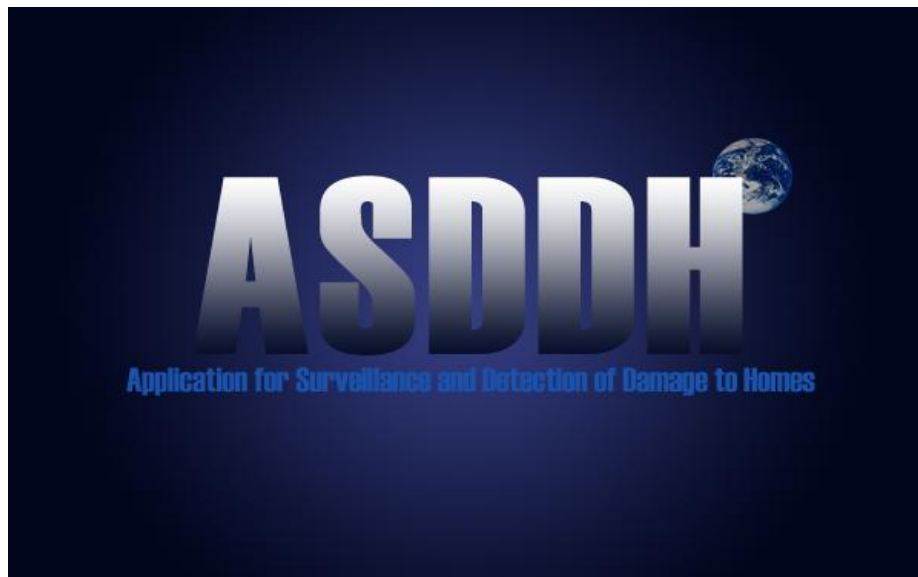




ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΙΟΝΙΩΝ ΝΗΣΩΝ

ΠΑΡΑΡΤΗΜΑ ΛΕΥΚΑΔΑΣ
ΤΜΗΜΑ ΕΦΑΡΜΟΓΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΣΤΗ ΔΙΟΙΚΗΣΗ ΚΑΙ ΣΤΗΝ ΟΙΚΟΝΟΜΙΑ

**Ανάπτυξη μίας εφαρμογής επίβλεψης και ανίχνευσης
ζημιών σε μια οικία με τη χρήση εργαλείων της Google**



των

Παναγιώτου Παναγιώτη

Παπαδόπουλος Νίκο

Επιβλέπων καθηγητής: Αθανάσιος Κακαρούντας

ΛΕΥΚΑΔΑ, 2011

ΠΡΟΛΟΓΟΣ

Η επιστήμη των υπολογιστών τα τελευταία χρόνια έχει εξελιχτεί πάρα πολύ, διεισδύοντας στην καθημερινότητά μας όλο και πιο πολύ. Χρησιμοποιούμε την τεχνολογία άλλοτε σε μικρό, και άλλοτε σε μεγάλο βαθμό. Η αγάπη μας για την επιστήμη των υπολογιστών μας «έσπρωξε» να θέλουμε να σπουδάσουμε σε ένα τμήμα πληροφορικής να γνωρίσουμε και να μάθουμε πως να δημιουργήσουμε μέσα από αυτή την επιστήμη.

Η στήριξη των δικών μας ανθρώπων αλλά και των συμφοιτητών μας, που με την τριβή του χρόνου έγιναν και εκείνοι δικοί μας άνθρωποι, μας ενδυνάμωσαν το ενδιαφέρον για την επιστήμη των υπολογιστών.

Η πρόκληση να δημιουργήσουμε μία εφαρμογή βασισμένη πάνω στη πλατφόρμα της Google και πιο συγκεκριμένα στο Google maps, είναι αυτή που μας ενέτεινε το ενδιαφέρον και μας οδήγησε σε πολύωρη και επίπονη ενασχόληση με το πολύ ενδιαφέρον θέμα που πραγματεύεται η παρούσα εργασία. Το αποτέλεσμα βέβαια θα το κρίνετε εσείς.

Κλείνοντας θα θέλαμε να ευχαριστήσουμε τον επιβλέποντα καθηγητή, κύριο Κακαρούντα Αθανάσιο, που μας εμπιστεύτηκε αυτή την εργασία και μας άντεξε όλες τις γρίνιες και τις απορίες μας, αλλά και τους καθηγητές μας του Τ.Ε.Ι Ιονίων Νήσων του τμήματος ΕΠΔΟ που μας βοήθησαν να αποκομίσουμε την απαραίτητη γνώση.

Επίσης θα θέλαμε να ευχαριστήσουμε τα παιδιά από την εταιρία interface-team που κατά τη διάρκεια τις πρακτικής μου άσκησης όσο και μετά από αυτή, ήταν κοντά μας για ότι απορία είχαμε σε σχέση με τον τρόπο υλοποίησης της πτυχιακής.

Αυτή η εργασία είναι αφιερωμένη στους γονείς μας, Χρυστάλλα, Ντίνο και Ελένη, Γιάννη.

Ευχόμαστε καλή ανάγνωση σε όλους,

Παναγιώτου Παναγιώτης,

Παπαδόπουλος Νίκος.

ΠΕΡΙΛΗΨΗ

Στην παρούσα εργασία θα πραγματοποιούμε την ανάπτυξη μιας εφαρμογής ενός συστήματος ελέγχου και ασφάλειας κτιρίων μέσω ενός υπολογιστή, βασισμένο στην υπηρεσία Google Maps και στην τεχνολογία GIS. Η εργασία είναι χωρισμένη σε 2 κύρια μέρη, το 1ο είναι το θεωρητικό κομμάτι και το 2ο το κομμάτι της υλοποίησης της εφαρμογής.

