



## Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

### Μεταπτυχιακή Διατριβή

<b>Τίτλος Διατριβής</b>	<i>Διαδικτυακή πλατφόρμα πολυεπίπεδης αρχιτεκτονικής για την πραγματοποίηση δυναμικών αναζητήσεων σε γεωπληροφοριακό σύστημα</i>  <i>A multidimensional web-application for dynamic querying into geographical information system</i>
<b>Όνοματεπώνυμο Φοιτητή</b>	<b>Κουγιουμτζής Δημήτρης</b>
<b>Πατρώνυμο</b>	<b>Αντώνιος</b>
<b>Αριθμός Μητρώου</b>	<b>ΜΠΠΛ/10047</b>
<b>Επιβλέπων</b>	<b>Χρήστος Δουληγέρης, Καθηγητής</b>
<b>Συνεπιβλέπων ερευνητής</b>	<b>Δρ. Μενεκλής Βασίλης</b>

Ημερομηνία Παράδοσης **Μάρτιος 2013**

**Τριμελής Εξεταστική Επιτροπή**

(υπογραφή)

(υπογραφή)

(υπογραφή)

Χρήστος Δουληγέρης  
Καθηγητής

Δημήτριος Αποστόλου  
Επίκουρος καθηγητής

Παναγιώτης Κοτζανικολάου  
Λέκτορας

## Περίληψη

Πριν την εξέλιξη της πληροφορικής για την επεξεργασία γεωγραφικών δεδομένων κάναμε χρήση των χαρτών. Η εξέλιξη της πληροφορικής μας έδωσε τη δυνατότητα δημιουργίας ψηφιακών χαρτών. Η διαχείριση των γεωγραφικών δεδομένων απαιτεί τη χρήση γεωγραφικών βάσεων δεδομένων ώστε να καταστεί δυνατή η εκτέλεση χωρικών ερωτημάτων. Στο πλαίσιο αυτής της εξέλιξης αναπτύχθηκε και η παρούσα εφαρμογή η οποία αξιοποιεί και συνδυάζει τις τεχνολογίες του διαδικτύου, των ψηφιακών χαρτών και των γεωγραφικών βάσεων δεδομένων. Η εφαρμογή δίνει τη δυνατότητα στο χρήστη να αλληλεπιδρά με το χάρτη ώστε να βρει συγκεκριμένα σημεία ενδιαφέροντος σε συγκεκριμένη περιοχή. Ο πρώτος τρόπος αλληλεπίδρασης είναι αρχικά η δημιουργία μια περιοχής (πολυγώνου) που ορίζει ο χρήστης και στη συνέχεια η επιλογή ενός σημείου ενδιαφέροντος που επιθυμεί να βρει στην περιοχή αυτή. Το σύστημα επιστρέφει τα αποτελέσματα εάν υπάρχουν στην περιοχή που έχει ορίσει. Ο δεύτερος τρόπος είναι η εισαγωγή γραμμής (διαδρομής) και η εμφάνιση σημείων ενδιαφέροντος που βρίσκονται κοντά στη γραμμή.

## Abstract

Before the evolution of computer science for the management of geographic data we made use of maps. The evolution of computer science although has allowed us to create digital maps. The management of geographic data requires the use of geographic databases for execution of spatial query. In this master thesis we present the development of an application which offers the ability to the user to find a specific point of interest in a specific area. The first way of doing this is by giving the ability to the user to draw a polygon on a google map and to select a point of interest he wants to find in that area. The database then returns the results of the points of interest that the polygon contains if any. The second way is by just drawing a line (path) on the map and selecting a point of interest. The database then returns the points of interest that are close to that line.

## Ευχαριστίες

Με την ολοκλήρωση της μεταπτυχιακής μου διατριβής θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή Δουληγέρη Χρήστο που δέχτηκε την εποπτεία της εργασίας μου. Επίσης θα ήθελα να ευχαριστήσω τον Δρ. Μενεκλή Βασίλη για την υποστήριξη και συνεργασία τόσο στην ανάπτυξη της εφαρμογής όσο και στη συγγραφή του κειμένου. Τέλος θα ήθελα να ευχαριστήσω την οικογενειά μου.

## Περιεχόμενα

Περίληψη .....	3
Abstract .....	3
Ευχαριστίες .....	4
Πίνακας εικόνων .....	6
1. Εισαγωγή.....	8
2. Θεωρητικό Υπόβαθρο .....	11
2.1. Γεωγραφικά συστήματα πληροφοριών .....	11
2.2. Περιπτώσεις χρήσης .....	18
3. Τεχνολογίες .....	24
3.1. PostGIS .....	24
3.2. PostgreSQL .....	24
3.3. Codeigniter .....	26
3.4. Jquery .....	28
3.5. Ajax.....	28
3.6. Json.....	29
3.6.1. GeoJson .....	29
3.7. Google maps API V.3 .....	30
4. Αρχιτεκτονική.....	31
4.1. Υλοποίηση εφαρμογής.....	31
4.2. Σχεδίαση βάσης δεδομένων .....	33
4.3. Διαγράμματα UML.....	34
4.3.1. Διαγράμματα περιπτώσεων-use case diagrams .....	34
4.3.2. Διαγράμματα σειράς-ακολουθίας diagrams .....	36
4.3.3. Διαγράμματα δραστηριοτήτων-activity diagrams.....	38
4.3.4. Διαγράμματα καταστάσεων-state diagrams .....	40
4.3.5. Διαγράμματα τάξεων-class diagrams.....	41

4.3.6.	Διαγράμματα αντικειμένων-object diagrams .....	42
4.3.7.	Διαγράμματα συνεργασίας-collaboration diagrams .....	43
4.3.8.	Διαγράμματα εξαρτημάτων-component diagrams .....	45
4.3.9.	Διαγράμματα διανομής-deployment diagram .....	46
4.4.	Περιγραφή Λειτουργικότητας.....	47
5.	Συμπεράσματα.....	49
6.	Παράρτημα.....	51
7.	Βιβλιογραφία .....	66

## Πίνακας εικόνων

Εικόνα 1	Απεικόνιση των γεωμετρικών σχημάτων σε μορφή διανύσματος ( <a href="http://egsc.usgs.gov/isb/pubs/gis_poster/">http://egsc.usgs.gov/isb/pubs/gis_poster/</a> ) .....	14
Εικόνα 2	Απεικόνιση των γεωμετρικών σχημάτων σε ψηφιδωτή μορφή ( <a href="http://egsc.usgs.gov/isb/pubs/gis_poster/">http://egsc.usgs.gov/isb/pubs/gis_poster/</a> ) .....	15
Εικόνα 3	Σύνδεση διανύσματος με βάση δεδομένων (Buckley,1997,p.9) .....	16
Εικόνα 4	Πρότυπο M-V-C .....	26
Εικόνα 5	Κώδικας μοντέλου .....	27
Εικόνα 6	Κώδικας ελεγκτή .....	27
Εικόνα 7	Κώδικας όψης.....	27
Εικόνα 8	Λειτουργία ajax .....	28
Εικόνα 9	Παράδειγμα json .....	29
Εικόνα 10	Αντικείμενο geojson .....	29
Εικόνα 11	Κώδικας google api –hello world .....	30
Εικόνα 12	Διάγραμμα υλοποίησης .....	32
Εικόνα 13	Εννοιολογικό(E-R) διάγραμμα εφαρμογής .....	33
Εικόνα 14	Λογικό διάγραμμα της βάσης δεδομένων .....	33
Εικόνα 15	Διάγραμμα περίπτωσης χρήσης-use case diagrams.....	35
Εικόνα 16	Διάγραμμα σειράς land.....	36
Εικόνα 17	Διάγραμμα σειράς query .....	37
Εικόνα 18	Διάγραμμα σειράς upolygon.....	37
Εικόνα 19	Διάγραμμα σειράς uline.....	38
Εικόνα 20	Διάγραμμα δραστηριοτήτων .....	39
Εικόνα 21	Διάγραμμα καταστάσεων .....	40
Εικόνα 22	Διάγραμμα τάξεων-class diagram .....	41

Εικόνα 23 Αντικείμενο land.....	42
Εικόνα 24 Αντικείμενο roi1 .....	42
Εικόνα 25 Αντικείμενο upolygon.....	42
Εικόνα 26 Αντικείμενο query .....	43
Εικόνα 27 Αντικείμενο uline.....	43
Εικόνα 28 Διάγραμμα συνεργασίας land.....	43
Εικόνα 29 Διάγραμμα συνεργασίας query .....	44
Εικόνα 30 Διάγραμμα συνεργασίας upolygon.....	44
Εικόνα 31 Διάγραμμα συνεργασίας uline.....	45
Εικόνα 32 Διάγραμμα εξαρτημάτων .....	46
Εικόνα 33 Διάγραμμα Διανομής .....	46
Εικόνα 34 Λειτουργία πολυγώνου .....	47
Εικόνα 35 Λειτουργία διαδρομής .....	48
Εικόνα 36 Κώδικας geo_model.php.....	52
Εικόνα 37 Κώδικας request.php .....	53
Εικόνα 38 Κώδικας view.php .....	54
Εικόνα 39 Κώδικας map.js.....	55
Εικόνα 40 Κώδικας polygon.js.....	58
Εικόνα 41 line.js .....	61
Εικόνα 42 Κώδικας map.php.....	62
Εικόνα 43 Κώδικας user.php.....	64
Εικόνα 44 Κώδικας line.php .....	65

## 1. Εισαγωγή

Η χρήση των υπολογιστών διευρύνεται σχεδόν σε κάθε τομέα. Τα σχετικά κόστη μειώνονται με ραγδαίους ρυθμούς λόγω της γρήγορης εξέλιξης των τεχνολογιών υλικού και λογισμικού. Ο υπολογιστής είναι μια συσκευή που εκτελεί υπολογισμούς και παίρνει λογικές αποφάσεις με ταχύτητα πολύ μεγαλύτερη από τον άνθρωπο. Η κύρια δουλειά του υπολογιστή με λίγα λόγια είναι η επεξεργασία δεδομένων. Όταν υπάρχει η ανάγκη για την επεξεργασία μεγάλου όγκου δεδομένων χρειαζόμαστε και ένα λογισμικό το οποίο θα μας εξασφαλίζει την οργάνωση των δεδομένων, την επεξεργασία τους και την ανάκτηση της καθώς και την προβολή των αποτελεσμάτων αυτής της επεξεργασίας. Αυτό το λογισμικό είναι τα συστήματα διαχείρισης βάσεων δεδομένων (ΣΔΒΔ). Στα συστήματα διαχείρισης βάσεων δεδομένων αποθηκεύεται και γίνεται επεξεργασία μεγάλου όγκου δεδομένων και αυτό διότι:

- Το ΣΔΒΔ εξασφαλίζει τη συνέπεια των δεδομένων και κάνει έλεγχο για πλεονάζοντα δεδομένα. Αν ένα δεδομένο είναι αποθηκευμένο μια φορά στη βάση δεδομένων τότε κάθε ενημέρωση που γίνεται στην τιμή του χρειάζεται να γίνει μόνο μια φορά και η νέα τιμή είναι άμεσα διαθέσιμη στους χρήστες.
- Μέσω των ΣΔΒΔ μπορεί να γίνει εξαγωγή περισσότερων πληροφοριών από το σύστημα. Αν για παράδειγμα μια επιχείρηση έχει τα τμήματα πωλήσεων και το τμήμα συμβολαίων τότε το τμήμα πωλήσεων μπορεί να προσπελάσει τα στοιχεία των εκμισθώσεων και αντίστοιχα το τμήμα συμβολαίων μπορεί να προσπελάσει τα στοιχεία των ιδιοκτητών.
- Επιτρέπουν τον διαμοιρασμό των δεδομένων διότι η βάση δεδομένων ανήκει σε όλο τον οργανισμό και μπορεί να προσπελαστεί από τους εξουσιοδοτημένους χρήστες. Με αυτόν τον τρόπο περισσότεροι χρήστες μοιράζονται περισσότερα δεδομένα.
- Παρέχουν βελτιωμένη ακεραιότητα δεδομένων που αναφέρεται στα αποθηκευμένα δεδομένα. Η ακεραιότητα συνήθως εκφράζεται με όρους περιορισμών. Οι περιορισμοί αυτοί μπορούν να εφαρμοστούν πάνω σε μοναδικές εγγραφές ή σε συσχετίσεις μεταξύ των εγγραφών. Για παράδειγμα ένας περιορισμός θα μπορούσε να δηλώνει ότι ο μισθός ενός υπαλλήλου δεν θα μπορούσε να είναι πάνω από 30.000€ ή ο κωδικός ενός υποκαταστήματος που περιέχεται στην εγγραφή του υπαλλήλου δεν θα μπορούσε να αντιστοιχεί σε ανύπαρκτο κατάστημα.
- Παρέχουν βελτιωμένη ασφάλεια με τη μορφή κωδικού πρόσβασης καθορίζονται τα δικαιώματα πρόσβασης στα δεδομένα του συστήματος και η προσπέλαση τους. Για παράδειγμα ο διευθυντής ενός υποκαταστήματος έχει πρόσβαση σε περισσότερα δεδομένα σε σχέση με κάποιον υπάλληλο του.
- Βελτιωμένη προσπέλαση και απόκριση των δεδομένων. Πολλά ΣΔΒΔ παρέχουν γλώσσες ερωτημάτων ή προγράμματα δημιουργίας αναφορών που επιτρέπουν στους χρήστες να κάνουν ερωτήσεις κατά περίπτωση χωρίς να απαιτείται από τον προγραμματιστή να γράψει κάποιο επιπλέον πρόγραμμα για αυτή τη δουλειά.
- Οικονομία στην κλιμάκωση. Η συγκέντρωση των λειτουργικών δεδομένων της επιχείρησης στη βάση δεδομένων και η δημιουργία ενός συνόλου από εφαρμογές οι οποίες χρησιμοποιούν τα δεδομένα αυτά έχει ως αποτέλεσμα τη μείωση του κόστους. Στην περίπτωση αυτή ο προϋπολογισμός που είναι καταναμημένος σε κάθε τμήμα για την ανάπτυξη και συντήρηση των συστημάτων που είναι βασισμένα σε αρχεία μπορεί να συνδυαστεί με πιθανό αποτέλεσμα τη μείωση του συνολικού κόστους που θα οδηγήσει στην οικονομία της κλιμάκωσης.



Βέβαια τα συστήματα διαχείρισης βάσεων δεδομένων (ΣΔΒΔ) έχουν και μειονεκτήματα όπως:

- Πολυπλοκότητα. Οι λειτουργικές δυνατότητες που περιμένει κάποιος να παρέχει το ΣΔΒΔ κάνει το ίδιο το ΣΔΒΔ ιδιαίτερα πολύπλοκο λογισμικό. Οι σχεδιαστές, οι προγραμματιστές, και οι τελικοί χρήστες πρέπει να κατανοούν αυτές τις λειτουργικές δυνατότητες προκειμένου να τις εκμεταλλευτούν στο μέγιστο βαθμό. Η αποτυχία κατανόησης του συστήματος μπορεί να οδηγήσει σε κακό σχεδιασμό, γεγονός που μπορεί να έχει σημαντικές συνέπειες για τον οργανισμό.
- Τα ΣΔΒΔ καταλαμβάνουν πολύ χώρο στον σκληρό δίσκο καθώς απαιτούν και μεγάλη ποσότητα σε μνήμη RAM έτσι ώστε να λειτουργήσουν αποτελεσματικά.
- Κατά κανόνα ένα σύστημα που βασίζεται σε αρχεία είναι γραμμένο για μια συγκεκριμένη εφαρμογή όπως η τιμολόγηση. Ως αποτέλεσμα η απόδοση είναι γενικά αρκετά καλή. Ωστόσο, το ΣΔΒΔ είναι γραμμένο έτσι ώστε να έχει ένα πιο γενικό χαρακτήρα και να εξυπηρετεί περισσότερες από μια εφαρμογές. Το αποτέλεσμα είναι ότι κάποιες από τις εφαρμογές ενδέχεται να μην λειτουργούν με την ίδια ταχύτητα.

Ανεξαρτήτως μειονεκτημάτων επειδή μας επιτρέπουν τη διαχείριση μεγάλου όγκου δεδομένων τα ΣΔΒΔ χρησιμοποιούνται σε πολλές επιστήμες. Μια από αυτές τις επιστήμες είναι και η γεωγραφία. Έτσι τα συστήματα διαχείρισης βάσεων δεδομένων που επεξεργάζονται και διαχειρίζονται γεωγραφικά δεδομένα έχουν πάρει την ονομασία Γεωγραφικά Πληροφοριακά Συστήματα (ΓΠΣ) (Buckley,1997,ρ.22). Πρόκειται για συστήματα που αποθηκεύουν τα δεδομένα των παραδοσιακών χαρτών σε ψηφιακή μορφή και με κατάλληλα εργαλεία παρουσιάζουν τα δεδομένα σε ψηφιακούς χάρτες.

Μπορούμε να εισάγουμε πληροφορίες σε ένα ΓΠΣ όχι μόνο σε τοπική βάση δεδομένων αλλά και σε απομακρυσμένη βάση δεδομένων. Αυτό γίνεται μέσω της τεχνολογίας του διαδικτύου. Το διαδίκτυο μας επιτρέπει να ενώσουμε τα υπολογιστικά συστήματα σε παγκόσμιο επίπεδο. Αυτό μας επιτρέπει να δημιουργήσουμε διαδικτυακές εφαρμογές όπου ο χρήστης μέσα από ένα πρόγραμμα περιηγητή θα συνδέεται απομακρυσμένα σε μια γεωγραφική βάση δεδομένων και θα εισάγει καινούργια δεδομένα ή θα εκτελεί ερωτήματα πάνω στη βάση δεδομένων και θα εξάγει είτε αναφορές είτε θα βλέπει τα ερωτήματα του οπτικοποιημένα πάνω σε ψηφιακούς χάρτες.

Αυτός ήταν και ο σκοπός εκπόνησης αυτής της εργασίας. Ο συνδυασμός των τεχνολογιών του διαδικτύου μαζί με τις τεχνολογίες των ΓΠΣ. Με τις τεχνολογίες και τα εργαλεία του διαδικτύου αναπτύσσονται εφαρμογές που είναι ανεξάρτητες πλατφόρμας διότι ο χρήστης μπορεί να τις προσπελάσει μέσα από έναν περιηγητή ανεξάρτητα από το λειτουργικό σύστημα που τρέχει ο υπολογιστής του (linux, mac os, windows). Αντίστοιχα, με τις τεχνολογίες των γεωγραφικών βάσεων δεδομένων μπορούν να αναπτυχθούν βάσεις δεδομένων με μεγάλο όγκο δεδομένων. Η εφαρμογή στην παρούσα εργασία είναι μια διαδικτυακή εφαρμογή όπου ο χρήστης ψάχνει να βρει δεδομένα (όπως για παράδειγμα σημεία ενδιαφέροντος) σε μια περιοχή την οποία ο ίδιος καθορίζει δυναμικά πάνω στον χάρτη. Στη συνέχεια γίνεται επεξεργασία των δεδομένων που έχει εισάγει ο χρήστης και εμφανίζονται τα αποτελέσματα πάνω στο χάρτη. Επίσης ο χρήστης καθορίζει μια διαδρομή και το σύστημα επιστρέφει τα σημεία ενδιαφέροντος που βρίσκονται κοντά στην διαδρομή.

Η δομή της μεταπτυχιακής διατριβής αποτελείται από τα παρακάτω τμήματα:

- Το κεφάλαιο 2 παρουσιάζει το θεωρητικό υπόβαθρο των γεωγραφικών πληροφοριακών συστημάτων. Γίνεται μια μελέτη για τα υποσυστήματα που αποτελούν ένα (ΓΠΣ). Αρχικά θα πρέπει να μελετηθεί τι είναι ένας χάρτης. Στην συνέχεια να γίνει μια ανάλυση για τον τύπο των γεωγραφικών δεδομένων που υπάρχουν καθώς και πως συνδέονται με μια τυπική βάση δεδομένων. Για να αποθηκευτούν τα γεωγραφικά δεδομένα χρειάζονται άλλους τύπους δεδομένων. Στην συνέχεια του κεφαλαίου αναφέρονται εφαρμογές που χρησιμοποιούν συστήματα (ΓΠΣ) και με ποιόν τρόπο αξιοποιούνται αυτά τα συστήματα στην ανθρώπινη δραστηριότητα.
- Το Κεφάλαιο 3 παρουσιάζει τις τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής καθώς και για τον λόγο που αναπτύχθηκε μια διαδικτυακή εφαρμογή.
- Το Κεφάλαιο 4 παρουσιάζει την αρχιτεκτονική της εφαρμογής. Γίνεται μια ανάλυση στην σχεδίαση της ΒΔ και παρουσιάζονται η γεωγραφικοί τύποι που αποθηκεύονται τα δεδομένα. Στην συνέχεια αναλύονται τα διαγράμματα UML της υλοποίησης της εφαρμογής και τέλος παρουσιάζεται η λειτουργικότητα με εικόνες σε επίπεδο εφαρμογής όπως την χειρίζεται ο τελικός χρήστης.
- Το Κεφάλαιο 5 παρουσιάζει τα συμπεράσματα που προκύπτουν από την ανάπτυξη της εφαρμογής.
- Το Κεφάλαιο 6 παρουσιάζει το κώδικα της εφαρμογής. Ο κώδικας της εφαρμογής αποτελείται από τον κώδικα που εκτελείται στο εξυπηρετητή καθώς και το κώδικα που εκτελείται σε επίπεδο πελάτη.

## 2. Θεωρητικό Υπόβαθρο

### 2.1. Γεωγραφικά συστήματα πληροφοριών

Η κύρια μέθοδος για την αναπαράσταση και την εύρεση μιας τοποθεσίας από γεωγραφικά αντικείμενα στη γη είναι ο χάρτης. Ο χάρτης είναι μια γραφική αναπαράσταση των αντικειμένων για το πόσο σχετικά κοντά βρίσκεται το ένα στο άλλο. Ο χάρτης αποτελείται από διαφορετικά γεωγραφικά αντικείμενα όπως σημεία, γραμμές, περιοχές. Κάθε αντικείμενο καθορίζεται από την τοποθεσία του στον χώρο (που καθορίζεται από ένα σύστημα συντεταγμένων) και από τα χαρακτηριστικά του (συνήθως τα αναφέρουμε ως γνωρίσματα). Με λίγα λόγια ο χάρτης είναι ένα μοντέλο του πραγματικού κόσμου.

Το υπόμνημα του χάρτη συνδέει τα γνωρίσματα με τα γεωγραφικά αντικείμενα. Τα γνωρίσματα, όπως για παράδειγμα το δάσος και η κατοικημένη περιοχή, τυπικά αναπαριστώνται στον χάρτη με διαφορετικό χρώμα και διαφορετική εικόνα. Για τα γεωγραφικά συστήματα πληροφοριών θα πρέπει να κωδικοποιηθούν σε μια μορφή που να είναι διαθέσιμα για επεξεργασία (Buckley, 1997, p.7). Αυτό προϋποθέτει τη φόρτωση των γνωρισμάτων σε ένα σύστημα διαχείρισης βάσεων δεδομένων και τη σύνδεση τους με τα γεωγραφικά αντικείμενα. Για τα γεωγραφικά δεδομένα που αναφέρονται συνήθως ως χωρικά δεδομένα τα χαρακτηριστικά τους αναφέρονται σε ένα σύστημα συντεταγμένων που μοντελοποιεί την τοποθεσία στην επιφάνεια της γης.

Ο χάρτης είναι η παραδοσιακή μέθοδος για την αποθήκευση και την αναπαράσταση γεωγραφικής πληροφορίας. Ο χάρτης αναπαριστά τα ακόλουθα 3 είδη πληροφορίας για τα γεωγραφικά αντικείμενα:

- Τοποθεσία του γεωγραφικού αντικειμένου.
- Γνωρίσματα του γεωγραφικού αντικειμένου.
- Συσχετισμούς μεταξύ των γεωγραφικών αντικειμένων.

Η γεωγραφία συνήθως περιγράφεται ως το αντικείμενο μελέτης που καθορίζει **Γιατί αυτό είναι εκεί**. Η περιγραφή είναι ακριβής αν υπολογίσουμε τα 3 είδη πληροφορίας που αναπαριστώνται από ένα παραδοσιακό χάρτη.

- Η τοποθεσία του γεωγραφικού αντικειμένου που καθορίζεται από ένα σύστημα συντεταγμένων που αναπαριστά την επιφάνεια της γης. Αυτό μας καθορίζει το **πού** βρίσκεται το γεωγραφικό αντικείμενο.
- Τα χαρακτηριστικά που περιγράφουν το γεωγραφικό αντικείμενο είναι το **τι** είναι το γεωγραφικό αντικείμενο.
- Η συσχέτιση του γεωγραφικού αντικειμένου με τα άλλα καθορίζεται από την τοποθεσία και τα γνωρίσματα όλων των χαρακτηριστικών, για παράδειγμα τους δρόμους που συνδέουν τις πόλεις. Οι σχέσεις περιγράφουν το **γιατί** του χαρακτηριστικού.

