τα δεδομένα που είναι χρήσιμα για την δημιουργία ομάδων (clusters)



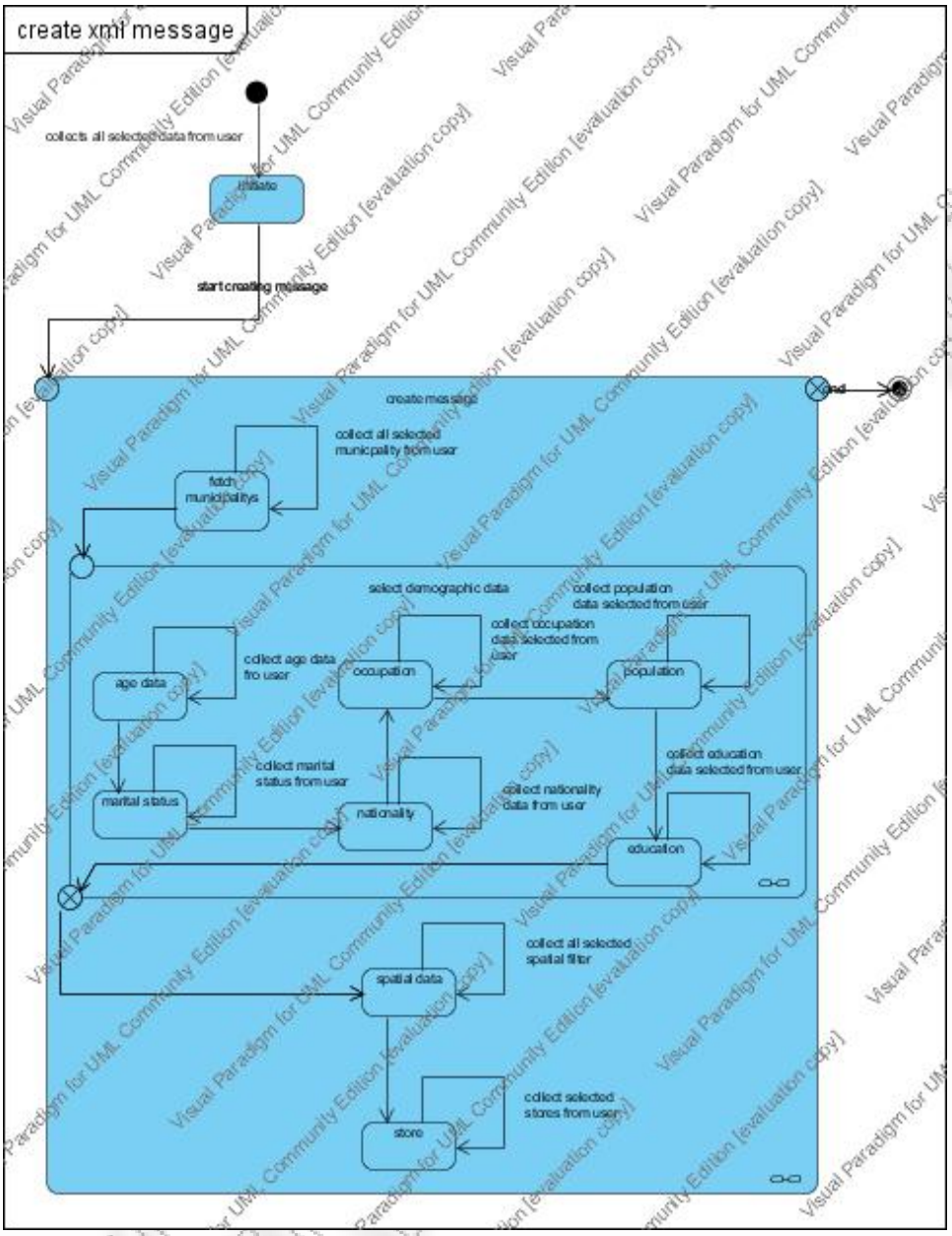
Εικόνα 57 Sequence diagram Αποστολή νέας αίτησης

Στο Sequence diagram της εικόνας 57 αναδεικνύεται όλη η διαδρομή που πρέπει να ακολουθηθεί για την δημιουργία ενός μηνύματος προς το σύστημα. Στο διάγραμμα αναδεικνύεται και οι διαφορετικές οντότητες που αλληλεπιδρούν στο σύστημα. Ένα μέρος της Business λογικής βρίσκεται στο presentation layer στην Silverlight εφαρμογής και επικοινωνεί μέσω web service με το Business logic της εφαρμογής. Στο επόμενο communication diagram αναδεικνύονται οι κλάσεις που συμμετέχουν για την δημιουργία του xml μηνύματος. Οι κλάσεις που χρησιμοποιούνται στο διάγραμμα έχουν αναλυθεί κεφάλαιο Business layer.



Εικόνα 58 Communication Diagram Δημιουργία Xml Μηνύματος

Στο διάγραμμα της εικόνας 58 εμφανίζονται οι κλάσεις που καλούνται για την δημιουργία του XML μηνύματος. Ο τρόπος με τον οποίο δημιουργείται το XML είναι αυτόματος και ελέγχει τα δεδομένα που έχουν δοθεί από τον χρήστη και παράγει το xml. Το επόμενο διάγραμμα State machine εμφανίζει αναλυτικά τον τρόπο που δημιουργείται το XML αρχείο.

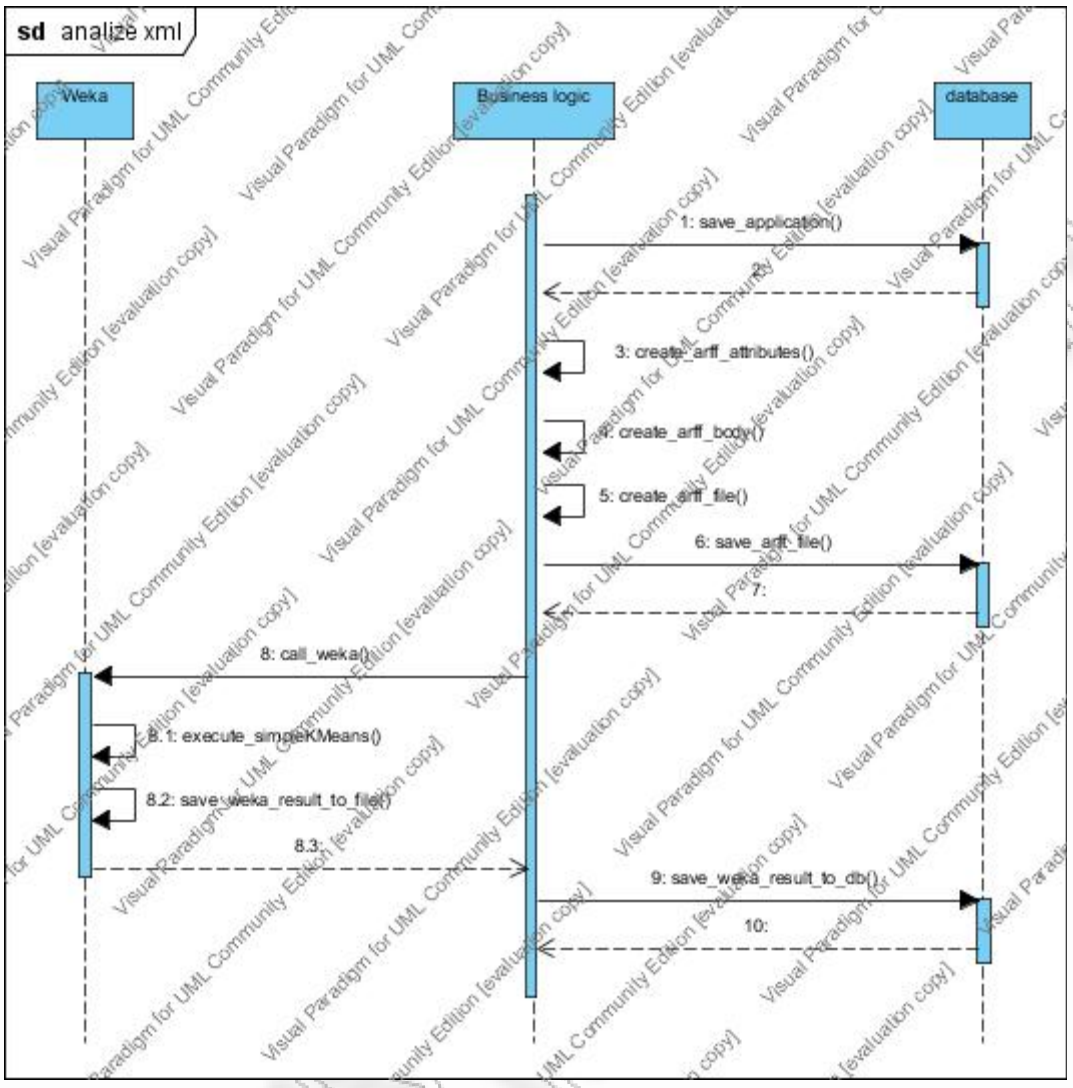


Εικόνα 59 State machine diagram create xml message

Το δεύτερο μέρος αναφέρεται στην ανάλυση του xml μηνύματος και την επεξεργασία του.

Ανάλυση υποσυστήματος - Επεξεργασία xml μηνύματος

Η ενότητα αυτή είναι ένα από τα πιο σημαντικά μέρη της εφαρμογής. Με βάση την ανάλυση του xml αρχείο εκτελείται ο αλγόριθμος επεξεργασίας και δημιουργούνται οι ομάδες των καταστημάτων. Το επόμενο Sequence διάγραμμα δείχνει την συγκεκριμένη εκτέλεση του αλγόριθμου.



Εικόνα 60 Analyze data with weka

Το Sequence διάγραμμα της εικόνας 60 αναλύει την διαδικασία ανάλυσης του XML αρχείου. Η διαδικασία ανάλυσης του xml ξεκινά με την αποθήκευση του μηνύματος στην βάση δεδομένων. Το επόμενο βήμα είναι η εύρεση ιδιοτήτων για την δημιουργία του header. Για να δημιουργηθεί το σωστό αρχείο για το Weka πρέπει να εισαχθούν και τα στιγμιότυπα με τις κατάλληλες τιμές. Αυτό ονομάζεται body του αρχείου. Όταν δημιουργηθεί όλο το arff αρχείο τότε αποθηκεύεται στην βάση δεδομένων.

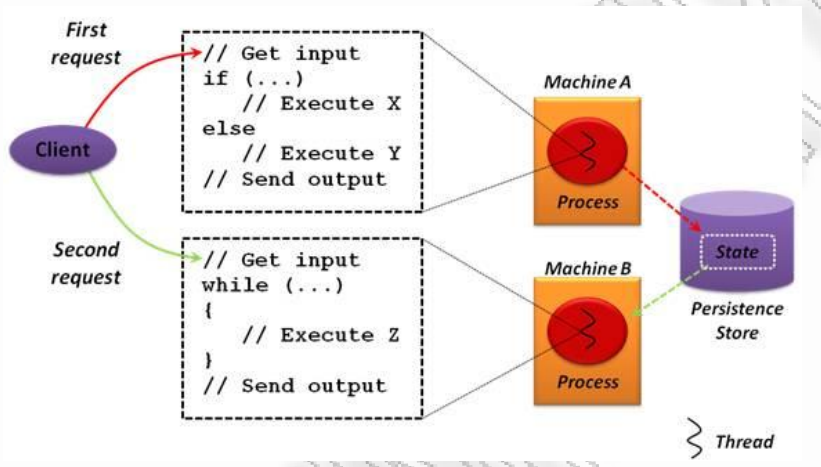
Εφόσον έχει δημιουργηθεί το αρχείο arff αυτά μπορεί να εκτελεστεί το Weka για να πράξει τις κλάσεις. Για την εκτέλεση του Weka με βάση το αλγόριθμο SimpleKMeans βρίσκετε στο Annex A Weka – SimpleKMeans.

Workflow

Πολλές εφαρμογές έχουν τα ακόλουθα κοινά χαρακτηριστικά. Όπως για παράδειγμα

- Διατήρηση της κατάστασης των μεταβλητών κατά την εκτέλεση του προγράμματος.

- Έχει την δυνατότητα να δέχεται δεδομένα από έναν client και να τα επεξεργάζεται. Μία windows εφαρμογή μπορεί να δέχεται δεδομένα μέσω του παραθύρου, μία Web εφαρμογή μπορεί να δέχεται δεδομένα μέσω του πρωτοκόλλου HTTP.
- Έχει την δυνατότητα να στέλνει δεδομένα μέσω Web Service.
- Η εφαρμογή επεξεργάζεται δεδομένα και έχουν υλοποιηθεί όλοι οι τρόποι αντιμετώπισες για την επίλυση του προβλήματος.

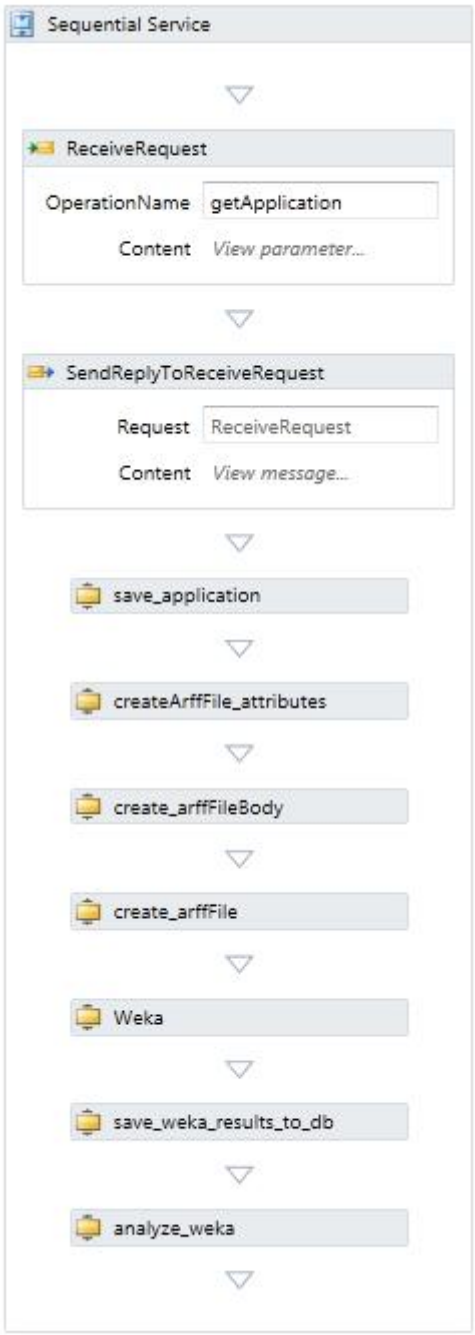


Εικόνα 61 Εκτέλεση κομμάτι κώδικα

Στο σχήμα της εικόνας 61 δείχνει την διαδικασία εκτέλεσης ενός κομμάτι κώδικα. Όταν ο Client ζητά να εκτελεστεί ο συγκεκριμένος κώδικας τότε το συγκεκριμένο κομμάτι φορτώνεται στην μνήμη. Όταν αίτημα εκτελείτε και αποστέλλεται στον client το αποτέλεσμα τότε αυτό βγαίνει από την μνήμη.

Με την χρήση του Workflow μέσα σε μία εφαρμογή λύνονται βασικά προβλήματα. Η συγκεκριμένη τεχνική χρησιμοποιείται πάρα πολύ όταν τα προβλήματα που πρέπει να λυθούν είναι πολύ σύνθετα. Για την επίλυση τους πρέπει να δημιουργηθεί κώδικας που είναι αρκετά πολύπλοκος χρησιμοποιώντας πολλές συνθήκες (όπως για παράδειγμα if – else, while…) για να υποστηρίξει όλες τις υποπεριπτώσεις.

Η δημιουργία του Arff αρχείου και η εκτέλεση του είναι ένα σύνθετο και πολύ σημαντικό πρόβλημα στην εφαρμογή. Η επίλυση για του συγκεκριμένο πρόβλημα έγινε χρησιμοποιώντας το Windows Work Flow. Στην επόμενη εικόνα 62 εμφανίζεται η σχεδίαση του Workflow στο Visual Studio 2010.



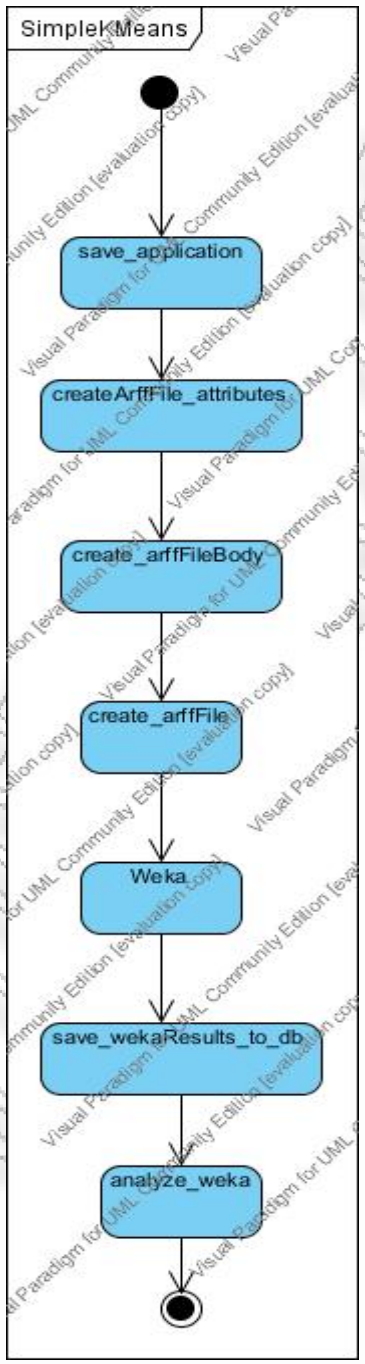
Εικόνα 62 Wokrlfow Εκτέλεση SimpleKMeans

Δύο είναι οι βασικοί λόγοι που χρησιμοποιήθηκε η συγκεκριμένη τεχνική.

- Ο πρώτος λόγος είναι ότι το πρόβλημα το οποίο είναι πάρα πολύ σύνθετο και πρέπει να υπάρχουν κομμάτια κώδικα ομαδοποιημένα για την σωστή εκτέλεση του κώδικα.
- Ο δεύτερος λόγος είναι ότι η συγκεκριμένη διαδικασία μπορεί να εκτελείτε για μεγάλο χρονικό διάστημα. Υπήρχαν δυο τρόποι επίλυσης του προβλήματος. Ο πρώτος τρόπος είναι όταν ο client ζητήσει να εκτελεστεί η συγκεκριμένη διαδικασία τότε να δημιουργηθεί ένα thread το οποίο θα εκτελέσει κομμάτια κώδικα σειριακά. Ο τρόπος αυτός δεν ενδείκνυται γιατί θα πρέπει να

υλοποιηθούν αλγόριθμοί οι οποίοι θα ελέγχουν την συγκεκριμένη διαδικασία. Ο δεύτερος τρόπος είναι χρησιμοποιώντας το Workflow της Microsoft το οποίο έχει την δυνατότητα εκτελεί κομμάτια κώδικα σειριακά ελεγχόμενα.

Η ανάλυση εκτέλεσης του SimpleKMeans αλγόριθμου αναπαριστάτε από το επόμενο διάγραμμα.



Εικόνα 63 State Machine diagram εκτέλεση του SimpleKMeans

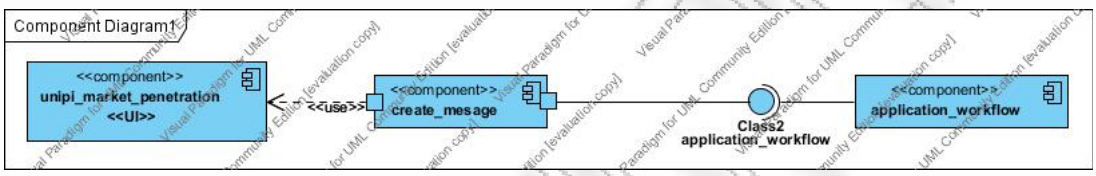
Όπως εμφανίζεται και στο State Machine Diagram για την εκτέλεση του SimpleKMeans πρέπει να δημιουργηθεί το arff αρχείο για να εκτελεστεί το Weka. Αυτό που πρέπει να αναλυθεί είναι το πώς μεταφέρεται το αντίστοιχο διάγραμμα στο Workflow της

Microsoft Πρώτα θα πρέπει να μελετηθούν τα βασικά στοιχεία του Microsoft Workflow Foundation.