Κατά τη χρονική περίοδο υλοποίησης της παρούσας εργασίας, η τεχνολογία GIS είναι μια καινούργια τεχνολογία και συνεχώς εξελισσόμενη θα αναφέρουμε τον τρόπο λειτουργίας της, που χρησιμοποιείτε και ποιες ανάγκες εξυπηρετεί. Στη συνέχεια θα αναφερθούμε στην υπηρεσία της Google, το Google Maps, το API Google Maps, που χρησιμοποιούνται για να αναπτυχτεί μια εφαρμογή GIS. Και τέλος θα κάνουμε μία μελέτη αγοράς για την εφαρμογή που θα αναπτύξουμε.

Στόχος της παρούσας διατριβής είναι να δημιουργήσουμε μια εφαρμογή που με τη βοήθεια των πιο πάνω τεχνολογιών, να παρέχει έλεγχο και ασφάλεια σε κτίρια από πιθανούς κινδύνους. Η εφαρμογή θα έχει τη δυνατότητα μέσω αισθητήρων να καταγράφει τι γίνεται στο κτίριο, να στέλνει τις πληροφορίες στη βάση δεδομένων (και εκεί να ελέγχονται τα δεδομένα) και αν διαπιστωθεί ότι υπάρχει κίνδυνος, να ενημερώνει το σύστημα ώστε να αποφύγουμε δυσάρεστες καταστάσεις(φωτιά, διαρροή φυσικού αερίου κ.α). Δευτερεύοντα στόχο αποτελεί η ανάπτυξη μιας εφαρμογής μέσα από τις δυνατότητες που μας δίνουν τα εργαλεία της Google.

Η παρούσα μελέτη απευθύνεται σε προπτυχιακούς και μεταπτυχιακούς φοιτητές, αλλά και σε επιστήμονες και τεχνολόγους της γνωστικής περιοχής της Τεχνολογίας Λογισμικού που ενδιαφέρονται να αντλήσουν πληροφοριακό υλικό, για ζητήματα της γνωστικής αυτής περιοχής, αλλά και εξειδικευμένες γνώσεις και δεξιότητες στο σχεδιασμό και ανάπτυξη εφαρμογής με χρήση της τεχνολογίας GIS, έχοντας ως στόχο μια ρεαλιστική εφαρμογή. Στις περιπτώσεις που είναι αναγκαία η παράθεση ορισμών και αποτελεσμάτων, αυτή γίνεται έτσι ώστε ο αναγνώστης να μην αποθαρρυνθεί αλλά να συνεχίσει την ανάγνωση της διατριβής χωρίς δυσκολία. Απαιτούνται ασφαλώς, κάποιες βασικές γνώσεις από την τεχνολογία λογισμικού των υπολογιστών.

Λέξεις κλειδιά: GIS, Google maps, Google api, ASDDH , modus

ABSTRACT

In this present work we will analyze the development of an implementation of a control system and buildings security with a software based on Goggle Maps service and GIS technology .

This work is divided into two main parts, the first part is the theoretical part and the second part is the implementation of the application. During the implementation of this project, the GIS technology is a new technology and constantly evolving, we will mention how it works, where it can be used and which needs it serves. Then we will refer to the services of Google, Google Maps, API Google Maps, where we can use them to develop a GIS application. And finally we will make a market study for the application that we will develop.

The aim of this present work is to create an application using the above technologies that will provide control and security to buildings against potential risks

The application will have the ability to record with sensor means what happens in the building, to send information to the database (and there to control data) and see if there is danger, to inform the system in order to avoid unpleasant situations (fire, leak of gas, etc.). Secondary aim is to develop an application through the possibilities that give us the tools of Google.

This study is aimed at undergraduate and postgraduate students, and scientists and technologists in the cognitive area of Software Engineering seeking to gain information on knowledge issues of this area, and specialized knowledge and skills in design and application development using technology, aiming for a realistic application. Where it is necessary to quote definitions and results, have been made with a way that the reader is not discouraged but to continue reading of this work without difficulty. Of course it requires, some basic knowledge of computer software technology.

Keywords: GIS, Google maps, Google api, ASDDH , modus

Περιεχόμενα

1.	Εισαγωγή	1
1.1	Γεωγραφικά Συστήματα Πληροφοριών (ΓΣΠ - GIS)	1
1.2	Google maps.....	2
1.3	Google Maps-Apí	2
1.4	Μεθοδολογία ολοκλήρωσης της εφαρμογής	2
1.5	Κύκλος ζωής- Προδιαγραφές συστήματος.....	2
1.6	Σχεδίαση Εφαρμογής	2
1.6	Αξιολόγηση	2
1.7	Συμπεράσματα	2
2.	Γεωγραφικά Συστήματα Πληροφοριών (GIS)	3
2.1.	Ιστορία	3
2.2	Ορισμός	3
2.3	Σκοπός των G.I.S	4
2.4	Τεχνολογίες	6
2.5	Κατηγορίες G.I.S.	12
2.6	Τα G.I.S στο Διαδίκτυο.....	14
2.7	Google Maps.....	15
2.8	Application Programming Interface (API).....	16
2.9	Google Maps API	17
3.	Μεθοδολογία ολοκλήρωσης της εφαρμογής	22
3.1	Εισαγωγή	22
3.2	Μελέτη αγοράς	22
3.3	Έρευνα με τη μέθοδο του ερωτηματολογίου - Συμπεράσματα	24
3.4	Μεθοδολογία υλοποίησης	35
4.	Κύκλος Ανάπτυξης λογισμικού.....	36