Ένα γεωγραφικό πληροφοριακό σύστημα κάνει διαχωρισμό μεταξύ των χωρικών δεδομένων και των χαρακτηριστικών των γεωγραφικών αντικειμένων.

Τα γεωγραφικά αντικείμενα στην επιφάνεια της γης μπορούν να καθοριστούν ως 3 είδη τύπων. Αυτά είναι *σημεία, γραμμές, πολύγωνα*.

- *Σημείο* είναι το χαρακτηριστικό που σχετίζεται με μια μοναδική τοποθεσία στην επιφάνεια της γης. Αυτό μπορεί να είναι για παράδειγμα ένα κτήριο.
- *Γραμμή* είναι το χαρακτηριστικό που σχετίζεται με μια γραμμή από γεωγραφικές συντεταγμένες. Αυτό μπορεί να είναι για παράδειγμα ένα ποτάμι, ένας δρόμος.
- *Επιφάνεια* είναι το χαρακτηριστικό που σχετίζεται από μια κλειστή γραμμή από γεωγραφικές συντεταγμένες συνήθως αναφέρεται ως *πολύγωνο*. Μπορούμε να χαρακτηρίσουμε ως πολύγωνο τα σύνορα ενός δήμου.

Με αυτόν τον τρόπο μπορούμε να κάνουμε αναπαράσταση των γεωγραφικών φαινομένων ως σημείο, ως γραμμή, η ως πολύγωνο.

Συνολικά, ένα γεωγραφικό πληροφοριακό σύστημα πρέπει να το μελετάμε σαν μια τεχνολογία και όχι σαν ένα απλό υπολογιστικό σύστημα (Buckley,1997,p.22).

Συνήθως καθορίζεται ότι γεωγραφικό πληροφοριακό σύστημα είναι ένα ολοκληρωμένο υπολογιστικό σύστημα. Ωστόσο ένα γεωγραφικό πληροφοριακό σύστημα μπορεί να δημιουργηθεί από μια ποικιλία λογισμικού και υλικού. Ο σημαντικός παράγοντας είναι οι ενοποίηση αυτών των εργαλείων για ένα αποτελεσματικό σύστημα επεξεργασίας γεωγραφικών δεδομένων.

Η ικανότητα να ενσωματώσουμε γεωγραφική πληροφορία, να την αναλύσουμε, να τη διαχειριστούμε και να απαντήσουμε σε χωρικά ερωτήματα είναι το διακριτό χαρακτηριστικό των γεωγραφικών πληροφοριακών συστημάτων (Buckley,1997,p.22).

Με βάση τους 2 ορισμούς μπορούμε να καταλήξουμε στο ότι :

Ένα γεωγραφικό πληροφοριακό σύστημα (ΓΠΣ) είναι το ενοποιημένο σύνολο από εργαλεία λογισμικού και υλικού που χρησιμοποιούνται για τη διαχείριση και τη χειραγώγηση των γεωγραφικών δεδομένων καθώς και της πληροφορίας που συνδέεται με αυτά.

Οι κατηγορίες των ΓΠΣ είναι οι παρακάτω:

- 4-διαστάσεων ΓΠΣ: Συνήθως τα ΓΠΣ αποθηκεύουν 2 διαστάσεις για τον χώρο και 1 διάσταση για τον χρόνο. Τα ΓΠΣ 4 διαστάσεων αποθηκεύουν 3 διαστάσεις για τον χώρο και 1 διάσταση για τον χρόνο.  
<[http://www.fig.net/pub/athens/papers/ts20/TS20\\_1\\_Celik\\_et\\_al.pdf](http://www.fig.net/pub/athens/papers/ts20/TS20_1_Celik_et_al.pdf)>[Προσπέλαση 28 Σεπτεμβρίου 2012 ]
- Πολυμεσικά ΓΠΣ: Μας επιτρέπουν τη διαχείριση εικόνας και βίντεο που μπορούν να αποθηκευτούν με τη μορφή αντικειμένων σε σχεσιακές βάσεις δεδομένων ή σε εξυπηρετητές με πολυμεσικό περιεχόμενο.  
<<http://geoinfo.sdsu.edu/mediaGIS/>>[Προσπέλαση 28 Σεπτεμβρίου 2012 ]
- Δικτυακά ΓΠΣ: Τα ΓΠΣ της συγκεκριμένης κατηγορίας έχουν μεγάλη ανάπτυξη λόγω της επέκτασης του διαδικτύου. Μια εφαρμογή αυτής της κατηγορίας ΓΠΣ είναι η παρακάτω:  
<<http://www.gmapgis.com/>>[Προσπέλαση 28 Σεπτεμβρίου 2012 ]
- ΓΠΣ εικονικής πραγματικότητας: Έχουν αναπτυχθεί εικονικά περιβάλλοντα που βασίζονται σε γεωγραφικά δεδομένα.  
<<http://proceedings.esri.com/library/userconf/proc01/professional/papers/pap105/p105.htm>>[Προσπέλαση 28 Σεπτεμβρίου 2012 ]

Ένα ΓΠΣ έχει 4 κύρια τμήματα. Αυτά είναι:

- Ένα σύστημα καταχώρισης των δεδομένων.
- Ένα σύστημα αποθήκευσης και ανάκτησης δεδομένων.
- Ένα σύστημα διαχείρισης και ανάλυσης δεδομένων.
- Ένα σύστημα οπτικοποίησης δεδομένων.

Το σύστημα καταχώρισης των δεδομένων επιτρέπει τον χρήστη να συλλέξει τα δεδομένα και να μετατρέψει τα χωρικά δεδομένα σε ψηφιακή μορφή. Τα δεδομένα είναι κυρίως από χάρτες, αναφορές, αεροφωτογραφίες.

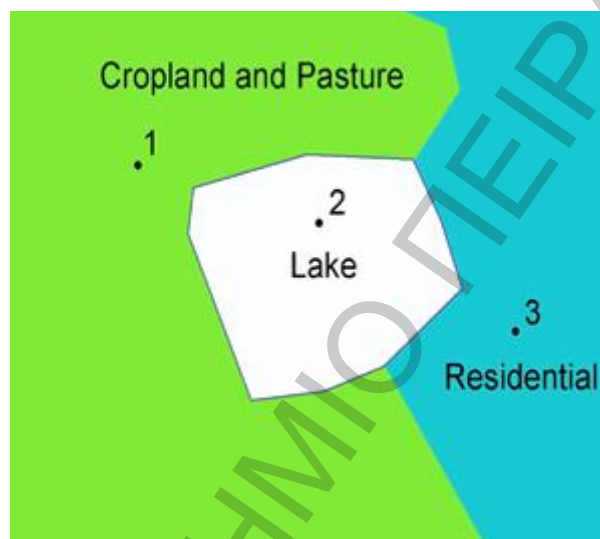
Το σύστημα αποθήκευσης και ανάκτησης των δεδομένων οργανώνει τα δεδομένα σε μια μορφή που επιτρέπει τη γρήγορη ανάκτηση τους. Συνήθως χρησιμοποιείται ένα σύστημα διαχείρισης βάσης δεδομένων για να έχει αποθηκευμένα τα δεδομένα των χαρακτηριστικών. Τα χωρικά δεδομένα συνήθως κωδικοποιούνται σε ένα κατάλληλο τύπο αρχείου.

Το σύστημα διαχείρισης και ανάλυσης των δεδομένων επιτρέπει στον χρήστη να εκτελεί λειτουργίες πάνω στα χωρικά δεδομένα καθώς και στα δεδομένα των χαρακτηριστικών έτσι ώστε να παράγει τις κατάλληλες αναφορές. Αυτό το σύστημα είναι η καρδιά του ΓΠΣ. Το σύστημα οπτικοποίησης των δεδομένων εξάγει τις γραφικές πληροφορίες και τις πληροφορίες των αναφορών.

Συνήθως τα χωρικά δεδομένα αναπαριστώνται σε μορφή χάρτη. Τρία είναι τα είδη των μοντέλων που έχουν εξελιχτεί για την ψηφιακή αποθήκευση των χωρικών δεδομένων. Αυτά είναι τα:

- Διανυσματικά μοντέλα (vectors)
- Ψηφιδωτά μοντέλα (rasters)
- Εικόνες.

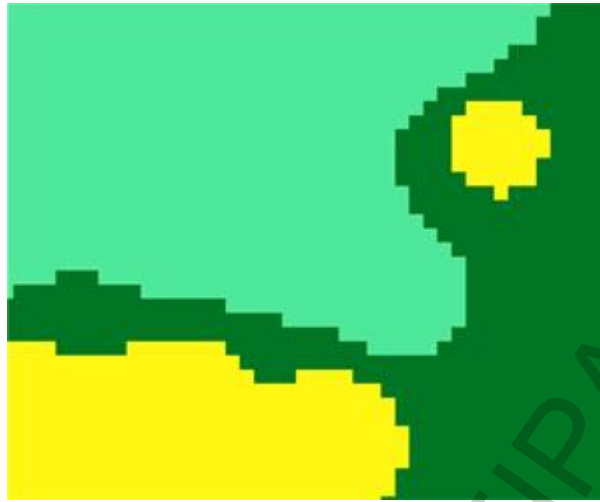
**Διανύσματα (Vector).** Όλα τα χωρικά δεδομένα μπορούν να αναπαρασταθούν με τρεις βασικούς τύπους γεωμετριών: σημεία, γραμμές, πολύγωνα. Έτσι για την απόδοση της θέσης μια πόλης σε ένα χάρτη μπορεί να χρησιμοποιηθεί ένα σημείο, για την αποτύπωση του οδικού δικτύου μια γραμμή αποτελούμενη από πολλές κορυφές και για την αποτύπωση μιας ιδιοκτησίας ένα πολύγωνο. Στην ουσία τα πάντα αναπαρίστανται από γραμμές. Το σημείο είναι μια γραμμή μηδενικού μήκους, ενώ το πολύγωνο είναι μια ακολουθία γραμμών με αρχή και τέλος την ίδια κορυφή. Η γεωμετρία που θα υιοθετηθεί για το συμβολισμό ενός αντικειμένου εξαρτάται από την κλίμακα απεικόνισης και το σκοπό της εφαρμογής που αναπτύσσεται. Έτσι π.χ. σε μια πολύ μεγάλη κλίμακα (1:1000) τα κτίσματα αποτυπώνονται ως πολύγωνα, ενώ σε μικρότερες κλίμακες (1:10.000) είναι ορθότερο να χρησιμοποιηθεί η γεωμετρία του σημείου. Τέλος, κάθε γεωμετρία συνδέεται με μια σχέση ένα-προς-ένα με μια εγγραφή σε ένα πίνακα περιγραφικών χαρακτηριστικών.



Εικόνα 1 Απεικόνιση των γεωμετρικών σχημάτων σε μορφή διανύσματος

Πηγή εικόνας 1 ([http://egsc.usgs.gov/isb/pubs/gis\\_poster/](http://egsc.usgs.gov/isb/pubs/gis_poster/)) [Προσπέλαση 2 Σεπτεμβρίου 2012]

**Ψηφιδωτά (Raster).** Η ψηφιδωτή δομή δεδομένων χρησιμοποιείται σε περιπτώσεις που το χωρικό φαινόμενο που αποτυπώνεται χαρακτηρίζεται ως συνεχής μεταβλητή (π.χ. το υψόμετρο του εδάφους, ή η κατανομή του θορύβου) ή σε περιπτώσεις που στο ΣΓΠ θέλουμε να ενσωματώσουμε μια δορυφορική εικόνα ή μια σαρωμένη αεροφωτογραφία. Οι ψηφιδωτές δομές δεδομένων έχουν περιορισμένες δυνατότητες σύνδεσης με περιγραφικά χαρακτηριστικά. Παράδειγμα πληροφορίας ψηφιδωτής δομής σε ένα ΓΠΣ είναι η εικόνα 2 ([http://egsc.usgs.gov/isb/pubs/gis\\_poster/](http://egsc.usgs.gov/isb/pubs/gis_poster/)) [Προσπέλαση 2 Σεπτεμβρίου 2012]



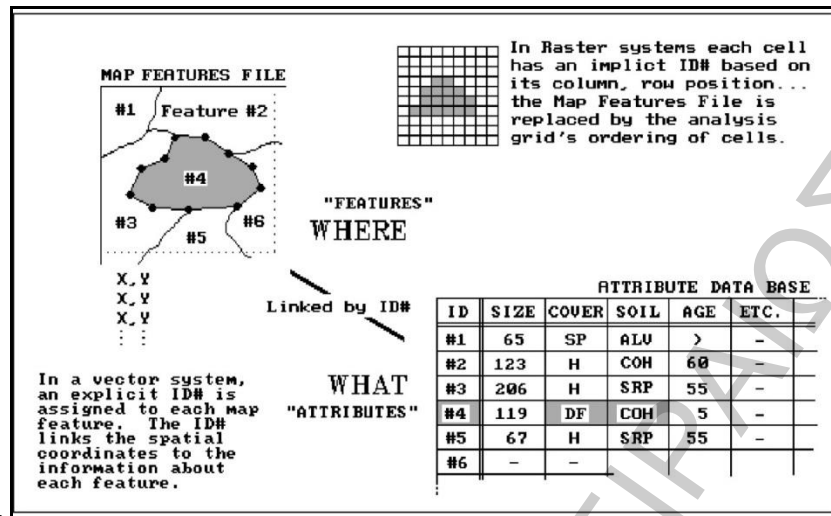
Εικόνα 2 Απεικόνιση των γεωμετρικών σχημάτων σε ψηφιδωτή μορφή

Τα χωρικά δεδομένα που έχουν τη μορφή εικόνας αποθηκεύονται στα πρότυπα που έχει καθορίσει η τεχνολογία της πληροφορικής όπως είναι τα αρχεία τύπου TIFF, GIF, PCX. Τα περισσότερα ΓΠΣ μπορούν να διαβάσουν αυτά τα είδη αρχείων και να οπτικοποιήσουν αυτή την πληροφορία.

Για να αποθηκεύσουμε τα χωρικά δεδομένα καθώς και τα δεδομένα των χαρακτηριστικών τους χρησιμοποιούμε τις σχεσιακές βάσεις δεδομένων. Η σχεσιακή βάση δεδομένων οργανώνει τα δεδομένα σε πίνακες. Κάθε πίνακας έχει ένα μοναδικό όνομα και είναι οργανωμένος σε γραμμές και στήλες. Κάθε στήλη μέσα στον πίνακα έχει επίσης ένα μοναδικό όνομα. Οι στήλες αποθηκεύουν τις τιμές για ένα συγκεκριμένο χαρακτηριστικό. Στα ΓΠΣ κάθε γραμμή αποθηκεύει ένα ξεχωριστό γεωγραφικό αντικείμενο. Το σχεσιακό μοντέλο αποθήκευσης είναι ευρέως διαδεδομένο για τη διαχείριση γεωγραφικών δεδομένων. Τα πλεονεκτήματα που το κάνουν διαδεδομένο είναι τα ακόλουθα:

- Απλότητα στην οργάνωση και μοντελοποίηση των δεδομένων.
- Ευελιξία, τα δεδομένα μπορούν να χειραγωγηθούν κατά περίπτωση με την ένωση πινάκων.
- Αποτελεσματικότητα στην αποθήκευση των δεδομένων με τον κατάλληλο σχεδιασμό πλεονάζοντα δεδομένα μπορούν να αφαιρεθούν.
- Τα ερωτήματα στη βάση δεδομένων δεν παίρνουν υπόψη τους την εσωτερική οργάνωση των δεδομένων.

Στην Εικόνα 3 φαίνεται ένα ψηφιδωτό σύστημα απεικόνισης, όπου κάθε κελί στη βάση δεδομένων έχει ένα μοναδικό κλειδί όπως και σε ένα διανυσματικό σύστημα κάθε κλειδί αντιστοιχεί στις γεωγραφικές συντεταγμένες του αντικειμένου.



Εικόνα 3 Σύνδεση διανύσματος με βάση δεδομένων (Buckley,1997,p.9)

Έχοντας αποθηκεύσει τα γεωγραφικά δεδομένα στη βάση το επόμενο στάδιο είναι η εκτέλεση χωρικών ερωτημάτων έτσι ώστε να γίνει η οπτικοποίηση των αποτελεσμάτων στον χάρτη της εκάστοτε εφαρμογής.

Συμπερασματικά, τα ΓΠΣ είναι συστήματα που αξιοποιούν γνώσεις από πολλά πεδία όπως αυτό της πληροφορικής, της γεωγραφίας, και των βάσεων δεδομένων. Λόγω του μεγάλου όγκου της πληροφορίας τα ΓΠΣ χρησιμοποιούνται σε εφαρμογές όπως:

- Περιβαλλοντική Διαχείριση (Environmental Management)
- Οργανισμούς Τοπικής Αυτοδιοίκησης
- Πολεοδομία και Χωροταξία
- Κατασκευές έργων μεγάλης κλίμακας (π.χ. οδοποιία κ.α.)
- Διαχείριση Δικτύων Κοινής Ωφελείας (AM/ FM)
- Κτηματολόγιο και Κτηματογραφήσεις
- Τοπογραφία, Γεωδαισία και Υδρογραφία
- Γεωλογία και Υδρογεωλογία
- Δίκτυα Μεταφορών και Επικοινωνιών
- Προγραμματισμό μεταφορών
- Προγραμματισμό δράσης έκτακτης ανάγκης
- Αυτόματη Πλοήγηση
- Διάθεση αποβλήτων
- Αρχαιολογία
- Γεωργία
- Δασονομία

Η τεχνολογία των ΓΠΣ μπορεί να χρησιμοποιηθεί για τις επιστημονικές έρευνες, τη διαχείριση των πόρων και τον προγραμματισμό της ανάπτυξης. Παραδείγματος χάριν, ένα ΓΠΣ μπορεί να επιτρέψει στους αρμόδιους το σχεδιασμό αντιμετώπισης μιας έκτακτης ανάγκης και τον υπολογισμό

*Διαδικτυακή πλατφόρμα πολυεπίπεδης αρχιτεκτονικής για την πραγματοποίηση δυναμικών αναζητήσεων σε γεωπληροφοριακό σύστημα*



των χρόνων απόκρισης σε μια έκτακτη ανάγκη όπως σε περίπτωση φυσικής καταστροφής, ή ακόμα μπορεί να χρησιμοποιηθεί για να βρει τους υγροτόπους που χρειάζονται προστασία από τη ρύπανση. Επίσης χρησιμοποιείται και στην επιλογή/εντοπισμό περιοχών που πληρούν κάποιες προϋποθέσεις ή έχουν κάποια συγκεκριμένα χαρακτηριστικά (π.χ. ποιές περιοχές καλύπτει μια επιχείρηση ύδρευσης: το ΓΠΣ επιλέγει τις περιοχές που δεν τις καλύπτει η εταιρία αυτή και αποκλείει τις περιοχές που καλύπτονται από αυτήν, ενώ μπορεί να αποκλείσει από την ανάλυση και περιοχές γύρω από τις οποίες υπάρχουν ακατάλληλα ρεύματα, μπορεί ακόμα να δείξει πηγές μόλυνσης και στοιχεία του εδάφους πάνω σε ένα χάρτη και τέλος να συνδυάσει αυτές τις πληροφορίες για να δώσει μια πλήρη εικόνα με αρκετές πληροφορίες για λήψη αποφάσεων). Ουσιαστικά ενσωματώνει παραδοσιακούς χαρτογράφους για την αυτοματοποιημένη παραγωγή των χαρτών. Ένα από τα πιο κοινά προϊόντα ενός ΓΠΣ είναι ένας χάρτης.

Ένα τέτοιο σύστημα έχει και εγγενή προβλήματα, τα οποία πρέπει να λύσουν οι χρήστες του προκειμένου να αποφευχθεί μια λανθασμένη ανάλυση. Μερικά από τα τυπικά προβλήματα που προκύπτουν μπορεί να αφορούν τα ακόλουθα:

- τη συλλογή ορθών και συγκεκριμένων χωρικών δεδομένων,
- την απόφαση ως προς τη μορφή των δεδομένων,
- την εισαγωγή των δεδομένων στο ΓΠΣ,
- τη διατήρηση της ποιότητας των δεδομένων,
- την ενσωμάτωση ενός ΓΠΣ στη διαδικασία λήψης αποφάσεων.

Υπάρχουν, επίσης, διάφοροι περιορισμοί, ή πιθανά προβλήματα, που μπορούν να υπονομεύσουν την ποιότητα οποιασδήποτε ανάλυσης με την χρήση ενός ΓΠΣ. Ενδεικτικά αναφέρονται:

- Ποιότητα δεδομένων.
- Διαθεσιμότητα δεδομένων.
- Κόστος του εξοπλισμού και του λογισμικού.
- Κόστος των δεδομένων.
- Επίπεδο κατάρτισης του χρήστη.

Προκειμένου η εφαρμογή ενός ΓΠΣ να στηρίξει τη λήψη αποφάσεων και να αποβούν χρήσιμα τα παραγόμενα από αυτό, είναι απαραίτητο να υλοποιηθεί με προσεκτικό τρόπο ώστε να διασφαλιστεί ότι διατηρείται η καλύτερη δυνατή ποιότητα δεδομένων σε όλο το φάσμα λήψης αποφάσεων. Ιδιαίτερη προσοχή συνιστάται στα ακόλουθα ζητήματα:

- Πολυπλοκότητα του αντικειμένου της χαρτογράφησης.
- Οργανωτικά ζητήματα, περιλαμβανομένων του προσωπικού και των αναγκών κατάρτισης.
- Ποιότητα δεδομένων.
- Κόστος.
- Πηγή δεδομένων.
- Χρονικές συγκεκριμένες εκτιμήσεις σχετικές με το παράκτιο περιβάλλον.

Επίσης, είναι αναγκαίο ο χρήστης να έχει κατανοήσει πλήρως τον τρόπο λειτουργίας του ΓΠΣ, αλλά και το πώς μπορούν να μεγιστοποιηθούν οι δυνατότητες του συστήματος και να ελαχιστοποιηθούν οι αδυναμίες του σε οποιαδήποτε δεδομένη μελέτη.

## 2.2. Περιπτώσεις χρήσης

Τα γεωγραφικά συστήματα παίζουν σημαντικό ρόλο στις ανθρώπινη δραστηριότητα και οι εφαρμογές τους σε ανάλυση γεωγραφικών δεδομένων είναι καθοριστικές. Οι περιπτώσεις χρήσης που εξετάζονται παρακάτω έχουν να κάνουν με τους εξής τομείς:

- Αγροτική ανάπτυξη.
- Τομείς συγκοινωνιών.
- Τομείς προστασίας περιβάλλοντος.
- Παρακολούθηση και καταγραφή σεισμών.
- Εύρεση σημείων ενδιαφέροντος.

Μια κατηγορία εφαρμογών των γεωγραφικών συστημάτων είναι οι υλοποιήσεις που σχετίζονται με τη γεωργία. Για παράδειγμα στη Galicia της Ισπανίας (*Ines Sante et all., 2004*) θέλουν να διαχειριστούν την παραγωγή της πατάτας. Το συμβούλιο που είναι υπεύθυνο για την παραγωγή της πατάτας αρχικά χρησιμοποιούσε τους κλασικούς χάρτες για να διαχειριστεί τη γεωγραφική πληροφορία. Το έτος 2003 αναπτύχθηκε μια διαδικτυακή εφαρμογή για να γίνει καταγραφή και οπτικοποίηση των αγροτεμαχίων για την παραγωγή πατάτας. Κατά τη διάρκεια όμως ανάπτυξης της εφαρμογής ενσωματώθηκαν λειτουργίες όπως ο έλεγχος της παραγωγής πατάτας καθώς και ο ποιοτικός έλεγχος. Με τη χρήση αυτού του προγράμματος είναι εφικτό ο έλεγχος κάθε αγροτεμαχίου για κάθε εποχή που γίνεται η καλλιέργεια της πατάτας.

Στο πλαίσιο ανάπτυξης της εφαρμογής χρησιμοποιήθηκε η σχεσιακή βάση δεδομένων sql server. Στη βάση δεδομένων έγινε αποθήκευση αλφαριθμητικών δεδομένων και γεωγραφικών δεδομένων. Για να γίνει ψηφιοποίηση των χαρτών χρησιμοποιήθηκε το εργαλείο GeoMedia. Η ιστοσελίδα της εφαρμογής αναπτύχθηκε με τις τεχνολογίες HTML, ASP, VBScript. Τα επίπεδα υλοποίησης της εφαρμογής είναι:

- **Πρώτο επίπεδο:** Το πρώτο επίπεδο είναι το επίπεδο του συστήματος υποστήριξης αποφάσεων όπου μπορεί ο χρήστης της εφαρμογής να εκτυπώσει αναφορές σχετικά με τα αγροτεμάχια καθώς και να δει οπτικοποιημένα γεωγραφικά δεδομένα πάνω στον χάρτη. Επίσης στο τμήμα του συστήματος αποφάσεων μέσω ερωτημάτων είναι αποθηκευμένα οι κανόνες της ρυθμιστικής αρχής για τον έλεγχο της παραγωγής πατάτας.
- **Δεύτερο επίπεδο:** Στο δεύτερο επίπεδο είναι η σχεσιακή βάση δεδομένων όπου είναι αποθηκευμένα τα γεωγραφικά στοιχεία των αγροτεμαχίων για παράδειγμα οι γεωγραφικές συντεταγμένες καθώς επίσης και ιστορικά δεδομένα σχετικά με την παραγωγή
- **Τρίτο επίπεδο:** Στο τρίτο επίπεδο είναι το επίπεδο της web εφαρμογής όπου οι αγρότες μέσω ερωτήσεων στο σύστημα βλέπουν οπτικοποιημένα δεδομένα στον χάρτη καθώς και πληροφορίες σχετικά με τα αγροτεμάχια τους που είναι η απόκριση από τη βάση δεδομένων.