Sequential Service : Το συγκεκριμένο Activity χρησιμοποιείται όταν στο πρόγραμμα πρέπει να εκτελεστούν ομάδες κώδικα σειριακά. Στο Sequential Workflow υπάρχει η δυνατότητα να σχεδιαστεί η ροή εκτέλεσης του κώδικα.

Code Activity : Το συγκεκριμένο Activity χρησιμοποιείται για την ομαδοποίηση κώδικα.

WCF Workflow Service : Για την συγκεκριμένη ανάλυση πρέπει πρώτα να σχεδιαστεί η αρχιτεκτονική επικοινωνίας της εφαρμογή με το Workflow. Στην επόμενη εικόνα υπάρχει είναι Component διάγραμμα για την ανάλυση του.



Εικόνα 64 Component Diagram

Ο χρήστης της εφαρμογής δημιουργεί το μήνυμα που πρέπει να στείλει ο χρήστης μέσω αυτής. Μέσα από το Visual studio δημιουργήθηκε ένα νέο Project WCF Workflow Service με το όνομα *ApplicationWorkflow*. Αυτόματα δημιουργείτε ένα activity το οποίο είναι τύπου *ResaveRequest* και *SendReplayResaveRequest*. Τα συγκεκριμένα Activities δημιουργούν τις κατάλληλες προϋποθέσεις για ένα Web Service. Όταν ο Client εκτελέσει το συγκεκριμένο Service τότε το Workflow ξεκινά να εκτελείτε. Στο Service που δημιουργείται μέσω των Activities που περιγράφτηκαν πριν μπορεί να πάρει και παραμέτρους για την αρχικοποίηση του.

Για την δημιουργία του workflow δημιουργήθηκε ο αντίστοιχος κώδικας ο οποίος όπως περιγράφτηκε και πριν ενσωματώθηκε σε Code Activities. Το αποτέλεσμα είναι να δημιουργηθούν τα αντίστοιχα Code Activities

Save_application : Το συγκεκριμένο activity αποθηκεύει το αίτημα του χρήστη στην βάση δεδομένων.

Create_ArffFileAttributes : Το συγκεκριμένο activity δημιουργεί τα χαρακτηριστικά που πρέπει να έχει το arff αρχείο

Create_ArffFileBody: Το συγκεκριμένο activity δημιουργεί το σώμα που πρέπει να έχει το αρχείο.

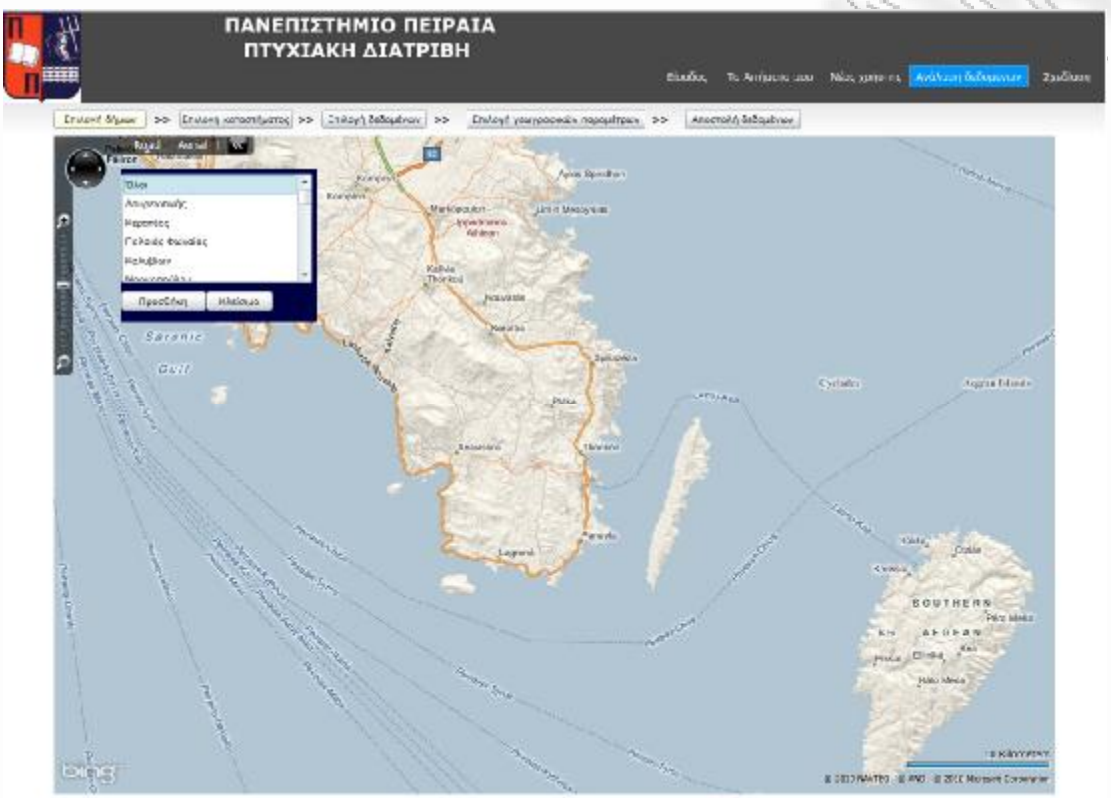
Weka : Σ' αυτό το activity εκτελείτε ο αλγόριθμος του SimpleKMeans για την δημιουργία των clusters

Save_weka_ti_db_results : Αποθηκεύεται το αποτέλεσμα στην βάση δεδομένων.

analyze_weka_fle : Το αποτέλεσμα που παραχθεί από το Weka διαβάζεται αυτόματα και αποθηκεύεται η ανάλυση του στην βάση δεδομένων.

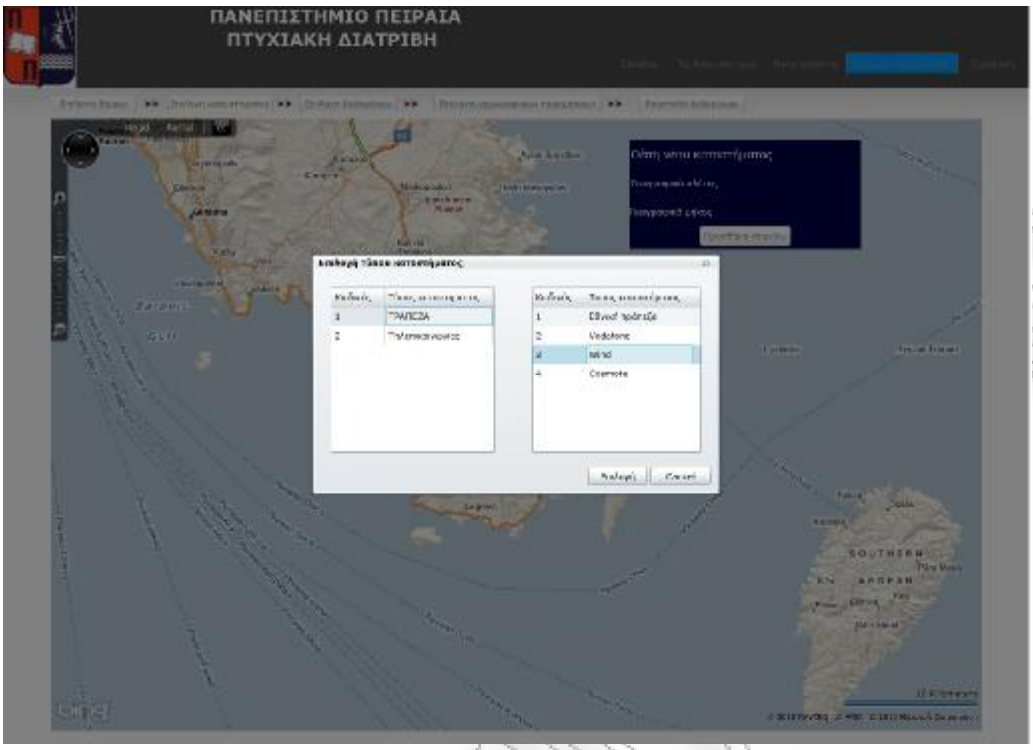
Παράδειγμα σεναρίου χρήσης

Με βάση την ανάλυση που έγινε στις επόμενες εικόνες αναπαριστάτε η εκτέλεση του.



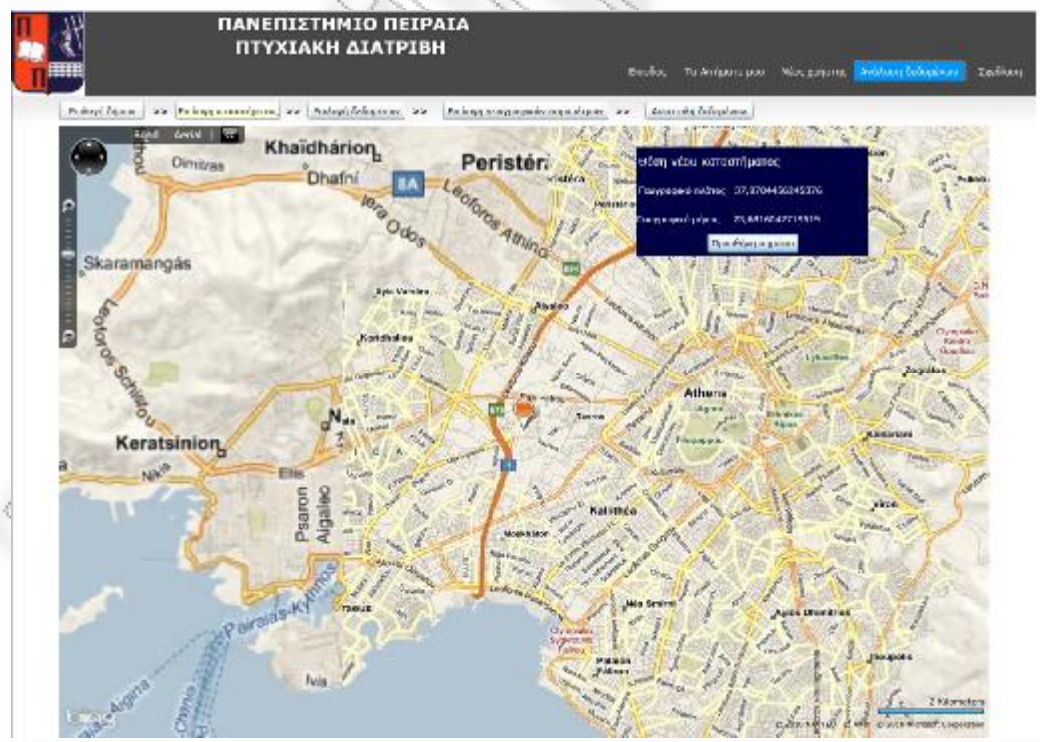
Εικόνα 65 Ανάλυση δεδομένων 1ο βήμα επιλογή δήμων

Στην πρώτη εικόνα επιλέγει ο χρήστης τους δήμους που επιθυμεί να συμπεριλάβει στην ανάλυση δεδομένων.



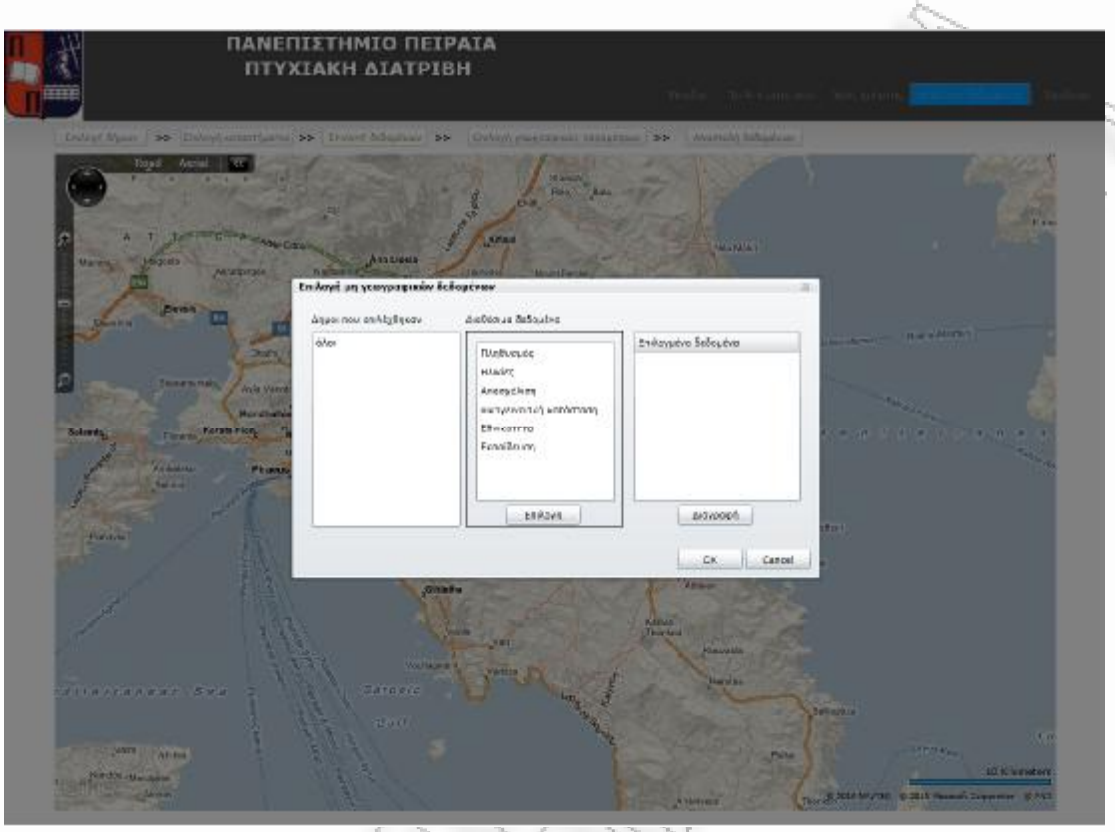
Εικόνα 66 Ανάλυση δεδομένων 2ο βήμα επιλογή τύπου καταστήματος και ονομασίας

Στην επόμενη εικόνα ο χρήστης επιλέγει την κατηγορία του καταστήματος που επιθυμεί να ανοίξει όπως και την ονομασία του. Η ονομασία του καταστήματος έχει ιδιαίτερη σημασία όταν το κατάστημα είναι franchise.



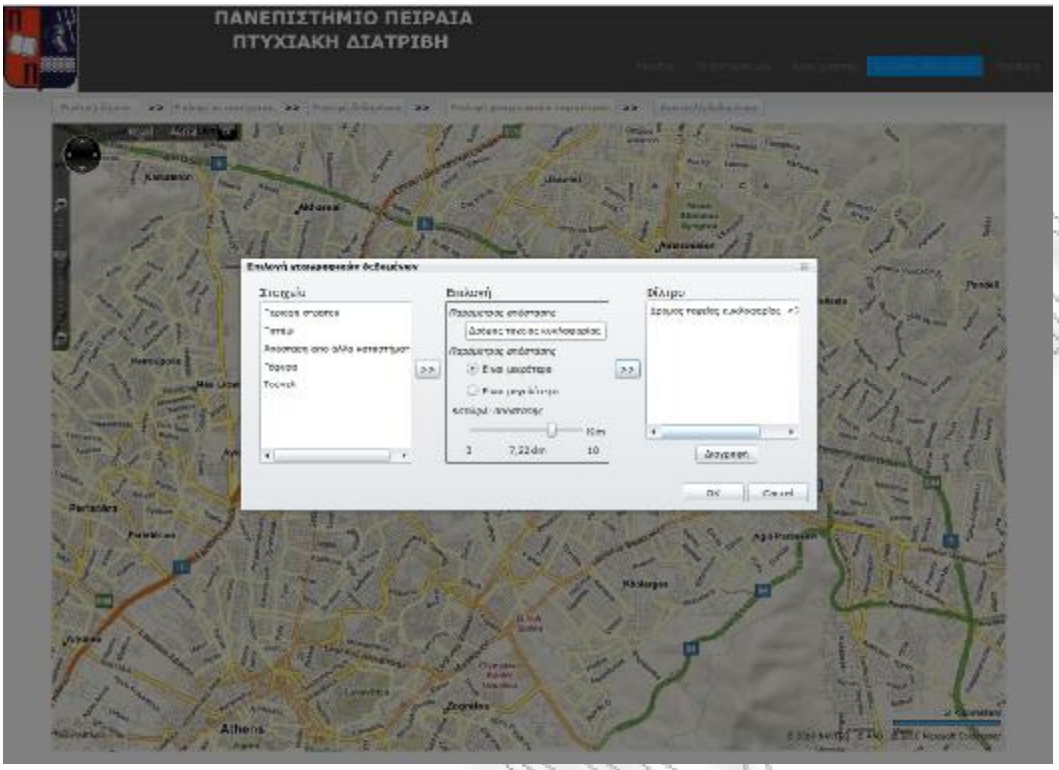
Εικόνα 67 Ανάλυση δεδομένων 2ο βήμα επιλογή τοποθετήσεων του νέου καταστήματος

Στο βήμα της εικόνας 68 ο χρήστης κάνοντας διπλό κλικ στο χάρτη επιλέγει την νέα τοποθεσία του καταστήματος. Πατώντας το κουμπί «Προσθήκη σημείου» εισάγεται ένα νέο pin στον χάρτη.



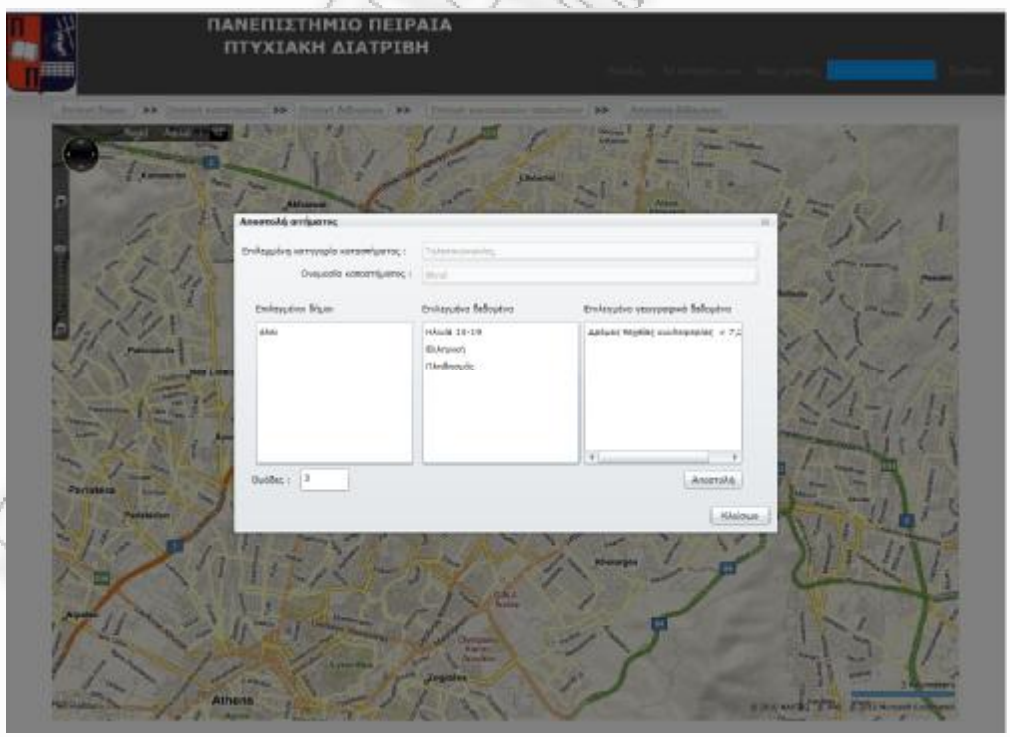
Εικόνα 68 Ανάλυση δεδομένων 2ο βήμα επιλογή δημογραφικών στοιχείων.

Στο σημείο αυτό ο χρήστης μπορεί να επιλέξει τα δημογραφικά στοιχεία που αφορούν το κατάστημα του.



Εικόνα 69 Ανάλυση δεδομένων 2ο βήμα επιλογή γεωγραφικών φίλτρων

Στην εικόνα 69 δείχνει την δυνατότητα του χρήστη να εισάγει στην ανάλυση του και γεωγραφικά φίλτρα.