4.1 Εισαγωγή	36
4.2 Έννοιες για την ανάπτυξη λογισμικού	36
4.3 Επιλογή μοντέλου κύκλου ζωής.....	43
5. Προδιαγραφές Συστήματος	45
5.1 Εισαγωγή	45
5.2 Προδιαγραφές Ανάπτυξης	45
5.3 Σχεδιαγράμματα	49
5.4 Λεξικό δεδομένων	54
6. Σχεδίαση Εφαρμογής	55
6.1 Εισαγωγή	55
6.2 Κατανόηση των απαιτήσεων.....	55
6.3 Σχεδίαση interface.....	55
6.4 Μετατροπή του Template σε κώδικα.....	57
7. Αξιολόγηση Εφαρμογής	75
7.1 Εισαγωγή	75
7.2 Εκπλήρωση απαιτήσεων	75
7.3 Λειτουργικότητα της εφαρμογής	75
7.4 Πλεονεκτήματα	75
7.5 Μειονεκτήματα	75
7.6 Συμπέρασμα	76
8. Συμπεράσματα	777
Πηγές	778
Παράρτημα Α.....	80
Παράρτημα Β.....	888

Κεφάλαιο 1

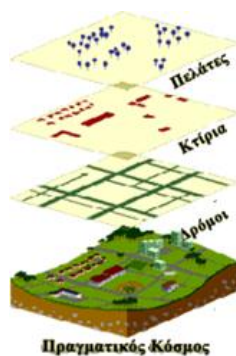
1. Εισαγωγή

Στις αρχές της δεκαετίας μπήκε για τα καλά το διαδίκτυο (internet) στη ζωή μας. Έτσι σιγά σιγά αναπτύχθηκαν εφαρμογές για την καλύτερη χρήση του και την εκμετάλλευση των δυνατοτήτων που μας προσφέρει. Έτσι η Google μία από τις πολλές υπηρεσίες που ανάπτυξε είναι και την υπηρεσία Google Maps. Χαρτογράφησε ψηφιακά σχεδόν όλο τον πλανήτη και συνάμα έδωσε την δυνατότητα στον απλό χρήστη στην περιπλάνηση του όπου αυτός θέλει αλλά και την δυνατότητα μεγαλύτερης ανάπτυξης των Γεωγραφικών Συστημάτων Πληροφοριών (GIS). Παράλληλα έδωσε νέα διάσταση στην εφαρμογή GPS. Με την ευκαιρία που μας δίνετε μέσω της πτυχιακής εργασίας θα προσπαθήσουμε να αναπτύξουμε μια εφαρμογή όπου θα εκμεταλλευτούμε της δυνατότητες που μας προσφέρουν τα GIS και το Google Maps. Θα αναπτύξουμε μία εφαρμογή επίβλεψης και ανίχνευσης ζημιών σε μια οικία. Πέραν από τις υπηρεσίες που προαναφέραμε θα καταπιαστούμε και με τις διαθέσιμες τεχνολογίες αισθητήρων, για σωστή επίβλεψη και ανίχνευση αιτιών που μπορούν να προκαλέσουν ζημιές σε ένα σπίτι ή ακόμη και να φέρουν σε κίνδυνο ανθρώπινες ζωές.

1.1 Γεωγραφικά Συστήματα Πληροφοριών (ΓΣΠ - GIS)

Όπως προαναφέραμε στην εισαγωγή τα γεωγραφικά συστήματα πληροφοριών αναπτύχθηκαν σημαντικά με την δυναμική εμφάνιση του internet. Τα GIS υπήρχαν πόλη πιο πριν από την εμφάνιση του internet και του υπολογιστή, ήταν ένα σύστημα ανάκτησης δεδομένων από χαρτογραφούμενες περιοχές. Συγκεκριμένα η πρώτη φορά που αναφέρθηκε ο όρος αυτός ήταν τη δεκαετία του '60 από τον Roger Tomlinson για να περιγράψει ένα Σύστημα Ανάλυσης Χαρτογραφικών Δεδομένων, που είχε αναπτυχθεί για την καναδική κυβέρνηση, καθώς και από τον Duane Marble για να περιγράψει ένα Σύστημα Μελέτης Αστικών Κυκλοφοριακών Προβλημάτων.

Πρακτικά ένα GIS σύστημα, είναι ένα σύστημα λογισμικού (software), υλικού (hardware), δεδομένων (data), και προσωπικού με σκοπό να βοηθήσει στο χειρισμό, στην ανάλυση και στην παρουσίαση των πληροφοριών που είναι συγκεντρωμένα σε μια γεωγραφική περιοχή. Οι τεχνολογία των Γεωγραφικών Συστημάτων Πληροφοριών (GIS) μπορεί γενικά να χρησιμοποιηθεί σε επιστημονικές έρευνες, στη διαχείριση πόρων και στη σχεδίαση αναπτυξιακών προγραμμάτων. Με απλά λόγια, ένα σύστημα GIS συνδυάζει «στρώσεις» πληροφοριών σχετικά με μια περιοχή για την καλύτερη κατανόηση αυτής της περιοχής. Το τι είδους «στρώσεις» πληροφοριών χρησιμοποιούμε έχει να κάνει με το σκοπό για τον οποίο θα χρησιμοποιήσουμε το σύστημα.



Σχήμα 1.1. Τα ΓΣΠ συνδυάζει στρώσεις πληροφοριών

1.2 Google maps

Η Google όπως προαναφέραμε εξελίχθηκε παράλληλα με την εξέλιξη του διαδικτύου. Έτσι ανέπτυξε διάφορες υπηρεσίες που προσφέρει στους χρήστες του διαδικτύου δωρεάν, μια από τις υπηρεσίες που προσφέρει είναι και το Google maps. Η υπηρεσία αυτή τα τελευταία 5 χρόνια (το 2005 πρωτοεμφανίστηκε) μας δίνει την δυνατότητα να έχουμε χάρτες σε ψηφιακή μορφή στο διαδίκτυο. Έτσι έχει τη δυνατότητα όποιος χρήστης θέλει να δει χάρτες από όλο το κόσμο.