Η χρήση αυτής της εφαρμογής κάλυψε τις ανάγκες των φορέων που εμπλέκονται στην παραγωγή πατάτας όπως τους αγρότες, τους αγρονόμους, και τους φορείς της πολιτείας. Η συγκεκριμένη εφαρμογή με κάποιες τροποποιήσεις θα μπορούσε να ωφελήσει και στη διαχείριση της παραγωγής άλλων αγροτικών προϊόντων. Συνολικά η εφαρμογή είχε μεγάλη αποδοχή τόσο από τους αγρότες όσο και από τους αγρονόμους και τους φορείς της πολιτείας διότι θα μπορούσε να βοηθήσει στη βελτιστοποίηση της παραγωγής της πατάτας.

Στην περιοχή Τοσκάνη της Ιταλίας (*Ricardo et all., 2010*) η επιτροπή της πόλης ήθελε να δημιουργήσει αιολικά πάρκα. Επειδή στη δημιουργία των αιολικών πάρκων υπάρχει σύγκρουση συμφερόντων μεταξύ περιβαλλοντικών οργανώσεων, των ιδιοκτητών και των φορέων θα έπρεπε να **Διαδικτυακή πλατφόρμα πολυεπίπεδης αρχιτεκτονικής για την πραγματοποίηση δυναμικών αναζητήσεων σε γεωπληροφοριακό σύστημα**

δημιουργηθεί ένα σύστημα το οποίο να υποστηρίζει πολλαπλά κριτήρια καθώς επίσης να συμπεριλαμβάνει και περιορισμούς. Για παράδειγμα δεν θα μπορούσε να δημιουργηθεί αιολικό πάρκο μέσα σε αρχαιολογικό χώρο. Η διαδικτυακή εφαρμογή που αναπτύχθηκε βασίστηκε σε βάση δεδομένων που θα είχε αποθηκευμένη γεωγραφική πληροφορία. Επίσης θα έπρεπε να ενσωματωθούν πολλαπλά κριτήρια έτσι ώστε η περιοχές που θα επιλεγόντουσαν για τη δημιουργία αιολικών πάρκων να μην έφερναν σε αντιπαράθεση τα μέλη με αντικρουόμενα συμφέροντα. Έτσι με αυτό τον τρόπο το σύστημα θα έπρεπε να επιλέγει περιοχές όπου θα συμφωνούσαν όλες οι ομάδες.

Το σύστημα αναπτύχθηκε με τη χρήση ελεύθερου λογισμικού όπως το Map Server και η ανάπτυξη έγινε με javascript και php. Επίσης προστέθηκαν 15 επίπεδα πληροφορίας που ο χρήστης μπορεί να δει για να επιλέξει την κατάλληλη περιοχή όπως ταχύτητα των ανέμων, χάρτες με την ενέργεια που μπορεί να παραχθεί από τους ανέμους σε μια περιοχή. Επίσης η εφαρμογή είναι διαθέσιμη δωρεάν από το διαδίκτυο. Η ιστοσελίδα της εφαρμογής είναι <<http://geoportale.lamma.rete.toscana.it/eng/windgis>[Προσπέλαση 2 Σεπτεμβρίου 2012].

Στην περιοχή της Κροατίας (*Jadraka,Ivicha., 2006*) δημιούργησαν ένα πληροφοριακό σύστημα με το οποίο παρακολουθούσαν το τμήμα του ποταμού Δούναβη που διέσχιζε την πόλη τους. Το σύστημα έδωσε τη δυνατότητα να παρακολουθούν δεδομένα που έστειλαν οι σταθμοί που είχαν εγκατασταθεί στο ποτάμι όπως το ύψος του νερού καθώς και δεδομένα σχετικά με τη μόλυνσή του. Επίσης υπάρχει η δυνατότητα να παρακολουθούν και τις κινήσεις των πλοίων που διασχίζουν το ποτάμι και αν αυτά συμμορφώνονται με τους περιβαλλοντικούς κανόνες. Το σύστημα ονομάζεται river IS και οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξή του είναι :

- Γεωγραφικό σύστημα πληροφοριών: Το σύστημα αυτό θα έχει ως σκοπό την οπτικοποίηση γεωγραφικών πληροφοριών στον χάρτη όπως για παράδειγμα μετρήσεις σχετικά με το ύψος του νερού στο ποτάμι καθώς και με δεδομένα που έχουν σχέση με τη μόλυνση του ποταμού.
- Σχεσιακή βάση δεδομένων: Η βάση που αποθηκεύονται δεδομένα που έχουν να κάνουν με τις αναφορές που θα εκτυπώνει το σύστημα.
- Autodesk MapGuide: Είναι το σύστημα για την απεικόνιση του χάρτη. Τα δεδομένα που χρησιμοποιεί είναι σε μορφή XML

Το πληροφοριακό σύστημα θα μπορούσε να χρησιμοποιηθεί από τις χώρες που βρίσκονται κοντά στον Δούναβη έτσι ώστε να έχουν μια καλύτερη και πιο αποδοτική εποπτεία του ποταμού Δούναβη, όπως έγινε με την πόλη της Κροατίας.

Το τμήμα των γεωγραφικών πληροφοριακών συστημάτων του πανεπιστημίου της πόλης Gdansk της Πολωνίας μαζί με το τμήμα του ελληνικού κέντρου ωκεανογραφίας (*Kulawiak et all., 2010*) συνεργάστηκαν για τη δημιουργία μιας διαδικτυακής εφαρμογής για την παρακολούθηση της περιβαλλοντικής μόλυνσης των ελληνικών χωρικών υδάτων. Η συνεργασία έγινε διότι λόγω της αυξανόμενης κίνησης των πλοίων στην περιοχή του Αιγαίου καθώς και στη Βαλτική Θάλασσα δημιουργήθηκε η ανάγκη για την ανάπτυξη ενός πληροφοριακού συστήματος που θα είχε ως στόχο τη μελέτη της μόλυνσης των θαλάσσιων περιοχών καθώς επίσης και μια εξομοίωση τις κίνησης των μολυσμένων υδάτων. Το πληροφοριακό σύστημα με αυτόν τον τρόπο βοήθησε στη μελέτη γεωγραφικών πληροφοριών καθώς και στην αναλύσή τους. Οι τεχνολογίες που χρησιμοποιήθηκαν ήταν οι PHP, Javascript, MapServer, και ArcIMS. Για να έχουν ακριβή μετεωρολογικά δεδομένα βασίστηκαν στα δεδομένα που παρέχει το πρόγραμμα POSEIDON. <[http://www.poseidon.hcmr.gr/weather\\_forecast\\_gr.php?area\\_id=gr](http://www.poseidon.hcmr.gr/weather_forecast_gr.php?area_id=gr)>.[Προσπέλαση 2 Σεπτεμβρίου 2012].

Στο πανεπιστήμιο τεχνολογίας της Μαλαισίας (*Zainuddin et all.,2010*) ανέπτυξαν ένα γεωγραφικό πληροφοριακό σύστημα με το οποίο θα έλυναν το πρόβλημα διαχείρισης χώρου των αιθουσών. Χρόνο με τον χρόνο ο αριθμός των φοιτητών στο πανεπιστήμιο αυξάνονταν έτσι το πανεπιστήμιο αντιμετώπιζε το πρόβλημα κατανομής των φοιτητών στις αίθουσες για την παρακολούθηση μαθημάτων σε συγκεκριμένες ώρες.

Το κύριο λογισμικό που χρησιμοποιήθηκε για την ανάπτυξη του γεωγραφικού συστήματος ήταν το ArcGIS. Χρησιμοποιήθηκε για τη δημιουργία των χαρτών και την εισαγωγή της τοπογραφικής πληροφορίας.

Τα δεδομένα στη συνέχεια τα επεξεργάστηκαν με το πρόγραμμα MapGuide που ανήκει στην κατηγορία του ελεύθερου λογισμικού. Για τη διαχείριση της πληροφορίας το MapGuide χρησιμοποιεί μια βάση δεδομένων XML. Τέλος, με το πρόγραμμα AutoCad ψηφιοποίησαν τους χάρτες των αιθουσών για την οπτικοποίηση των τοπολογικών δεδομένων που στη συγκεκριμένη περίπτωση ήταν τα κτήρια.

Μια διαδικτυακή εφαρμογή για τη μελέτη των σεισμών αναπτύχθηκε στο Πανεπιστήμιο του Πειραιά με την ονομασία Seismo-Surfer (*Γκέργι, 2005*). Ο σκοπός της συγκεκριμένης εφαρμογής είναι η ολοκλήρωση, οπτικοποίηση και ανάλυση σεισμικών δεδομένων που προέρχονται από τον ελληνικό και τον παγκόσμιο χώρο. Ουσιαστικά πρόκειται για τη δημιουργία μιας πύλης πληροφοριών και γνώσης σχετικά με το φαινόμενο του σεισμού. Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκαν τεχνολογίες Java (Swing, JavaBeans, applets στην πλευρά του πελάτη και Servlets, JSP, JDBC, Tomcat στον εξυπηρετητή), βάσεων δεδομένων (Oracle 9i with Spatial), GIS (OpenMap toolkit) και η ανάλυση και ο σχεδιασμός βασίστηκαν σε αντικειμενοστρεφείς τεχνικές (π.χ. UML διαγράμματα). Η ανάπτυξη έγινε με το JBuilder 9, και τα UML διαγράμματα έγιναν με το ArgoUML και το Jude.

Μια διαδικτυακή εφαρμογή που είχε ως στόχο την μέτρηση της στάθμης του νερού στον ποταμό Ευρώτα (*Νάκου, nd*). Βασίστηκε σε γεωγραφική βάση δεδομένων καθώς και σε τεχνολογίες PHP σε επίπεδο διακομιστή, και σε τεχνολογίες javascript και html σε επίπεδο πελάτη. Η βάση δεδομένων που χρησιμοποιήθηκε ήταν η MySQL και για την απεικόνιση των αποτελεσμάτων χρησιμοποιήθηκαν τα προγράμματα της εταιρίας google, δηλαδή το google maps και το google earth. Εγκαταστάθηκαν αισθητήρες στον ποταμό Ευρώτα έτσι ώστε να υπάρχουν μετρήσεις σχετικά με τη στάθμη του νερού. Ανάλογα με τη στάθμη του νερού ο αισθητήρας άλλαζε χρώμα έτσι ώστε αν σε κάποιον σταθμό αυξανόταν πάνω από κάποια όρια η στάθμη του νερού να μπορούσε να απεικονιστεί αντίστοιχα και στον χάρτη.

Η συγκεκριμένη υλοποίηση είναι πάρα πολύ χρήσιμη στους κατοίκους που μένουν κοντά σε ποταμούς έτσι ώστε να είναι ενήμεροι για το σημείο που έχει πλημμύρα ένα ποτάμι καθώς και τους εναλλακτικούς δρόμους για να αποφύγουν το σημείο της πλημμύρας.

Ένα άλλο παράδειγμα γεωγραφικού πληροφοριακού συστήματος είναι η καθοδήγηση χρηστών για να βρουν σημεία ενδιαφέροντος σε χάρτη (*Μανδουράρη, 2005*). Η ανάπτυξη της εφαρμογής έγινε με τη γλώσσα PHP όσο αφορά το κομμάτι του εξυπηρετητή, σε επίπεδο πελάτη χρησιμοποιήθηκε η html και η javascript. Η βάση δεδομένων που χρησιμοποιήθηκε ήταν η MySQL. Ο χρήστης της εφαρμογής έχει τη δυνατότητα να κάνει αναζήτηση για σημεία ενδιαφέροντος.

Μια εφαρμογή που αναπτύχθηκε με γεωγραφικά συστήματα πληροφοριών και συγκεκριμένα το πρόγραμμα ArcGIS Server είχε ως σκοπό την αξιοποίηση και τη διαχείριση της αγοράς ακινήτων **Διαδικτυακή πλατφόρμα πολυεπίπεδης αρχιτεκτονικής για την πραγματοποίηση δυναμικών αναζητήσεων σε γεωπληροφοριακό σύστημα**

(Σχοινιά, 2012). Η περιοχή που επιλέχθηκε για την εφαρμογή ήταν η ευρύτερη περιοχή του Δ. Λαρισίων. Από την ανάπτυξη της συγκεκριμένης εφαρμογής υπήρξαν πολλαπλά οφέλη που δεν περιορίστηκαν μόνο στους φορείς διαχείρισης ακινήτων αλλά επεκτάθηκαν και σε μια πλειάδα χρηστών. Ειδικότερα επηρεάζονται οι δημόσιοι οργανισμοί, οι οργανισμοί τοπικής αυτοδιοίκησης και οι χρηματοπιστωτικοί οργανισμοί.

Στο πλαίσιο ανάπτυξης του συστήματος, επιλέχθηκε ως εξυπηρετητής γεωγραφικών δεδομένων το λογισμικό ArcGIS Server 10, της ESRI. Οι λόγοι για τη χρήση του συγκεκριμένου εμπορικού λογισμικού σχετίζονται με:

- τις αυξημένες δυνατότητες και λειτουργίες του
- την εμπορικότητα και το ποσοστό χρησιμοποίησής του στην εγχωρία αγορά
- την άμεση διαθεσιμότητά του
- την δυνατότητα τεχνικής υποστήριξης, απόρροια της εμπορικότητας

Για την ανάπτυξη υπηρεσιών διαδικτύου χρησιμοποιούνται δύο ευρέως γνωστές πλατφόρμες, αυτή της Microsoft .NET framework και η πλατφόρμα της Java. Και οι δύο πλατφόρμες υποστηρίζουν πλήθος εργασιών για τη σύνταξη εφαρμογών και υπηρεσιών διαδικτύου. Το βασικό πλεονέκτημα της πλατφόρμας της Java, είναι ότι αποτελεί ένα ανοιχτό πρότυπο, κάτι που την κάνει ελκυστική, παρόλο που η πλατφόρμα της Microsoft θεωρείται μάλλον πιο εύχρηστη. Η επιλογή των windows server 2003 της Microsoft, ως λειτουργικού συστήματος, ουσιαστικά είχε ως αποτέλεσμα τη χρησιμοποίηση του λογισμικού ArcGIS Server 10 for the Microsoft .NET Framework, για την ανάπτυξη της εφαρμογής.

Αντίστοιχα, λόγω των τεχνικών απαιτήσεων, ως κατάλληλη έκδοση λογισμικού επιλέχθηκε η Enterprise Advanced, η οποία βάσει προδιαγραφών, παρέχει ένα ολοκληρωμένο περιβάλλον για τη διαχείριση, την οπτικοποίηση των χωρικών δεδομένων καθώς και για την επεξεργασία τους μέσω του διαδικτύου.

Η βάση δεδομένων για την υλοποίηση της συγκεκριμένης εφαρμογής ήταν η PostgreSQL. Η συγκεκριμένη βάση δεδομένων επιλέχθηκε επειδή έχει αυξημένη δημοτικότητα και επειδή είναι ανοικτό λογισμικό.

Στον Πίνακα 1 υπάρχουν οι τεχνολογίες που χρησιμοποιηθήκαν για την ανάπτυξη της κάθε εφαρμογής

Τεχνολογίες	Παραγωγή Πατάτας	Παρακολούθηση ποταμού Δούναβη	Εγκατάσταση ανεμογεννητριών	Διαχείριση χώρου πανεπιστημίου	Διαχείριση απόβλητων στο Αιγαίο	Seismo-surfer	Web εφαρμογή εργασίας	Εφαρμογή πλημμύρων	Εφαρμογή σημείων ενδιαφέροντος	Εφαρμογή διαχείρισης ακινήτων
Sql server	ok	-	-	-	-	-	-	-	-	-
Oracle database	-	ok	-	-	-	ok	-	-	-	-
GeoMedia	ok	-	-	-	-	-	-	-	-	-
Visual basic	ok	-	-	-	-	-	-	-	-	-
ActiveX	ok	-	-	-	-	-	-	-	-	-
VBScript	ok	-	-	-	-	-	-	-	-	-
MapGuide	-	-	-	ok	-	-	-	-	-	-
Maestro	-	-	-	ok	-	-	-	-	-	-
Geo-SMS	-	-	-	ok	-	-	-	-	-	-
ArcGis	-	-	-	-	ok	-	-	-	-	ok
ArchIMS	-	-	-	-	ok	-	-	-	-	-
Asp	-	ok	-	-	-	-	-	-	-	-
Asp.net	-	ok	-	-	-	-	-	-	-	-
Ado	ok	ok	-	-	-	-	-	-	-	-
Ado.net	-	ok	-	-	-	-	-	-	-	-
Xml	-	ok	-	ok	ok	ok	-	-	-	-
Xsit	-	ok	-	-	-	-	-	-	-	-
Map server	-	-	ok	-	-	-	-	-	-	-
php	-	-	ok	-	-	-	ok	ok	ok	-
javascript	-	-	ok	-	-	-	ok	ok	ok	-
Postgresql	-	-	-	-	-	-	ok	-	-	ok
PostGIS	-	-	-	-	-	-	ok	-	-	-
Json	-	-	-	-	-	-	ok	-	-	-
java	-	-	-	-	-	ok	-	-	-	-
Mysql	-	-	-	-	-	-	-	ok	ok	-
Google maps	-	-	-	-	-	-	-	ok	ok	-

Πίνακας 1 Συγκριτικός πίνακας τεχνολογιών περιπτώσεων χρήσης

Όπως παρατηρούμε στον παραπάνω πίνακα 1 για την ανάπτυξη των εφαρμογών το κοινό χαρακτηριστικό είναι ότι η διακίνηση της πληροφορίας μεταξύ των συστατικών του πληροφοριακού συστήματος της κάθε εφαρμογής γίνεται με την τεχνολογία XML.

Η παρούσα εργασία χρησιμοποιεί τις τεχνολογίες php και javascript όπως η υλοποίηση για τη μέτρηση της στάθμης του ποταμού Ευρώτα και η υλοποίηση για την εύρεση σημείων ενδιαφέροντος. Ωστόσο τα δύο συστήματα χρησιμοποιούν σαν αποθήκευση των δεδομένων το σύστημα βάσεων δεδομένων MySQL το οποίο δεν δίνει τη δυνατότητα εκτέλεσης χωρικών ερωτημάτων. Σε επίπεδο οπτικοποίησης των δεδομένων και οι τρεις εργασίες χρησιμοποιούν το πρόγραμμα google maps. Η κύρια διαφορά της παρούσας εργασίας είναι ότι η βάση δεδομένων έχει αναπτυχθεί σε postgresql με την επέκταση postgis που δίνει τη δυνατότητα εκτέλεσης χωρικών ερωτημάτων. Στα 2 προηγούμενα παραδείγματα ο χρήστης απλά βλέπει τα αποτελέσματα στον χάρτη. Στην παρούσα εργασία ο χρήστης μπορεί να επιλέξει μια περιοχή (ένα πολύγωνο) πάνω στον χάρτη και η βάση δεδομένων να του επιστρέψει σημεία ενδιαφέροντος που βρίσκονται μέσα στην περιοχή όπου έχει επιλέξει. Αυτή η λειτουργία προϋποθέτει στη βάση δεδομένων την εκτέλεση ενός χωρικού ερωτήματος για την επιστροφή των αποτελεσμάτων. Όπως επίσης ο χρήστης μπορεί να επιλέξει μια διαδρομή πάνω στον χάρτη και να δει σημεία ενδιαφέροντος πάνω στον χάρτη. Για την υλοποίησή τους μπορείτε να ανατρέξετε στο κομμάτι του παραρτήματος της εφαρμογής στο κώδικα του αρχείου(geo\_model.php).

Η εφαρμογή με τη διαχείριση των ακινήτων χρησιμοποιεί τη βάση δεδομένων postgresQL όπως και η υπάρχουσα εργασία αλλά η οπτικοποίηση στην εφαρμογή της διαχείρισης ακινήτων χρησιμοποιεί το πακέτο ArcGIS.

Στα παραπάνω παραδείγματα βλέπουμε πόσο σημαντικό ρόλο παίζουν τα γεωγραφικά πληροφοριακά συστήματα στη λήψη αποφάσεων καθώς και στην καλύτερη διαχείριση γεωγραφικών πληροφοριών που παλιότερα γίνονταν με τους χάρτες. Τα παραδείγματα που αναφέρθηκαν ήταν ξεχωριστές εφαρμογές από διάφορους τομείς της ανθρώπινης δραστηριότητας, όπως στην πόλη της Ισπανίας για τη βελτιστοποίηση της παραγωγής πατάτας, στην πόλη της Ιταλίας για τη μέτρηση των ανέμων και την κατασκευή αιολικών πάρκων, στην περιοχή της Κροατίας για τη μελέτη του περιβάλλοντος στον ποταμό Δούναβη, στην περιοχή των ελληνικών χωρικών υδάτων για τη μόλυνση των υδάτων από την έλευση των πλοίων, στο πανεπιστήμιο της Μαλαισίας για τη διαχείριση των εγκαταστάσεων του, η διαχείριση της στάθμης του ποταμού Ευρώτα, η εφαρμογή για την εύρεση σημείων ενδιαφέροντος και τέλος η εφαρμογή για τη διαχείριση των ακινήτων. Τα γεωγραφικά συστήματα στις παραπάνω εφαρμογές παίζουν έναν καθοριστικό ρόλο στη διαχείριση της γεωγραφικής πληροφορίας.

### 3. Τεχνολογίες

Ένα από τα μεγαλύτερα πλεονεκτήματα του Παγκόσμιου Ιστού ως μέσο επικοινωνίας είναι η διασυστημική (cross platform) συμβατότητα. Διαφορετικές πλατφόρμες και λειτουργικά συστήματα μπορούν να επικοινωνήσουν μεταξύ τους. Οι επισκέπτες του Ιστού έχουν πρόσβαση στις πληροφορίες στον Ιστό με διαφορετικά λειτουργικά συστήματα. Αυτό είναι ένα πλεονέκτημα έτσι ώστε η εφαρμογή να είναι προσπελάσιμη από πολλούς χρήστες.

Παρακάτω αναφέρονται οι τεχνολογίες που χρησιμοποιηθήκαν για την ανάπτυξη της εφαρμογής.

#### 3.1.PostGIS

Το PostGIS(<http://postgis.net/>) είναι μια χωρική βάση δεδομένων που επεκτείνει τη σχεσιακή βάση δεδομένων PostgreSQL, είναι η πιο ισχυρή ανοικτού λογισμικού χωρική βάση δεδομένων. Προσθέτει στη PostgreSQL χωρικούς τύπους δεδομένων και πάνω από 300 συναρτήσεις για την επεξεργασία των χωρικών δεδομένων.

Επίσης υπάρχουν και οι τύποι δεδομένων που αποθηκεύουν πολλούς «απλούς» τύπους δεδομένων όπως πολυσημείο (multipoint), πολύ-πολύγωνα (multipolygon) και πολυγραμμή (multilinestring).

Μια χωρική βάση δεδομένων παρέχει τη δυνατότητα της ανάμειξης των χωρικών δεδομένων μαζί με εταιρικά δεδομένα όπως με οικονομικά δεδομένα ή δεδομένα που αφορούν τις πωλήσεις. Αυτό μπορεί να γίνει εισάγοντας τα χωρικά δεδομένα μέσω αρχείων τύπου shapfile ή kmf. Παρέχεται επίσης η δυνατότητα για την εκτέλεση συναρτήσεων που έχουν να κάνουν με στατιστικά δεδομένα. Επίσης όταν έχει αναπτυχθεί η διαδικτυακή εφαρμογή σε οποιαδήποτε γλώσσα προγραμματισμού, php, java, .net, perl, python σε κάθε μια από αυτές αναφέρεται στο εγχειρίδιο χρήσης η σύνδεση με τη βάση δεδομένων PostgreSQL.