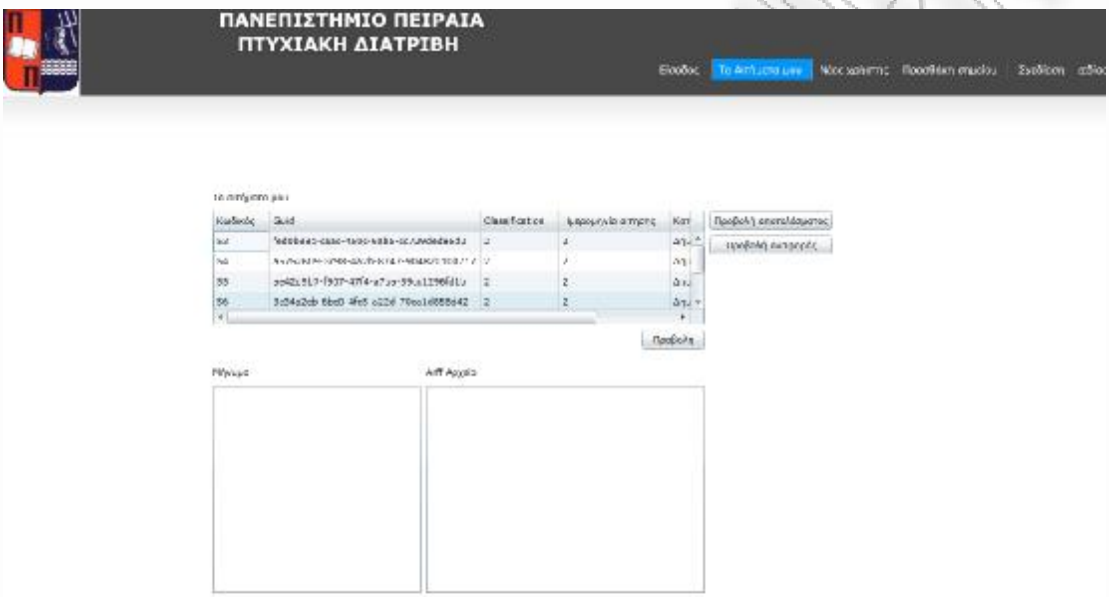
Εικόνα 70 Ανάλυση δεδομένων 2ο βήμα επιλογή γεωγραφικών φίλτρων

Τέλος, πατώντας το κουμπί «Αποστολή δεδομένων» εμφανίζεται η διεπαφή της εικόνας 70. Στην συγκεκριμένη διεπαφή στον χρήστη προβάλλονται όλα τα δεδομένα που έχει

επιλέξει. Υπάρχει ένα πεδίο με το όνομα «Ομάδες» στο οποίο ο χρήστης επιλέγει τις ομάδες που θέλει να δημιουργηθούν μετά την εκτέλεση της ανάλυσης δεδομένων.

Υποσύστημα Αιτήματα χρήστη

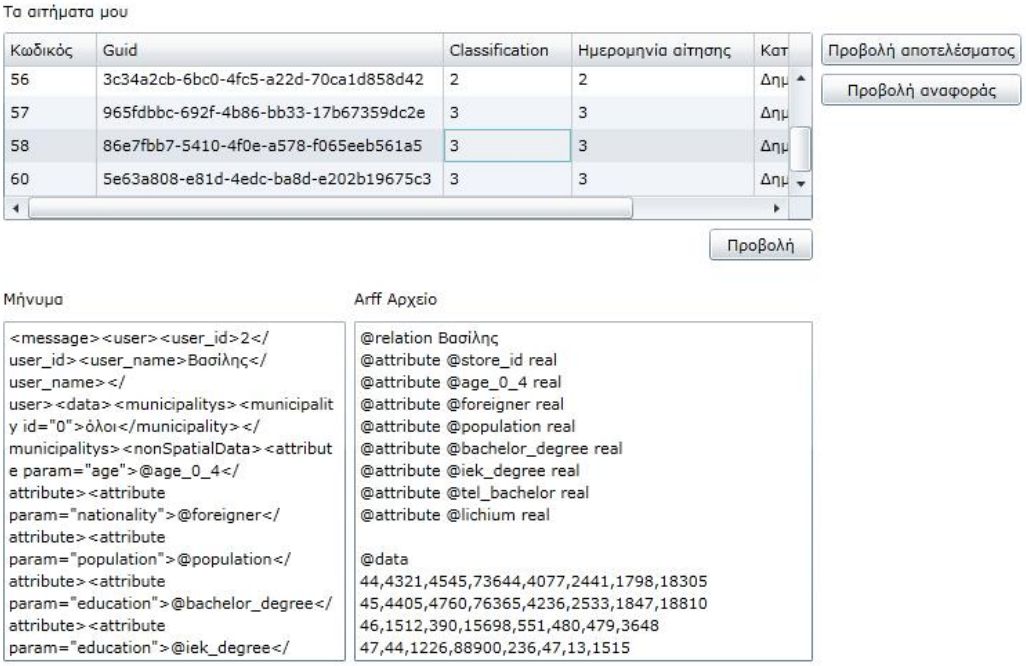
Ο χρήστης όταν εισέλθει στο σύστημα μπορεί να δει τα αιτήματα που έχει κάνει. Η διεπαφή που διαχειρίζεται τα αιτήματα που έχει κάνει ο χρήστης εμφανίζεται στην εικόνα 71 . Αίτημα στο πληροφοριακό σύστημα είναι το αποτέλεσμα της διαδικασίας του Data Mining. Το αποτέλεσμα εκτέλεσης Data Mining αποθηκεύεται στην βάση δεδομένων. Δημιουργήθηκαν οι αντίστοιχες διεπαφές για την προβολή του αιτήματος.



Εικόνα 71 Αιτήματα χρήστη

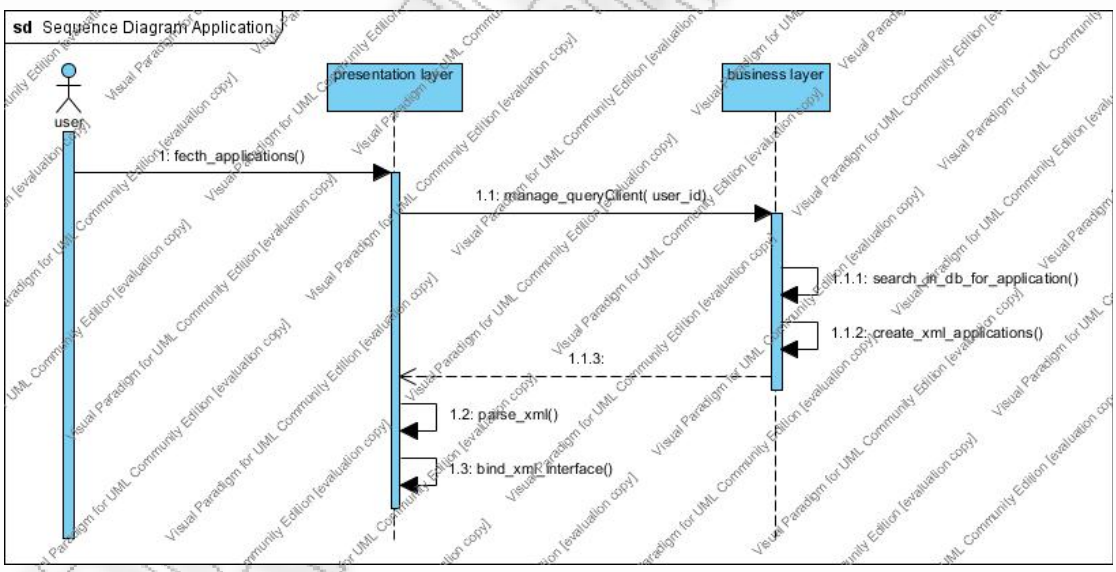
Στην εικόνα 72 δείχνει τα αιτήματα που έχει κάνει ένας χρήστης. Κάθε νέο αίτημα στο σύστημα έχει έναν μοναδικό κωδικό που τον χαρακτηρίζει, ένα GUID. Τον αριθμό κλάσεων που έχει ζητήσει ο χρήστης και την κατάσταση που βρίσκεται το αίτημα του. Η συγκεκριμένη διεπαφή έχει πολλές λειτουργίες που χρειάζονται ανάλυση.

Στο κάτω μέρος της διεπαφής εμφανίζονται δύο περιοχές, η μία δείχνει το μήνυμα που έχει αποστείλει ο χρήστης και το άλλο πεδίο είναι το arff αρχείο που έχει δημιουργηθεί. Η επόμενη εικόνα δείχνει την συγκεκριμένη λειτουργία όταν ο χρήστης πατήσει σε ένα από τα αιτήματα του και πατήσει το κουμπί προβολή.



Εικόνα 72 Προβολή λεπτομερών αιτήματος

Η ανάλυση εκτέλεσης της παραπάνω λειτουργίας αναλύεται στο παρακάτω sequence διάγραμμα.

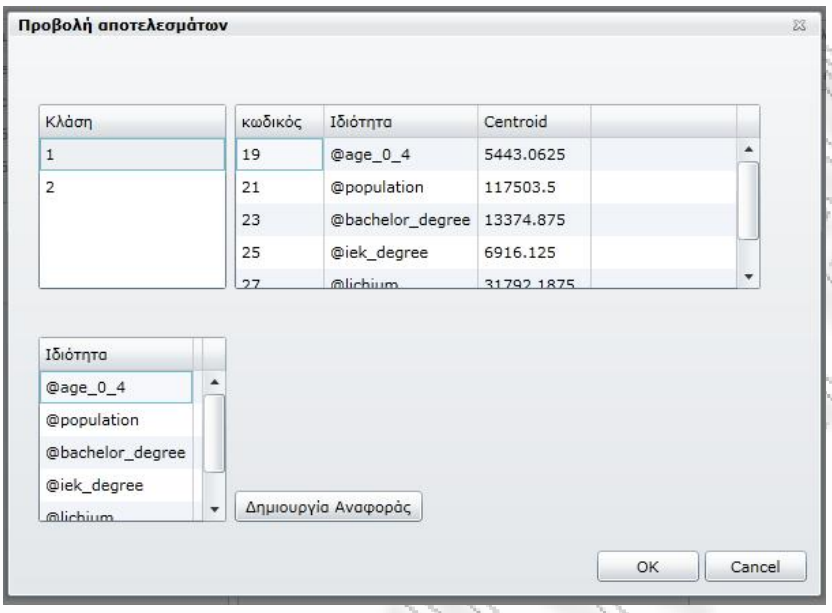


Εικόνα 73 Sequence diagram προβολή αιτημάτων

Προβολή αποτελέσματος

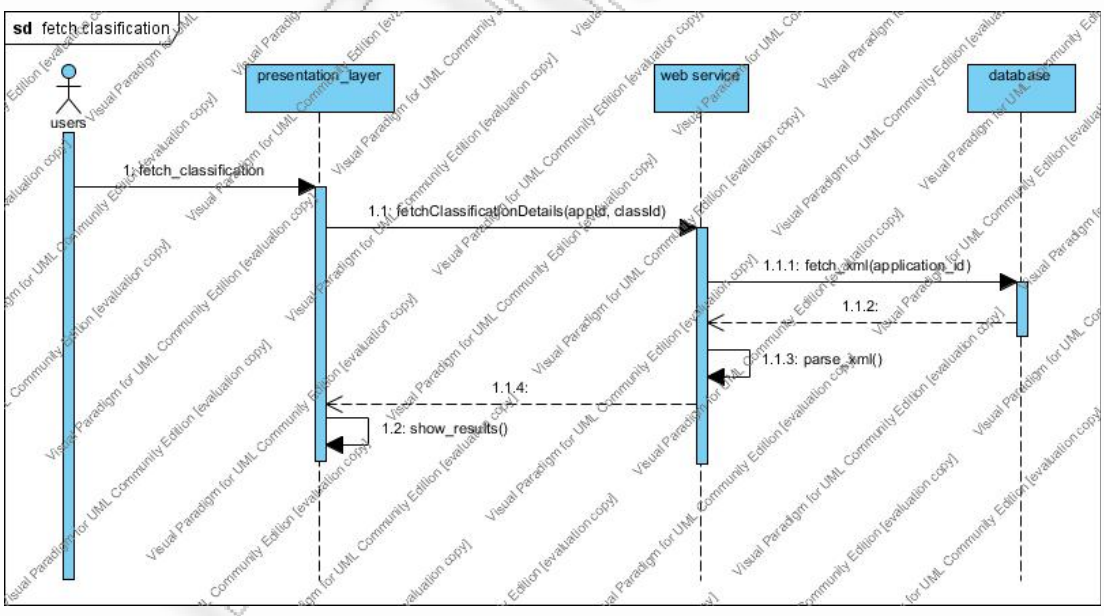
Η λειτουργία προβολή αποτελέσματος μπορεί να εκτελεστεί μόνο όταν ο έχει τελειώσει η εκτέλεση του αλγόριθμου SimpleKMeans. Όπως αναφέρθηκε στην ενότητα εκτέλεσης αλγόριθμου SimpleKMeans όταν εκτελεστεί ο αλγόριθμος τότε τα αποτελέσματα του αναλύονται και αποθηκεύονται στην βάση δεδομένων.

Δημιουργήθηκε η συγκεκριμένη διεπαφή για την προβολή ανάλυσης που έγινε αυτόματα από τον αλγόριθμο. Η επόμενη εικόνα δείχνει την συγκεκριμένη διεπαφή.



Εικόνα 74 Διεπαφή αποτελεσμάτων

Για το συγκεκριμένο αποτέλεσμα ο χρήστης έχει ζητήσει δύο κλάσεις αυτό εμφανίζεται στον πίνακα «*κλάση*». Ο χρήστης πατώντας στην συγκεκριμένη κλάση στον πίνακα δίπλα εμφανίζομαι οι ιδιότητες που έχει και τα κέντα (centroids) που έχουν δημιουργηθεί. Στο επόμενο Sequence διάγραμμα αναλύεται το τρόπος με τον οποίο εκτελείτε η συγκεκριμένη λειτουργία.



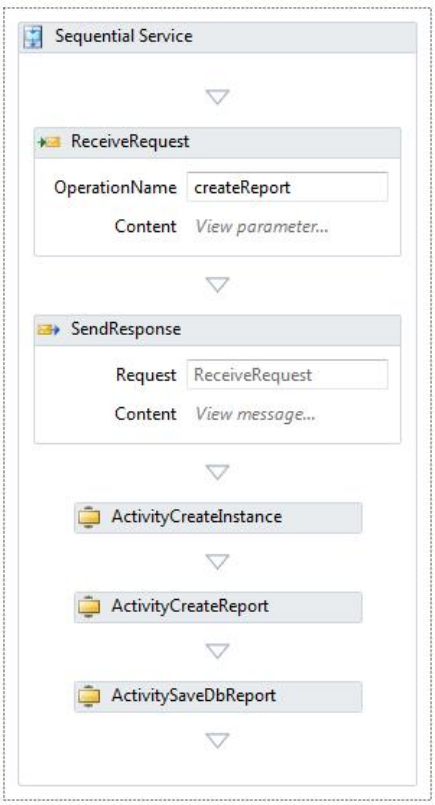
Εικόνα 75 Sequence Diagram Προβολή classifiation

Μέσα από την συγκεκριμένη διεπαφή ο χρήστης έχει ακόμα μία λειτουργία. Μπορεί να επιλέξει μία ιδιότητα μέσα από τις είδη υπάρχουσες και να εκτελέσει την ανάλυσή της.

Ανάλυση δεδομένων

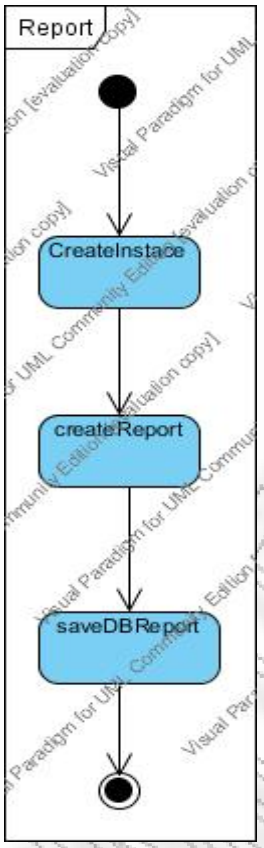
Η ομαδοποίηση των δεδομένων έχει πραγματοποιηθεί σε ένα πολυδιάστατο χώρο. Μέσα τον χώρο των διαστάσεων μπορεί ο χρήστης να επιλέξει ανάλυση της συγκεκριμένης ιδιότητας ως προς τις ομάδες.

Είναι πάρα πολύ σημαντικό σημείο της εφαρμογής είναι η αναπαράσταση των δεδομένων με βάση την ανάλυση. Ο χρήστης επιλέγοντας ένα από τα δημογραφικά χαρακτηριστικά μπορεί να «δει» σε πια ομάδα ανοίκει. Από την διεπαφή της εικόνας 74 ο χρήστης επιλέξει την ιδιότητα που επιθυμεί να γίνει η ανάλυση και πατάει το κουμπί «Δημιουργία αναφοράς». Τότε θα εκτελεστεί το Workflow με όνομα *createRepot* το οποίο θα δημιουργήσει την συγκεκριμένη αναφορά. Το Workflow δημιουργήθηκε χρησιμοποιώντας το Visual Studio 2010 και έχει την μορφή της εικόνα 76.



Εικόνα 76 Workflow Report

Δημιουργήθηκε ένα Sequence Workflow το οποίο μπορεί να το καλέσει ο client μέσω ενός Web Service. Το Web Service θα έχει το όνομα *createReport* και θα δέχεται τρείς παραμέτρους το id της αίτησης η ιδιότητα που θα ελεγχθεί και τέλος τον τύπο του καταστήματος. Το επόμενο State Machine Διάγραμμα αναλύει το πώς δημιουργείτε το Report.



Εικόνα 77 State Machine Diagram create Report

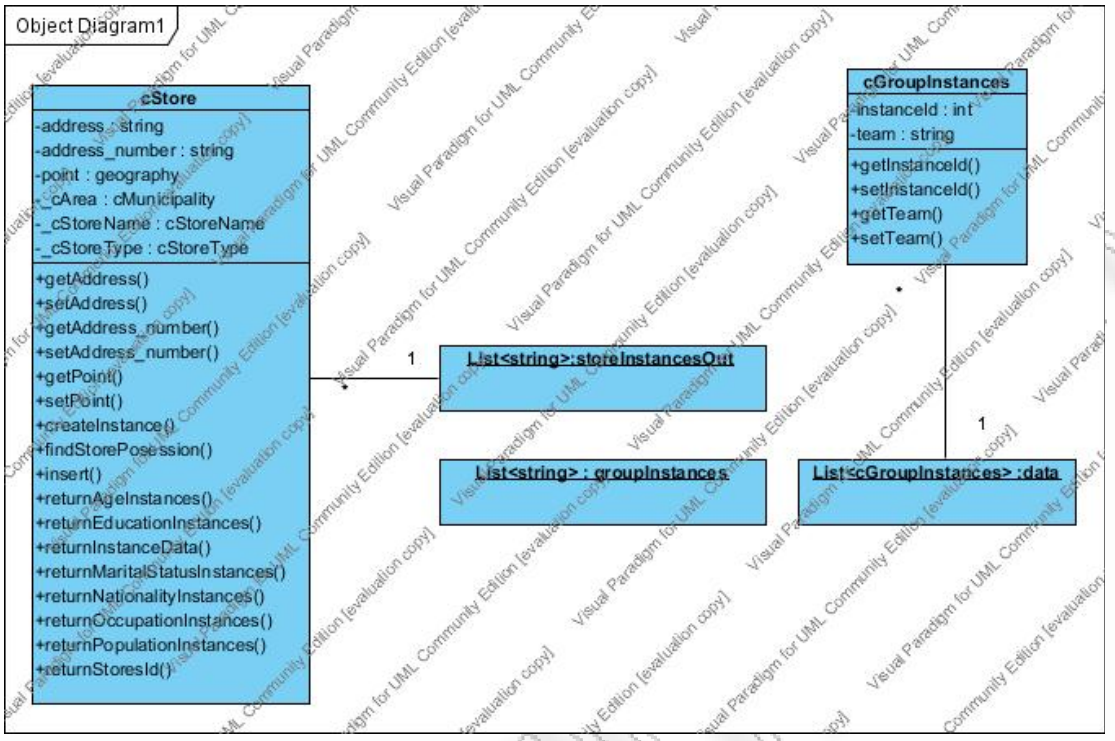
Για την δημιουργία του Report υλοποιήθηκαν τα αντίστοιχα Code Activities.

createInstance: Το συγκεκριμένο κομμάτι κώδικα δημιουργεί μία λίστα στην μνήμη με τον τύπο καταστημάτων που έχει επιλέξει ο χρήστης.

createReport: Το συγκεκριμένο κομμάτι κώδικα αναλύει τα δεδομένα από το Data Mining και βρίσκει σε ποια ομάδα ανήκει το συγκεκριμένο instance.

saveDBReport: Το συγκεκριμένο activity σώζει την ανάλυση των δεδομένων στην βάση δεδομένων.