1.3 Google Maps-API

Το Google Maps – Api είναι μια διεπαφή προγραμματισμού εφαρμογών (Application Programming Interface – API) της Google το οποίο δίνει τη δυνατότητα ενσωμάτωσης ενός δυναμικού χάρτη Google σε ιστοσελίδες (web σελίδες) με τη χρήση της javascript. Το API παρέχει έναν αριθμό εργαλείων το χειρισμό χαρτών και την προσθήκη περιεχομένου σε αυτούς μέσω διαφόρων υπηρεσιών, όπως και στην επίσημη ιστοσελίδα του Google Maps (<http://maps.google.com>), επιτρέποντας έτσι τη δημιουργία εύρωστων εφαρμογών που εκμεταλλεύονται γεωγραφικές πληροφορίες.

1.4 Μεθοδολογία ολοκλήρωσης της εφαρμογής

Στη μεθοδολογία ολοκλήρωσης κάνουμε μία μελέτη αγοράς και παραθέτουμε μερικά στοιχεία που δείχνουν τη χρήση των GIS στην αγορά. Σε ποιούς τομείς υπάρχει μεγάλη χρήση των συστημάτων αυτών. Επίσης θα αναφερθούμε στην έρευνα που κάναμε με τη μέθοδο του ερωτηματολογίου για να δούμε τη διάθεση της αγοράς προς τις υπηρεσίες που προσφέρουν τα GIS και το Google. Να δούμε τι άποψη επικρατεί για την εφαρμογή που εμείς έχουμε σκοπό να ανάπτυξη και σε πιο βαθμό είναι διευθετημένο το αγοραστικό κοινό να την αποδεχτεί και με ποιο κόστος θα ήταν πρόθυμο να την χρησιμοποιήσει.

1.5 Κύκλος ζωής- Προδιαγραφές συστήματος

Στο κύκλο ζωής καθορίζει το πώς θα κινηθούμε για να υλοποιήσουμε την εφαρμογή. Δηλαδή την μεθοδολογία που θα ακολουθήσουμε και με ποία σειρά θα ακολουθηθεί για την γρήγορη και σωστή λειτουργία της εφαρμογής. Στη συνέχεια στις προδιαγραφές του συστήματος θα αναφερθούμε στις απαιτήσεις για την εφαρμογή τον τρόπο λειτουργίας της. Επίσης θα δήξουμε τον τρόπο λειτουργία τις με σχεδιαγράμματα για να έχουμε μια πλήρη και λεπτομερή εικόνα για το τη έχουμε σκοπό να κάνουμε.

1.6 Σχεδίαση Εφαρμογής

Θα αναφέρουμε πως σχεδιάστηκε η εφαρμογή και με πια λογική ακολουθήθηκε συγκεκριμένος τρόπος συγγραφής του κώδικα της με παραδείγματα.

1.6 Αξιολόγηση

Θα αξιολογήσουμε την εφαρμογή που φτιάξαμε για να δούμε αν είναι ικανοποιητική η απόδοση τις, αν περιλαμβάνει τις προδιαγραφές που θέσαμε, και ποία είναι τα μειονεκτήματα της (αν υπάρχουν).

1.7 Συμπεράσματα

Στα συμπεράσματα θα αναφερθούμε γενικά στην εφαρμογή και αν θεωρούμε πως θα ανταποκριθεί στην αγορά.

Κεφάλαιο 2

2. Γεωγραφικά Συστήματα Πληροφοριών (GIS)

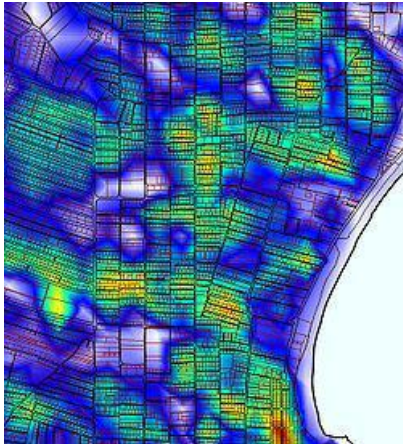
2.1. Ιστορία

Ο όρος **Γεωγραφικά Συστήματα Πληροφοριών (GIS)** χρησιμοποιήθηκε για πρώτη φορά τη δεκαετία του '60 από τον Roger Tomlinson για να περιγράψει ένα Σύστημα Ανάλυσης Χαρτογραφικών Δεδομένων, που είχε αναπτυχθεί για την καναδική κυβέρνηση, καθώς και από τον Duane Marble για να περιγράψει ένα Σύστημα Μελέτης Αστικών Κυκλοφοριακών Προβλημάτων. Αλλά στην ουσία τους, τα γεωγραφικά συστήματα πληροφοριών αναπτύχθηκαν και εξελίχθηκαν κατά τη διάρκεια πολλών αιώνων, βασικά μέσω της δημιουργίας χαρτών, καθώς και της συλλογής γεωγραφικών πληροφοριών και της αποθήκευσής τους σε υλικά μέσα της εκάστοτε εποχής. Την τελευταία δεκαετία τα GIS αναπτύχθηκαν και εδρεοθηκαν αισθητά στο χώρο της τεχνολογίας και τον ψηφιακών μέσων. Τα 2 πιο τρανά παραδείγματα αυτής της τεχνολογικής εξέλιξης είναι τα GPS που εβαλαν στην καθημερινότητα μας του χάρτες και το πιο γνωστο τους διαδικτυακους χρήστες Google Maps.