#### 3.2.PostgreSQL

Η PostgreSQL (<http://www.postgresql.org/>) είναι μια ισχυρή ανοικτού λογισμικού αντικείμενο-σχεσιακή βάση δεδομένων. Ο κώδικας και οι λειτουργίες της δεν αναπτύσσονται από μια και μόνο εταιρία αλλά αναπτύσσονται από την κοινότητα των χρηστών της. Η PostgreSQL τρέχει σε όλα τα βασικά λειτουργικά συστήματα, περιλαμβάνοντας του Linux, UNIX και Windows. Είναι συμβατή με το πρότυπο ACID (<http://www.postgresql.org/about/>), έχει ολοκληρωμένη υποστήριξη για ξένα κλειδιά, ενώσεις, όψεις, σκανδαλιστές, και αποθηκευμένες διαδικασίες (σε διάφορες γλώσσες προγραμματισμού). Η PostgreSQL υλοποιεί εξαιρετικά χαρακτηριστικά όπως Multi-Version Concurrency Control (το MVCC είναι μια διαδικασία με την οποία όταν η βάση δεδομένων θέλει να κάνει ενημέρωση μια εγγραφή στη βάση δεδομένων τότε δεν σβήνει την παλιά εγγραφή και προσθέτει την καινούργια αλλά μαρκάρει την παλιά εγγραφή ως διαγραμμένη και προσθέτει σε νέα εγγραφή την καινούργια τιμή), tablespaces, ασύγχρονη αντιγραφή, εμφωλευμένες συναλλαγές, αντίγραφα ασφαλείας ενώ η βάση λειτουργεί, βελτιστοποιημένα ερωτήματα και καταγραφή συμβάντων για ελαχιστοποίηση των σφαλμάτων. Υποστηρίζει διεθνή σύνολα χαρακτήρων, κωδικοποίηση χαρακτήρων σε πολλά byte, Unicode καθώς και δυνατότητα ταξινόμησης δεδομένων ανεξάρτητα από την τοποθεσία. Η PostgreSQL μπορεί να διαχειριστεί εύκολα μεγάλους αριθμούς ταυτόχρονων χρηστών

*Διαδικτυακή πλατφόρμα πολυεπίπεδης αρχιτεκτονικής για την πραγματοποίηση δυναμικών αναζητήσεων σε γεωπληροφοριακό σύστημα*



καθώς και μεγάλο όγκο δεδομένων. Υπάρχουν ενεργές εγκαταστάσεις σε περιβάλλοντα παραγωγής που διαχειρίζονται πάνω από 4 terabytes δεδομένων. Η PostgreSQL είναι συνεπής με τις προδιαγραφές.

Η υλοποίησή της είναι απολύτως σύμφωνη με τις προδιαγραφές ANSI-SQL 92/99 (<http://gerardnico.com/wiki/language/sql/ansi>). Έχει ολοκληρωμένη υποστήριξη για εμφωλευμένα υποερωτήματα. Η PostgreSQL αποτελεί ένα πλήρες σχεσιακό σύστημα που υποστηρίζει πολλαπλά σχήματα ανά βάση δεδομένων. Ο κατάλογος (πληροφορίες σχετικά με τους πίνακες, στήλες, όψεις) είναι διαθέσιμος διαμέσου του Information Schema όπως ορίζεται στο SQL standard. Στα χαρακτηριστικά για ακεραιότητα δεδομένων συμπεριλαμβάνονται: πρωτεύοντα κλειδιά, και ξένα κλειδιά. Η PostgreSQL έχει αρκετά προηγμένα χαρακτηριστικά όπως: αυτόματη αρίθμηση μέσω σειρών, και LIMIT/OFFSET που επιτρέπουν την επιστροφή μερικών αποτελεσμάτων. Πάνω στο GiST και τη PostgreSQL έχουν θεμελιωθεί πολλά projects όπως τα OpenFTS (<http://openfts.sourceforge.net/>) και PostGIS (<http://postgis.net/>). Το OpenFTS (Open Source Full Text Search engine) παρέχει online indexing και σχετική κατηγοριοποίηση για αναζήτηση. Το PostGIS αποτελεί ένα project το οποίο προσθέτει υποστήριξη για γεωγραφικά αντικείμενα στη PostgreSQL, επιτρέποντας τη χρήση της σαν γεωγραφική βάση δεδομένων (ΓΠΣ), περίπου όπως τα ESRI's SDE ή τις χωρικές επεκτάσεις της Oracle .

Άλλα προηγμένα χαρακτηριστικά της PostgreSQL είναι: table inheritance, rules systems και database events. Το Table inheritance (κληρονομικότητα πινάκων) προσθέτει μια αντικειμενοστρεφή διάσταση στη δημιουργία πινάκων, επιτρέποντας στους σχεδιαστές ΒΔ να δημιουργούν νέους πίνακες από άλλους πίνακες χρησιμοποιώντας τους ως βάση. Ακόμα καλύτερα η PostgreSQL υποστηρίζει και μονή και πολλαπλή κληρονομικότητα με τον δικό της τρόπο. Το rules system, το οποίο καλείται the query rewrite system, επιτρέπει στον σχεδιαστή βάσεων να δημιουργήσει κανόνες που ορίζουν συγκεκριμένες λειτουργίες για έναν πίνακα ή view, και να τις μετατρέπει δυναμικά την ώρα που εκτελούνται λειτουργίες σε άλλες εναλλακτικές. Το σύστημα γεγονότων αποτελεί ένα διαδραστικό σύστημα επικοινωνίας στο οποίο μηνύματα και γεγονότα μπορούν να μεταδοθούν μεταξύ πελατών (clients) χρησιμοποιώντας τις εντολές LISTEN και NOTIFY, επιτρέποντας από την απλή peer to peer επικοινωνία ως ένα εξελιγμένο συντονισμό βασισμένο σε γεγονότα ΒΔ. Εφόσον οι ιδιοποιήσεις μπορεί να προέρχονται από triggers, stored procedures, και PostgreSQL clients μπορούν να επιβλέπουν λειτουργίες όπως: ενημερώσεις, εισαγωγές ή διαγραφές πινάκων όταν αυτά γίνονται. Οι λειτουργίες τις οποίες υποστηρίζει είναι:

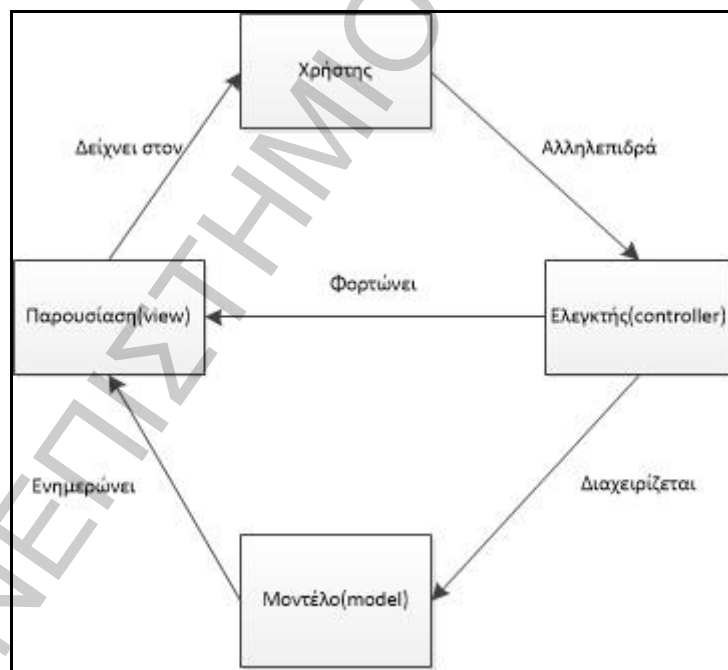
- Συναρτήσεις: Εκτός από τη γλώσσα προγραμματισμού SQL υποστηρίζει τις παρακάτω γλώσσες προγραμματισμού (PL/pgSQL, PL/Tcl, PL/Perl, PL/Python, PL/Java PL/Js, PL/LOLCODE, PL/Lua, PL/OpenCL, PL/Parrot, PL/PHP, PL/Proxy, PL/PSM, PL/R, PL/Ruby, PL/scheme, PL/sh, PL/V8). Μέσω αυτών μπορούμε να αναπτύξουμε συναρτήσεις για να εκτελέσουμε πιο πολύπλοκα ερωτήματα για ανάλυση των δεδομένων που με απλή sql δεν θα μπορούσαμε να εκτελέσουμε.
- GiST (generalized indexing search trees)-Γενικευμένα δέντρα αναζήτησης δεικτών: Το GiST (Generalized indexing Search Tree) αποτελεί ένα προηγμένο σύστημα το οποίο συνδυάζει ένα μεγάλο εύρος από διαφορετικούς αλγόριθμους ταξινόμησης και αναζήτησης όπως B-tree, B+-tree, R-tree, partial sum trees, ranked B+-trees και αρκετούς ακόμα. Επίσης διαθέτει διεπαφή το οποίο επιτρέπει τόσο τη δημιουργία custom data types όσο και επεκτάσιμους τρόπους ερωτήσεων (query) για την αναζήτηση τους. το GiST προσφέρει την ευελιξία στον προσδιορισμό του τι και με πιο τρόπο να το αποθηκεύσεις, και τη δυνατότητα να ορίσεις νέους τρόπους αναζήτησης.
- Gin (generalized inverted indexes)-Γενικευμένοι αντίθετοι δείκτες: Είναι σχεδιασμένο για να χειρίζεται περιπτώσεις όπου τα αντικείμενα που πρέπει να ταξινομηθούν είναι πολύπλοκες

τιμές και τα ερωτήματα που πρέπει να εκτελεστούν ψάχνουν τιμές που είναι μέσα σε πολύπλοκα αντικείμενα για παράδειγμα ψάχνοντας για συγκεκριμένες λέξεις μέσα σε κείμενα.

### 3.3.Codeigniter

Το codeigniter (<http://ellislab.com/codeigniter>) είναι ένα πλαίσιο ανάπτυξης web εφαρμογών για τους προγραμματιστές που αναπτύσσουν ιστοσελίδες χρησιμοποιώντας τη γλώσσα προγραμματισμού php. Για την ανάπτυξη των εφαρμογών το codeigniter χρησιμοποιεί την προσέγγιση MVC (Model-View-Controller). Αυτή η προσέγγιση διαχωρίζει την εφαρμογή σε 3 επίπεδα :

- Το πρώτο επίπεδο είναι το επίπεδο του μοντέλου (model) όπου σε αυτό το επίπεδο βάζουμε εκείνο το κομμάτι του κώδικα όπου συνδέεται η εφαρμογή μας με μια βάση δεδομένων.
- Το δεύτερο επίπεδο είναι το επίπεδο του ελεγκτή (controller) όπου σε αυτό το επίπεδο γράφουμε το τμήμα του κώδικα όπου τα αποτελέσματα της επεξεργασίας από το επίπεδο του μοντέλου θα τα πάρει το δεύτερο επίπεδο έτσι ώστε να τα μεταφέρει στο επίπεδο της παρουσίασης (view)
- Το τρίτο επίπεδο είναι το επίπεδο της παρουσίασης (view) όπου ουσιαστικά είναι το επίπεδο εκείνο όπου βάζουμε τον κώδικα της παρουσίασης δηλαδή είναι αυτό που βλέπει ο χρήστης στην εφαρμογή του περιηγητή. Σε αυτό το επίπεδο παρουσιάζονται τα δεδομένα που έχουν τύχει επεξεργασίας από το μοντέλο.



Εικόνα 4 Πρότυπο M-V-C

Πηγή εικόνας 4 (Myer,2008,ρ 5)

Παρακάτω θα αναλυθεί η λειτουργία αυτού του μοντέλου με παράδειγμα.

*Διαδικτυακή πλατφόρμα πολυεπίπεδης αρχιτεκτονικής για την πραγματοποίηση δυναμικών αναζητήσεων σε γεωπληροφοριακό σύστημα*

```

<?php
class Ex_model extends CI_Model {

    public function __construct()
    {

    }

    public function getVar()
    {

        $ex="Hello MVC programming world !!!";
        return $ex ;

    }

}

```

Εικόνα 5 Κώδικας μοντέλου

Στην Εικόνα 5 έχουμε δημιουργήσει το πρώτο επίπεδο του προγραμματιστικού μοντέλου MVC δηλαδή το μοντέλο της εφαρμογής μας. Έχουμε δημιουργήσει μια συνάρτηση getVar όπου σε αυτή έχουμε ορίσει μια μεταβλητή την ex και η συνάρτηση απλά την επιστρέφει.

```

<?php
class Ex_view extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
        $this->load->model('ex_model');
    }

    public function view() {
        $data['exdata']=$this->ex_model->getVar();
        $this->load->view('view/header.php');
        $this->load->view('view/view.php', $data);
        $this->load->view('view/footer.php');
    }

}

```

Εικόνα 6 Κώδικας ελεγκτή

Στην εικόνα 6 έχουμε δημιουργήσει τον ελεγκτή, το δεύτερο επίπεδο του προγραμματιστικού μοντέλου MVC το οποίο μέσω της συνάρτησης construct επικοινωνεί με το μοντέλο που δημιουργήσαμε για να πάρει την τιμή της μεταβλητής μας και να την προωθήσει στο τρίτο επίπεδο που είναι το view.

```

<html>
    <head>
        <title>Παράδειγμα</title>
    </head>
    <body>
        <?php echo $exdata ;?>
    </body>
</html>

```

Εικόνα 7 Κώδικας όψης

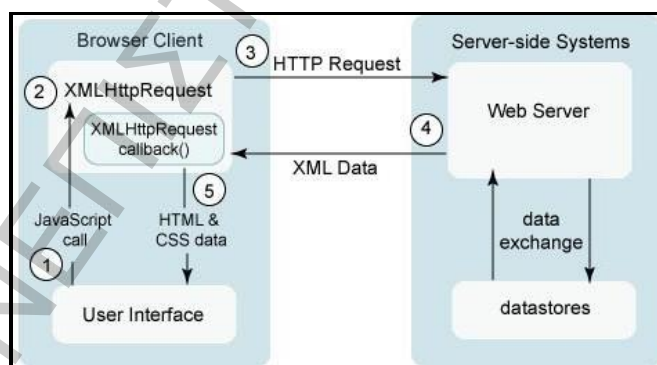
Στην εικόνα 7 έχουμε δημιουργήσει την όψη, το τρίτο επίπεδο του προγραμματιστικού μοντέλου MVC το οποίο μέσω του κώδικα php μας εμφανίζει το περιεχόμενο της μεταβλητής μας.

### 3.4.Jquery

Η javascript είναι μια γλώσσα προγραμματισμού που μας βοηθά στην ανάπτυξη εφαρμογών σε επίπεδο πελάτη. Η JQuery είναι ουσιαστικά μια βιβλιοθήκη της javascript. Ενσωματώνοντας αυτή τη βιβλιοθήκη στον κώδικα μπορούμε να αναπτύσσουμε κώδικα javascript πολύ πιο εύκολα και πολύ πιο ευέλικτα. Η ευελιξία αυτή ώθησε τους προγραμματιστές ανάπτυξης περιεχομένου σε επίπεδο πελάτη να δημιουργήσουν πλούσια plugins που κάνουν πιο διασκεδαστική την περιήγηση των χρηστών στο διαδίκτυο. Έτσι η βιβλιοθήκη έγινε ιδιαίτερα δημοφιλής. Αν κάνουμε μια αναζήτηση σε μηχανή αναζήτησης για jquery plugins τότε θα δούμε ότι υπάρχουν εκατοντάδες και μπορούμε να καταλάβουμε πόσο μεγάλο οικοσύστημα έχει δημιουργηθεί στην ανάπτυξη εφαρμογών σε επίπεδο πελάτη γύρω από αυτή τη βιβλιοθήκη. Σε μεγάλες εταιρίες όπως οι IBM, Netflix, Amazon, Dell σε κρίσιμες εφαρμογές χρησιμοποιούν jquery. Επίσης η Microsoft και η Nokia έχουν ενσωματώσει στα εργαλεία ανάπτυξης εφαρμογών τη συγκεκριμένη βιβλιοθήκη. Η JQuery είναι συμβατή με όλους τους περιηγητές που είναι διαθέσιμοι αυτή τη στιγμή. Αυτό είναι ένα πλεονέκτημα όταν γίνεται ανάπτυξη εφαρμογών να υπάρχει η συμβατότητα έτσι ώστε ο προγραμματιστής να μην χρειάζεται να γράφει ξεχωριστό κώδικα για κάθε περιηγητή ξεχωριστά. Έτσι αντί να καταναλώνει τον χρόνο του στον έλεγχο της συμβατότητας μπορεί να τον καταναλώνει στο να φτιάχνει πιο πλούσιες εφαρμογές που θα διευκολύνουν το χρήστη στη λειτουργία της κάθε εφαρμογής.

### 3.5. Ajax

Με τη javascript οι web developers μπορούσαν να φτιάξουν πολύ ωραία γραφικά στις ιστοσελίδες αλλά υπήρχε ένα εμπόδιο. Το εμπόδιο ήταν ότι δεν μπορούσε να επικοινωνήσει μια λειτουργία που εκτελούσε ο κώδικας javascript με τον εξηρητητή. Για να γίνει αυτό θα έπρεπε να γίνει με λειτουργίες pηr όπως η εντολή `$_POST` ή `$_GET`. Το αποτέλεσμα της λειτουργίας το εμφάνιζε ο εξυρητητής σε νέα σελίδα που άνοιγε. Το 2005 η εταιρία Google εισήγαγε το προγραμματιστικό μοντέλο AJAX (Asynchronous Javascript And Xml) ή αλλιώς XMLHttpRequest. Αυτό μας επιτρέπει να πάρουμε τα αποτελέσματα της επεξεργασίας του εξυρητητή στη σελίδα που είναι ο χρήστης χωρίς να χρειάζεται να ανοίγει μια καινούρια σελίδα.



Εικόνα 8 Λειτουργία ajax

Στην Εικόνα 8([http://images.drccsystems.com/ajax\\_img.jpg](http://images.drccsystems.com/ajax_img.jpg)) πρώτα καλείται μια συνάρτηση javascript (1), στη συνέχεια δημιουργείται ένα αντικείμενο XMLHttpRequest (2) το οποίο έχει τα δεδομένα που θα σταλούν στον εξυρητητή (3). Ο εξυρητητής επιστρέφει τα δεδομένα με μορφή XML (4). Τα δεδομένα αυτά αποθηκεύονται μέσα σε μια συνάρτηση του αντικειμένου XMLHttpRequest

*Διαδικτυακή πλατφόρμα πολυεπίπεδης αρχιτεκτονικής για την πραγματοποίηση δυναμικών αναζητήσεων σε γεωπληροφορικό σύστημα*

(5). Μετά τα αποτελέσματα θα εμφανιστούν στον περιηγητή ανάλογα με τις λειτουργίες που θα έχει η συνάρτηση επιστροφής (5).

### 3.6.Json

Το Json το οποίο σημαίνει Javascript Object Notation, είναι μια μορφή αποθήκευσης και ανταλλαγής πληροφορίας. Το Json σαν αντικείμενο είναι πιο μικρό σε σχέση με ένα αντικείμενο xml. Επίσης η σύνταξη της δημιουργίας ενός αντικειμένου Json είναι ίδιος με τη δημιουργία ενός αντικειμένου σε javascript.

```
{
  "employees": [
    { "firstName": "John", "lastName": "Doe" },
    { "firstName": "Anna", "lastName": "Smith" },
    { "firstName": "Peter", "lastName": "Jones" }
  ]
}
```

Εικόνα 9 Παράδειγμα json

Στην Εικόνα 9 το αντικείμενο employees είναι ένας πίνακας που αποτελείται από 3 εγγραφές (αντικείμενα).

#### 3.6.1. Geojson

Το geojson είναι ένα αντικείμενο τύπου Json που περιλαμβάνει πληροφορία σχετικά με γεωγραφικές συντεταγμένες. Αυτό το αντικείμενο μας επιτρέπει πληροφορία που είναι αποθηκευμένη σε γεωγραφική βάση δεδομένων όπως είναι η PostgreSQL να την αναπαραστήσουμε σε εφαρμογή χαρτών όπως για παράδειγμα το Google maps.

```
{ "type": "Polygon",
  "coordinates": [
    [ [100.0, 0.0], [101.0, 0.0], [101.0, 1.0], [100.0, 1.0], [100.0, 0.0] ]
  ] }
```

Εικόνα 10 Αντικείμενο geojson

Στην εικόνα 10 παρουσιάζεται ένα αντικείμενο geojson. Στο παράδειγμα περιγράφεται ένα πολύγωνο με τις συντεταγμένες του.

### 3.7. Google maps API V.3

Στην παρούσα εργασία χρησιμοποιήθηκε το Google maps <<https://developers.google.com/maps/documentation/javascript/>>, το οποίο είναι η πλατφόρμα χαρτών της εταιρίας Google για την απεικόνιση των αποτελεσμάτων στο χρήστη. Το προγραμματιστικό μοντέλο του εργαλείου που χρησιμοποιήθηκε ήταν η έκδοση 3. Στην Εικόνα 11 παρουσιάζεται ο κώδικας του google api για τους χάρτες. Πρόκειται για μια συνάρτηση javascript που φορτώνει το χάρτη μόλις φορτωθεί η σελίδα. Στις ρυθμίσεις του χάρτη ελέγχουμε σε ποιο σημείο θα δείχνει ο χάρτης, δηλαδή ορίζουμε τις γεωγραφικές συντεταγμένες του, επίσης ορίζουμε το επίπεδο εστίασης καθώς και τον τύπο του χάρτη.

```
<html>
  <head>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
    <style type="text/css">
      html { height: 100% }
      body { height: 100%; margin: 0; padding: 0 }
      #map_canvas { height: 100% }
    </style>
    <script type="text/javascript"
      src="http://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&sensor=SET_TO_TRUE_OR_FALSE">
    </script>
    <script type="text/javascript">
      function initialize() {
        var mapOptions = {
          center: new google.maps.LatLng(-34.397, 150.644),
          zoom: 8,
          mapTypeId: google.maps.MapTypeId.ROADMAP
        };
        var map = new google.maps.Map(document.getElementById("map_canvas"),
          mapOptions);
      }
    </script>
  </head>
  <body onload="initialize()">
    <div id="map_canvas" style="width:100%; height:100%"></div>
  </body>
</html>
```

Εικόνα 11 Κώδικας google api –hello world

## 4. Αρχιτεκτονική

### 4.1. Υλοποίηση εφαρμογής

Η υλοποίηση της εφαρμογής είχε ως σκοπό τον συνδυασμό των τεχνολογιών που αναλύθηκαν στο Κεφάλαιο 3. Αρχικά έγινε η εγκατάσταση της βάσης δεδομένων PostgreSQL. Στη συνέχεια έγινε η εγκατάσταση της επέκτασης PostGIS που ενσωματώνεται στη PostgreSQL για την αποθήκευση χωρικών δεδομένων. Επίσης μέσω των λειτουργιών της PostGIS δίνεται η δυνατότητα για την εκτέλεση χωρικών ερωτημάτων πάνω στα δεδομένα της βάσης.

Αρχικά η βάση δεδομένων είχε ένα πίνακα όπου αποθηκεύονταν τα σημεία ενδιαφέροντος. Όταν ο χρήστης εκτελούσε τις δυο λειτουργίες του προγράμματος, δηλαδή τη λειτουργία του πολυγώνου που θα επέστρεφε τα σημεία ενδιαφέροντος που βρίσκονταν μέσα στην περιοχή που είχε ορίσει και τη δεύτερη λειτουργία όπου το σύστημα θα επέστρεφε σημεία ενδιαφέροντος που βρισκόταν κοντά στη διαδρομή που είχε ορίσει, το σύστημα δεν ήταν τόσο βελτιστοποιημένο διότι υπήρχε εξάρτηση από το πλήθος των εγγράφων που είχε ο πίνακας. Ο χρόνος απόκρισης του συστήματος αυξάνονταν αναλογικά με το πλήθος των εγγράφων στον πίνακα. Για το λόγο αυτό για να βελτιστοποιηθεί η απόκριση του συστήματος δημιουργήθηκε ο δεύτερος πίνακας, όπου εκεί ορίστηκε μια περιοχή όπου ανήκουν τα σημεία ενδιαφέροντος έτσι ώστε όταν έχουμε μεγάλο πλήθος σημείων ενδιαφέροντος να είναι κατηγοριοποιημένα ανά περιοχές έτσι ώστε να έχει το σύστημα καλύτερη απόκριση. Γι' αυτό το λόγο ορίστηκε στα πλαίσια της παρούσας εργασίας η περιοχή του Πειραιά καθώς και τα σημεία ενδιαφέροντος που ανήκουν σε αυτή τη περιοχή.