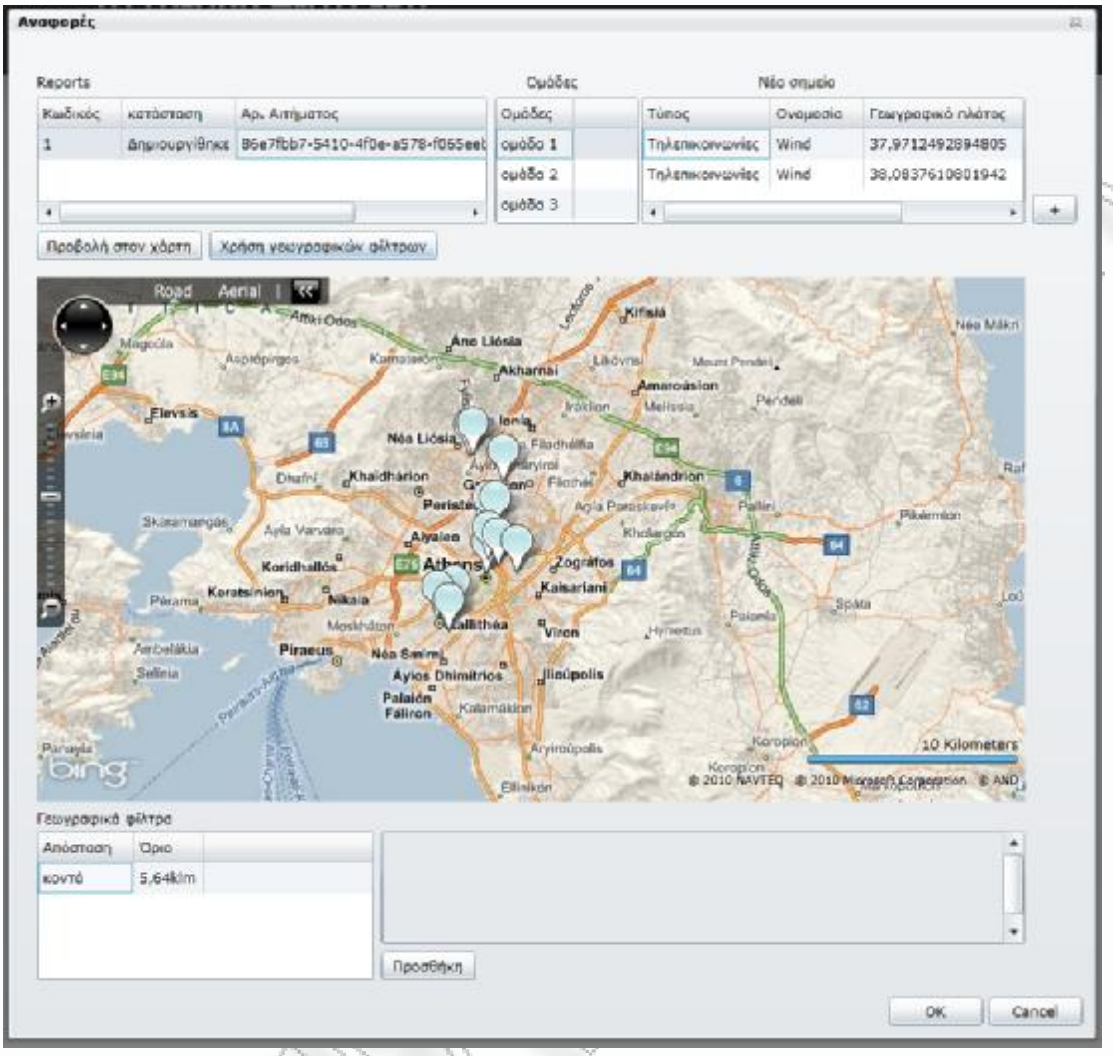
Το επόμενο object διάγραμμα αναλύονται τα αντικείμενα που χρησιμοποιούνται για την δημιουργία του Report.



Εικόνα 78 Object diagram

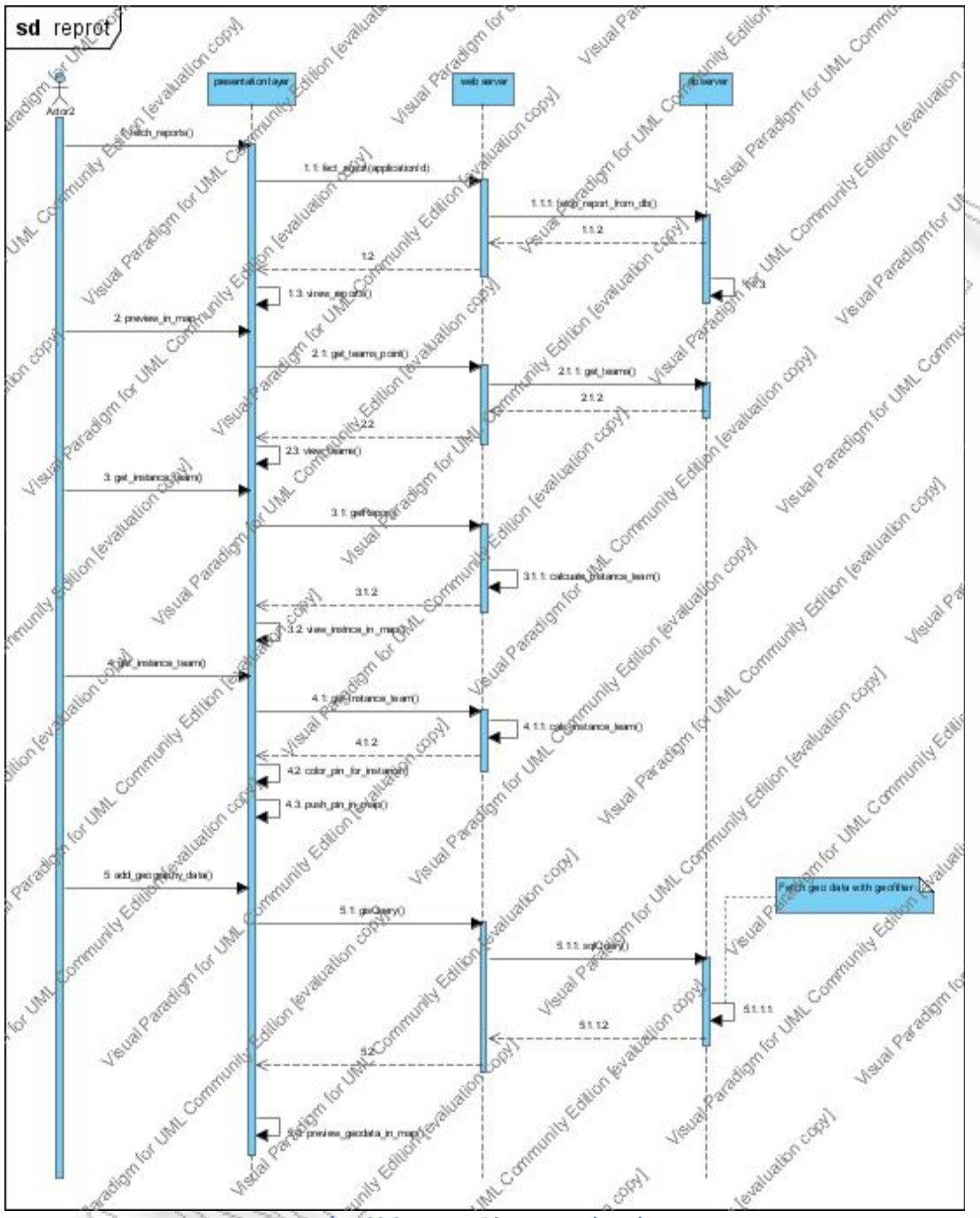
Το τελευταίο βήμα είναι η συγκεκριμένη ανάλυση να γίνει αναπαράσταση στον χάρτη. Με βάση την διεπαφή της εικόνας 71 ο χρήστης μπορεί να πατήσει το κουμπί «Προβολή αναφοράς» θα του εμφανιστεί η επόμενη εικόνα.

Προβολή δεδομένων στον χάρτη



Εικόνα 79 Διεπαφή προβολή αναφοράς.

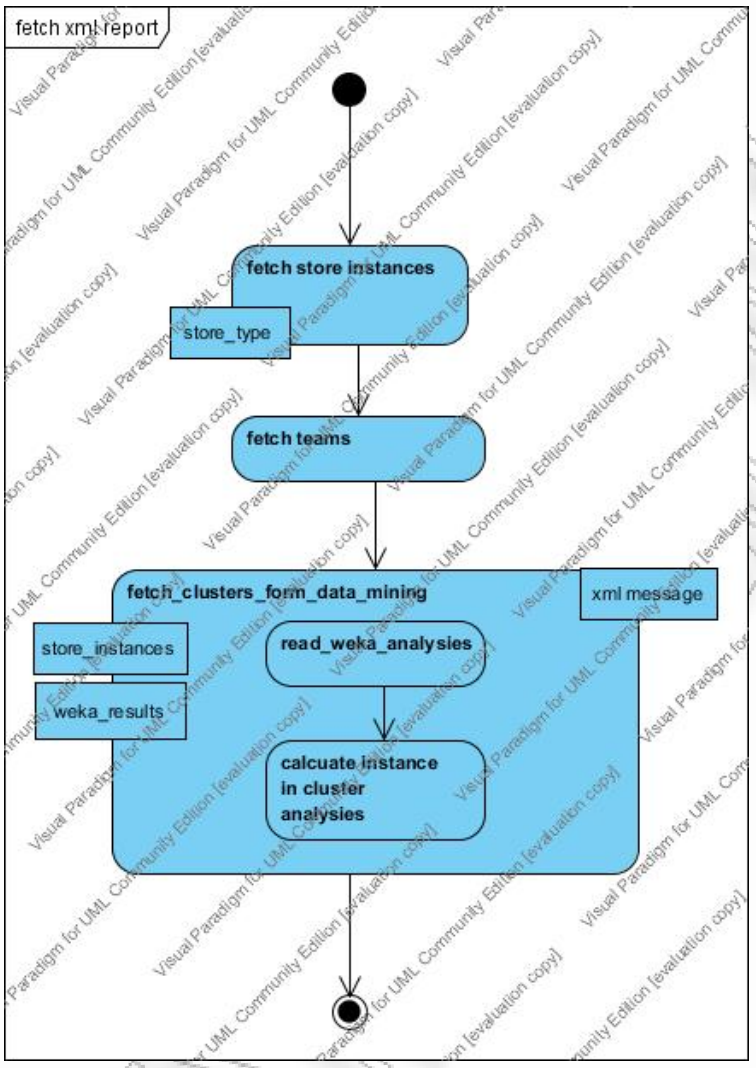
Ο χρήστης πατώντας το κουμπί «Προβολή στον χάρτη» τότε εκτελείτε το Web Service *view_report_user* . Το συγκεκριμένο Service θα φέρει αποτέλεσμα μόνο όταν έχει τελειώσει η διαδικασία του Report. Το αποτέλεσμα που θα φέρει είναι σε μορφή xml και έχει τα καταστήματα που έχει επιλέξει ο χρήστης βάση της κατηγορίας τους και σε ποια ομάδα ανήκουν. Το επόμενο Sequence διάγραμμα αναλύει την ροή εκτέλεσης των Web Service.



Εικόνα 80 Sequence Diagram analyze data

Στο συγκεκριμένο διάγραμμα ο χρήστης εκτελεί μέσω τις διεπαφης πέντε λειτουργίες. Η πρώτη είναι όταν φορτώνετε η διεπαφή εκτελείτε η συνάρτηση *fetch_reports* η οποία φέρνει για το συγκεκριμένο αίτημα που έχει κάνει ο χρήστη τις αναφορές που έχει ζητήσει.

Η δεύτερη και η Τρίτη εκτελούνται μαζί. Μόλις ο χρήστης πατήσει το κουμπί «Προβολή στον χάρτη» τότε θα ζητήσει τις ομάδες που έχουν δημιουργηθεί από την ανάλυση δεδομένων και σε τι ομάδες ανήκουν τα δεδομένα που εκπαίδευσαν τον αλγόριθμο. Το συγκεκριμένο service θα επιστρέψει ένα xml αρχείο. Ο αλγόριθμος που εκτελείτε για την εύρεση του κάθε instance σε ποια ομάδα ανήκει εμφανίζεται στον επόμενο state machine διάγραμμα.

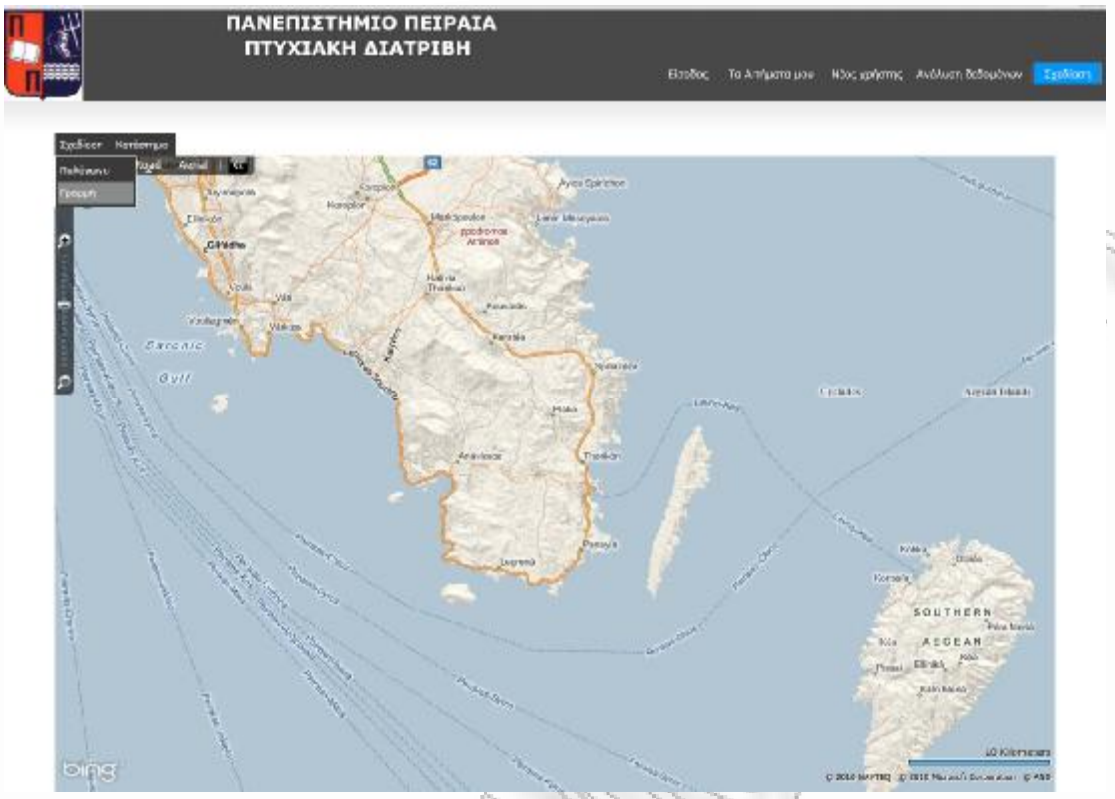


Εικόνα 81 Activity diagram

Με το διάγραμμα της εικόνας 80 αναπαριστάτε ο τρόπος που δημιουργείτε το xml αρχείου.

Υποσύστημα σχεδίασης

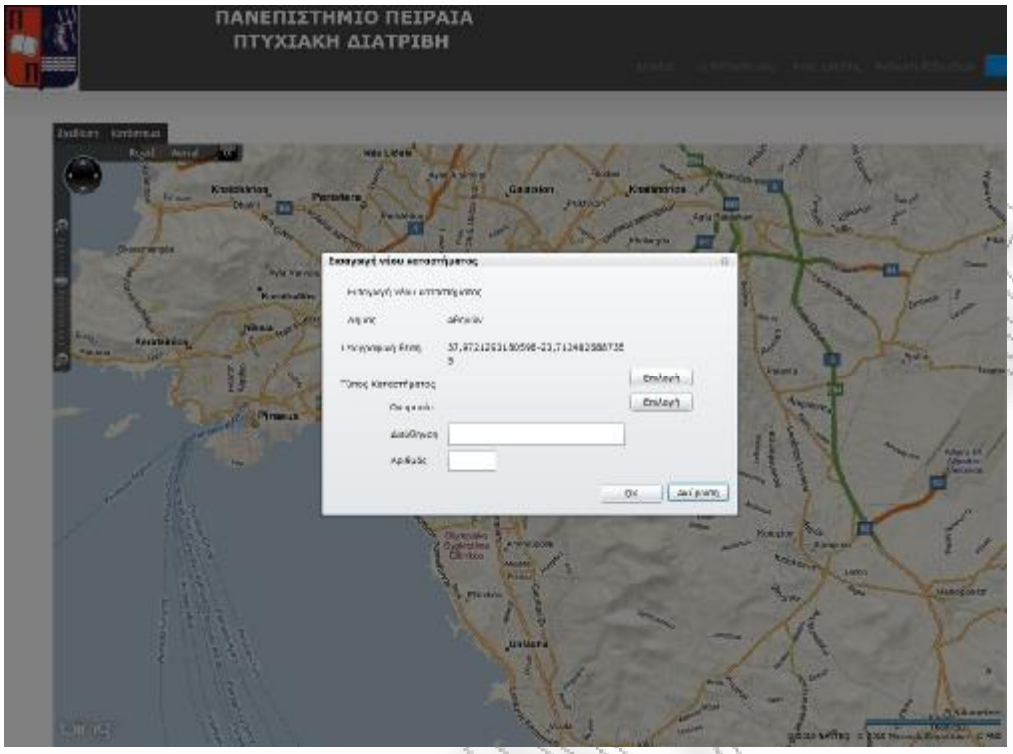
Το υποσύστημα σχεδίασης επιτρέπει στον εγγεγραμμένο χρήστη να προσθέτει νέα γεωγραφικά δεδομένα στο πληροφοριακό σύστημα. Εφόσον ο χρήστης έχει κάνει Login στην εφαρμογή μπορεί μέσω του menu να επιλέγει την «Σχεδίαση» Τότε θα του εμφανιστεί η επόμενη διεπαφή.



Εικόνα 82 Διεπαφή σχεδίαση

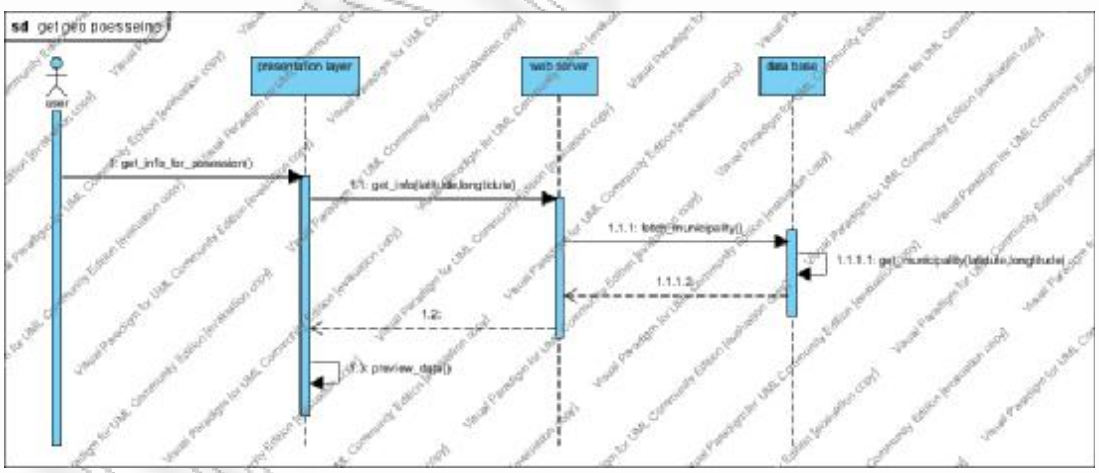
Μέσω της συγκεκριμένης διεπαφής ο χρήστης μπορεί να εισάγει ένα νέο γεωγραφικό σημείο ή μία νέα περιοχή ή μία νέα γραμμή. Χρησιμοποιώντας την δυνατότητα του tagging μπορεί να βάζει πληροφορίες για την συγκεκριμένη γεωγραφική θέση. Ακόμα έχει την δυνατότητα να προσθέσει ένα νέο κατάστημα στην βάση δεδομένων.