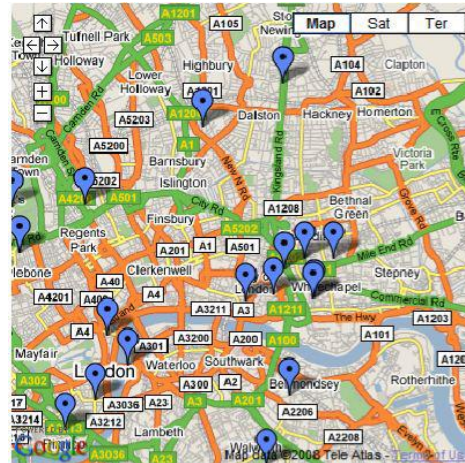
2.2 Ορισμός

Το Γεωγραφικών Σύστημα Πληροφοριών (ΓΣΠ), γνωστό ευρέως και ως G.I.S. Geographic Information Systems, είναι σύστημα διαχείρισης χωρικών δεδομένων (spatial data) και συσχετισμένων ιδιοτήτων. Στην πιο αυστηρή μορφή του είναι ένα ψηφιακό σύστημα, ικανό να ενσωματώσει, αποθηκεύσει, προσαρμόσει, αναλύσει και παρουσιάσει γεωγραφικά συσχετισμένες (geographically-referenced) πληροφορίες. Σε πιο γενική μορφή, ένα ΓΣΠ είναι ένα εργαλείο "έξυπνου χάρτη", το οποίο επιτρέπει στους χρήστες του να αποτυπώσουν μια περίληψη του πραγματικού κόσμου, να δημιουργήσουν διαδραστικά ερωτήσεις χωρικού ή περιγραφικού χαρακτήρα (αναζητήσεις δημιουργούμενες από τον χρήστη), να αναλύσουν τα χωρικά δεδομένα (spatial data), να τα προσαρμόσουν και να τα αποδώσουν σε αναλογικά μέσα (εκτυπώσεις χαρτών και διαγραμμάτων) ή σε ψηφιακά μέσα (αρχεία χωρικών δεδομένων, διαδραστικοί χάρτες στο Διαδίκτυο).

Τα συστήματα GIS, όπως και τα συστήματα CAD, αποτυπώνουν χωρικά δεδομένα σε γεωγραφικό ή χαρτογραφικό ή καρτεσιανό σύστημα συντεταγμένων. Βασικό χαρακτηριστικό των ΓΣΠ είναι ότι τα χωρικά δεδομένα συνδέονται και με περιγραφικά δεδομένα, π.χ. μια ομάδα σημείων που αναπαριστούν θέσεις πόλεων συνδέεται με ένα πίνακα όπου κάθε εγγραφή εκτός από τη θέση περιέχει πληροφορίες όπως ονομασία, πληθυσμός κλπ. η): Δεδομένα και πληροφορίες, η σημασία (αξία) και η χρήση τους.



Σχήμα 2.1. Παράδειγμα :Χάρτης Οικιστικής Πυκνότητας Κατάλληλος για λήψη αποφάσεων, π.χ.ανάπτυξη υποδομών ΟΤΑ



Σχήμα 2.2. Παράδειγμα : Google Maps (σημεία ενδιαφέροντος) Τα σημεία όπου 'έδρασε' ο "Τζακ ο Αντεροβγάλτης" (Λονδίνο, 1888)

Τα Συστήματα Γεωγραφικών Πληροφοριών (ΓΠΣ) είναι πληροφοριακά συστήματα (Information Systems) που παρέχουν την δυνατότητα συλλογής, διαχείρισης, αποθήκευσης, επεξεργασίας, ανάλυσης και οπτικοποίησης, σε ψηφιακό περιβάλλον, των δεδομένων που σχετίζονται με τον χώρο. Τα δεδομένα αυτά συνήθως λέγονται γεωγραφικά ή χαρτογραφικά ή χωρικά (spatial) και μπορεί να συσχετίζονται με μια σειρά από περιγραφικά δεδομένα τα οποία και τα χαρακτηρίζουν μοναδικά.

Σύμφωνα με τον γενικό ορισμό του G.I.S θα μπορούσαμε να πούμε ότι και το αποτέλεσμα που παρουσιάζουμε και στη πτυχιακή εργασία είναι ένα Γεωγραφικό πληροφοριακό σύστημα με περιορισμένες δυνατότητες κάνοντας χρήση των Google Maps και Google Earth.

2.3 Σκοπός των G.I.S

Τα ΓΣΠ συνδυάζουν δεδομένα και συνεργάζονται μ' ένα μεγάλο αριθμό άλλων επιστημονικών κατευθύνσεων, όπως:

- Στη Γεωγραφία
- Στη Χαρτογραφία
- Στη Τηλεπισκόπηση
- Την Πληροφορική
- Την Επιχειρησιακή Έρευνα
- Την Τεχνητή Νοημοσύνη κ.τ.λ.

Οι πληροφορίες χρησιμοποιούνται, είτε για να αναγνωρίσουν και να επισημάνουν την ύπαρξη και τη θέση ενός προβλήματος, είτε για να ανιχνεύσουν και να αναλύσουν τις διάφορες εναλλακτικές λύσεις ή και για να βοηθήσουν στην εκτέλεση μιας απόφασης.

Έτσι σκοπός των ΓΣΠ είναι να εφοδιάζουν τις διαδικασίες λήψης αποφάσεων με τις απαραίτητες πληροφορίες.

Σε αυτή την εξέλιξη είναι αναπόφευκτο να ακολουθήσουν τρία βασικά στάδια έως την πλήρη αποδοχή τους:

1. Άρνηση χρήσης και αποδοχής
2. Προσεκτική αποδοχή και χρήση
3. Πλήρης αποδοχή και εφαρμογή των πλεονεκτημάτων που παρέχουν.