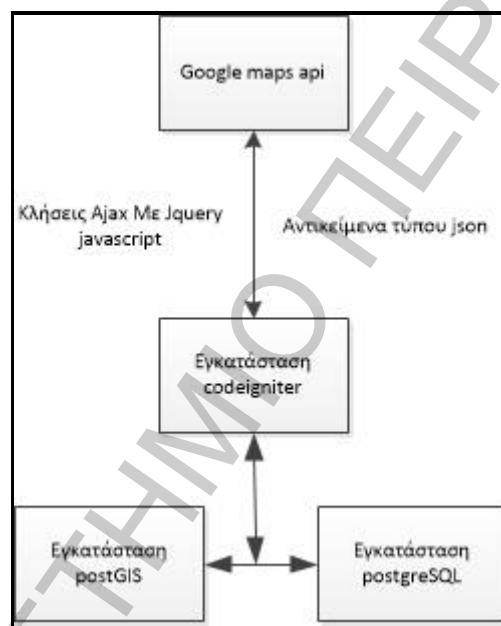
Για να βρεθούν οι γεωγραφικές συντεταγμένες των σημείων ενδιαφέροντος καθώς και της περιοχής του Πειραιά χρησιμοποιήθηκε το πρόγραμμα google earth όπου με τις διευθύνσεις των σημείων ενδιαφέροντος το πρόγραμμα επέστρεφε τις γεωγραφικές συντεταγμένες. Στη συνέχεια με τα ερωτήματα sql (insert) πέρασαν τα σημεία ενδιαφέροντος, και η περιοχή του Πειραιά μαζί με γεωγραφικά τους δεδομένα μέσα στους πίνακες της βάσης δεδομένων

Αφού ολοκληρώθηκε η δημιουργία της βάσης δεδομένων με τις εγγραφές, το επόμενο βήμα ήταν το χτίσιμο της εφαρμογής γύρω από το πλαίσιο ανάπτυξης λογισμικού codeigniter. Για να γίνει αυτό έγινε σε πρώτο στάδιο η σύνδεση της βάσης δεδομένων με το codeigniter. Στη συνέχεια δημιουργήθηκε το μοντέλο όπου εκεί αναπτύχθηκε ο κώδικας των ερωτημάτων της εφαρμογής. Το επόμενο βήμα ήταν η δημιουργία των ελεγκτών για τη μεταφορά δεδομένων από το μοντέλο στην όψη του προγράμματος και το αντίθετο. Τέλος δημιουργήθηκαν αρχεία σε jquery και οι κλήσεις Ajax της εφαρμογής και η ανταλλαγή των δεδομένων μεταξύ των επίπεδων της εφαρμογής μέσω αντικειμένων τύπου json. Με αυτό το τρόπο αναπτύχθηκαν οι λειτουργίες που εκτελεί ο χρήστης στην παρούσα εργασία.

Ο χρήστης μέσω των λειτουργιών εισαγωγής πολυγώνου, και εισαγωγής γραμμής έχει τη δυνατότητα να δει τα σημεία που τον ενδιαφέρουν στο χάρτη. Για να πραγματοποιηθούν αυτές οι δύο λειτουργίες συνδυάζονται τεχνολογίες τόσο του διαδικτύου όσο και των γεωγραφικών βάσεων δεδομένων. Σε πρώτο επίπεδο όταν ο χρήστης για παράδειγμα επιλέγει μια περιοχή (ένα πολύγωνο) και ένα σημείο ενδιαφέροντος μέσω της τεχνολογίας AJAX αποστέλλονται αυτά τα δεδομένα σε ένα χωρικό ερώτημα. Η βάση δεδομένων εκτελεί αυτό το ερώτημα και αποστέλλει μέσω της τεχνολογίας Ajax τα δεδομένα στην εφαρμογή μας όπου μέσω της πλατφόρμας για τους χάρτες google maps ο χρήστης βλέπει τα αποτελέσματα στο χάρτη. Ο χρήστης αλληλεπιδρά με το χάρτη και η βάση δεδομένων κάθε φορά επιστρέφει τα δεδομένα εφόσον υπάρχουν. Μια επιπρόσθετη λειτουργία που αναπτύχθηκε κατά τη διάρκεια ανάπτυξης της εφαρμογής ήταν η τεχνολογία της ομαδοποίησης των

αποτελεσμάτων. Εάν το σύστημα επέστρεφε σε ένα πολύγωνο πολλά δεδομένα μαζί για να μην υπάρχουν διάσπαρτα πολλά σημεία ενδιαφέροντος τότε το σύστημα ομαδοποιεί αυτά τα δεδομένα και όταν ο χρήστης κάνει κλικ σε μια ομάδα τότε η εφαρμογή του δείχνει τα σημεία ενδιαφέροντος που βρίσκονται μέσα στην ομάδα. Με αυτόν τον τρόπο υπάρχει μια ομαδοποίηση των δεδομένων. Ένα άλλο σημείο που βελτιώθηκε κατά τη διάρκεια ανάπτυξης της εφαρμογής σχετίζεται με τη βελτιστοποίηση της βάσης δεδομένων. Αρχικά ο χρόνος απόκρισης της βάσης δεδομένων όταν γινόταν εκτέλεση του χωρικού ερωτήματος ήταν απαγορευτικός για την οπτικοποίηση των δεδομένων στο χάρτη. Αυτό βελτιώθηκε με τη δημιουργία ταξινόμησης (Indexing) των δεδομένων.

Η Εικόνα 12 δείχνει την εφαρμογή και τη σύνδεση μεταξύ των τεχνολογιών που χρησιμοποιήθηκαν στην ανάπτυξη της. Τα δεδομένα της εφαρμογής μέσω της τεχνολογίας ajax και των αντικειμένων τύπου json οπτικοποιούνται στον χάρτη μέσω του google maps.



Εικόνα 12 Διάγραμμα υλοποίησης

Ο κώδικας της εφαρμογής παρουσιάζεται αναλυτικά στο παράρτημα της εργασίας.

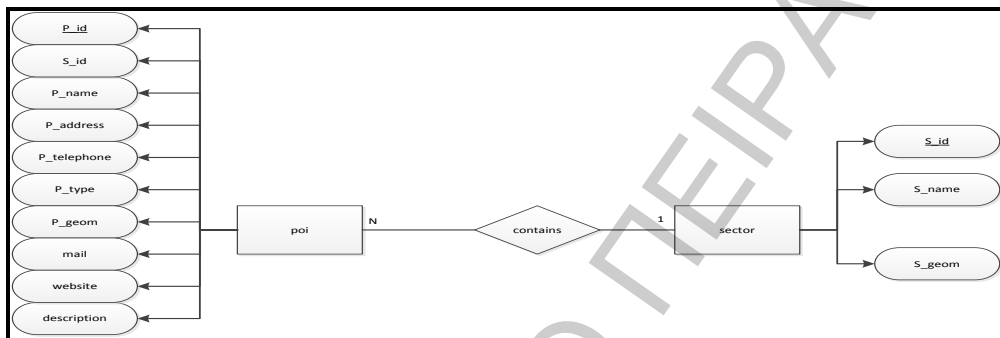


## 4.2. Σχεδίαση βάσης δεδομένων

Η βάση δεδομένων για την υλοποίηση έγινε στο σύστημα βάσεων δεδομένων PostgreSQL με τις λειτουργίες postGIS έτσι ώστε να μπορούμε να εκτελέσουμε ερωτήματα με χωρικές λειτουργίες.

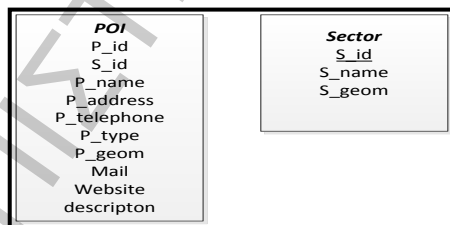
Η βάση αποτελείται από δύο πίνακες. Η σχέση μεταξύ των δύο πινάκων είναι σχέση ένα προς ένα. Ο πρώτος πίνακας είναι ο πίνακας sector όπου η γεωμετρία του είναι τύπου πολυγώνου. Ο δεύτερος πίνακας είναι ο πίνακας poi όπου η γεωμετρία του είναι τύπου σημείου. Όπως αναφέρθηκε η σχέση είναι ένα προς ένα διότι ένα σημείο ανήκει σε έναν τομέα.

Το εννοιολογικό σχήμα της βάσης (E-R) εμφανίζεται στην εικόνα 13



Εικόνα 13 Εννοιολογικό(E-R) διάγραμμα εφαρμογής

Στην Εικόνα 13 παρουσιάζονται οι οντότητες με τα χαρακτηριστικά τους. Ένας τομέας μπορεί να περιέχει πολλά σημεία ενδιαφέροντος (poi) ενώ ένα σημείο ενδιαφέροντος (poi) περιέχεται μόνο σε έναν τομέα.



Εικόνα 14 Λογικό διάγραμμα της βάσης δεδομένων

Στην Εικόνα 14 βλέπουμε τους πίνακες που αποτελούν τη βάση δεδομένων καθώς και τα πεδία από τα οποία αποτελούνται. Η βάση δεδομένων είναι μια γεωγραφική βάση δεδομένων διότι στον πίνακα που αποθηκεύονται τα σημεία ενδιαφέροντος (POI) εκτός από τα πεδία που περιγράφουν ένα γεωγραφικό χαρακτηριστικό όπως το (P\_id, S\_id, P\_name, P\_address, P\_telephone, P\_type, mail, website, description) υπάρχει και το πεδίο (P\_geom). Αυτό το πεδίο σε σχέση με τα άλλα πεδία του πίνακα βρίσκεται αποθηκευμένη η γεωγραφική πληροφορία (δηλαδή γεωγραφικές συντεταγμένες) για την τοποθεσία ενός σημείου ενδιαφέροντος στον χάρτη. Αντίστοιχα στον πίνακα τομέα (Sector) έχουμε

τα πεδία (S\_id, S\_name) και το πεδίο που έχει αποθηκευμένη γεωγραφική πληροφορία είναι το πεδίο (S\_geom).

## 4.3. Διαγράμματα UML

### 4.3.1. Διαγράμματα περιπτώσεων-use case diagrams

Τα διαγράμματα περιπτώσεων χρήσης αναπαριστούν τις λειτουργίες ενός συστήματος από την οπτική γωνία ενός χρήστη. Ακόμη βοηθάει να κατανοήσουμε τον γενικό σκοπό του συστήματος και μας καθιστά ικανούς να αντιληφθούμε το μέγεθος της συνολικής εφαρμογής. Οι περιπτώσεις χρήσης περιγράφουν τη συμπεριφορά ενός συστήματος από την οπτική γωνία ενός χρήστη. Επιτρέπουν τον ορισμό των ορίων του συστήματος και του περιβάλλοντος. Μια περίπτωση χρήσης αντιστοιχεί σε ένα συγκεκριμένο είδος χρήσης του συστήματος. Είναι μια εικόνα της λειτουργικότητας ενός συστήματος το οποίο ενεργοποιείται για να ανταποκριθεί σε έναν εξωτερικό ενεργοποιό (actor).

Το μοντέλο περιπτώσεων χρήσης περιλαμβάνει:

- τις ίδιες τις περιπτώσεις χρήσης, και
- τους ενεργοποιούς (actors)

Το σύνολο της λειτουργικότητας ενός δεδομένου συστήματος καθορίζεται διαμέσου της μελέτης των λειτουργικών απαιτήσεων κάθε ενεργοποιού. Αυτές εκφράζονται με τις περιπτώσεις χρήσης σε μορφή οικογενειακών διεπαφών. Οι ενεργοποιοί συμβολίζονται με ανθρωπάκια που ενεργοποιούν τις περιπτώσεις χρήσης. Οι περιπτώσεις χρήσης συμβολίζονται με ελλείψεις. Ένας ενεργοποιός αναπαριστά ένα ρόλο που παίζεται από ένα άτομο ή πράγμα που αλληλεπιδρά με το σύστημα. Οι ενεργοποιοί προσδιορίζονται παρατηρώντας τους άμεσους χρήστες του συστήματος (αυτούς που το χρησιμοποιούν και το συντηρούν). Επίσης παρατηρώντας κάθε άλλο σύστημα που αλληλεπιδρά με αυτό που αναπτύσσεται. Το φυσικό πρόσωπο μπορεί να παίξει το ρόλο πολλών ενεργοποιών, ενώ πολλοί άνθρωποι μπορεί να παίξουν τον ίδιο ρόλο. Το όνομα του ενεργοποιού περιγράφει το ρόλο που παίζει ο χρήστης.

Ακόμη μια περίπτωση χρήσης είναι δυνατόν να χρησιμοποιεί (uses) μια άλλη περίπτωση χρήσης ή να επεκτείνεται (extends) σε αυτή.

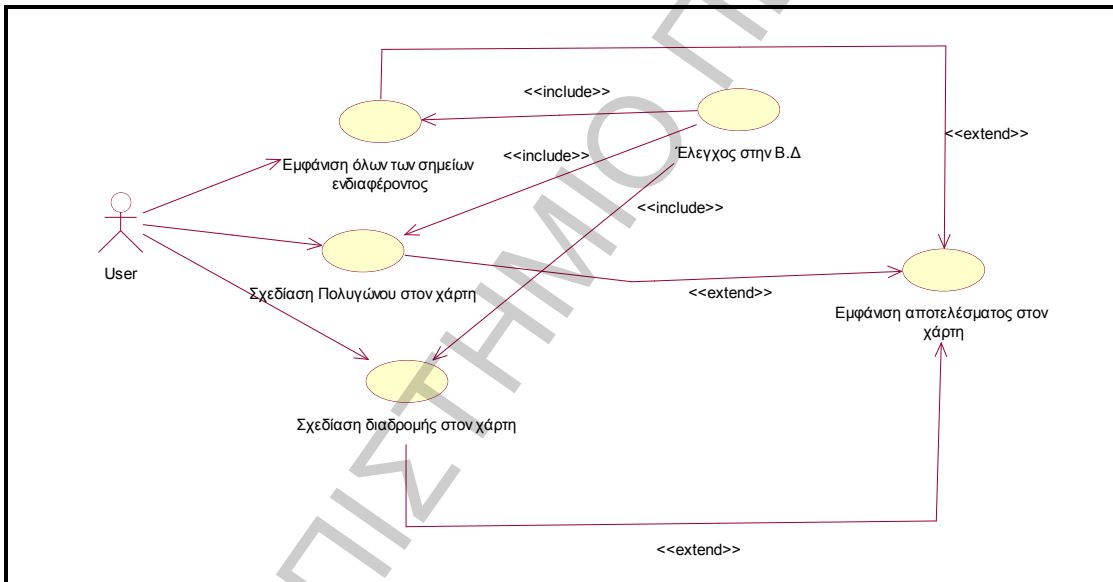
Η σχέση use ορίζεται μεταξύ περιπτώσεων χρήσης και σημαίνει ότι ένα στιγμότυπο της πηγής συμπεριλαμβάνει τη συμπεριφορά του στόχου. Η σχέση uses χρησιμοποιείται σε περιπτώσεις όπου περισσότερες από μία περιπτώσεις χρήσεις μοιράζονται στοιχεία από την ίδια λειτουργικότητα. Αυτή η λειτουργικότητα τοποθετείται σε μια ξεχωριστή περίπτωση χρήσης αντί να τεκμηριώνεται σε κάθε περίπτωση χρήσης που τη χρειάζεται. Οι σχέσεις uses δημιουργούνται μεταξύ της νέας περίπτωσης χρήσης και κάθε άλλης που χρησιμοποιεί τη λειτουργικότητά της.

Η σχέση extend μεταξύ δύο περιπτώσεων χρήσης σημαίνει ότι η πηγή επεκτείνει τη συμπεριφορά του στόχου. Μια σχέση extends χρησιμοποιείται για να δείξει:

- προαιρετική συμπεριφορά
- συμπεριφορά που υπάρχει μόνο κάτω από ορισμένες συνθήκες
- αρκετές διαφορετικές ροές που μπορεί να υπάρχουν βασισμένες σε επιλογές ενεργοποιών.

Η UML χρησιμοποιεί την έννοια στιγμιότυπο, η οποία παρέχει τη δυνατότητα να επεκτείνονται τα βασικά στοιχεία μοντελοποίησης έτσι ώστε να δημιουργούν νέα στοιχεία. Το πλεονέκτημα της έννοιας αυτής είναι ότι επιτρέπει στη UML να έχει ένα μικρό σύνολο στοιχείων τα οποία μπορούν να επεκταθούν ανάλογα με τις ανάγκες του σχεδιασμού. Τα ονόματα των στερεοτύπων μπαίνουν σε ελληνικά εισαγωγικά («»). Τα στερεότυπα χρησιμοποιούνται για να δημιουργήσουν τις σχέσεις των use cases. Οι σχέσεις uses και extends πρέπει υποχρεωτικά να χρησιμοποιούν στερεότυπα εφόσον και τα δύο αναπαριστώνται από το τόξο της γενίκευσης.

Στην εικόνα 15 παρουσιάζεται το διάγραμμα περίπτωσης χρήσης για την εφαρμογή.



Εικόνα 15 Διάγραμμα περίπτωσης χρήσης-use case diagrams

Στην Εικόνα 15 απεικονίζονται οι λειτουργίες που μπορεί να κάνει ο χρήστης της εφαρμογής. Η πρώτη περίπτωση είναι να δει στο χάρτη τα σημεία ενδιαφέροντος που έχει επιλέξει. Όταν επιλέξει να τα δει τότε η εφαρμογή επικοινωνεί με τη βάση δεδομένων, η οποία του επιστρέφει τα σημεία ενδιαφέροντος και η εφαρμογή αναλαμβάνει να τα οπτικοποιήσει στο χάρτη. Η δεύτερη περίπτωση είναι να σχεδιάσει ένα πολύγωνο στο χάρτη. Αυτό το πολύγωνο αποθηκεύεται και στη συνέχεια γίνεται ο έλεγχος στη βάση δεδομένων για την εμφάνιση των σημείων ενδιαφέροντος που υπάρχουν στο πολύγωνο που έχει οριστεί από το χρήστη και αν υπάρχουν εμφανίζονται στο χάρτη. Η τρίτη λειτουργία είναι ο χρήστης να επιλέξει μια διαδρομή. Η διαδρομή αποθηκεύεται και γίνεται έλεγχος στη βάση δεδομένων για να επιστρέψει τα σημεία τα οποία βρίσκονται κοντά σε αυτήν τη διαδρομή. Τα δεδομένα που έχουν να κάνουν με τις γεωγραφικές συντεταγμένες και την απεικονισή τους στο χάρτη

*Διαδικτυακή πλατφόρμα πολυεπίπεδης αρχιτεκτονικής για την πραγματοποίηση δυναμικών αναζητήσεων σε γεωπληροφοριακό σύστημα*

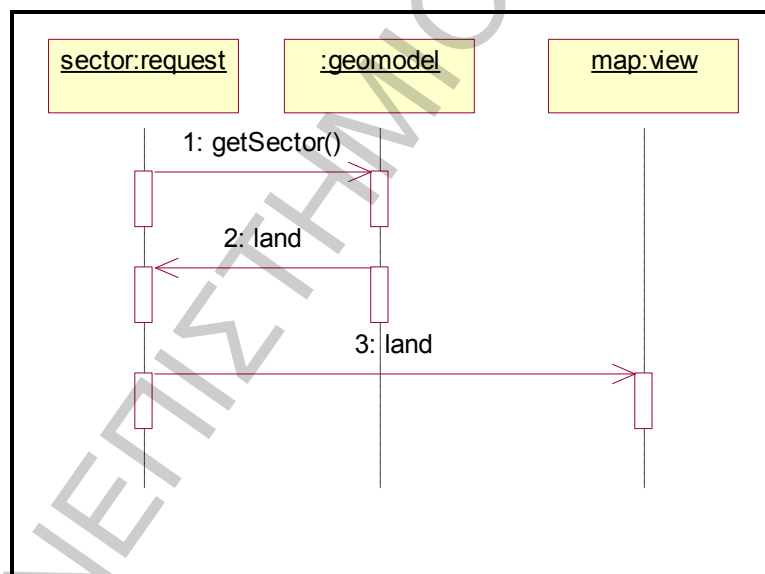
είναι αποθηκευμένα στη βάση δεδομένων με χωρικούς τύπους δεδομένων και για την επιστροφή των αποτελεσμάτων στο χάρτη εκτελούνται χωρικά ερωτήματα (κώδικας στο παράρτημα στο αρχείο geo\_model.php).

#### 4.3.2. Διαγράμματα σειράς-ακολουθίας diagrams

Τα διαγράμματα Σειράς απεικονίζουν τη σειρά των ενεργειών ενός σεναρίου στον άξονα του χρόνου. Εξηγούν ποιος δράστης εκκινεί μια ενέργεια και με ποιον δράστη συνεργάζεται.

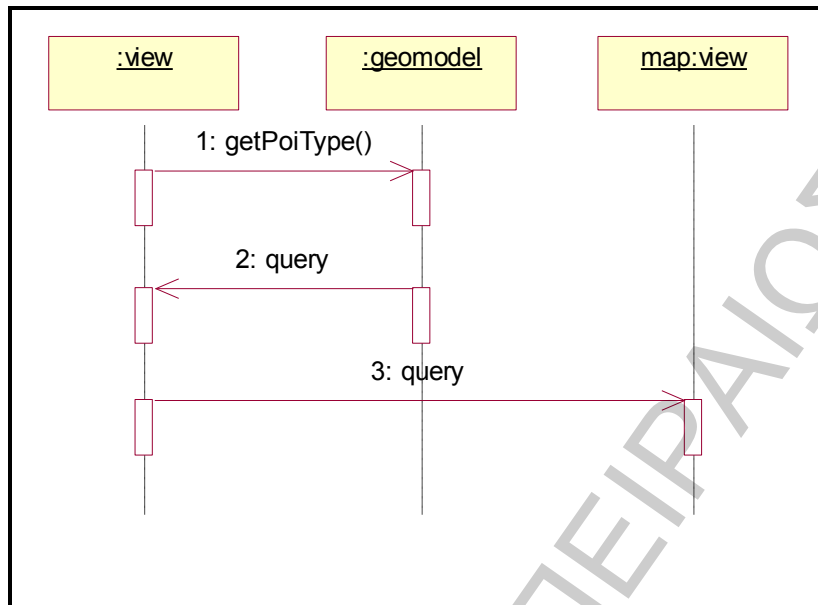
Σε κάθε αντικείμενο αντιστοιχεί μία κάθετη γραμμή που ονομάζεται γραμμή ζωής (lifeline). Για όσο χρόνο ένα αντικείμενο υφίσταται, η γραμμή αυτή είναι διακεκομμένη ενώ για όσο χρόνο μία διαδικασία του εν λόγω αντικειμένου είναι ενεργή, η γραμμή ζωής σχεδιάζεται ως μία διπλή γραμμή.

Ένα μήνυμα συμβολίζεται ως μία ακμή από τη γραμμή ζωής ενός αντικειμένου προς τη γραμμή ζωής ενός άλλου. Η θέση των ακμών αντιστοιχεί στην τοποθέτησή τους σε σχέση με τον χρόνο (μία ακμή χαμηλότερα από μία άλλη αντιστοιχεί σε μεταγενέστερο χρόνο). Οι απαντήσεις σε μηνύματα υποδηλώνονται ως οριζόντιες διακεκομμένες ακμές.



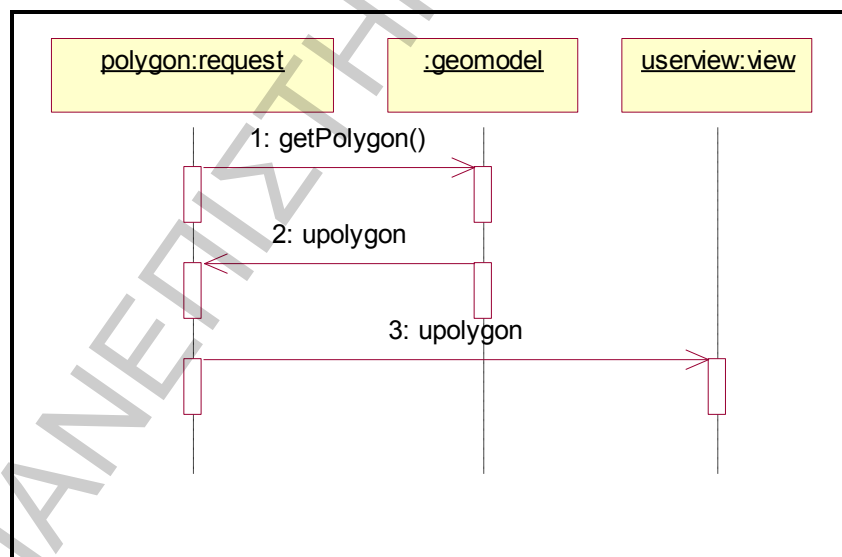
Εικόνα 16 Διάγραμμα σειράς land

Στην Εικόνα 16 η συνάρτηση sector της κλάσης request επικοινωνεί με την κλάση geomodel μέσω της συνάρτησης getSector. Το μοντέλο επιστρέφει ένα αντικείμενο land το οποίο στη συνέχεια αποστέλλεται στη συνάρτηση map της κλάσης view.