Μέσα από την διεπαφή ο χρήστης ένα επιλέξει Κατάστημα → Νέα εισαγωγή κάνοντας κλικ στην θέση που θέλει να τοποθετήσει το νέο κατάστημα του εμφανίζεται η επόμενη εικόνα.



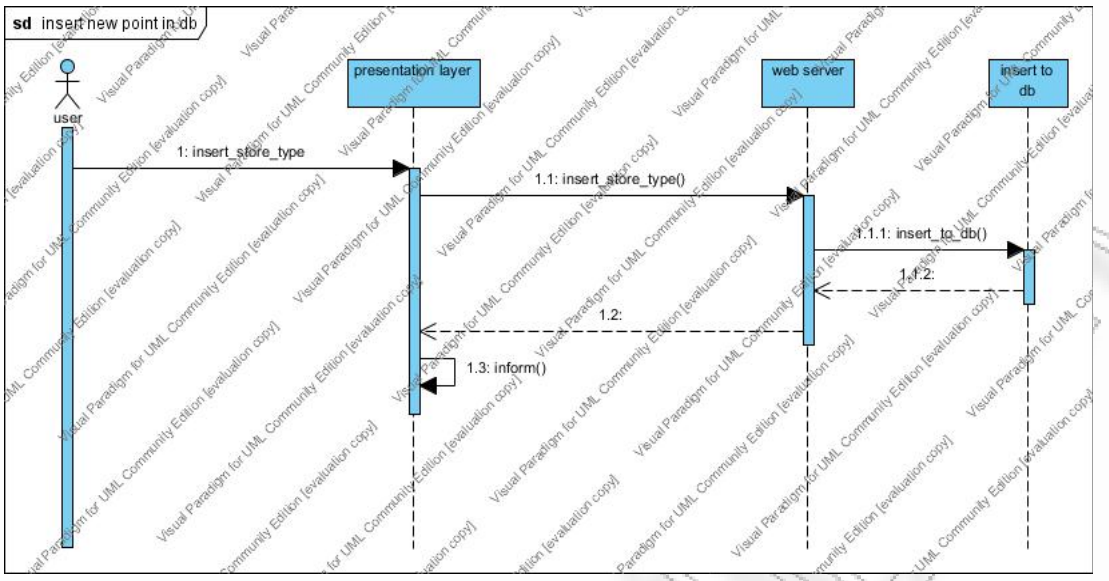
Εικόνα 83 Διεπαφή Εισαγωγή νέου καταστήματος

Ο χρήστης κάνοντας ένα κλικ πάνω στον χάρτη του εμφανίζεται η διεπαφή της εικόνας 82. Του επιστρέφει το γεωμετρικό πλάτος και μήκος και τον δήμο που επέλεξε. Αυτές οι πληροφορίες έρχονται στον client μέσω ενός Web Service και από την βάση δεδομένων. Το επόμενο Sequence διάγραμμα αναλύει το πώς έρχονται στον client οι πληροφορίες αυτές.



Εικόνα 84 Sequence diagram προβολή δήμου - γεωγραφικού σημείου

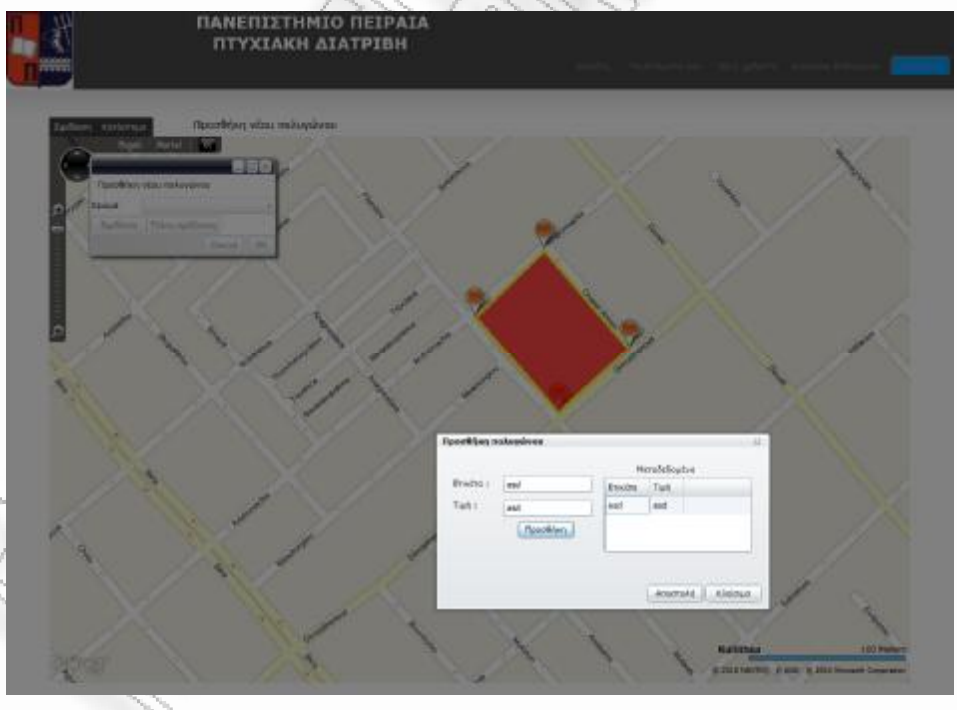
Ο χρήστης θα πρέπει να επιλέξει Τον τύπο του καταστήματος και το όνομα που θα έχει. Αυτό μπορεί να γίνει πατώντας το κουμπί «Επιλογή». Πατώντας το κουμπί «OK» γίνετε η εισαγωγή στην βάση δεδομένο το νέο σημείο. Το επόμενο Sequence διάγραμμα περιγράφεται ο τρόπος που πραγματοποιείται η εισαγωγή του νέου σημείου στην εφαρμογή.



Εικόνα 85 Sequence Diagram εισαγωγή νέου καταστήματος

Εισαγωγή νέου πολυγώνου

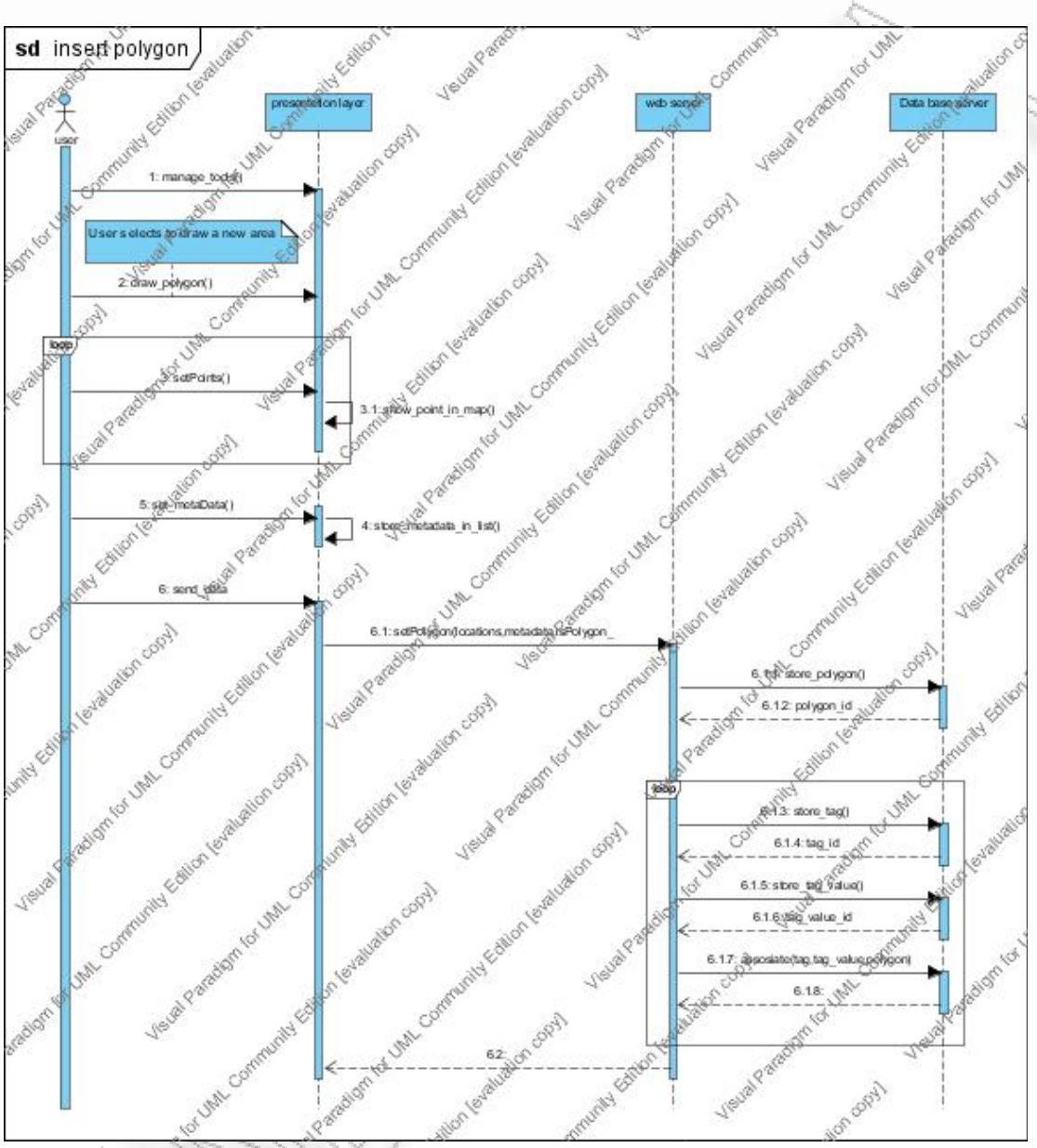
Μέσα από την επιλογή «διαχείριση» ο χρήστης μπορεί να εισάγει μία νέα περιοχή και να προσθέσει πληροφορία για αυτή. Η επόμενη εικόνα δείχνει την συγκεκριμένη διεπαφή.



Εικόνα 86 Διεπαφή εισαγωγή νέας περιοχής

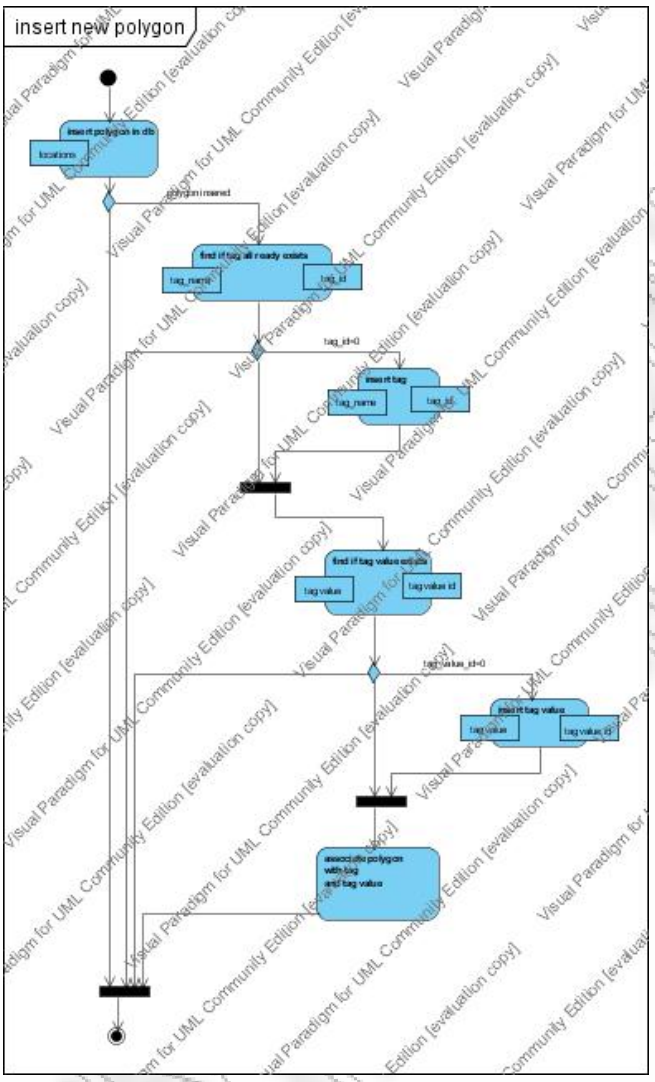
Ο χρήστης από τον menu επιλέγει Σχεδίαση → Πολύγωνο και εμφανίζεται το παράθυρο εισαγωγής νέου πολυγώνου. Ο χρήστης πατάει το κουμπί «Σχεδίαση» και κάνοντας κλικ στο κουμπί «Τέλος σχεδίασης» ολοκληρώνεται η σχεδίαση του. Τότε εμφανίζεται το παράθυρο εισαγωγής πληροφορίας για την συγκεκριμένη περιοχή. Στο συγκεκριμένο

παράθυρο ο χρήστης μπορεί να εισάγει ετικέτες για την συγκεκριμένη περιοχή. Πατώντας το κουμπί «Αποστολή» στέλνετε στον server το συγκεκριμένο αίτημα για επεξεργασία. Το επόμενο Sequence διάγραμμα αναλύει την συγκεκριμένη διαδικασία.



Εικόνα 87 Sequence diagram προσθήκη πολυγώνου

Ο αλγόριθμος που εκτελείται για την εισαγωγή πολυγώνου και την εισαγωγή πληροφορίας για αυτό εμφανίζεται στο επόμενο Activity diagram.

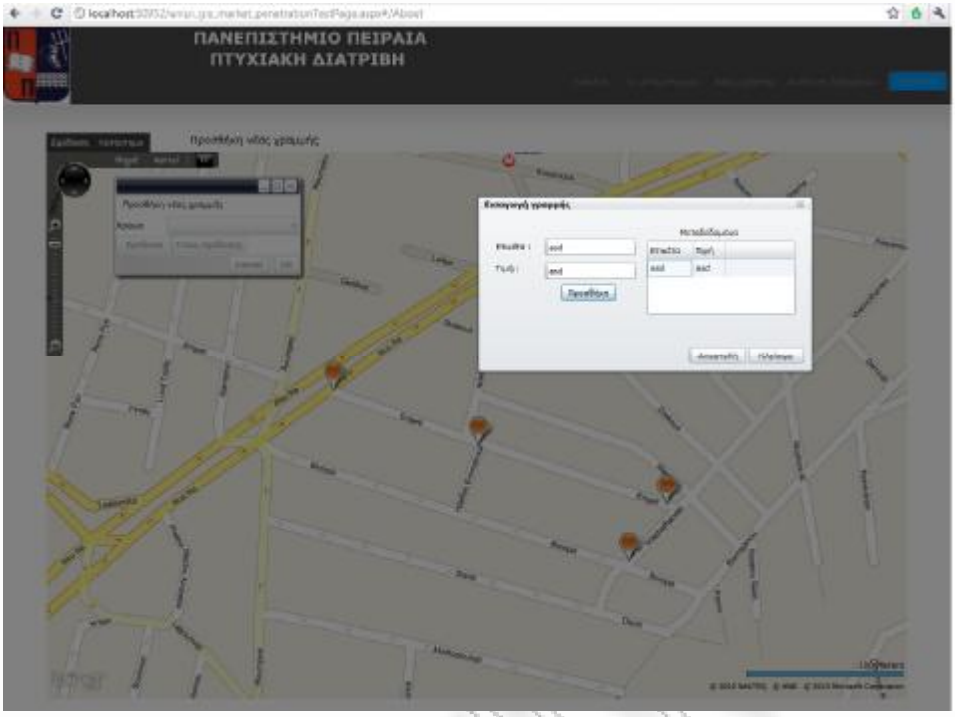


Εικόνα 88 Activity diagram εισαγωγή νέου πολυγώνου

Στο σχήμα της εικόνας 88 αναλύεται ο αλγόριθμος εισαγωγής νέου πολυγώνου στην βάση δεδομένων. Πρώτα εισάγεται το πολύγωνο στην βάση δεδομένων. Το δεύτερο βήμα είναι η εισαγωγή της ετικέτας και τέλος πρέπει να γίνει η εισαγωγή της τιμής. Εφόσον εισαχθούν όλες αυτές οι πληροφορίες στην βάση δεδομένων τότε πρέπει να γίνει και η συσχέτιση τους.

Εισαγωγή νέας γραμμής

Μέσα από την διαχείριση ο χρήστης μπορεί να εισάγει μία νέα γραμμή στην βάση δεδομένων. Ο χρήστης μπορεί να εισάγει μία νέα γραμμή μέσω από το menu Σχεδίαση → Γραμμή τότε θα εμφανιστεί η επόμενη διεπαφή.



Η διαδικασία εκτέλεση του για την εισαγωγή νέας γραμμή είναι ίδια με την εισαγωγή πολυγώνου. Η μόνη διαφορά είναι στο script του κεντρικού server, στο ερώτημα για την εισαγωγή νέου γεωγραφικού δεδομένων είναι τύπου γραμμής και όχι πολυγώνου.

Συλλογή γεωγραφικών δεδομένων μέσω του OpenStreetMap

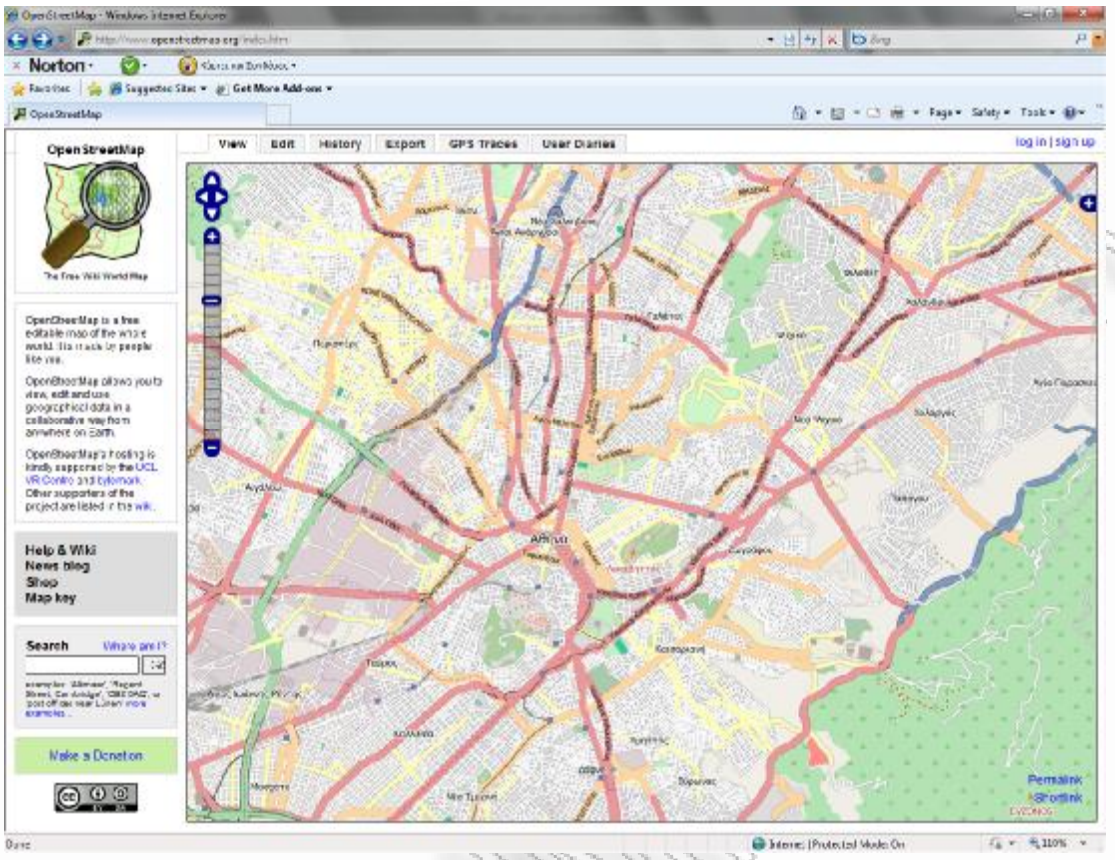
Εισαγωγή

Το openstreetmap είναι ένα μία εφαρμογή στο internet στην οποία έχουν συλλέξει γεωγραφικά δεδομένα από όλο τον κόσμο για τους δρόμους που έχει κάθε χώρα και σημεία ενδιαφέροντος . Μέσα από την εφαρμογή που παρέχει η σελίδα μπορούμε να εξάγουμε τα δεδομένα σε xml μορφή. Θα δημιουργηθούν τα αντίστοιχα script με τα οποία θα πραγματοποιηθεί η εξαγωγή δεδομένων. Αποτέλεσμα θα είναι να γεμίσει η βάση δεδομένων της εφαρμογής με γεωργικά δεδομένα.