2.3.1 Εφαρμογές των G.I.S

Τα συστήματα αυτά χρησιμοποιούνται ευρέως σε πολλές διοικητικές και παραγωγικές δραστηριότητες, οι οποίες ταξινομούνται σε τρεις βασικές κατηγορίες

- Κοινωνικό-οικονομικές εφαρμογές (π.χ. πολεοδομικός και χωροταξικός σχεδιασμός, κτηματολόγιο, αρχαιολογία, φυσικοί πόροι, ανάλυση αγοράς κ.λπ.)
- Περιβαλλοντικές εφαρμογές (π.χ. δασολογία, έλεγχος πυρκαγιών, έλεγχος επιδημιών κ.λπ.)
- Εφαρμογές διαχείρισης (π.χ. οργάνωση δικτύων ύδρευσης, διαχείριση οδικού δικτύου, οργάνωση επικοινωνιών και ενέργειας, πλοήγηση πλοίων και αεροπλάνων κ.λπ.)

Ο ρόλος των GIS στις παραπάνω εφαρμογές είναι να προσφέρουν στους χρήστες και υπεύθυνους στη λήψη αποφάσεων, ισχυρά εργαλεία για την επίλυση των σύνθετων και όχι πάντα πλήρως δομημένων χωρικών προβλημάτων. Επιπλέον, τα συστήματα αυτά οφείλουν να έχουν μια αποτελεσματική απόδοση (π.χ. ταχεία απόκριση), ώστε να υποστηρίξουν τις ανάγκες των χρηστών σε ένα περιβάλλον παραγωγής.

Η τεχνολογία των Γ.Σ.Π. προσφέρει μια σειρά ωφελειών σε κάθε δραστηριότητα οικονομική ή διοικητική. Παρακάτω δίνεται το πλαίσιο ειδικότερων εφαρμογών που δύναται να αναπτυχθούν.

- Περιφερειακός Προγ/σμός - Σχεδιασμός
- Αστικός Προγραμματισμός - Σχεδιασμός
- Συγκοινωνίες - Μεταφορές
- Τεχνική υποδομή
- Διαχείριση Περιβάλλοντος
- Φορολογία
- Τηλεπικοινωνίες
- Εκπαίδευση και Υγεία – Κοινωνική Πρόνοια
- Πυροσβεστική, Δασική Υπηρεσία, Αστυνομία
- Τουρισμός – Αναψυχή

- Ανάλυση Αγοράς
- Αγορά Εργασίας
- Δίκτυα διανομών, πωλήσεων και χωροθετήσεις κατανομών

Στον αστικό χώρο τα προβλήματα για τα οποία μπορούν να χρησιμοποιηθούν τα συστήματα λήψης αποφάσεων είναι αρκετά και έχουν σχέση κυρίως με τον στρατηγικό σχεδιασμό της πόλης. Οι πιθανές εφαρμογές αφορούν τον δημόσιο αλλά και τον ιδιωτικό τομέα. Οι πλέον συνηθισμένες είναι:

- Προβλέψεις για την μελλοντική ανάπτυξη της πόλης
- Κυκλοφοριακές μελέτες
- Δρομολόγηση περιπολικών και άλλων οχημάτων άμεσης βοήθειας
- Χάραξη δρομολογίων για φορτηγά με επικίνδυνες ουσίες
- Μελέτη επιπτώσεων στο οδικό δίκτυο από το «κλείσιμο» ενός δρόμου
- Χωροθέτηση υπηρεσιών κοινής ωφελείας (σχολεία, νοσοκομεία άλλες υπηρεσίες)
- Σχεδιασμός συλλογής απορριμμάτων
- Χωροθέτηση καταστημάτων, υπηρεσιών ιδιωτικού τομέα
- Οριοθέτηση περιοχών ευθύνης (territories assignment)
- ΟΛΟΚΛΗΡΩΜΕΝΟ ΣΥΣΤΗΜΑ ΑΠΕΙΚΟΝΙΣΗΣ ΚΙΝΔΥΝΩΝ ΠΛΗΜΜΥΡΩΝ 34
- Δρομολόγηση στόλου φορτηγών για διανομές
- Προσδιορισμός περιοχών πωλήσεως

2.4 Τεχνολογίες

2.4.1 Αρχιτεκτονικές G.I.S

Η χαρακτηριστική δυνατότητα που παρέχουν τα GIS είναι αυτή της σύνδεσης της χωρικής με την περιγραφική πληροφορία (η οποία δεν έχει από μόνη της χωρική υπόσταση). Η τεχνολογία που χρησιμοποιείται για την λειτουργία αυτή βασίζεται:

- Είτε στο σχεσιακό (relational) μοντέλο δεδομένων, όπου τα περιγραφικά δεδομένα πινακοποιούνται χωριστά και αργότερα συσχετίζονται με τα χωρικά δεδομένα μέσω κάποιων μοναδικών τιμών που είναι κοινές και στα δύο είδη δεδομένων.
- Είτε στο αντικειμενοστραφές (object-oriented) μοντέλο δεδομένων, όπου τόσο τα χωρικά όσο και τα περιγραφικά δεδομένα συγχωνεύονται σε αντικείμενα, τα οποία μπορεί να μοντελοποιούν κάποια αντικείμενα με φυσική υπόσταση (π.χ. κατηγορία = "δρόμος", όνομα = "Πανεπιστημίου", γεωμετρία = "[X1,Y1],[X2,Y2]...", πλάτος = "20μέτρα").

Το αντικειμενοστραφές μοντέλο τείνει να χρησιμοποιείται όλο και περισσότερο σε εφαρμογές GIS εξαιτίας των αυξημένων δυνατοτήτων του σε σχέση με το σχεσιακό μοντέλο της δυνατότητας που παρέχει για την εύκολη και απλοποιημένη ΟΛΟΚΛΗΡΩΜΕΝΟ ΣΥΣΤΗΜΑ ΑΠΕΙΚΟΝΙΣΗΣ ΚΙΝΔΥΝΩΝ ΠΛΗΜΜΥΡΩΝ 27 μοντελοποίηση σύνθετων φυσικών φαινομένων και αντικειμένων με χωρική διάσταση.