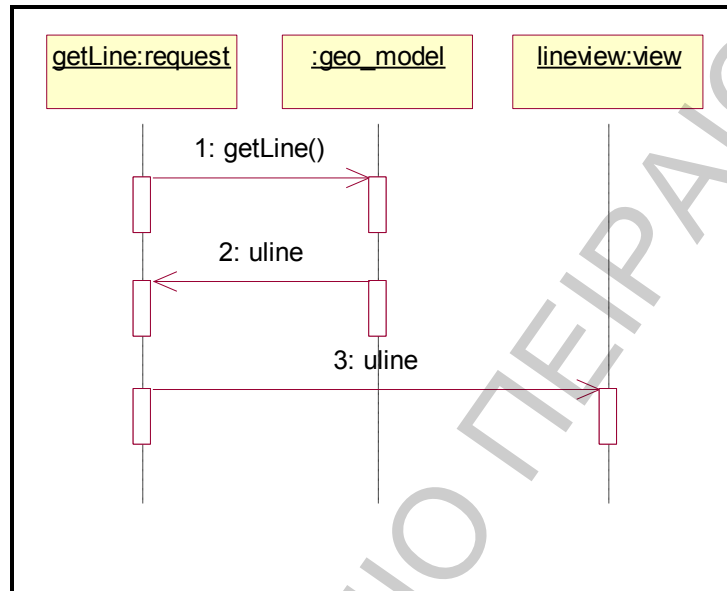
Εικόνα 17 Διάγραμμα σειράς query

Στην εικόνα 17 η κλάση view επικοινωνεί με την κλάση geomodel μέσω της συνάρτησης getPoiType. Το μοντέλο επιστρέφει ένα αντικείμενο query το οποίο στη συνέχεια αποστέλλεται στη συνάρτηση map της κλάσης view.



Εικόνα 18 Διάγραμμα σειράς upolygon

Στην εικόνα 18 η συνάρτηση polygon της κλάσης request επικοινωνεί με την κλάση geomodel μέσω της συνάρτησης getPolygon. Το μοντέλο επιστρέφει ένα αντικείμενο upolygon το οποίο στη συνέχεια αποστέλλεται στη συνάρτηση userview της κλάσης view.

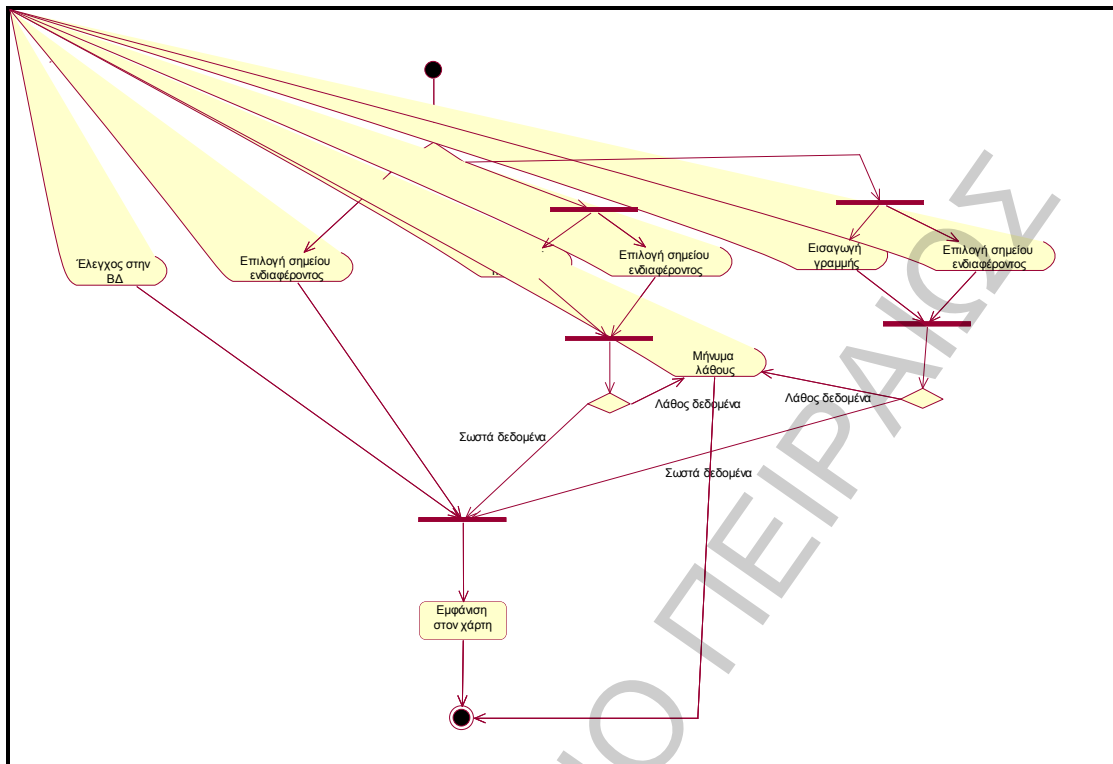


Εικόνα 19 Διάγραμμα σειράς uline

Στην εικόνα 19 η συνάρτηση getLine της κλάσης request επικοινωνεί με την κλάση geomodel μέσω της συνάρτησης getLine. Το μοντέλο επιστρέφει ένα αντικείμενο uline το οποίο στη συνέχεια αποστέλλεται στη συνάρτηση lineview της κλάσης view.

#### 4.3.3. Διαγράμματα δραστηριοτήτων-activity diagrams

Το διάγραμμα δραστηριοτήτων είναι «παραλλαγή» του διαγράμματος καταστάσεων. Στο διάγραμμα δραστηριοτήτων οι κόμβοι αναπαριστούν καταστάσεις ενεργειών και οι μεταβάσεις λαμβάνουν χώρα με την ολοκλήρωση αυτών των ενεργειών και όχι ως συνέπεια ενός συμβάντος (όπως στο διάγραμμα καταστάσεων). Αναπαριστούν τη ροή ενεργειών (ροή εργασίας - workflow) που εκτελούνται στο σύστημα, ποιες δραστηριότητες μπορεί να γίνουν παράλληλα και πότε υπάρχουν εναλλακτικές διαδρομές: συνθήκες (conditions), σημεία αποφάσεων (decisions) και παράλληλες εκτελέσεις ενεργειών. Επιπλέον καταγράφουν μια συγκεκριμένη επιχειρησιακή διαδικασία ή τη δυναμική μιας ομάδας αντικειμένων. Κύριο στοιχείο τους είναι οι δραστηριότητες (activities).



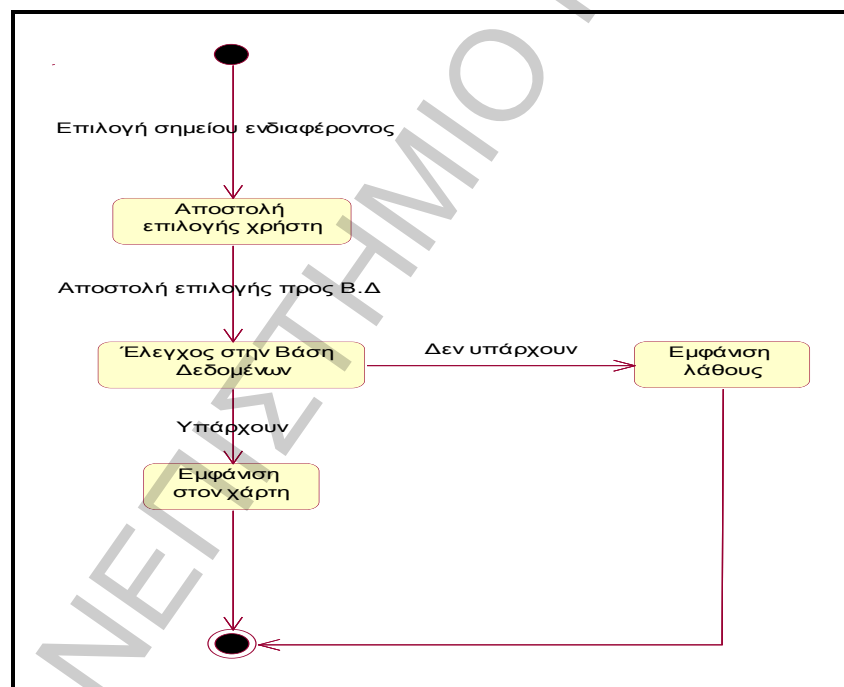
Εικόνα 20 Διάγραμμα δραστηριοτήτων

Στην εικόνα 20 απεικονίζονται οι λειτουργίες που προσφέρει το πρόγραμμα στο χρήστη. Στην πρώτη περίπτωση ο χρήστης επιλέγει ένα σημείο ενδιαφέροντος και γίνεται έλεγχος στη βάση δεδομένων. Εάν υπάρχουν αποτελέσματα τότε εμφανίζονται στο χάρτη, ενώ εάν δεν υπάρχουν τότε εμφανίζεται μήνυμα λάθους. Στη δεύτερη λειτουργία ο χρήστης «σχεδιάζει» ένα πολύγωνο στον χάρτη, και επιλέγει το είδος του σημείου ενδιαφέροντος που θέλει να βρει στο πολύγωνο. Μετά γίνεται έλεγχος στη βάση. Εάν υπάρχουν αποτελέσματα εμφανίζονται στον χάρτη αλλιώς εμφανίζεται μήνυμα λάθους. Η τρίτη λειτουργία είναι η εύρεση σημείων ενδιαφέροντος κατά μήκος μιας διαδρομής. Ο χρήστης σχεδιάζει μια διαδρομή στον χάρτη και το πρόγραμμα επιστρέφει σημεία ενδιαφέροντος κατά μήκος αυτής της διαδρομής.

#### 4.3.4. Διαγράμματα καταστάσεων-state diagrams

Τα διαγράμματα καταστάσεων παρουσιάζουν όλες τις πιθανές καταστάσεις που μπορεί να έχει ένα συγκεκριμένο αντικείμενο (σε όλη τη διάρκεια ύπαρξής του) και πώς αυτές αλλάζουν ανάλογα με τα γεγονότα (events) που φθάνουν στο αντικείμενο. Ένα διάγραμμα καταστάσεων αφορά μόνο μία κλάση. Συνήθως δεν σχεδιάζουμε διαγράμματα κατάστασης για όλες τις κλάσεις σε ένα σύστημα αλλά μόνο για εκείνες που παρουσιάζουν μία έντονα δυναμική συμπεριφορά. Τα διαγράμματα αυτά παρουσιάζουν τις διάφορες καταστάσεις και τους τρόπους αλλαγών της κατάστασης ενός αντικειμένου. Τα βασικά στοιχεία που ορίζονται στα διαγράμματα κατάστασης είναι:

- Καταστάσεις ενός αντικειμένου σε διάφορες χρονικές στιγμές
- Ειδικούς συμβολισμούς έχουν η αρχική και τελική κατάσταση ενός αντικειμένου
- Μεταβάσεις μεταξύ των καταστάσεων ενός αντικειμένου
- Συνθήκες μετάβασης μεταξύ των καταστάσεων
- Γεγονότα και ενέργειες που προκαλούν τις μεταβάσεις
- Ομαδοποίηση καταστάσεων προς σύνθεση μεγαλύτερων



Εικόνα 21 Διάγραμμα καταστάσεων

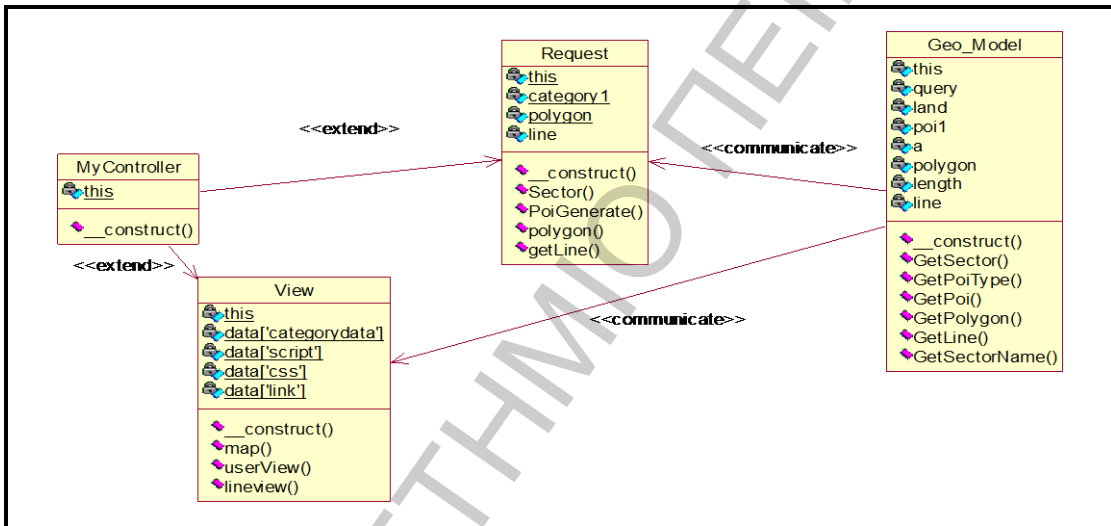
Στην Εικόνα 21 απεικονίζονται οι καταστάσεις του προγράμματος όταν ο χρήστης θέλει να δει τα συγκεκριμένα σημεία ενδιαφέροντος στον χάρτη. Αρχικά επιλέγεται το σημείο ενδιαφέροντος που θέλει να δει ο χρήστης, αποστέλλεται στη βάση δεδομένων και γίνεται έλεγχος εάν υπάρχουν τα



συγκεκριμένα σημεία ενδιαφέροντος. Εάν υπάρχουν, απεικονίζονται στον χάρτη αλλιώς εμφανίζεται μήνυμα λάθους που ενημερώνει στον χρήστη ότι δεν υπάρχουν.

#### 4.3.5. Διαγράμματα τάξεων-class diagrams

Τα διαγράμματα τάξεων αναπαριστούν τη στατική δομή όσον αφορά τις τάξεις και τις σχέσεις τους. Οι τάξεις αναπαρίστανται ως ορθογώνια τα οποία περιέχουν διαχωρισμούς. Το πρώτο τμήμα περιέχει το όνομα, το δεύτερο τα χαρακτηριστικά της τάξης και το τρίτο τις λειτουργίες της. Τα χαρακτηριστικά και οι μέθοδοι (λειτουργίες) της τάξης μπορεί να είναι δημόσια (ορατά σε όλους), προστατευόμενα (ορατά μόνο στις υποτάξεις της) είτε ιδιωτικά (ορατά δηλαδή μόνο στην ίδια την τάξη). Μερικά χαρακτηριστικά και λειτουργίες μπορεί να είναι ορατά σφαιρικά σε ολόκληρη την ισχύ της τάξης. Αυτά τα στοιχεία ονομάζονται μεταβλητές τάξης και λειτουργίες τάξης. Μια μεταβλητή τάξης μπορεί να μοιάζει σαν ένα αντικείμενο το οποίο μοιράζονται τα στιγμιότυπα μιας τάξης.



Εικόνα 22 Διάγραμμα τάξεων-class diagram

Στην εικόνα 22 απεικονίζονται οι τάξεις από τις οποίες αποτελείται η εφαρμογή. Στο διάγραμμα η τάξη Geo\_Model περιέχει όλες τις λειτουργίες που επικοινωνεί με τη βάση δεδομένων για την εκτέλεση των χωρικών ερωτημάτων της εφαρμογής. Η τάξη request περιέχει τις λειτουργίες εκείνες για την αποστολή των δεδομένων από και προς της βάση δεδομένων, καθώς και την αποστολή των αποτελεσμάτων στον χάρτη για την απεικόνιση τους. Η τάξη view έχει τις λειτουργίες για την εμφάνιση των αποτελεσμάτων στον περιηγητή και κατ'επέκταση στον χρήστη.

#### 4.3.6. Διαγράμματα αντικειμένων-object diagrams

Τα διαγράμματα αντικειμένων αναπαριστούν αντικείμενα και τις σχέσεις τους. Κάθε αντικείμενο αναπαρίσταται με ένα ορθογώνιο, το οποίο περιέχει είτε το όνομα και την τάξη του αντικειμένου (χωρισμένα με άνω και κάτω τελεία), ή μόνο την τάξη του αντικειμένου (σε αυτή την περίπτωση το αντικείμενο θεωρείται ότι είναι ανώνυμο). Το όνομα από μόνο του αντιστοιχεί σε ένα ελλειπές μοντέλο, στο οποίο η τάξη του αντικειμένου δεν έχει καθοριστεί ακόμα. Η τάξη από μόνη της αποφεύγει την εισαγωγή μη απαραίτητων ονομάτων στα διαγράμματα, ενώ επιτρέπει την έκφραση γενικών μηχανισμών που είναι έγκυροι για πολλά αντικείμενα. Το όνομα της τάξης μπορεί να περιέχει ολόκληρο το μονοπάτι που δημιουργείται από ονόματα διαφόρων εμπειροχόμενων πακέτων χωρισμένα με άνω και κάτω τελεία.

Η εφαρμογή αποτελείται από τις κλάσεις του ελεγκτή, του μοντέλου, και της παρουσίασης. Κάθε κλάση περιέχει τις δικές της συναρτήσεις. Για να επικοινωνήσουν οι κλάσεις της εφαρμογής έτσι ώστε να εκτελεστούν οι λειτουργίες του προγράμματος χρησιμοποιούνται αντικείμενα, και συγκεκριμένα χρησιμοποιούνται τα αντικείμενα που έχουν την μορφή τύπου JSON για την επεξεργασία τους.



Εικόνα 23 Αντικείμενο land

Στην εικόνα 23 η πληροφορία του αντικειμένου land από την κλάση geomodel επικοινωνεί με το αντικείμενο land στην κλάση request.



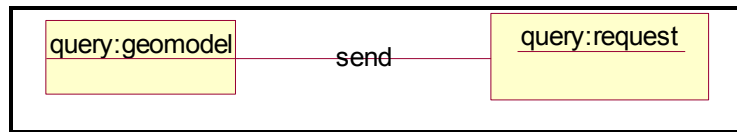
Εικόνα 24 Αντικείμενο poi1

Στην εικόνα 24 η πληροφορία του αντικειμένου poi1 από την κλάση geomodel επικοινωνεί με το αντικείμενο poi1 στην κλάση request.



Εικόνα 25 Αντικείμενο upolygon

Στην εικόνα 25 η πληροφορία του αντικειμένου upolygon από την κλάση geomodel επικοινωνεί με το αντικείμενο upolygon στην κλάση request.



Εικόνα 26 Αντικείμενο query

Στην εικόνα 26 η πληροφορία του αντικειμένου query από την κλάση geomodel μεταφέρεται σε ένα αντικείμενο query στην κλάση request.

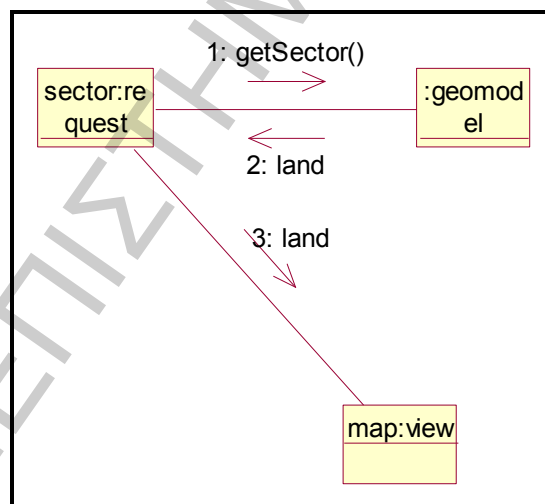


Εικόνα 27 Αντικείμενο uline

Στην εικόνα 27 η πληροφορία του αντικειμένου uline από την κλάση geomodel μεταφέρεται σε ένα αντικείμενο uline στην κλάση request.

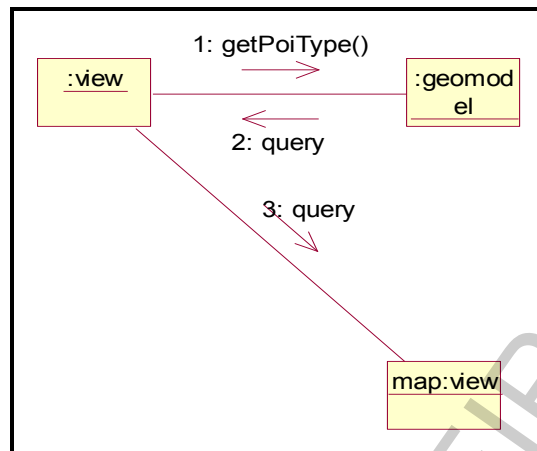
#### 4.3.7. Διαγράμματα συνεργασίας-collaboration diagrams

Τα διαγράμματα συνεργασίας είναι μια εναλλακτική αποτύπωση των διαγραμμάτων ακολουθίας. Αντί να διαρθρώσουμε τα μηνύματα κατά τον άξονα του χρόνου, τα τοποθετούμε πάνω σε ένα διάγραμμα αντικειμένων. Τα διαγράμματα αυτά δείχνουν την αλληλεπίδραση ανάμεσα στις κλάσεις για μια περίπτωση χρήσης, χωρίς να είναι διαρθρωμένα κατά τον άξονα του χρόνου. Όπως και στα διαγράμματα ακολουθίας, χρησιμοποιείται αρίθμηση για την απεικόνιση της ακολουθίας των μηνυμάτων.



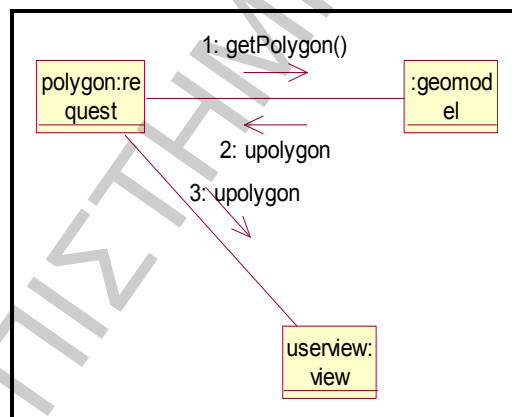
Εικόνα 28 Διάγραμμα συνεργασίας land

Στην εικόνα 28 η συνάρτηση sector της κλάσης request μέσω της λειτουργίας getSector επικοινωνεί με το geomodel για την επιστροφή του αντικειμένου land. Στη συνέχεια το αντικείμενο land μεταφέρεται στη συνάρτηση map της κλάσης view.



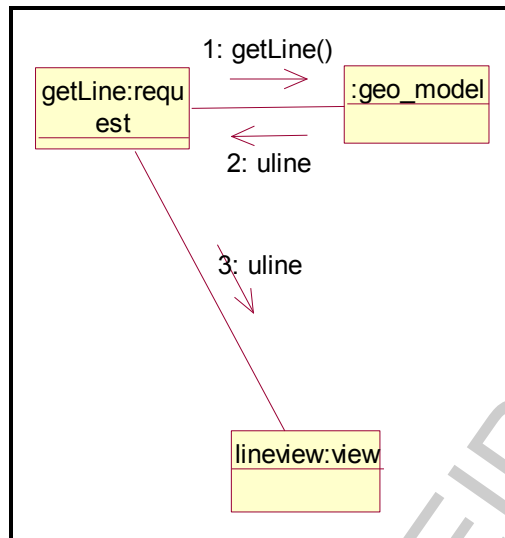
Εικόνα 29 Διάγραμμα συνεργασίας query

Στην εικόνα 29 η κλάση view μέσω της λειτουργίας `getPoiType` επικοινωνεί με το `geomodel` για την επιστροφή του αντικειμένου `query`. Στη συνέχεια το αντικείμενο `query` μεταφέρεται στη συνάρτηση `map` της κλάσης `view`.



Εικόνα 30 Διάγραμμα συνεργασίας upolygon

Στην εικόνα 30 η συνάρτηση `polygon` της κλάσης `request` μέσω της λειτουργίας `getPolygon` επικοινωνεί με το `geomodel` για την επιστροφή του αντικειμένου `upolygon`. Στη συνέχεια το αντικείμενο `upolygon` μεταφέρεται στη συνάρτηση `userview` της κλάσης `view`.



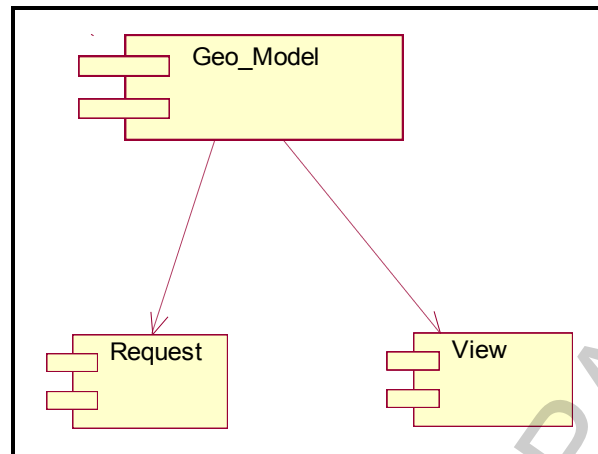
Εικόνα 31 Διάγραμμα συνεργασίας uline

Στην εικόνα 31 η συνάρτηση `getLine` της κλάσης `request` μέσω της λειτουργίας `getLine` επικοινωνεί με το `geomodel` για την επιστροφή του αντικειμένου `uline`. Στη συνέχεια το αντικείμενο `uline` μεταφέρεται στη συνάρτηση `lineview` της κλάσης `view`.