Ανάλυση του XML Αρχείου

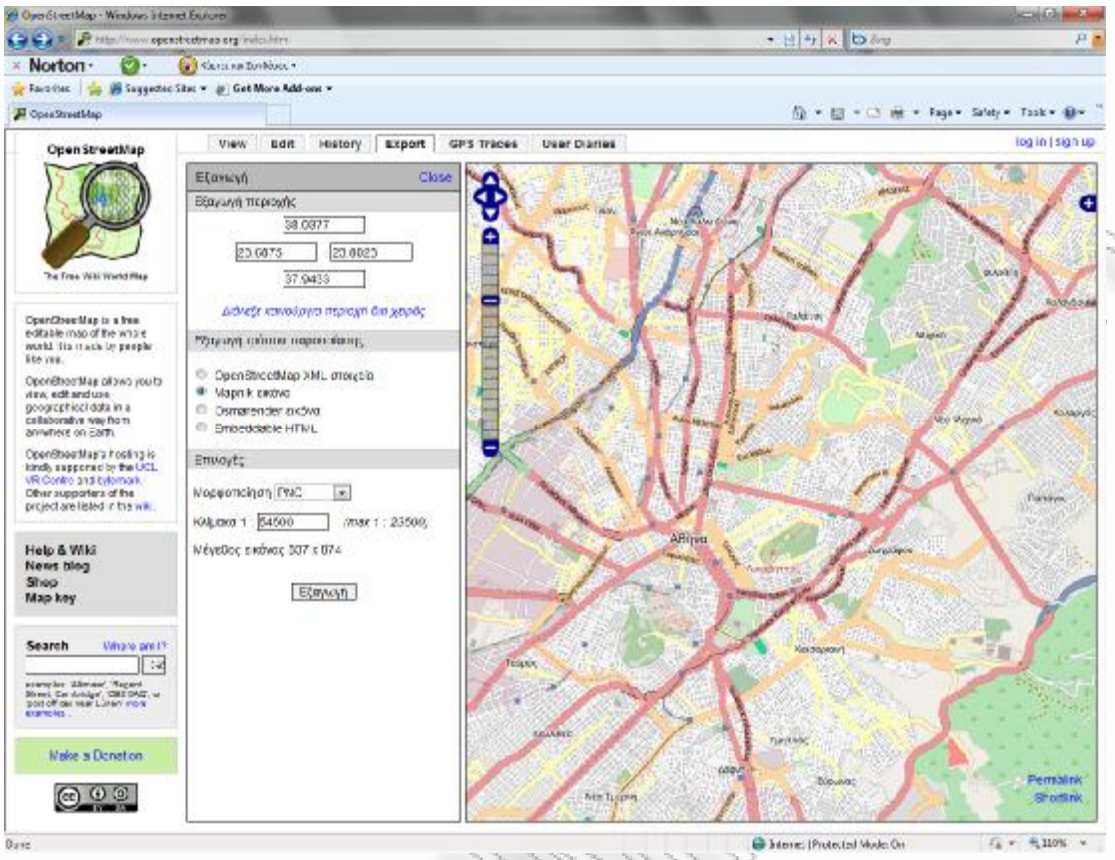
Export Data

Στην πρώτη φάση πρέπει να εξάγουμε το xml αρχείο από την εφαρμογή. Θα επιλέξουμε την περιοχή της Αθήνας για να εξάγουμε τα δεδομένα όπως εμφανίζεται στην επόμενη εικόνα.



Εικόνα 89 Open Street Map

Επιλέγουμε το export και εμφανίζεται η επόμενη εικόνα.



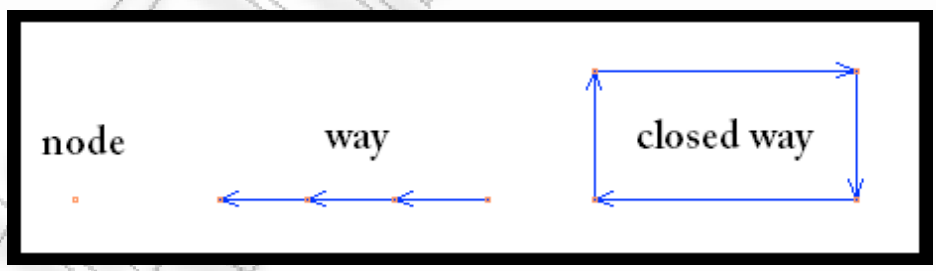
Εικόνα 90 Open Street Map export step 2

Πατάμε το κουμπί «Εξαγωγή».

Πατώντας το κουμπί «Εξαγωγή» πραγματοποιείται το download του αρχείου «map.osm».

Δομή XML αρχείου

Στο Xml αρχείο τα πρωταρχικά στοιχεία που έχει είναι οι κόμβοι, οι δρόμοι και οι σχέσεις. Την αναπαράσταση αυτή την βλέπουμε και στο επόμενο σχήμα.



Εικόνα 91 Xml Structure

Θα πραγματοποιήσουμε μία περεταίρω ανάλυση των βασικών στοιχείων που έχουμε στο xml αρχείο.

Κόμβος (node)

Ο κόμβος είναι ένα από τα βασικά γεωγραφικά δεδομένα. Ο κόμβος στο xml αρχείο περιέχει δύο πληροφορίες το latitude και longitude. Στο xml αρχείο για να δηλώσουμε ένα κόμβο έχουμε την ακόλουθη σύνταξη.

```
<node id="25496583" lat="51.5173639" lon="-0.140043" version="1"
changeset="203496" user="80n" uid="1238" visible="true"
timestamp="2007-01-28T11:40:26Z">
    <tag k="created_by" v="JOSM"/>
</node>
```

Τα στοιχεία που μας είναι χρήσιμα για περεταίρω επεξεργασία είναι τα ακόλουθα.

Id Η τιμή που περιέχει το attribute id είναι τύπου ακεραίου και είναι μοναδικό για κάθε κόμβο. Με αυτό τον τρόπο μπορούμε να χαρακτηρίσουμε μοναδικά ένα κόμβο.

Lat Διατηρεί την πληροφορία για το γεωγραφικό πλάτος.

Lng Διατηρεί την πληροφορία για το γεωγραφικό μήκος.

Δρόμος (Way)

Με την δήλωση αυτή στο xml αρχείο μπορεί να περιγραφεί ένας δρόμος, μία διαδρομή, ένας σιδηρόδρομος, ένα ποτάμι, ένας φράκτης, μία γραμμή εταιρίας παροχής ηλεκτρισμού, ένα οικοδομικό τετράγωνο. Για να πραγματοποιηθεί αυτή η αναπαράσταση πρέπει να έχουν οριστεί παραπάνω από έναν κόμβο.

Κλειστός Δρόμος (Close Way)

Όταν δηλώνεται στο xml αρχείο τότε περιγράφεται μία διαδρομή ή οποία τελειώνει εκεί που ξεκίνα. Για να δηλωθεί ένας κλειστός δρόμος πρέπει να έχει ένα μοναδικό id μία λίστα με τους κόμβους (nd) που περιέχει και κάποια πεδία tag που έχουν ένα κλειδί και μία τιμή. Ένα παράδειγμα είναι το ακόλουθο.

```
  <way id="5090250" visible="true" timestamp="2009-01-19T19:07:25Z"
version="8" changeset="816806" user="Blumpsy" uid="64226">
    <nd ref="822403"/>
    <nd ref="21533912"/>
    <nd ref="821601"/>
    <nd ref="21533910"/>
    <nd ref="135791608"/>
    <nd ref="333725784"/>
    <nd ref="333725781"/>
    <nd ref="333725774"/>
    <nd ref="333725776"/>
    <nd ref="823771"/>
    <tag k="created_by" v="Potlatch 0.10e"/>
    <tag k="highway" v="unclassified"/>
    <tag k="name" v="Clipstone Street"/>
    <tag k="oneway" v="yes"/>
  </way>
```

Σχέσεις

Μία σχέση είναι μία ομάδα με ένα ή περιστερά βασικά στοιχεία που συνδέονται μεταξύ τους. Με αυτό τον τρόπο μπορούμε να περιγράψουμε σχέσεις που υπάρχουν μεταξύ δεδομένων.

```
  <relation id="12176" visible="true" timestamp="2008-08-
09T17:55:39+01:00" user="Andy Allan">
    <member type="way" ref="4068452" role="forward"/>
    <member type="way" ref="4082616" role="forward"/>
    <member type="way" ref="4253466" role=""/>
    <member type="way" ref="4254996" role="backward"/>
    <member type="way" ref="4254997" role="forward"/>
    <member type="way" ref="4255009" role="forward"/>
    <member type="way" ref="4255103" role="forward"/>
    <member type="way" ref="4255109" role="forward"/>
    <member type="way" ref="4256129" role="forward"/>
    <member type="way" ref="5090250" role="forward"/>
    <member type="way" ref="17926237" role="backward"/>
    <member type="way" ref="17926240" role=""/>
    <member type="way" ref="17926241" role=""/>
    <member type="way" ref="17929790" role="forward"/>
    <member type="way" ref="17929795" role="forward"/>
    <member type="way" ref="17944924" role="forward"/>
    <member type="way" ref="17944925" role="forward"/>
    <member type="way" ref="17944927" role="forward"/>
    <member type="way" ref="17944928" role="forward"/>
    <member type="way" ref="19211502" role=""/>
    <member type="way" ref="19212465" role="forward"/>
    <member type="way" ref="26164616" role=""/>
    <tag k="ref" v="0"/>
    <tag k="type" v="route"/>
    <tag k="created_by" v="Potlatch 0.10b"/>
    <tag k="network" v="lcn"/>
    <tag k="route" v="bicycle"/>
  </relation>
```

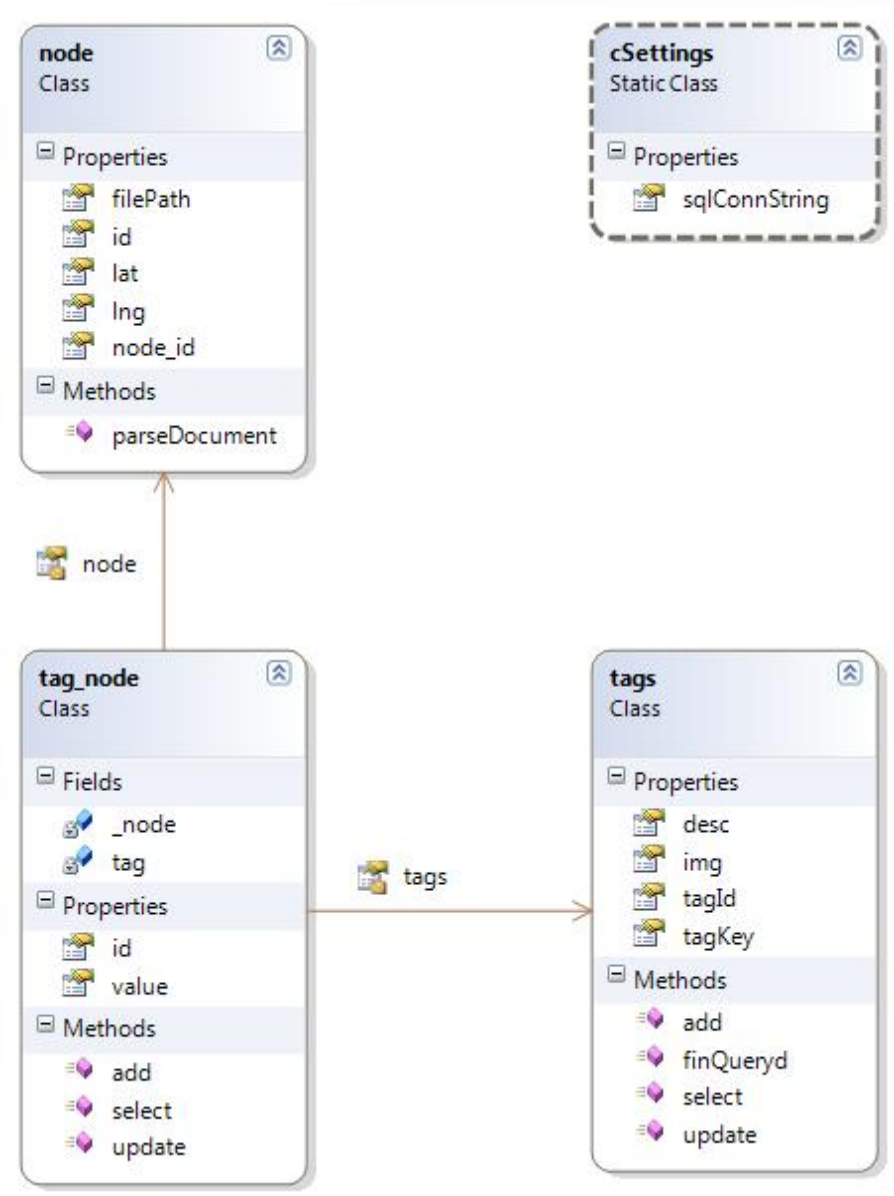
Σχεδίαση Βάσης Δεδομένων – Data layer.

Η ανάλυση που πραγματοποιήθηκε από το xml θα μας οδηγήσει σε μία σχεσιακή βάση δεδομένων. Με την ανάπτυξη της βάσης δεδομένων θα δημιουργήσουμε και τα αντίστοιχα αντικείμενα που θα διαχειρίζονται τους πίνακες (οντότητες). Η σχεδίαση – ανάπτυξη θα εξαρτηθεί πάρα πολύ από την δομή του xml.

Στο διάγραμμα αυτό η βασική οντότητα είναι ο κόμβος «OSM_node» ο οποίος έχει ένα βασικό κλειδί το id και το node_id που είναι η μοναδική τιμή που διαβάζουμε από το xml αρχείο. Στην οντότητα αυτή αποθηκεύουμε τις βασικές τιμές που έχουμε για κάθε κόμβο. Ο κάθε κόμβος έχει ένα η περισσότερα tags και το κάθε tag έχει ένα κλειδί «key» και μία τιμή «value».

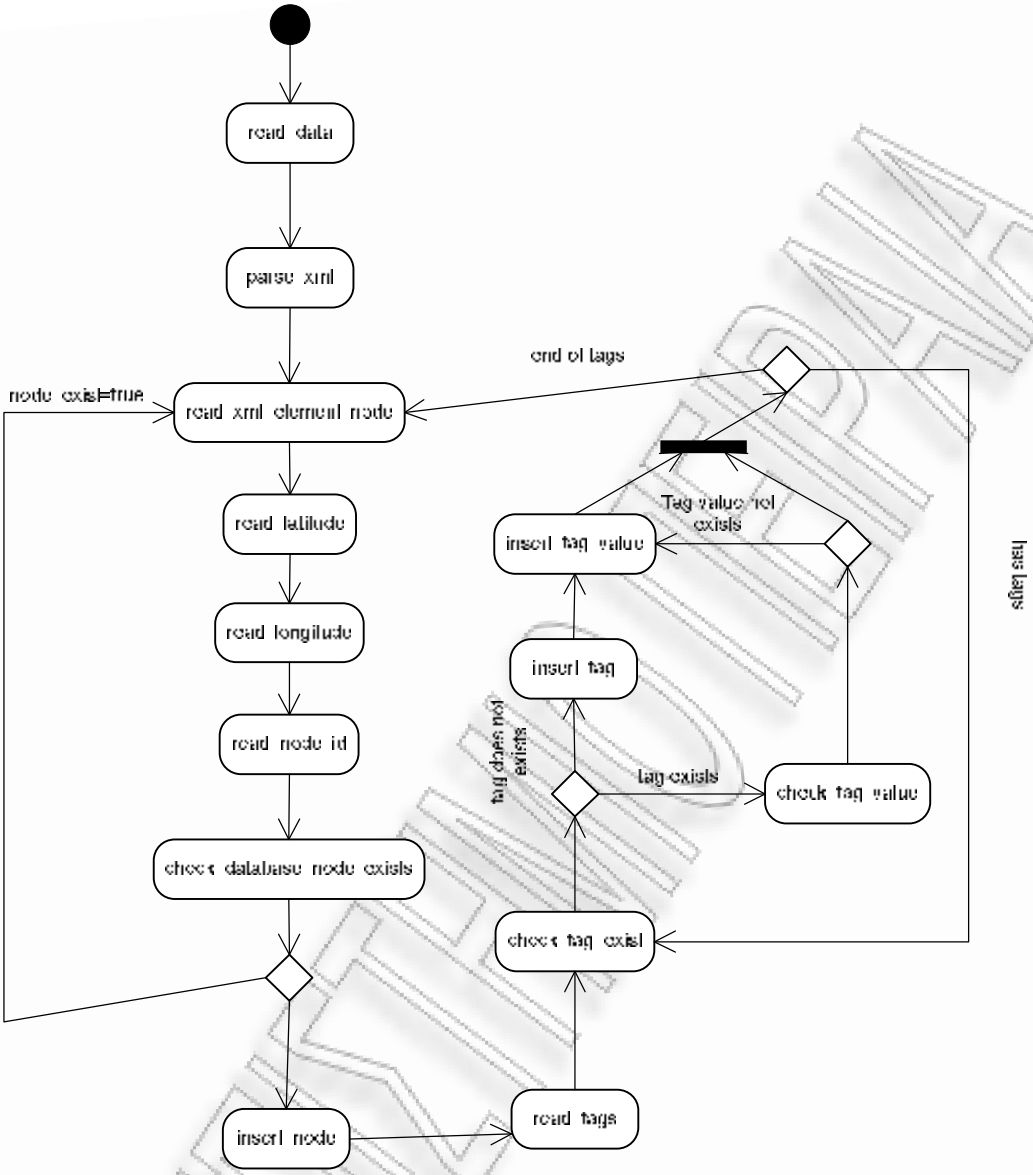
Στο project που έχουμε δημιουργήσει υπάρχει μία βιβλιοθήκη με όνομα OSMLib η οποία θα έχει τα βασικά αντικείμενα για την εφαρμογή.

Στην επόμενη εικόνα εμφανίζεται το διάγραμμα κλάσεων για το σχήμα βάσης που περιγράφτηκε πριν.



Εικόνα 92 Διάγραμμα κλάσεων

Η συνάρτηση η **parseDocument** η οποία το **xml** αρχείο και εισάγει τα **nodes** στην Β.Δ. Ο αλγόριθμος που υλοποιείται για να εκτελεστεί η συνάρτηση είναι ο ακόλουθος.