Πολλές φορές η ολοκληρωμένη έννοια των GIS (integrated GIS concept) επεκτείνεται για να συμπεριλάβει τόσο τα δεδομένα (που αποτελούν ουσιαστικά τον πυρήνα τους), το λογισμικό και τον μηχανικό εξοπλισμό, όσο και τις διαδικασίες και το ανθρώπινο δυναμικό, που αποτελούν αναπόσπαστα τμήματα ενός οργανισμού, ο οποίος έχει σαν πρωταρχική του δραστηριότητα την διαχείριση πληροφορίας με την βοήθεια GIS.

2.4.2 Πως λειτουργεί ένα G.I.S:

• Συσχετισμός των πληροφοριών από διαφορετικές πηγές.

Ένα απ' τα βασικά χαρακτηριστικά των GIS είναι η δυνατότητα τους να «διασταυρώνουν» τις πληροφορίες από διαφορετικές πηγές. Αν για παράδειγμα μπορούσαμε να συνδέσουμε τις πληροφορίες βροχοπτώσεων της χώρας μας με αεροφωτογραφίες της, ίσως να μπορούσαμε να προσδιορίσουμε ποιοι υδροβιότοποι της χώρας μας στερεύουν συγκεκριμένες χρονικές περιόδους με σε ένα χρόνο. Βάση αυτής της λογικής λειτουργούν και τα συστήματα GIS, μόνο που μπορούν να συσχετίσουν τις πληροφορίες από πολύ περισσότερες και σε διάφορες μορφές πηγές, βοηθώντας μας στην ανάλυση τους.

Δε μπορούν όμως όλες ανεξαρτήτως οι πηγές πληροφοριών να χρησιμοποιηθούν από τα συστήματα GIS. Απαραίτητη προϋπόθεση είναι, τα δεδομένα των πηγών να καθορίζουν τη θέση του κάθε χαρακτηριστικού που περιγράφουν. Η θέση αυτή μπορεί να περιγραφεί από τις συντεταγμένες γεωγραφικού μήκους, πλάτους και υψόμετρου ή συστημάτων όπως Ταχυδρομικών Κωδικών ή χιλιομετρήσεων εθνικών οδών. Κάθε λοιπόν μεταβλητή που μπορεί να τοποθετηθεί γεωγραφικά αποτελεί αποδεκτή-χρήσιμη πληροφορία για τα Γεωγραφικά Συστήματα Πληροφοριών. Όπως αναφέραμε και παραπάνω τα GIS μπορούν να αξιοποιήσουν πληροφορίες διαφόρων μορφών, αφού τις μετατρέψουν σε επεξεργάσιμη πληροφορία για τα ίδια. Έτσι διαφορετικού είδους δεδομένα σε μορφή χάρτη μπορούν να εισαχθούν στο σύστημα.



Σχήμα 2.3. Πηγή Πληροφοριών – Χάρτης

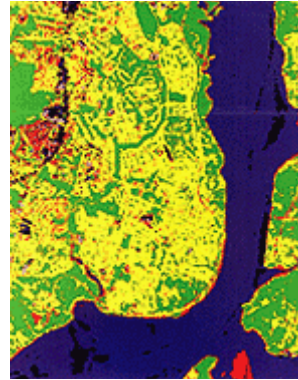


Σχήμα 2.4. Επεξεργάσιμη μορφή μετά την ανάλυση απ' το GIS (layer Οδικού δικτύου)

Ένα τέτοιο σύστημα μπορεί ακόμη να μετατρέψει ψηφιακές πληροφορίες που δεν είναι ακόμα σε μορφή χάρτη, σε μορφές που μπορεί να αναγνωρίσει και να χρησιμοποιήσει. Όπως για παράδειγμα ψηφιακές φωτογραφίες από δορυφόρους μπορούν να αναλυθούν και να παράγουν ένα χάρτη-στρώμα (layer) ψηφιακής πληροφορίας σχετικά με την κάλυψη του πράσινου στην περιοχή.



Σχήμα 2.5. Ψηφιακή Φωτογραφία από δορυφόρο



Σχήμα 2.6. Ανάλυση από GIS που περιγράφει την κάλυψη της χλωρίδας στην περιοχή(RAW DATA)

Ομοίως μια συλλογή πληροφοριών σε μορφή στατιστικών στοιχείων όπως η απογραφή ή οι υδρολογικές μετρήσεις ενός υδροβιότοπου μπορούν να μετατραπούν σε οπτικοποιημένες πληροφορίες για την αμεσότερη κατανόηση τους.

- **Συλλογή Δεδομένων**

Όπως προαναφέραμε τα γεωγραφικά συστήματα πληροφοριών μπορούν να δεχθούν πληροφορίες από διαφόρων μορφών πηγές. Όταν αυτή η πληροφορία είναι σε ψηφιακή μορφή, η τροφοδότηση του στο σύστημα είναι σχετικά απλή. Στην αντίθετη περίπτωση οι χάρτες πρέπει να ψηφιοποιηθούν για να μπορεί το σύστημα να τους αναγνωρίσει.