#### 4.3.8. Διαγράμματα εξαρτημάτων-component diagrams

Τα διαγράμματα εξαρτημάτων περιγράφουν τμήματα λογισμικού και τη σχέση τους με το περιβάλλον υλοποίησης. Υποδηλώνουν τις επιλογές που γίνονται κατά τη χρονική στιγμή υλοποίησης. Τα εξαρτήματα αναπαριστούν όλα τα είδη των στοιχείων που αναφέρονται στην τμηματοποίηση των εφαρμογών λογισμικού. Μεταξύ άλλων, αυτά μπορεί να είναι απλά αρχεία ή βιβλιοθήκες που διαβάζονται και χρησιμοποιούνται δυναμικά. Εξ ορισμού, κάθε τάξη στο λογικό μοντέλο έχει έναν προσδιορισμό και ένα κυρίως σώμα. Ο προσδιορισμός περιέχει το σύστημα διεπαφής της τάξης, η οποία αναπαρίσταται με τον συμβολισμό της UML με ένα μικρό κύκλο που συνδέεται μέσω μιας γραμμής με το σώμα της τάξης. Το σώμα περιέχει την υλοποίηση της ίδιας της τάξης και αναπαρίσταται με το ίδιο το σύμβολο του εξαρτήματος.

Στην εικόνα 32 απεικονίζεται το διάγραμμα εξαρτημάτων της εφαρμογής. Ουσιαστικά βασίζεται στο μοντέλο που περιγράφεται στην ενότητα του `codeigniter` δηλαδή του μοντέλου `mvc(model-view-controller)`. Έτσι έχουμε το μοντέλο που είναι το `geo_model` που συνδέεται με τη βάση δεδομένων και τους 2 controllers `request` και `view`. Ο `request` χειρίζεται τα αντικείμενα που του στέλνει το μοντέλο και ο `view` χειρίζεται αυτά που βλέπει ο χρήστης στην οθόνη του.

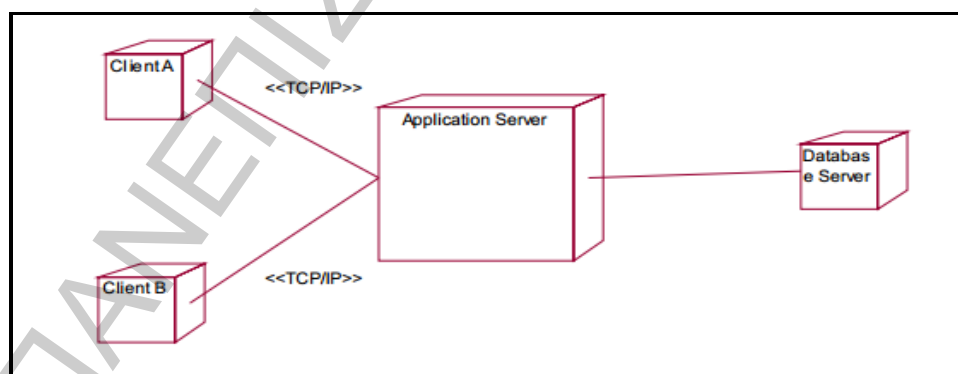


Εικόνα 32 Διάγραμμα εξαρτημάτων

#### 4.3.9. Διαγράμματα διανομής-deployment diagram

Τα διαγράμματα διανομής δείχνουν τη φυσική διάταξη των διαφόρων εξαρτημάτων (κόμβων) υλικού, που αποτελούν ένα σύστημα, όπως και τη διαμονή των εκτελέσιμων προγραμμάτων σε αυτό το υλικό. Κάθε σύστημα μπορεί να περιγραφεί με ένα μικρό αριθμό διαγραμμάτων διανομής. Πολλές φορές ένα μόνο διάγραμμα είναι αρκετό. Η φύση του εξοπλισμού μπορεί να καθοριστεί χρησιμοποιώντας ένα στερεότυπο. Οι διάφοροι κύβοι που εμφανίζονται στο διάγραμμα διανομής συνδέονται μεταξύ τους με απλές γραμμές που αναπαριστούν μία υποδομή επικοινωνίας, η οποία είναι διπλής κατεύθυνσης. Τα διαγράμματα διανομής μπορεί να δείχνουν τάξεις κόμβων ή στιγμιότυπα κόμβων.

Στην εικόνα 33 απεικονίζεται η λειτουργία της εφαρμογής. Υπάρχουν οι πελάτες που μέσω του πρωτοκόλλου TCP/IP συνδέονται στον εξυπηρετητή που είναι εγκατεστημένη η εφαρμογή. Ο εξυπηρετητής συνδέεται ξεχωριστά με τον εξυπηρετητή της βάσης δεδομένων που βρίσκονται αποθηκευμένα τα σημεία ενδιαφέροντος.

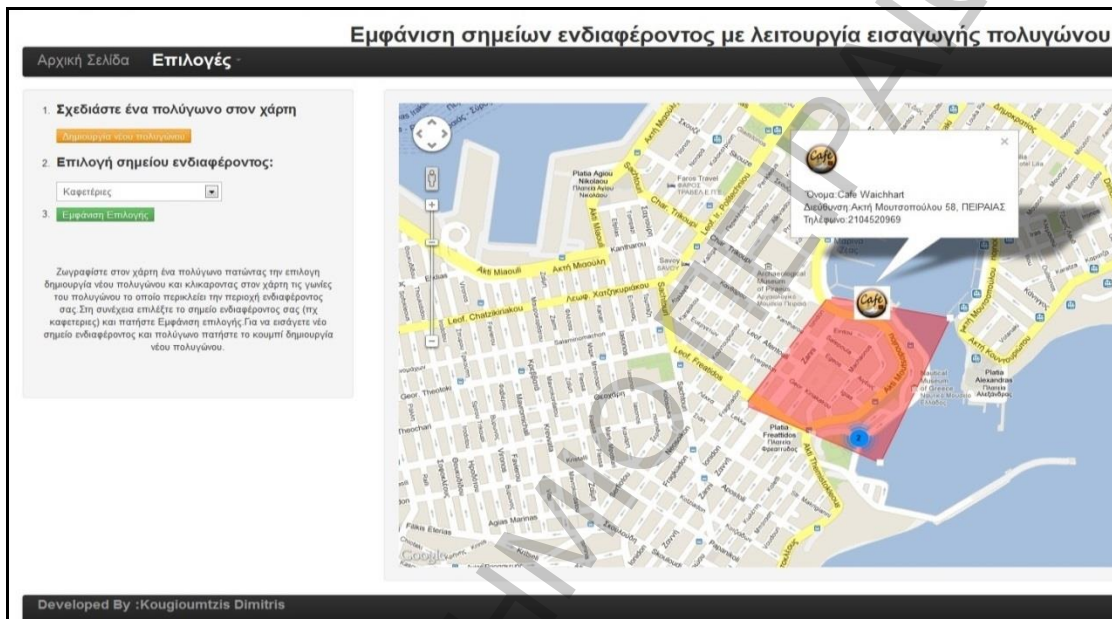


Εικόνα 33 Διάγραμμα Διανομής

*Διαδικτυακή πλατφόρμα πολυεπίπεδης αρχιτεκτονικής για την πραγματοποίηση δυναμικών αναζητήσεων σε γεωπληροφοριακό σύστημα*

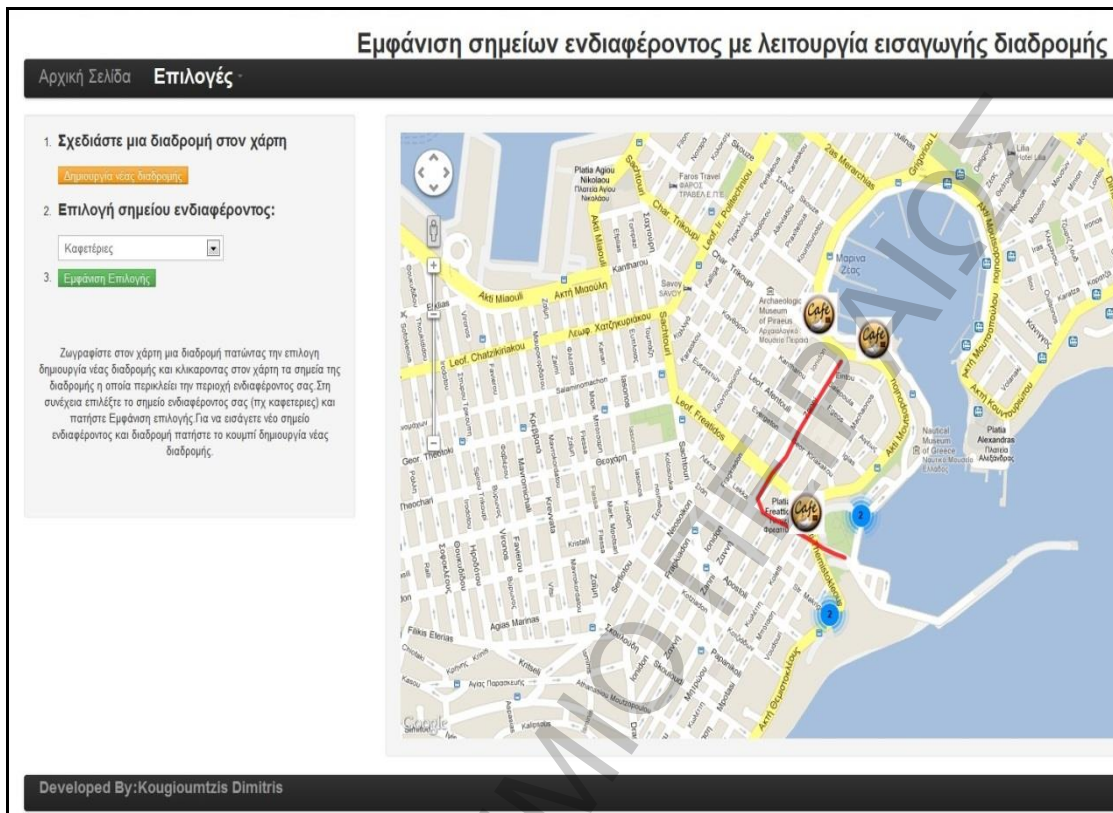
#### 4.4. Περιγραφή Λειτουργικότητας

Η εφαρμογή έχει ως σκοπό την αλληλεπίδραση του χρήστη με τον χάρτη για την εύρεση σημείων ενδιαφέροντος. Η αλληλεπίδραση γίνεται με 2 τρόπους. Ο πρώτος τρόπος είναι η δημιουργία μιας περιοχής (ενός πολυγώνου) που ορίζει ο χρήστης. Στη συνέχεια ο χρήστης επιλέγει ένα σημείο ενδιαφέροντος που επιθυμεί και το σύστημα επιστρέφει τα σημεία ενδιαφέροντος αν υπάρχουν στην περιοχή που έχει ορίσει. Ο δεύτερος τρόπος είναι η εισαγωγή γραμμής και η εμφάνιση σημείων ενδιαφέροντος που βρίσκονται κοντά στη γραμμή.



Εικόνα 34 Λειτουργία πολυγώνου

Στην εικόνα 34 ο χρήστης έχει επιλέξει ότι θέλει να βρει κάποιες καφετέριες. Στη συνέχεια έχει σχεδιάσει στον χάρτη ένα πολύγωνο που απεικονίζεται με το κόκκινο χρώμα, και στη συνέχεια το σύστημα του επιστρέφει τις καφετέριες που βρίσκονται μέσα στο πολύγωνο. Το μπλε εικονίδιο με τον αριθμό 2 σημαίνει ότι οι 2 καφετέριες που βρίσκονται κοντά έχουν ομαδοποιηθεί. Αυτό γίνεται για να μην υπάρχουν πολλά σημεία και επίσης για λόγους ευελιξίας στην χρήση του χάρτη.



Εικόνα 35 Λειτουργία διαδρομής

Στην εικόνα 35 ο χρήστης έχει επιλέξει ότι θέλει να βρει κάποιες καφετέριες. Στη συνέχεια έχει ζωγραφίσει στον χάρτη μια διαδρομή που απεικονίζεται με το κόκκινο χρώμα, και στη συνέχεια το σύστημα του επιστέφει τις καφετέριες που βρίσκονται κοντά στη διαδρομή. Το μπλε εικονίδιο με τον αριθμό 2 σημαίνει ότι 2 καφετέριες που βρίσκονται κοντά έχουν ομαδοποιηθεί αυτό γίνεται για να μην υπάρχουν πολλά σημεία και επίσης για λόγο ευελιξίας του χάρτη.



## 5. Συμπεράσματα

Ο σκοπός αυτής της εργασίας ήταν η μελέτη των τεχνολογιών του διαδικτύου και των τεχνολογιών των γεωγραφικών πληροφοριακών συστημάτων (ΓΠΣ). Το διαδίκτυο επιτρέπει τη δημιουργία εφαρμογών οι οποίες θα τρέχουν σε όλες τις πλατφόρμες επειδή εκτελούνται μέσα από έναν περιηγητή. Οι σύγχρονοι περιηγητές υποστηρίζουν όλα τα τελευταία πρότυπα έτσι ο χρήστης δε χρειάζεται να ανησυχεί για ασυμβατότητες. Στον τομέα των συστημάτων βάσεων δεδομένων τα ΓΠΣ επιτρέπουν την αποθήκευση και τη διαχείριση χωρικών δεδομένων. Με τη χρήση χωρικών ερωτημάτων μπορούμε να εξαγάγουμε τις γεωγραφικές πληροφορίες και να τις οπτικοποιήσουμε σε χάρτη. Με τις δύο αυτές τεχνολογίες δημιουργήθηκε η εφαρμογή που σαν σκοπό έχει τη δυναμική οπτικοποίηση ερωτημάτων που εκτελεί ο χρήστης.

Η μελέτη των τεχνολογιών των βάσεων δεδομένων και των τεχνολογιών του διαδικτύου έπαιξε σημαντικό ρόλο στην ανάπτυξη της εφαρμογής. Συγκεκριμένα η μελέτη στο επίπεδο του σχεσιακού μοντέλου των βάσεων δεδομένων ήταν καθοριστική. Από την υλοποίηση της εφαρμογής έγινε σαφές ότι η βάση δεδομένων πρέπει να είναι υλοποιημένη με τέτοια κριτήρια έτσι ώστε να γίνεται γρήγορη εκτέλεση των ερωτημάτων. Σε περίπτωση που μια βάση δεν είναι σωστά υλοποιημένη τότε η εκάστοτε εφαρμογή θα έχει χαμηλή απόκριση.

Η μελέτη των χωρικών βάσεων δεδομένων ήταν επίσης ένα σημαντικό τμήμα της εργασίας. Αυτό σε συνδυασμό με την τεχνολογία postGIS επέτρεψε τη δημιουργία χωρικών ερωτημάτων που θα εκτελούνταν με τέτοιο τρόπο έτσι ώστε η συνολική απόκριση της βάσης δεδομένων να είναι όσο το δυνατόν καλύτερη.

Στο επίπεδο των τεχνολογιών του διαδικτύου σημαντικό ρόλο έπαιξε η μελέτη του προγραμματιστικού μοντέλου Μοντέλο-Ελεγκτής-Παρουσίαση (MVC). Μέσα από την ενασχόληση με το συγκεκριμένο μοντέλο και με τη λεπτομερή κατανόησή του έγινε μοντελοποίηση για την αρχιτεκτονική της εφαρμογής και πώς θα γίνει εφικτή η διασύνδεση των τμημάτων αυτής. Με την ολοκλήρωση της εφαρμογής έγινε σαφές ότι το προγραμματιστικό μοντέλο δίνει τη δυνατότητα ανάπτυξης εφαρμογών που θα είναι χωρισμένες σε τμήματα. Με αυτό τον τρόπο ο προγραμματιστής θα μπορεί να αναπτύξει καινούργια τμήματα της εφαρμογής χωρίς να είναι αναγκασμένος να δει το πρόγραμμα από την αρχή. Επιπλέον, μπορεί να γίνει καλύτερη τεκμηρίωση του κώδικα ανά επίπεδο. Με τις γνώσεις χρήσης αυτού του μοντέλου χρησιμοποιήθηκε το πλαίσιο ανάπτυξης διαδικτυακών εφαρμογών codeigniter για την ανάπτυξη της εφαρμογής.

Για την σύνδεση των δεδομένων από και προς τη βάση δεδομένων χρησιμοποιήθηκαν τα αντικείμενα τύπου json. Η μελέτη τους κρίθηκε αναγκαία κατά τη διάρκεια υλοποίησης της εφαρμογής διότι ήταν ο τρόπος με τον οποίο διακινούνται τα δεδομένα στα τμήματα της εφαρμογής. Επέκταση αυτών των αντικειμένων είναι τα αντικείμενα τύπου geoJSON τα οποία μεταφέρουν γεωγραφική πληροφορία.

Η μελέτη της τεχνολογίας ajax σε συνδυασμό με τη βιβλιοθήκη jquery της javascript έδωσε τη δυνατότητα ανάπτυξης της εφαρμογής που θα είναι ευέλικτη στον περιηγητή του χρήστη. Αυτό είναι σημαντικό διότι όταν ο χρήστης επιλέγει να δει τα σημεία ενδιαφέροντος, το πρόγραμμα του περιηγητή θα τα εμφανίζει χωρίς να χρειάζεται να ξανά-φορτώνει η σελίδα.

Το τελευταίο στάδιο της μελέτης ήταν η μελέτη του προγραμματιστικού μοντέλου της google για τους χάρτες. Ήταν σημαντικό τόσο για την αλληλεπίδραση του χρήστη πάνω στον χάρτη όσο και για την οπτικοποίηση των δεδομένων πάνω στον χάρτη.

Ένας περιορισμός στην υλοποίηση της εφαρμογής ήταν η εύρεση πραγματικών δεδομένων για τα σημεία ενδιαφέροντος. Αυτό περιόρισε το πλήθος των δεδομένων που αποθηκεύτηκαν στη βάση δεδομένων. Επίσης μια άλλη δυσκολία που ξεπεράστηκε κατά τη διάρκεια της υλοποίησης ήταν πώς το πολύγωνο που εισάγει ο χρήστης στον χάρτη θα ενσωματωθεί στο χωρικό ερώτημα που τρέχει το μοντέλο στη βάση δεδομένων για να επιστρέψει τα αποτελέσματα.

Μέσα από την εμπειρία που αποκτήθηκε κατά τη διάρκεια της υλοποίησης θα ήταν εφικτό να βελτιωθεί η εφαρμογή. Αρχικά θα μπορούσε ο πίνακας που αποθηκεύεται ο τομέας (sector) να περιλαμβάνει περισσότερες πόλεις, επίσης ο πίνακας των σημείων ενδιαφέροντος (poi) να περιλαμβάνει σημεία ενδιαφέροντος για κάθε πόλη ξεχωριστά. Έτσι με τη δημιουργία μιας βάσης δεδομένων ο χρήστης θα έχει περισσότερες επιλογές στην λειτουργικότητα του προγράμματος, πώς π.χ θα είναι εφικτό να ψάχνει σημεία ενδιαφέροντος στην πόλη που θέλει. Με αυτόν τον τρόπο εμπλουτίζοντας περισσότερο τη βάση δεδομένων θα ήταν δυνατόν να υλοποιηθεί μια εφαρμογή που θα έτρεχε σε κινητές συσκευές και ανάλογα με την τοποθεσία του χρήστη να βρίσκει τα σημεία ενδιαφέροντος που επιθυμεί.

## 6. Παράρτημα

Παρακάτω παρουσιάζεται ο κώδικας της εφαρμογής. Η σειρά με την οποία παρουσιάζεται είναι η έξης:

- Geo\_model.php
- Request.php
- View.php
- Map.js
- polygon.js
- Line.js
- Map.php
- User.php
- Line.php

```
<?php
class Geo_model extends CI_Model {
    public function __construct()
    {
        $this->load->database();
    }
    public function getSector()
    {
        header('Content-type: application/json');
        $query = $this->db->query('select st_asgeojson(st_centroid(s_geom)) as
center,st_asgeojson(s_geom) as sector from services.sector ');
        $land=array();
        foreach ($query->result() as $row)
        {
            array_push($land,json_decode($row->sector,true));
            array_push($land,json_decode($row->center,true));
        }
        echo json_encode($land);
    }
    public function getPoiType()
    {
        $query = $this->db->query('SELECT distinct p_type from services.poi
;');
        return $query->result_array();
    }
    public function getPoi($poi)
    {
        header('Content-type: application/json');
        $query = $this->db->query('select
p.p name,p.p type,st_asgeojson(s.s geom) as sector,st_asgeojson(p.p geom) as
```

*Διαδικτυακή πλατφόρμα πολυεπίπεδης αρχιτεκτονικής για την πραγματοποίηση δυναμικών αναζητήσεων σε γεωπληροφορικό σύστημα*

```

poi,p.p_address,p.p_telephone
from services.poi
as p,services.sector as s
where
s.s_id=p.s_id and p.p_type='\'' . $poi .'\'';');
foreach ($query->result() as $row)
{
    $poi[] = array('name'=>$row->p_name, 'type'=>$row-
>p_type, 'sector'=>json_decode($row->sector, true), 'poi'=>json_decode($row-
>poi, true), 'address'=>$row->p_address, 'telephone'=>$row->p_telephone);
}
echo json_encode($poi);
}
public function getPolygon($user,$category) {
header('Content-type: application/json');
$polygon="";
$length=count($user);
for ($i=0;$i<$length;$i++){
    $polygon.=$user[$i].', ';
}
$polygon.=$user[0];
$query = $this->db->query('select
p.p_name,p.p_type,st_asgeojson(p.p_geom) as poi,p.p_address,p.p_telephone
from services.poi
as p,services.sector as s
where
s.s_id=p.s_id and st_contains(st_geomfromtext('\''POLYGON((' . $polygon
.'))\'' ,4326),p.p_geom) and p.p_type='\'' . $category .'\'';');
foreach ($query->result() as $row)
{
    $polygon[] = array('name'=>$row->p_name, 'type'=>$row-
>p_type, 'poi'=>json_decode($row->poi, true), 'address'=>$row-
>p_address, 'telephone'=>$row->p_telephone);
}
echo json_encode($polygon);
}
public function getLine($user,$category) {
header('Content-type: application/json');

$line="";
$uline="";
$length=count($user);
for ($i=0;$i<$length-1;$i++){

    $line.=$user[$i].', ';
}
$line.=$user[$length-1];
$query = $this->db->query('select
p.p_name,p.p_type,st_asgeojson(p.p_geom) as poi,p.p_address,p.p_telephone
from services.poi as p,services.sector as s where
s.s_id=p.s_id and st_contains(st_buffer(st_geomfromtext('\''LINESTRING(' . $line
.'))\'' ,4326),0.002),p.p_geom) and p.p_type='\'' . $category .'\'';');
foreach ($query->result() as $row)
{
    $uline[] = array('name'=>$row->p_name, 'type'=>$row-
>p_type, 'poi'=>json_decode($row->poi, true), 'address'=>$row-
>p_address, 'telephone'=>$row->p_telephone);
}

echo json_encode($uline);
}
}

```

Εικόνα 36 Κώδικας geo\_model.php

```

<?php
if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Request extends MY_Controller {
    public function __construct()
    {
        parent::__construct();
        $this->load->model('geo_model');
    }
    public function sector() {
        $this->geo_model->getSector();
    }
    public function poiGenerate() {
        $category1=$this->input->post('poi');
        $this->geo_model->getPoi($category1);
    }
    public function polygon() {
        header('Content-type: application/json');
        $polygon=$this->input->post('user');
        $category1=$this->input->post('poi');
        $this->geo_model->getPolygon($polygon,$category1);
    }
    public function getLine() {
        header('Content-type: application/json');
        $line=$this->input->post('user');
        $category1=$this->input->post('poi');
        $this->geo_model->getLine($line,$category1);
    }
}

```

Εικόνα 37 Κώδικας request.php

```

<?php
if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Request extends MY_Controller {
    public function __construct()
    {
        parent::__construct();
        $this->load->model('geo_model');
    }
    public function sector() {
        $this->geo_model->getSector();
    }
    public function poiGenerate() {
        $category1=$this->input->post('poi');
        $this->geo_model->getPoi($category1);
    }
    public function polygon() {
        header('Content-type: application/json');
        $polygon=$this->input->post('user');
        $category1=$this->input->post('poi');
        $this->geo_model->getPolygon($polygon,$category1);
    }
    public function getLine() {
        header('Content-type: application/json');
        $line=$this->input->post('user');
        $category1=$this->input->post('poi');
        $this->geo_model->getLine($line,$category1);
    }
}