Εικόνα 93 statechart diagram - import node

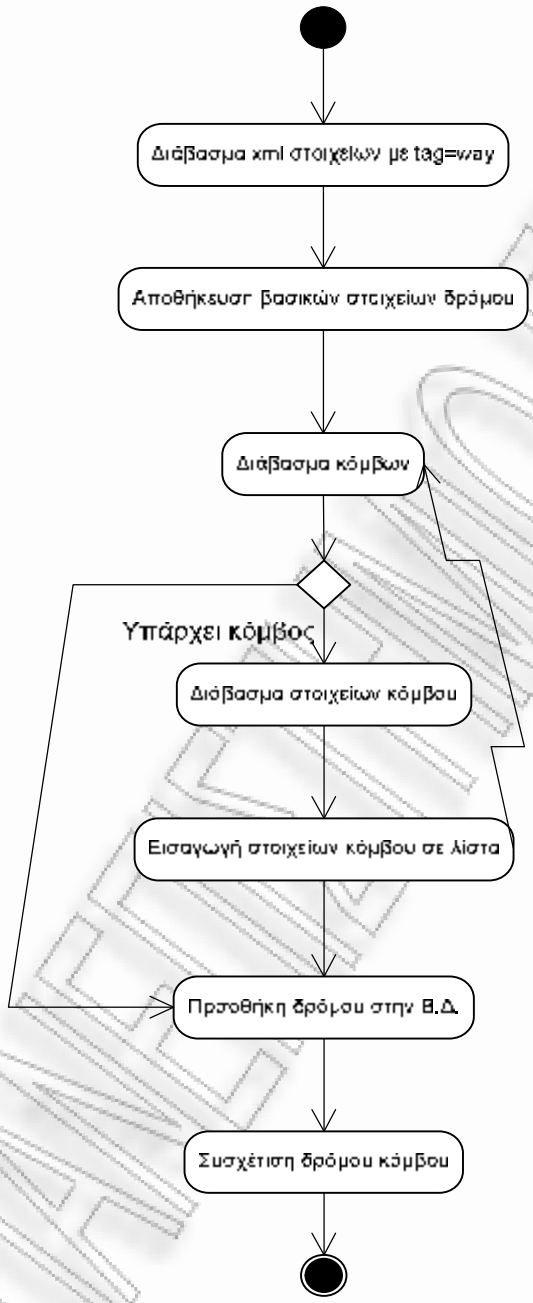
Ο αντίστοιχος κώδικας που έχει γραφτεί για να υλοποιηθεί το script βρίσκετε στο αρχείο node.cs και είναι η κλάση με το όνομα node.

Το XML αρχείο έχει και πληροφορίες για δρόμους. Θα πρέπει να δημιουργηθεί το αντίστοιχό scrip το οποίο θα μεταφέρει την πληροφορία από το αρχείο στον SQL Server. Στο script πρέπει να οριστούν κάποιες βασικές αρχές. Η μορφή που έχει ένας δρόμος είναι η ακόλουθη.

```
<way id="55363478" user="Simon Kokolakis" uid="23785" visible="true"
version="1" changeset="4415591" timestamp="2010-04-13T17:22:40Z">
  <nd ref="695578781"/>
  <nd ref="695578784"/>
  <nd ref="695578786"/>
  <nd ref="695578789"/>
  <nd ref="695578791"/>
  <nd ref="695578793"/>
  <nd ref="695578795"/>
  <nd ref="695578797"/>
  <nd ref="695578781"/>
```

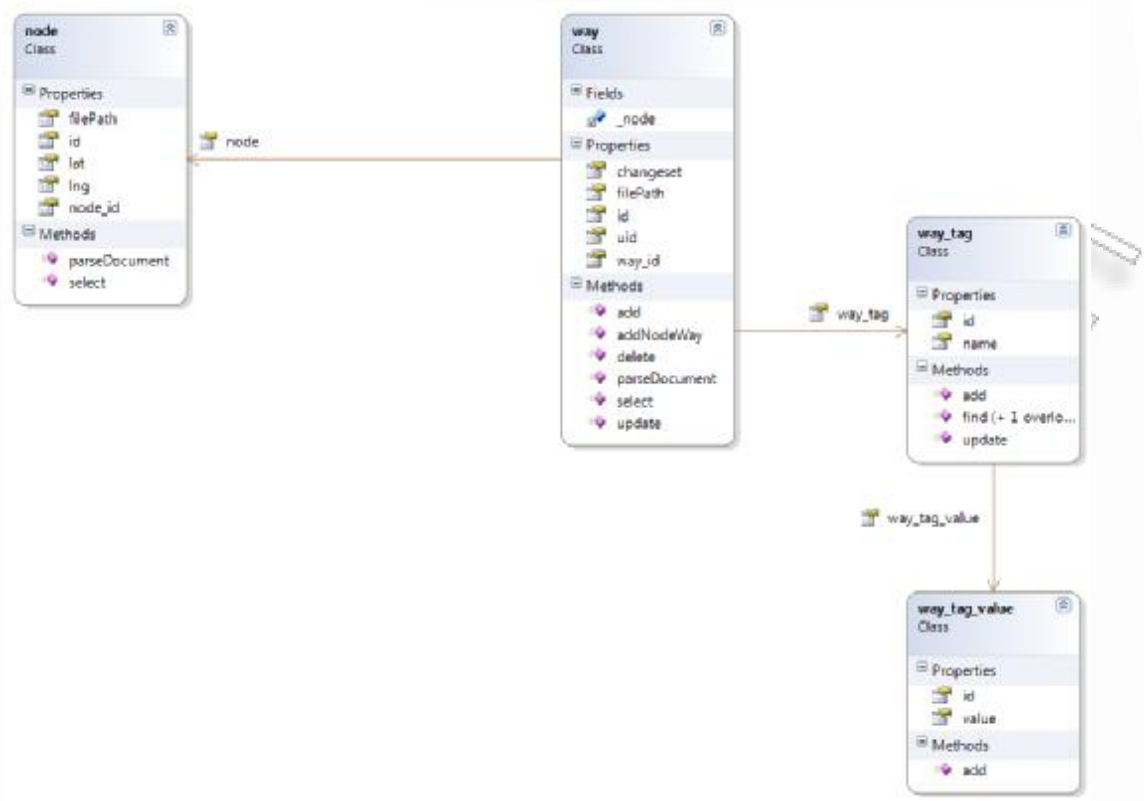
```
  <tag k="landuse" v="residential"/>
 </way>
```

Ο κόμβος έχει ένα μοναδικό id μέσα στο tag way έχουμε τους κόμβους nd που ανήκει ο δρόμος. Ο αλγόριθμος που δημιουργήθηκε για την μετάπτωση των δεδομένων είναι ό ακόλουθος.



Εικόνα 94 Αλγόριθμος προσθήκης δρόμου

Τα αντίστοιχα αντικείμενα και διαγράμματα που δημιουργήθηκαν είναι τα ακόλουθα :



Εικόνα 95 Κλάση way (δρόμος)

Η συνάρτηση parseDocument() ανοίγει το xml αρχείο και προσθέτει νέους δρόμους στην Β.Δ μέσα από τον αλγόριθμο που περιγράφτηκε πριν.

Η συνάρτηση add() προσθέτει ένα νέο δρόμο στην βάση δεδομένων. Η συνάρτηση αυτή εκτελεί ένα SQL query για την εισαγωγή του δρόμου. Στο To Sql Query είναι της μορφής

```
INSERT INTO [unipi_OpenStreetMap].[dbo].[OSM_way]
([way_id]
,[uid]
,[changeset],position)
VALUES
 (1
,'123'
,'123'
, geography::STGeomFromText('LINESTRING("32.12346 32,256788, 32.5665
32.25874")',4120)) SELECT @@IDENTITY AS 'newId' ");
```

Η συνάρτηση LINESTRING μπορεί να εισάγει μία γραμμή στον SQL Server το μόνο που χρειάζεται σε κάθε σημείο είναι το γεωμετρικό πλάτος και μήκος.

Τέλος πρέπει να δημιουργηθεί ο αντίστοιχος αλγόριθμος που θα αποθηκεύει τα πολύγονα. Ένα πολύγονο περιλαμβάνει μία περιοχή στην οποία μπορεί να διατηρούνται πολλές πληροφορίες.

Δημογραφικά στοιχεία

Η εφαρμογή θα καλύπτει γεωγραφικά τον νομή Αττικής. Πρέπει να πραγματοποιηθεί ανάλυση των δήμων που έχει η Αθήνα. Στον επόμενο πίνακα ακολουθεί μία λίστα με τους δήμους

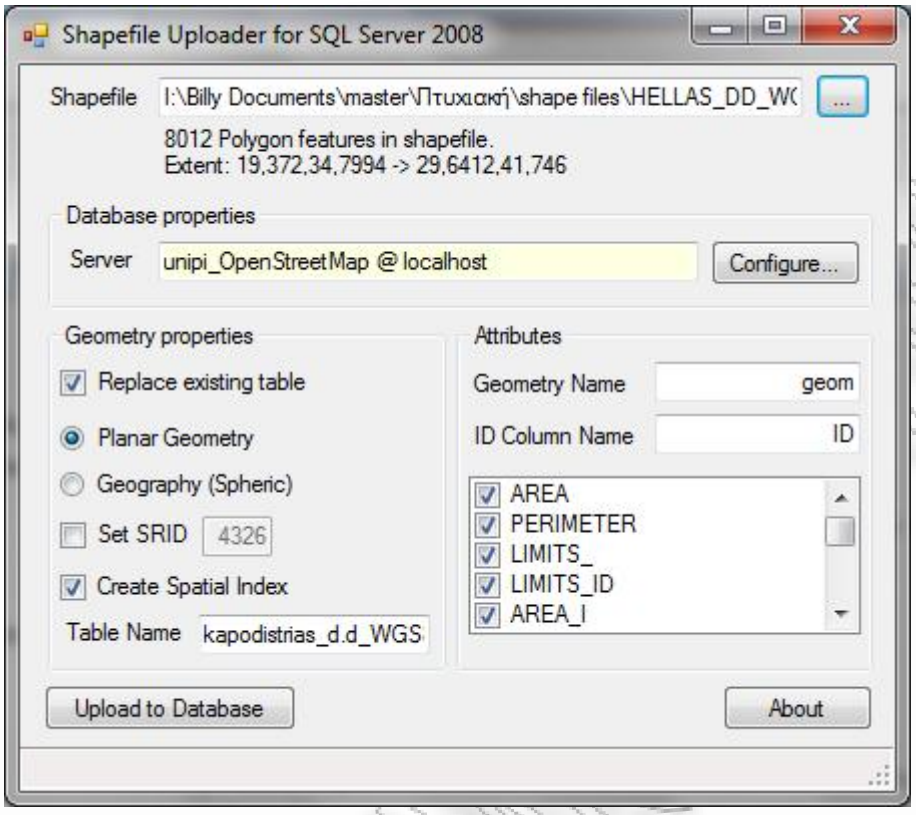
Δήμος – Κοινότητα	Κωδικός
Δήμος Λαυρεωτικής	5601
Δήμος Κερατέας	5351
Δήμος Παλαιάς Φωκαίας	5597
Κοινότητα Αναβύσσου	5570
Κοινότητα Σαρωνίδας	5546
Δήμος Καλυβίων	5329
Κοινότητα Κουβαρά	5346
Δήμος Μαρκοπούλου	5186
Δήμος Κορωπίου	5166
Δήμος Βάρης	5312
Δήμος Βουλιαγμένης	5418
Δήμος Βούλας	5318
Δήμος Γλυφάδας	5233
Δήμος Ελληνικού	5240
Δήμος Αργυρουπόλεως	5212
Δήμος Αλύμου	5201
Δήμος Ηλιουπόλεως	5156
Δήμος Βύρωνος	5120
Δήμος Καισαριανής	5107
Δήμος Παιανίας	5074
Δήμος Σπάτων Λούτσας	5046
Δήμος Αρτέμιδος	5056
Δήμος Αλίμου	5201
Δήμος Ηλιουπόλεως	5156
Δήμος Παιανίας	5074
Δήμος Αγίου Δημητρίου	5157
Δήμος Παλαιού Φαλήρου	5172
Δήμος Δάφνης	5164,5137
Δήμος Υμηττού	5136
Δήμος Νέας Σμύρνης	5140
Δήμος Παιανίας	5074
Δήμος Καλλιθέας	5127
Δήμος Αθηνών	5016
Δήμος Παπάγου	5060
Δήμος Χολαργού	5049
Δήμος Αγίας Παρασκευής	5031

Δήμος Γλυκών νερών	5044
Δήμος Παλλήνης	4994
Δήμος Πικερμίου	4989
Δήμος Ραφήνας	4995
Δήμος Ζωγράφου	5085
Κοινότητα Ανθούσης	4996
Δήμος Γέρακα	4984
Δήμος Νέου Ψυχικού	5045
Δήμος Ψυχικού	5035
Δήμος Γαλατσίου	5011
Δήμος Φιλοθέη	5013
Δήμος Βριλησσίων	4975
Δήμος Γέρακα	4984
Δήμος Ραφήνα	4995
Δήμος Νέα Μάκρη	4893
Κοινότητα Πεντέλης	4930
Κοινότητα Νέας Πεντέλης	4928
Δήμος Μελισσίων	4946
Δήμος Αμαρουσίου	4948
Δήμος Διονύσου	4884
Δήμος Κηφισιάς	4892
Δήμος Πεύκη	4940
Δήμος Ηρακλείου	4953
Δήμος Νέας Ιωνίας	4978
Δήμος Μεταμόρφωσης	4926
Δήμος Νέας Φιλαδέλφειας	4966
Δήμος Διονύσου	4884
Δήμος Αχαρνών	4705
Δήμος Ζεφυρίου	4923
Δήμος Καματερού	4951
Δήμος Νέας Χαλκηδόνας	5017
Δήμος Ηλίου	4954
Δήμος Πετρουπόλεως	4962
Δήμος Περιστερίου	5010
Δήμος Αιγάλεω	5048
Δήμος Αγίας Βαρβάρας	5064
Δήμος Κορυδαλλού	5065
Δήμος Αγίων Αναργύρων	4976
Δήμος Χαϊδαρίου	5005
Δήμος Νέας Χαλκηδόνας	5017
Δήμος Περάματος	5073

Δήμος Κερατσινίου	5071
Δήμος Δραπετσώνας	5051
Δήμος Πειραιώς	5121
Δήμος Μοσχάτου	5124
Δήμος Ταύρου	5094
Δήμος Άγιου Ιωάννου Ρέντη	5095
Δήμος Νίκαιας	5069
Δήμος Δραπετσώνας	5150
Δήμος Ασπροπύργου	4782
Δήμος Άνω Λιοσίων	4818
Δήμος Φυλής	4718
Κοινότητα Θρακομακεδόνων	4829
Δήμος Νέας Ερυθραίας	4868
Κοινότητα Εκάλης	4864
Κοινότητα Διονύσου	4884
Κοινότητα Ροδοπόλεως	4855
Δήμος Μαραθώνας	4722
Κοινότητα Σταμάτας	4804
Κοινότητα Δροσιάς	4862
Κοινότητα Κρυονερίου	4816
Δήμος Άγιος Στέφανος	4809
Δήμος Καπανδρίτι	4654
Κοινότητα Αφήνων	4709

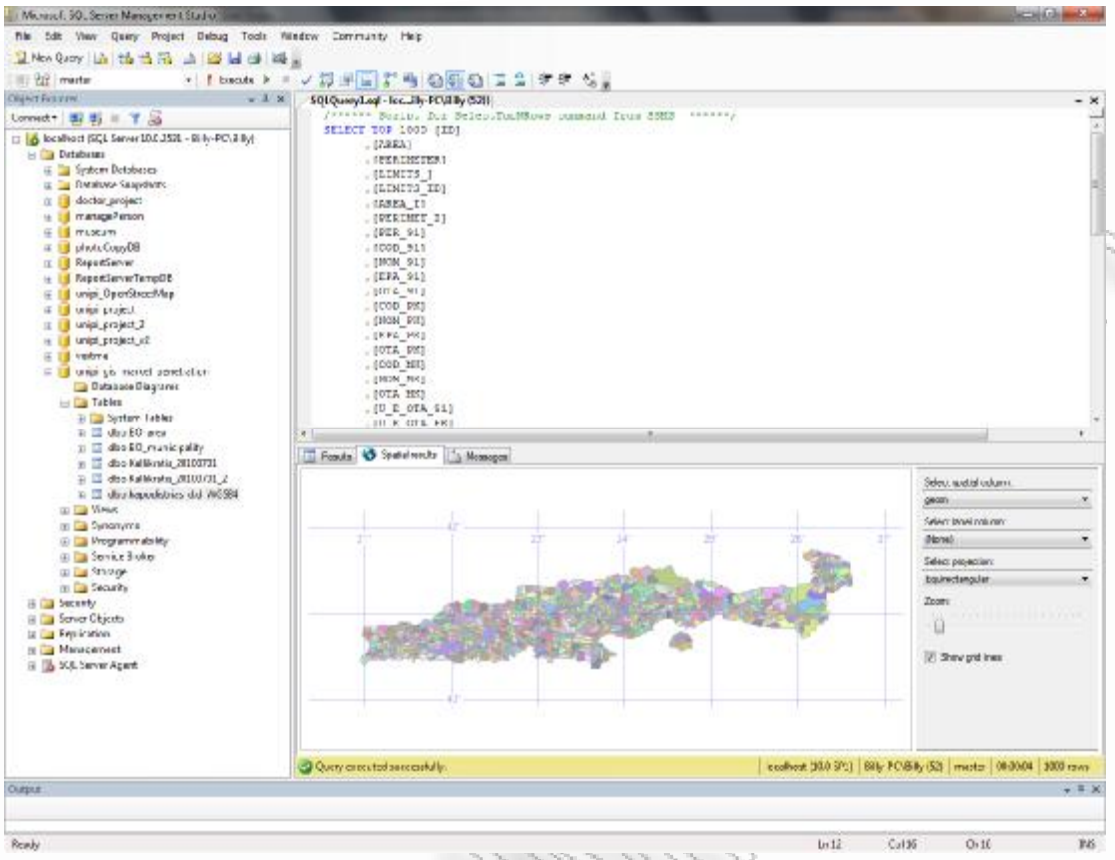
Μεταφορά δεδομένων shape file **στον** SQL Server 2008

Για την μεταφορά δεδομένων χρησιμοποιήθηκε μία εφαρμογή με που βρέθηκε μέσα από το codeplex.com και το όνομά της είναι «Shape2SQL». Με την χρήση του προγράμματος δημιουργήθηκε ο πίνακας «kapodistrias_d.d_WGS84» .



Εικόνα 96 Shape2SQL.exe

Το δεύτερο βήμα είναι η εισαγωγή δεδομένων στον πίνακα των δήμων. Για την μεταφορά δεδομένων από τον ένα πίνακα στον άλλον δημιουργήθηκε ένα μικρό πρόγραμμα που έχει το όνομα «convert_db_spatia». Το συγκεκριμένο πρόγραμμα διαβάζει τον πίνακα που δημιουργήθηκε από την μεταφορά δεδομένων του shapefile και εκτελεί ένα update query στον πίνακα των δήμων με βάση τον μοναδικό κωδικό. Με την εκτέλεση του προγράμματος το αποτέλεσμα του εμφανίζεται στην επόμενη εικόνα.

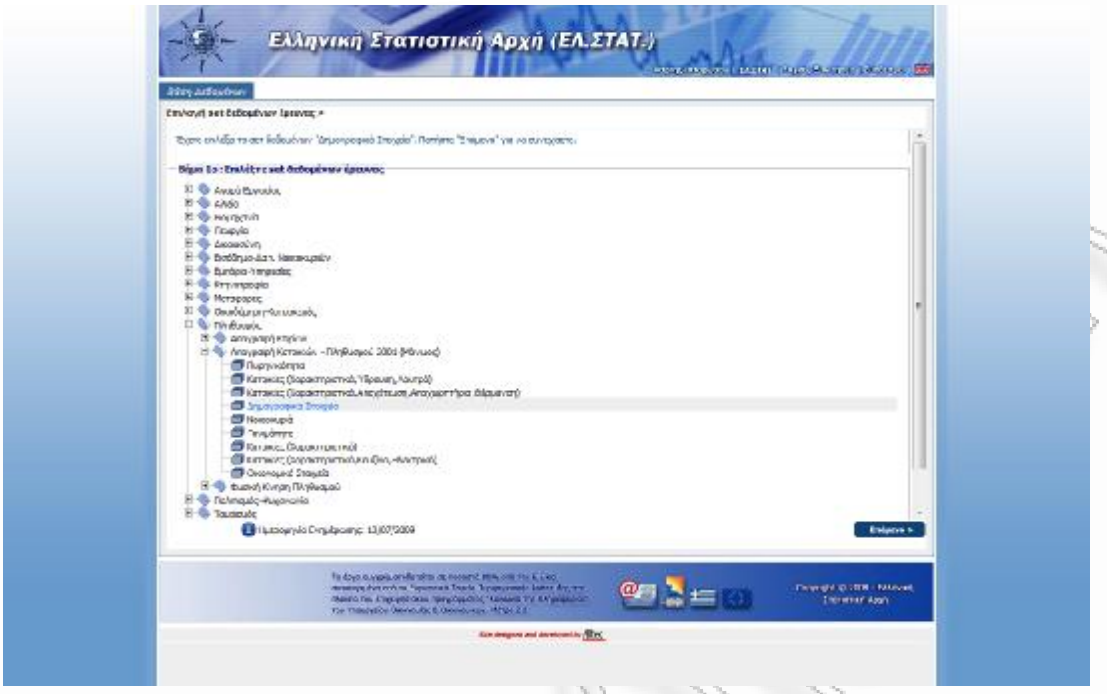


Εικόνα 97 Δήμοι Νομού Αττικής

Το επόμενο βήμα είναι η εισαγωγή δεδομένων (δήμων –κοινοτήτων) στο πίνακα *BO_municipality*. Για την εισαγωγή δημιουργήθηκε η παρακάτω store procedure.

Εξαγωγή Δημογραφικών στοιχείων

Η συλλογή δημογραφικών στοιχείων πραγματοποιήθηκε μέσω της στατιστικής υπηρεσίας. (Το link της υπηρεσίας είναι το ακόλουθο www.statistics.gr).



Εικόνα 98 Στατιστική υπηρεσία

Μέσα από ένα Wizard που έχει το site πραγματοποιήθηκε εξαγωγή των δεδομένων σε μορφή excel. Από την βάση δεδομένων έγιναν οι παρακάτω επιλογές Πληθυσμός → Απογραφή κατοικιών → Δημογραφικά στοιχεία. Το επόμενο βήμα είναι η επιλογή των χαρακτηριστικών που χρειαζόμαστε.

Χαρακτηριστικά που επιλέχθηκαν είναι τα ακόλουθα:

Πληθυσμός
Κατανομή ηλικιών
Οικογενειακή κατάσταση
Κύρια ασχολία
Επίπεδο εκπαίδευσης
Ηλικίες

Περιγραφή χαρακτηριστικών.

Πληθυσμός : Ο πληθυσμός είναι μία αριθμητή τιμή.

Κατανομή ηλικιών : Περιγράφει πόσοι άνθρωποι ανήκουν σε ένα εύρος ηλικίας. Τα δεδομένα ομαδοποιούνται με βάση το εύρος ηλικίας το οποίο καθορίζεται από τον επόμενο πίνακα.

Εύρος ηλικίας
0-4
5-9
10-14
15-19

20-24
25-29
30-34
35-39
40-44
45-49
50-54
55-59
60-64
65-69
70-74
75-79
80-84
85+

Κύρια ασχολία : Το συγκεκριμένο χαρακτηριστικό περιγράφει την κύρια ασχολία που έχουν οι πολίτες στους δήμους της Αθήνας. Στον επόμενο πίνακα περιγράφεται το λεξιλόγιο που προσφέρεται για το συγκεκριμένο χαρακτηριστικό.

Άτομα κάτω των 10 ετών
Εργαζόμενος (-ή)
Ζητούσε εργασία
Ζητούσε εργασία για πρώτη φορά
Μαθητής (-τρια) / Σπουδαστής (-τρια)
Συνταξιούχος
Εισοδηματίας
Οικιακά
Άλλη περίπτωση

Οικογενειακή κατάσταση: Το συγκεκριμένο χαρακτηριστικό περιγράφει την οικογενειακή κατάσταση που έχουν οι πολίτες σε ένα δήμο. Το λεξιλόγιο περιγράφεται στον επόμενο πίνακα.

Άγαμος
Έγγαμος
Διαζευγμένος
Σε διάσταση
Χήρος

Υπηκοότητα : Το χαρακτηριστικό περιγράφει την υπηκοότητα που έχουν οι πολίτες στον συγκεκριμένο δήμο και έχει δύο τιμές Ελληνική, Ξένη.

Annex A Weka – SimpleKMeans

Το WEKA είναι ένα ελεύθερο open source λογισμικό για Data-mining. Θα χρησιμοποιηθεί για την εξαγωγή αποτελεσμάτων. Όπως αναφέρθηκε και πριν πρέπει να δημιουργηθεί ένα αρχείο το οποίο θα περιέχει τα δεδομένα που θα επεξεργαστούν. Το αρχείο θα έχει την μορφή arff Η μορφοποίηση του θα είναι η ακόλουθη ◎

Παράδειγμα ARFF αρχείου

@relation marketing

@attribute instance_id ◎

@attribute municipality {athina, kallithea, marousi} ◎

@attribure population ◎

@attribute metro_stop {close, far} ◎

@attribute vodafone_store {close, far} ◎

@data

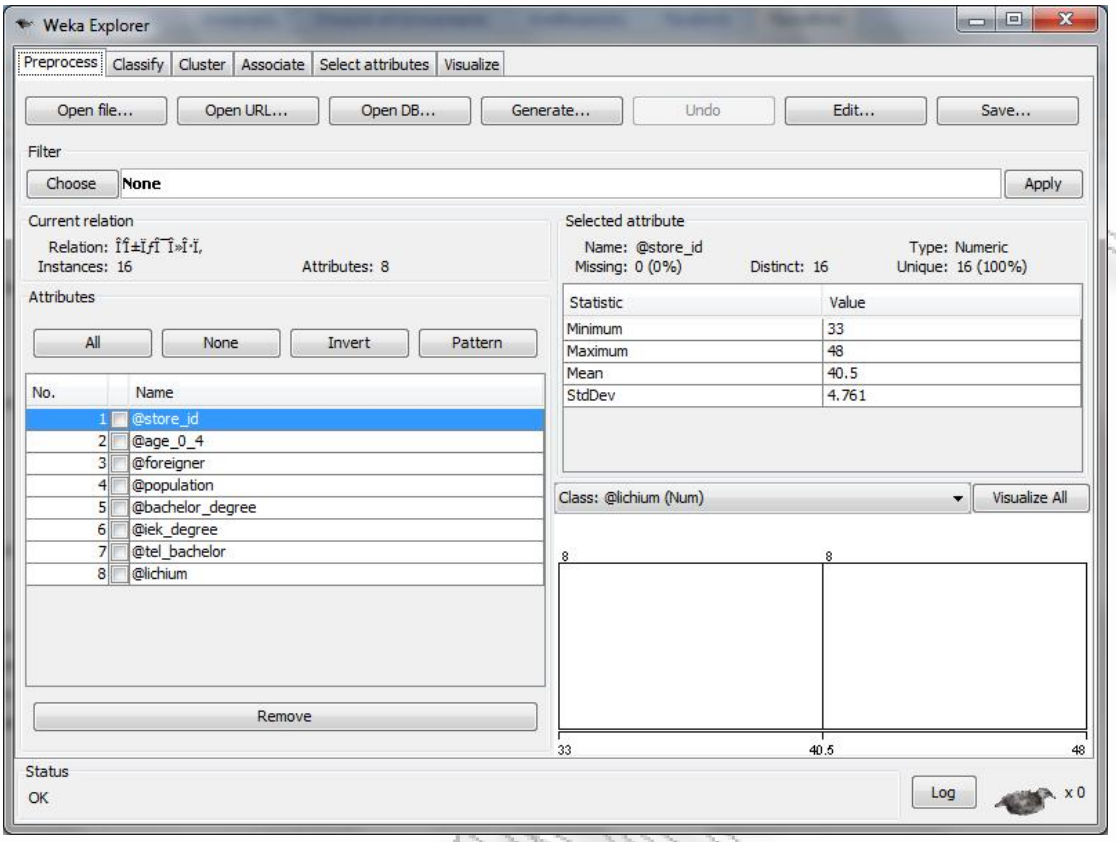
1,athina,300.000, close,far

2,marousi,10.000,far,close

Η δομή του αρχείου είναι η ακόλουθη.

Με την δήλωση @attribute δηλώνουμε στο WEKA ένα νέο χαρακτηριστικό, αν είναι οι ◎ τιμές που μπορεί να πάρει η ιδιότητα είναι αριθμός τότε δεν χρειάζεται να δηλωθεί το εύρος τιμών. Αν όμως οι τιμές που μπορεί είναι λέξεις τότε θα πρέπει να δηλωθούν με την ακόλουθη σύνταξη {.....}.

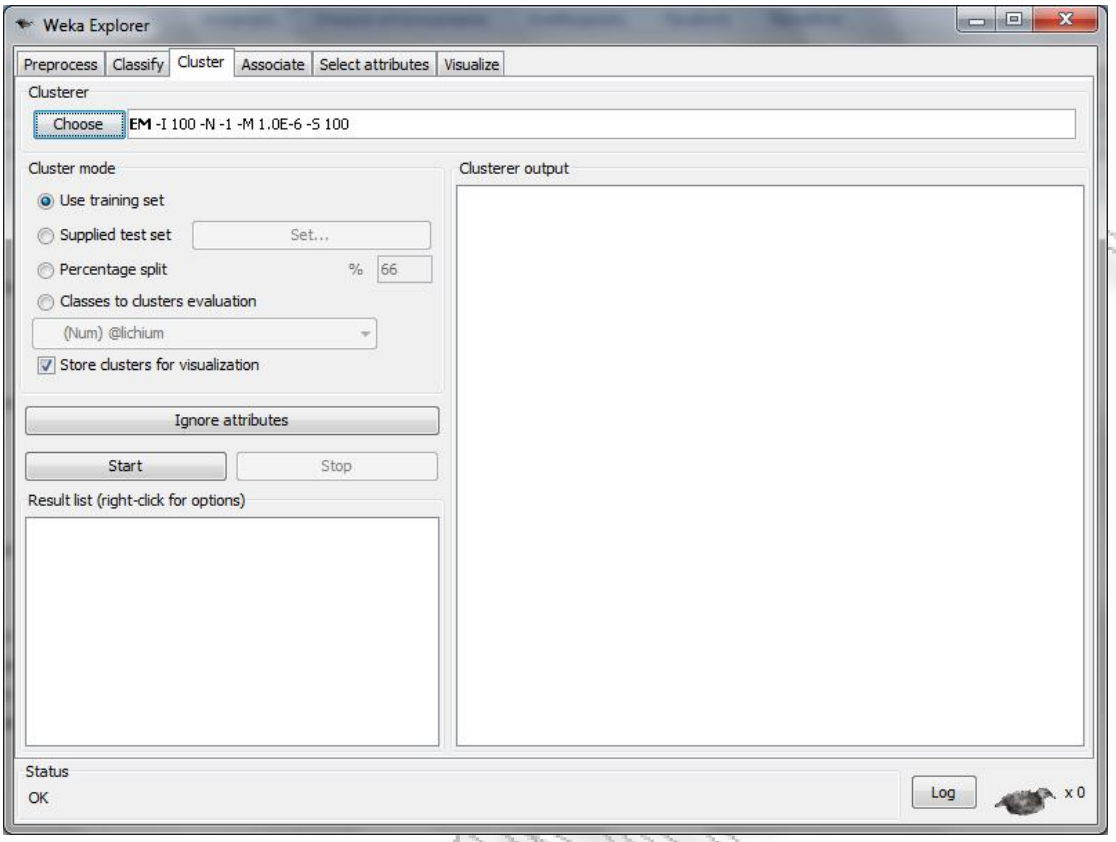
Το επόμενο βήμα είναι η δήλωση των στιγμιότυπων και των δεδομένων τους. Η δήλωση ξεκινά με την λέξη @data. Αν ανοιχτεί ένα αρχείο arff και υπάρχει στον υπολογιστή ◎ εγκατεστημένο το Weka θα εμφανιστεί η επόμενη οθόνη.



Εικόνα 99 Weka εισαγωγική οθόνη

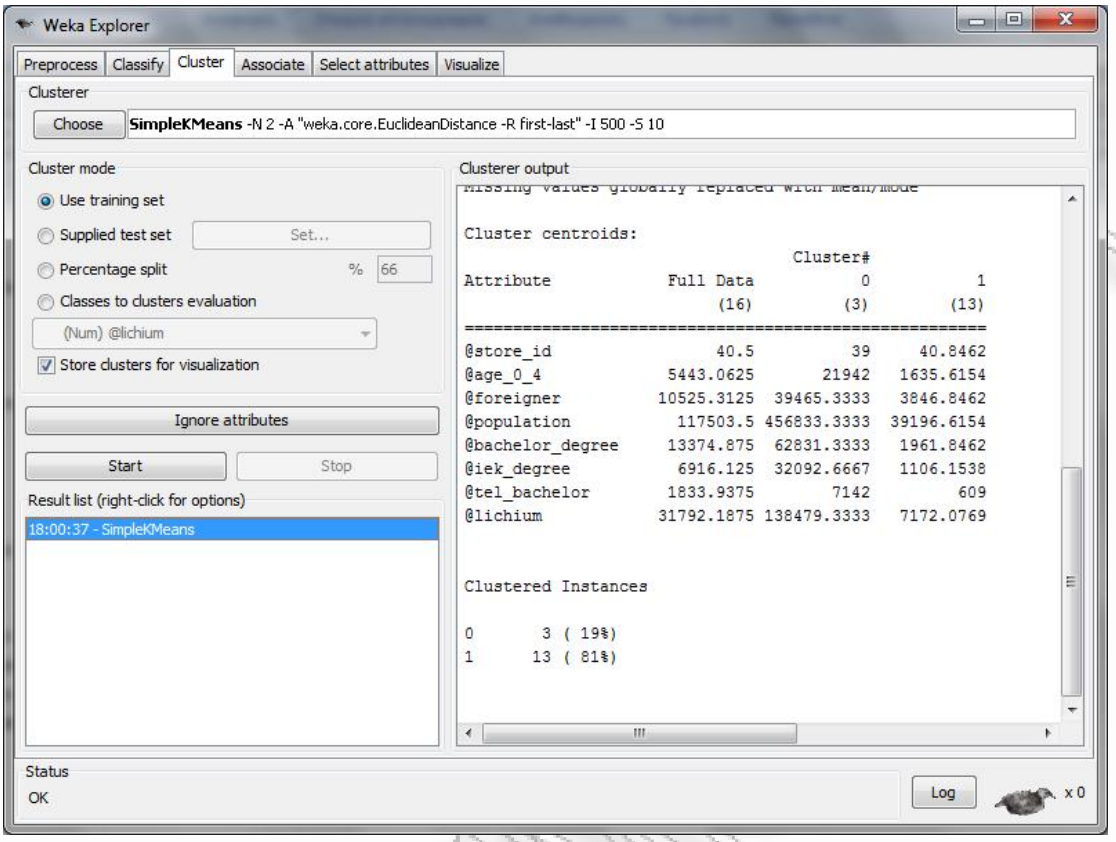
Στην εικόνα 99 εμφανίζονται οι ιδιότητες που έχει το αρχείο *store_id,age_0_4,foreigner, population,bachelor_degree,iek_degree,tei_bachelor,lichium.* Αν επιλεχτεί μία ιδιότητα τότε στον αριστερό πίνακα «Selected Attributes» εμφανίζονται χρήσιμα στοιχεία όπως το όνομα η ελάχιστη τιμή, η μέγιστη τιμή. Όλες αυτές οι πληροφορίες είναι πολύ χρήσιμες για την επεξεργασία των δεδομένων.

Αν επιλέξουμε το δεύτερο tab «Cluster» τότε εμφανίζεται η επόμενη εικόνα.



Εικόνα 100 Weka Cluster

Εδώ ο χρήστης μπορεί να επιλέξει το αλγόριθμο για την ανάλυση δεδομένων. Για παράδειγμα αν επιλέξουμε τον SimpleKMeans και πατήσουμε το κουμπί start θα εμφανιστεί η επόμενη εικόνα.



Εικόνα 101 Weka SimpleKMeans

Στο παράθυρο Cluster Output εμφανίζει τις ομάδες που δημιουργήθηκαν και τα Centroids που έχουν δημιουργηθεί για κάθε ιδιότητα.

Συμπεράσματα

Με την ανάπτυξη του συγκεκριμένου πληροφοριακού συστήματος συνδυάστηκε η χρήση γεωγραφικών συστημάτων με αλγόριθμους εξόρυξης δεδομένων. Ο επαγγελματίας που θέλει να ανοίξει ένα νέο κατάστημα μπορεί να πληροφορηθεί για την γεωγραφική θέση που επέλεξε αν ανήκει σε κάποια από τις ομάδες των πετυχημένων καταστημάτων της ίδιας κατηγορίας.

Χρησιμοποιώντας την εφαρμογή αναλύονται τα δεδομένα μέσα από τον αλγόριθμο SimpleKMeans και αναπαριστάται το αποτέλεσμα της ανάλυσης μέσα από τον χάρτη. Με αυτό τον τρόπο ο τελικός χρήστης μπορεί να έχει σωστή αντίληψη για τις ομάδες που δημιουργήθηκαν από την εκτέλεση του αλγόριθμου. Αυτό είναι πάρα πολύ σημαντικό γιατί τα δημογραφικά δεδομένα που έχει η εφαρμογή δεν έχουν άμεση σχέση με γεωγραφική πληροφορία των καταστημάτων. Μέσω του πληροφοριακού συστήματος που δημιουργήθηκε στα πλαίσια της μεταπτυχιακής διατριβής συνδυάζονται όλες αυτές οι πληροφορίες, αναλύονται και τέλος πραγματοποιείται η αναπαράσταση του αποτελέσματος ώστε να γίνει αντιληπτή η λύση στο πρόβλημα.

Πίνακας εικόνων

Βιβλιογραφία

Armstrong, D. (n.d.). *.NET Application Architecture: the Data Access Layer*. Ανάκτηση από simple - talk: http://www.simple-talk.com/dotnet/.net-framework/.net-application-architecture-the-data-access-layer/

Control, G. U. (n.d.). *GoogleMaps User Control*. Ανάκτηση από http://en.googlemaps.subgurim.net/

Microsoft (n.d.). *The Workflow Way: Understanding Windows Workflow Foundation*. Ανάκτηση από http://msdn.microsoft.com/library/dd851337.aspx

wiki. (n.d.). *Software architecture*. Ανάκτηση από wikipedia: http://en.wikipedia.org/wiki/Software_architecture

Wikipedia. (n.d.). *Interpolation*. Ανάκτηση από http://en.wikipedia.org/wiki/Interpolation

Wikipedia. (n.d.). *Software architecture*. Ανάκτηση από http://en.wikipedia.org/wiki/Software_architecture

Workflow, T. C. (n.d.). *Tutorial: Create a Sequential Workflow*. Ανάκτηση από http://msdn.microsoft.com/en-us/library/ms734794(VS.90).aspx

(2009). Beginning Spatial with SQL Server 2008 . Στο A. AITCHISON, *Beginning Spatial with SQL Server 2008* .

architecture, M. (n.d.). Ανάκτηση 10 21, 2010, από Wikipedia : http://en.wikipedia.org/wiki/Multitier_architecture

Boston Geographic Information Systems. (n.d.). *Σύγκριση Β.Δ. για διαχείρηση γεωγραφικής πληροφορίας*. Ανάκτηση Μάρτιος 2010, από Τοποθεσία Web Boston Geographic Information Systems:

http://www.bostongis.com/?content_name=sql server2008_postgis_mysql_compare#1◎78

Control, G. U. (n.d.). *GoogleMaps User Control*. Ανάκτηση από http://en.googlemaps.subgurim.net/◎

Map, O. S. (n.d.). *Open Street Map*. Ανάκτηση από http://www.openstreetmap.org/◎

wiki. (n.d.). *Κατάλογος δήμων του σχεδίου Καλλικράτης*. Ανάκτηση 07 31, 2010, από http://el.wikipedia.org/wiki/%CE%9A%CE%B1%CF%84%CE%AC%CE%BB%CE%BF%CE%B3%CE%BF%CF%82_%CE%B4%CE%AE%CE%BC%CF%89%CE%BD_%CF%84%CE%BF%CF%85_%CF%83%CF%87%CE%B5%CE%B4%CE%AF%CE%BF%CF%85_%CE%9A%CE%B1%CE%BB%CE%BB%CE%B9%CE%BA%CF%81%CE%AC%CF%84%CE%B7%CF%82#.CE.94.CE.A◎

Αρχή, Ε. Σ. (n.d.). Ανάκτηση 06 27, 2010, από http://www.statistics.gr/◎