```

Εικόνα 38 Κώδικας view.php

```

$(document).ready(function() {
    var mc=null;
    var infowindow = new google.maps.InfoWindow();
    var markers=[];
    $.ajax({
        url: 'http://localhost/findpoi/public/request/sector',
        type: 'post',
        datatype: 'json',
        success: function(data) {
            var mapOptions = {
                center: new
google.maps.LatLng(data[1].coordinates[1], data[1].coordinates[0]),
                zoom: 15,
                mapTypeId: google.maps.MapTypeId.ROADMAP
            };
            var map = new
google.maps.Map(document.getElementById("map_canvas"), mapOptions);
            var land=[];
            for(i=0;i<data[0].coordinates[0].length;i++){
                land.push(new
google.maps.LatLng(data[0].coordinates[0][i][1], data[0].coordinates[0][i][0]));
            }
            var land1;
            land1 = new google.maps.Polygon({
                paths: land,
                strokeColor: "#FF0000",
                strokeOpacity: 0.8,
                strokeWeight: 1,
                fillColor: "#FF0000",
                fillOpacity: 0
            });
            land1.setMap(map);
            var options2 = {
                url:
'http://localhost/findpoi/public/request/poiGenerate',
                type: 'post',
                datatype: 'json',
                success: showResponse2
            };
            $('#myform').submit(function() {
                var selection=$('#myselect').val();
                if (selection=='1'){
                    alert('Δεν έχετε επιλέξει σημείο
ενδιαφέροντος');
                }
                else
                {
                    $(this).ajaxSubmit(options2);
                    return false;
                }
            });
            function showResponse2(data) {
                if(mc){
                    mc.clearMarkers();
                    markers.length=0;
                }
                var marker, i;
                for (i = 0; i < data.length; i++) {
                    var image;
                    if
                    (data[i].type=='Καφετέριας'){

```

Διαδικτυακή πλατφόρμα πολυεπίπεδης αρχιτεκτονικής για την πραγματοποίηση δυναμικών αναζητήσεων σε γεωπληροφοριακό σύστημα

```

var image='http://localhost/findpoi/public/img/cafe.jpg';
};
if(data[i].type=='Φαρμακεία'){
var image='http://localhost/findpoi/public/img/pharmacy.jpg';
};
if(data[i].type=='Ξενοδοχεία'){
var image='http://localhost/findpoi/public/img/hotel.jpg';
};
if(data[i].type=='κλινικές-Νοσοκομεία'){
var image='http://localhost/findpoi/public/img/hospital.png';
};
if(data[i].type=='Μπυραρίες'){
var image='http://localhost/findpoi/public/img/beer.png';
};
marker = new google.maps.Marker({
    position: new google.maps.LatLng(data[i].poi.coordinates[1],
data[i].poi.coordinates[0]),
    map: map,
    icon:image
});
markers.push(marker);

google.maps.event.addListener(marker,
'click', (function(marker, i) {
    return function() {
        infowindow.close();
        infowindow.setContent("<br><br>Όνομα:"+data[i].name+"<br>"+"Διεύθυνση:"+data[i].address+"<br
>"+"Τηλέφωνο:"+data[i].telephone);

        infowindow.setPosition(marker.latLng);
        infowindow.open(map, marker);
    };
})(marker, i));
});
mc = new MarkerClusterer(map, markers);
});
});

```

Εικόνα 39 Κώδικας map.js

```

$(document).ready(function() {
    var mc=null;
    $("#new").click(function() {
        location.reload();
        if (mc) {
            mc.clearMarkers();
        }
    });
    var polygon=new Array();
    var infowindow = new google.maps.InfoWindow();
    var markers=[];
    var shape;
    var a;
    var obj = {};

    var x = new Array();
    var polygon2 = new Array();

    $.ajax({
        url: 'http://localhost/findpoi/public/request/sector',
        type: 'post',

```

*Διαδικτυακή πλατφόρμα πολυεπίπεδης αρχιτεκτονικής για την πραγματοποίηση δυναμικών αναζητήσεων σε γεωπληροφορικό σύστημα*

```

        datatype: 'json',
        success: function(data) {
            var mapOptions = {
                center: new
google.maps.LatLng(37.934185,23.648375),
                zoom: 16,
                mapTypeId: google.maps.MapTypeId.ROADMAP
            };
            var map = new
google.maps.Map(document.getElementById("map_canvas"), mapOptions);
            var markerCluster = new MarkerClusterer(map, markers);
            var land=[];
            for(i=0;i<data[0].coordinates[0].length;i++){
google.maps.LatLng(data[0].coordinates[0][i][1], data[0].coordinates[0][i][0]);
                land.push(new
            }
            var land1;
            land1 = new google.maps.Polygon({
                clickable:true,
                paths: land,
                strokeColor: "#FF0000",
                strokeOpacity: 0.8,
                strokeWeight: 1,
                fillColor: "#FF0000",
                fillOpacity: 0
            });
            land1.setMap(map);
            shape = new google.maps.Polygon({
                Editable:true,
                strokeColor: 'FF0000',
                strokeOpacity: 0.8,
                strokeWeight: 1,
                fillColor: 'FF0000',
                fillOpacity: 0.4
            });

            google.maps.event.addListener(land1,'mousemove',function(event) {
event.latLng.lng());
                $('#coord').val(event.latLng.lat() + ', ' +

            });
            shape.setMap(map);
            google.maps.event.addListener(map,'click',error);
            google.maps.event.addListener(land1,'click',addPoint);
            function error(){
                alert("Not drawing area");
            }
            function addPoint(e) {
                shape.setMap(map);

                $('#poly').val(e.latLng.lat() + ', ' +

                vertices= shape.getPath();
                vertices.push(e.latLng);
                polygon.push(e.latLng);
            }
            $("#polygon").click(function(e){
                a=JSON.stringify(polygon);
                var c=JSON.parse(a);
                for (i=0;i<c.length;i++){
                    x.push(c[i].Za+ " "+c[i].Ya)
                }
                $('#poly').val(a);
                var selection=$("#myselect").val();
                if (selection=='1' || a=="[]"){

```



```

σημείο ενδιαφέροντος');
alert('Σχεδίασε περιοχή ή επέλεξε
}
else{
$.ajax({
url:
'http://localhost/findpoi/public/request/polygon',
type: 'post',
data: {'poi':selection , 'user':x},
datatype: 'json',
success: function(data) {
google.maps.event.clearListeners(land1, 'click');
shape.setOptions({ Editable:
false });
if(mc){
mc.clearMarkers();
markers.length=0;
}
if (data.length=='0'){
alert('Δεν βρέθηκαν
αποτελέσματα');
}
else
{
for (i = 0; i <
data.length; i++) {
var image;
if
(data[i].type=='Καφετέριες'){
var image='http://localhost/findpoi/public/img/cafe.jpg';
};
if(data[i].type=='Φαρμακεία'){
var image='http://localhost/findpoi/public/img/pharmacy.jpg';
};
if(data[i].type=='Ξενοδοχεία'){
var image='http://localhost/findpoi/public/img/hotel.jpg';
};
if(data[i].type=='Κλινικές-Νοσοκομεία'){
var image='http://localhost/findpoi/public/img/hospital.png';
};
if(data[i].type=='Μπυραρίες'){
var image='http://localhost/findpoi/public/img/beer.png';
};
marker
= new google.maps.Marker({
position: new google.maps.LatLng(data[i].poi.coordinates[1],
data[i].poi.coordinates[0]),

```



```

        zoom: 16,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    });
    var map = new
google.maps.Map(document.getElementById("map_canvas"), mapOptions);
    var land=[];
    for(i=0;i<data[0].coordinates[0].length;i++){
        land.push(new
google.maps.LatLng(data[0].coordinates[0][i][1], data[0].coordinates[0][i][0]));
    }
    var land1;
    land1 = new google.maps.Polygon({
        clickable:true,
        paths: land,
        strokeColor: "#FF0000",
        strokeOpacity: 0.5,
        strokeWeight: 1,
        fillColor: "#FF0000",
        fillOpacity: 0
    });
    land1.setMap(map);

    shape = new google.maps.Polyline({
        Editable:true,
        strokeColor: '#FF0000',
        strokeOpacity: 0.8,
        strokeWeight: 5,
        fillColor: '#FF0000',
        fillOpacity: 0.5
    });
    google.maps.event.addListener(land1,'mousemove',function(event) {
        $('#coord').val(event.latLng.lat() + ', ' +
event.latLng.lng());
    });

    shape.setMap(map);
    google.maps.event.addListener(land1,'click',addPoint);
    google.maps.event.addListener(map,'click',error);
    function error(){
        alert("Not drawing area");
    }
    function addPoint(e) {
        shape.setMap(map);
        shape.setOptions({ Editable: true });
        $('#line2').val(e.latLng.lat() + ', ' +
e.latLng.lng());

        vertices= shape.getPath();
        vertices.push(e.latLng);
        polygon.push(e.latLng);
    }
    $("#line").click(function(e){
        a=JSON.stringify(polygon);
        var c=JSON.parse(a);
        for (i=0;i<c.length;i++){
            x.push(c[i].Za+ " "+c[i].Ya)
        }
        $('#line2').val(a);
        var selection=$("#myselect").val();
        var c=JSON.parse(a);
        for (i=0;i<c.length;i++){
            polygon2.push(c[i].Ya);
            polygon2.push(c[i].Za);
        }
    }
}

```



```

        if(data[i].type=='Μπυραρίες') {

            var image='http://localhost/findpoi/public/img/beer.png';

            marker

        = new google.maps.Marker({

            position: new google.maps.LatLng(data[i].poi.coordinates[1],
            data[i].poi.coordinates[0]),

            map: map,

            icon:image

        });

        markers.push(marker);

        google.maps.event.addListener(marker, 'click', (function(marker, i) {

            return

        function() {

            var geojson=JSON.stringify(data[i].poi);

            $('#geojson').val(geojson);

            $('#coord').val( data[i].poi.coordinates[1]+ ',' + data[i].poi.coordinates[0]);

            infowindow.setContent("<br><br>Όνομα:" +data[i].name+"<br>" + "Διεύθυνση:" +data[i].address+"<br>
            "+"Τηλέφωνο: " +data[i].telephone);

            infowindow.open(map, marker);

        }

        })(marker, i));

        mc = new

    MarkerClusterer(map, markers);

    });

});

```

Εικόνα 41 line.js

```

<div class="container-fluid">
    <div class="row-fluid">
        <div class="navbar">
            <div class="navbar-inner">
                <div class="container">
                    <a class="brand"
href="http://localhost/findpoi/public/view/map">
                        Αρχική Σελίδα
                    </a>
                    <ul class="nav">
                        <li
class="dropdown">
                            <a
href="#"

```

Διαδικτυακή πλατφόρμα πολυεπίπεδης αρχιτεκτονικής για την πραγματοποίηση δυναμικών αναζητήσεων σε γεωπληροφοριακό σύστημα



```

<h1 align="center">Εμφάνιση σημείων ενδιαφέροντος με λειτουργία εισαγωγής πολυγώνου
</h1>
<div class="container-fluid">
    <div class="row-fluid">
        <div class="navbar">
            <div class="navbar-inner">
                <div class="container">
                    <a class="brand"
href="http://localhost/findpoi/public/view/userview">
                        Αρχική Σελίδα
                    </a>
                    <ul class="nav">
                        <li
class="dropdown">
                            <a
href="#"
                                class="dropdown-toggle"
                                data-toggle="dropdown">
                                    <strong><font color="white" size="5">Επιλογές</font></strong>
                                    <b class="caret"></b>
                                </a>
                            <ul
class="dropdown-menu">
                                <?php echo $link ;?>
                            </ul>
                        </li>
                    </ul>
                </div>
            </div>
        </div>
        <div class="row-fluid">
            <div class="span3 well">
                <ol>
                    <li><strong><font
size="4">Σχεδιάστε ένα πολύγωνο στον χάρτη</font></strong></li>
                    <li><strong><font
class="btn-warning" value="Δημιουργία νέου πολυγώνου" /></strong></li>
                    <li><strong><font size="4">Επιλογή
σημείου ενδιαφέροντος:</font></strong></li>
                </ol>
                <select name="poi" id="myselect">
                    <option
value="1">Επιλογή... </option>
                    <?php foreach
($categorydata as $category1):?>
                        <option value='<?php echo
$category1['p_type']?'><?php echo $category1['p_type']?'></option>
                    <?php endforeach ;?>
                </select>
                <li><input id="polygon"
class="btn-success" type="button" value="Εμφάνιση Επιλογής" /></li>
                </ol>
            </div>
        </div>
    </div>
    <p align="center">Ζωγραφίστε στον χάρτη
ένα πολύγωνο πατώντας την επιλογή δημιουργία νέου πολυγώνου και κλικαροντας
στον χάρτη τις γωνίες του πολυγώνου το
οποίο περικλείει την περιοχή ενδιαφέροντος σας. Στη συνέχεια επιλέξτε το σημείο
ενδιαφέροντος σας (πχ καφετεριες) και
πατήστε Εμφάνιση επιλογής. Για να εισάγετε νέο σημείο ενδιαφέροντος και

```





```

σημείου ενδιαφέροντος:</font></strong></li></br>
<li><strong><font size="4">Επιλογή
</strong></li></br>
<select name="poi" id="myselect">
  <option
value="1">Επιλογή...</option>
  <?php foreach
($categorydata as $category1):?>
    <option value='<?php echo
$category1['p_type']?'><?php echo $category1['p_type']?'></option>
  <?php endforeach ;?>
</select>

<li><input id="line" class="btn-
success" type="button" value="Εμφάνιση Επιλογής" /></li></br> </br>

</ol>
</br>
<p align="center">Ζωγραφίστε στον χάρτη
για διαδρομή πατώντας την επιλογή δημιουργία νέας διαδρομής και κλικαροντας
στον χάρτη τα σημεία της διαδρομής η
οποία περικλείει την περιοχή ενδιαφέροντος σας.Στη συνέχεια επιλέξτε το σημείο
ενδιαφέροντος σας (πχ καφετεριες) και
πατήστε Εμφάνιση επιλογής.Για να εισάγετε νέο σημείο ενδιαφέροντος και
διαδρομή πατήστε το κουμπί
δημιουργία νέας διαδρομής.</p>

</br></br>
</div>
<div class="span9 well">
  <div id="map_canvas"style="width:100%;
height:70%"></div>
</div>
</div>
<div class="navbar">
  <div class="navbar-inner">
    <h3>Developed By:Kougioumtzis
Dimitris</h3>
  </div>
</div>
</div>
</div>

```

Εικόνα 44 Κώδικας line.php

## 7. Βιβλιογραφία

Οδηγός χρήσης Codeigniter. Διαθέσιμο στο:

<[http://codeigniter.com/user\\_guide/toc.html](http://codeigniter.com/user_guide/toc.html)> [Προσπέλαση 2 Σεπτεμβρίου 2012]

Jquery how it works. Διαθέσιμο στο:

<[http://docs.jquery.com/Tutorials:How\\_JQuery\\_Works](http://docs.jquery.com/Tutorials:How_JQuery_Works)>[Προσπέλαση 2 Σεπτεμβρίου 2012]

Αρχιτεκτονική Ajax. Διαθέσιμο στο:

<[http://images.drccsystems.com/ajax\\_img.jpg](http://images.drccsystems.com/ajax_img.jpg)> [Προσπέλαση 2 Σεπτεμβρίου 2012]

Οδηγός χρήσης Ajax. Διαθέσιμο στο:

<<http://www.tizag.com/ajaxTutorial/index.php>>[Προσπέλαση 2 Σεπτεμβρίου 2012]

Οδηγός χρήσης Json . Διαθέσιμο στο:

<<http://www.w3schools.com/json/default.asp>>[Προσπέλαση 2 Σεπτεμβρίου 2012]

Οδηγός χρήσης Geojson. Διαθέσιμο στο:

<<http://www.geojson.org/geojson-spec.html>> [Προσπέλαση 2 Σεπτεμβρίου 2012]

Οδηγός χρήσης Google maps api. Διαθέσιμο στο:

<<https://developers.google.com/maps/documentation/javascript/tutorial>>[Προσπέλαση 2 Σεπτεμβρίου 2012]

Παράδειγμα Google maps. Διαθέσιμο στο:

<<https://google-developers.appspot.com/maps/documentation/javascript/examples/map-simple>> [Προσπέλαση 2 Σεπτεμβρίου 2012]

Οδηγός χρήσης προγραμματισμού Ajax. Διαθέσιμο στο:

<[http://en.wikipedia.org/wiki/Ajax\\_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming))>[Προσπέλαση 2 Σεπτεμβρίου 2012]

PostgreSQL Advantages. Διαθέσιμο στο:

<<http://www.postgresql.org/about/advantages/>>[Προσπέλαση 2 Σεπτεμβρίου 2012]

PostgreSQL Documentation. Διαθέσιμο στο:

<<http://www.postgresql.org/docs/devel/static/gin-intro.html>>[Προσπέλαση 2 Σεπτεμβρίου 2012]

About postgresQL. Διαθέσιμο στο:

<<http://postgresql.gr/node/3>>[Προσπέλαση 2 Σεπτεμβρίου 2012]

Wind GIS Tuscany region Διαθέσιμο στο:

<<http://geoportale.lamma.rete.toscana.it/eng/windgis>>[Προσπέλαση 2 Σεπτεμβρίου 2012]

Aitkenhead M.J., Aalders I.H, 2009. Predicting land cover using GIS, Bayesian and evolutionary algorithm methods, p 236-250. Sciencedirect. Διαθέσιμο στο:

< <http://www.sciencedirect.com/science/article/pii/S0301479707003465>>

[Προσπέλαση 2 Σεπτεμβρίου2012]

Anon, 2000. Handbook on geographic information systems and digital mapping (United Nations).

Διαθέσιμο στο:

<url:[http://unstats.un.org/unsd/publication/SeriesF/SeriesF\\_79E.pdf](http://unstats.un.org/unsd/publication/SeriesF/SeriesF_79E.pdf)>

[Προσπέλαση 2 Σεπτεμβρίου2012]

Buckley David J., 1997. The GIS primer [pdf] Διαθέσιμο στο:

<[http://www.innovativegis.com/basis/primer/The\\_GIS\\_Primer\\_Buckley.pdf](http://www.innovativegis.com/basis/primer/The_GIS_Primer_Buckley.pdf)> [Προσπέλαση 2

Σεπτεμβρίου2012]

Connolly Thomas, Begg Carolyn E, 1998. Database systems. A practical approach to design, implementation, and management (2 ed) Boston, MA, USA.

Eastman, J. Ronald, Fulk, Michele , Toledano James , 1993. The GIS Handbook .The University of Arizona. Διαθέσιμο στο:

< <http://www.afr-sd.org/publications/a11gis.pdf>>

[Προσπέλαση 2 Σεπτεμβρίου2012]

Huang Bo, Liu Nan, Chandramouli Magesh, 2005. A GIS supported Ant algorithm for the linear feature covering problem with distance constraints, p 1063-1075. Sciencedirect. Διαθέσιμο στο:

<<http://www.sciencedirect.com/science/article/pii/S0167923605001326>>

[Προσπέλαση 2 Σεπτεμβρίου2012]

Inés Santé, Rafael Crecente, David Miranda, Juan Touriño, Fernando Canzobre, Ramón Doallo,2004. A GIS web-based tool for the management of the PGI potato of Galicia, p 161-171. Sciencedirect. Διαθέσιμο στο:

<<http://www.sciencedirect.com/science/article/pii/S016816990400047X>>

[Προσπέλαση 2 Σεπτεμβρίου2012]

Jadranka Pecar-Ilic, Ivica Ruzic, 2006. Application of GIS and Web technologies for Danube waterway data management in Croatia, p 1562-1571. Sciencedirect. Διαθέσιμο στο:

< <http://www.sciencedirect.com/science/article/pii/S1364815206001150>>

[Προσπέλαση 2 Σεπτεμβρίου2012]

Kulawajak M., Prospathopoulos A., Perivoliotis L., Łuba M., Kioroglou S., Stepnowski A., 2010. Interactive visualization of marine pollution monitoring and forecasting data via a Web-based GIS, p 1069-1080. Sciencedirect. Διαθέσιμο στο:

[Προσπέλαση 2 Σεπτεμβρίου2012]

<<http://www.sciencedirect.com/science/article/pii/S0098300410001603>>  
[Προσπέλαση 2 Σεπτεμβρίου2012]

Myer Thomas,2008. Professional codeigniter. Willey Publishing, Inc. Indiana, USA.

Riccardo Maria, Lorenzo Bottaia, Caterina Busillo, Francesca Calastrini, Bernardo Gozzini, Giovanni Gualtieri, 2010. A GIS-based interactive web decision support system for planning wind farms in Tuscany (Italy), p 754-763. Sciencedirect. Διαθέσιμο στο:  
<<http://www.sciencedirect.com/science/article/pii/S0960148110003290>>  
[Προσπέλαση 2 Σεπτεμβρίου2012]

Sayar A., Pierce M., Fox G.C., n.d. Developing GIS visualization web services for geophysical applications, p 10. Διαθέσιμο στο:  
<[http://grids.ucs.indiana.edu/ptliupages/publications/isprs\\_asayar.pdf](http://grids.ucs.indiana.edu/ptliupages/publications/isprs_asayar.pdf)>  
[Προσπέλαση 2 Σεπτεμβρίου2012]

Zainuddin K, Mokhtar E.S., Wan Yusof K., 2010. Developing a UiTM (Perlis) Web-Based of Building Space Management System: A Preliminary Study in Locating a Specified Space/Room Area Using Open Source GIS Tool, p 154-158. Sciencedirect. Διαθέσιμο στο:  
<<http://www.sciencedirect.com/science/article/pii/S1877705811029596>>  
[Προσπέλαση 2 Σεπτεμβρίου2012]

Γκατζόφλιας Δημήτριος, 2007. Γεωγραφικά πληροφοριακά συστήματα στο διαδίκτυο-Μελέτη περίπτωσης MapServer, Μεταπτυχιακή εργασία, Πανεπιστήμιο Μακεδονίας. Διαθέσιμο στο:  
<<http://dspace.lib.uom.gr/bitstream/2159/3785/1/GkatzofliasMsc2007.pdf>>  
[Προσπέλαση 2 Σεπτεμβρίου2012]

Γκέργκι Στρακόσα, 2005. Seismo-Surfer: Ένα Σύστημα Γεωγραφικών Πληροφοριών για τον Ιστό με σκοπό την ολοκλήρωση, οπτικοποίηση και ανάλυση σεισμικών δεδομένων, Πτυχιακή εργασία, Πανεπιστήμιο Πειραιά. Διαθέσιμο στο:  
< <http://seimosurfer.googlecode.com/files/thesis.pdf>>  
[Προσπέλαση 2 Σεπτεμβρίου2012]

Νάκου Βασιλείου,2010. Ολοκληρωμένο σύστημα απεικόνισης κινδύνων πλημμύρων, Διπλωματική εργασία, Πανεπιστήμιο Πατρών. Διαθέσιμο στο:  
<<http://nemertes.lis.upatras.gr/jspui/bitstream/10889/2912/1/%CE%B4%CE%B9%CF%80%CE%B%CF%89%CE%BC%CE%B1%CF%84%CE%B9%CE%BA%CE%AE.pdf>>  
[Προσπέλαση 2 Σεπτεμβρίου2012]

Ζώταλης Αντώνιος,2008. Γεωγραφικά συστήματα πληροφοριών σε δίκτυα ύδρευσης οικισμών, εφαρμογή: Μανδράκι, Νίσυρος, Διπλωματική εργασία, Εθνικό Μετσόβιο πολυτεχνείο. Διαθέσιμο στο:

<[http://dspace.lib.ntua.gr/bitstream/123456789/2540/3/zotalisa\\_gis.pdf](http://dspace.lib.ntua.gr/bitstream/123456789/2540/3/zotalisa_gis.pdf)>

[Προσπέλαση 2 Σεπτεμβρίου2012]

Καραπαναγιωτή Ευθαλία, 2005. Συνδυασμός γεωγραφικών συστημάτων πληροφοριών και μαθηματικού προγραμματισμού για τη χωροθέτηση εμπορικών επιχειρήσεων, Πτυχιακή εργασία, Πανεπιστήμιο Πατρών. Διαθέσιμο στο:

<<http://nemertes.lis.upatras.gr/jspui/bitstream/10889/973/1/%CE%94%CE%B9%CF%80%CE%BB%CF%89%CE%BC%CE%B1%CF%84%CE%B9%CE%BA%CE%AE%20%CE%95%CF%81%CE%B3%CE%B1%CF%83%CE%AF%CE%B1.pdf>>

[Προσπέλαση 2 Σεπτεμβρίου2012]

Σχοινά Κωνσταντίνου Σ, 2012. Ανάπτυξη ενός πληροφοριακού συστήματος διαχείρισης αστικών ακινήτων με την τεχνολογία των Web services, Διπλωματική εργασία, Εθνικό Μετσόβιο Πολυτεχνείο. Διαθέσιμο στο:

<[http://dspace.lib.ntua.gr/bitstream/123456789/6964/3/sxoinask\\_gis.pdf](http://dspace.lib.ntua.gr/bitstream/123456789/6964/3/sxoinask_gis.pdf)>

[Προσπέλαση 2 Σεπτεμβρίου2012]

Μανδουράρη Δήμητρα, 2005. Πληροφοριακό σύστημα για την καθοδήγηση χρηστών με χρήση GIS, Πτυχιακή εργασία, Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης. Διαθέσιμο στο:

<<http://nefeli.lib.teicrete.gr/browse/stef/epp/2011/MandourariDimitra/attached-document-1308739738-773291-16209/Mandourari2011.pdf>>

[Προσπέλαση 2 Σεπτεμβρίου2012]