Αυτή η εργασία γίνεται με διάφορες μεθόδους. Για τη συλλογή των δεδομένων ενός Χάρτη για παράδειγμα, μπορεί να χρησιμοποιηθεί ένα ειδικά τροποποιημένο ποντίκι, που εισάγει τις συνταγμένες των διαφόρων χαρακτηριστικών (πόλεις, οδούς, λίμνες, κ.α.) του, στο σύστημα. Συσκευές σάρωσης (scanner) μπορούν επίσης να ψηφιοποιήσουν ορισμένων τύπων χάρτες. Ένα σύστημα GIS μπορεί να δώσει έμφαση στη χωρική σχέση μεταξύ των αντικειμένων που χαρτογραφούνται. Ενώ δηλαδή ένα απλό υπολογιστικό σύστημα χαρτογράφησης, θα παρουσιάσει το δρόμο ως μία απλή γραμμή, το GIS μπορεί πιθανόν να αναγνωρίσει τον δρόμο και ως σύνορο μεταξύ υγροτόπου και κατοικημένης περιοχής ή ως τον σύνδεσμο του δρόμου αυτού με ένα σοκάκι.



Σχήμα 2.7. Εισαγωγή Συντεταγμένων με τη βοήθεια χειροκίνητου ποντικιού



Σχήμα 2.8. Οι σαρωτές μπορούν να βοηθήσουν στην συλλογή των δεδομένων, χωρίς όμως 100% επιτυχία πάντα

Η συλλογή δεδομένων – η εισαγωγή δηλαδή των πληροφοριών στο σύστημα – είναι το πιο χρονοβόρο συστατικό των συστημάτων GIS. Και αυτό γιατί στο στάδιο αυτό πρέπει να προσδιοριστούν οι ταυτότητες των αντικειμένων του χάρτη καθώς και οι χωρικές τους σχέσεις. Και η επεξεργασία όμως των πληροφοριών που εισάγονται αυτόματα μπορεί τελικά να αποδειχθεί ομοίως χρονοβόρα διαδικασία, καθυστερώντας ακόμη περισσότερο τη διαδικασία συλλογής των δεδομένων. Ένα σκουπιδάκι για παράδειγμα στον σαρωτή (scanner) θα σαρωθεί με την ίδια λεπτομέρεια που θα σαρωθούν τα χαρακτηριστικά γνωρίσματα του χάρτη. Έτσι είναι πολύ πιθανό αυτή λανθασμένη πληροφορία να συνδέσει για παράδειγμα δυο δρόμους που στην πραγματικότητα δε συναντώνται πουθενά.

• Ενοποίηση Δεδομένων

Το μεγαλύτερο ίσως πλεονέκτημα των GIS είναι οι δυνατότητά τους να συνδέσουν και να ενοποιήσουν μεταξύ τους τα δεδομένα απ' τις διάφορες πηγές (αφού έχουν πρώτα εισαχθεί στο σύστημα σε κατανοητή μορφή)



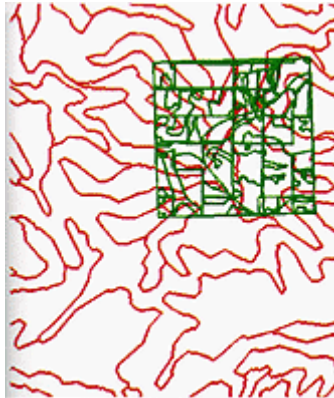
Σχήμα 2.9 Ενοποίηση Δεδομένων(Data Integration) είναι η σύνδεση των διάφορων μορφών πληροφορίας μέσω ενός συστήματος GIS

Σε τέτοιο βαθμό μάλιστα, που ενώ αρχικά το σύνολο των δεδομένων δε φαίνεται να οδηγεί σε κάποιο ασφαλές συμπέρασμα, τελικά, μετά την ανάλυσή τους απ' το σύστημα να προκύψουν νέες μεταβλητές που δεν ήταν εμφανείς εξ αρχής. Για παράδειγμα, χρησιμοποιώντας την τεχνολογία των Γεωγραφικών Συστημάτων Πληροφοριών και τις πληροφορίες μετρήσεων κατανάλωσης μιας εταιρείας ύδρευσης σε μια περιοχή, μπορούμε να εξομοιώσουμε την σηπτική απόθεση σε ένα γειτονικό υδρότοπο. Οι λογαριασμοί της εταιρείας ύδρευσης παρουσιάζουν το ποσό νερού που καταναλώνει κάθε διεύθυνση. Το ποσό νερού που χρησιμοποιεί ο κάθε πελάτης θα προσδιορίσει προσεγγιστικά την ποσότητα των υλικών που θα αποτεθούν τελικά στο σηπτικό σύστημα (υδρότοπος). Έτσι το σύστημα GIS θα μπορέσει να εντοπίσει τις περιοχές βαριάς σηπτικής απόθεσης.

• Προβολή και Καταγραφή

Η κλίμακα ενός χάρτη ιδιοκτησίας, μπορεί να είναι διαφορετική από αυτή ενός εδαφολογικού χάρτη, που χρησιμοποιούνται για την τροφοδότηση ενός GIS. Από αυτό και μόνο, αλλά και για αρκετούς ακόμη λόγους, τα δεδομένα των χαρτών που εισάγονται, πρέπει πρώτα να

δεχθούν την κατάλληλη επεξεργασία ώστε για τις ίδιες περιοχές, διαφορετικές πηγές δεδομένων να δίνουν τι ίδιες πληροφορίες πριν καταχωρηθούν στο σύστημα.



Σχήμα 2.10 Ένας χάρτης ιδιοκτησίας(πράσινος) πάνω από έναν εδαφολογικό(κόκκινος). Οι δύο χάρτες έχουν διαφορετική κλίμακα και προβολή

Μια θεμελιώδης μέθοδος δημιουργίας χαρτών (mapmaking) είναι η προβολή (projection). Η προβολή με τη μαθηματική της έννοια, σκοπό έχει να μεταφέρει τις τρισδιάστατες πληροφορίες της κυρτής γης, σε δισδιάστατο μέσο όπως το χαρτί ή την οθόνη του υπολογιστή. Διαφορετικοί τύποι προβολής χρησιμοποιούνται σε κάθε χάρτη, γιατί κάθε προβολή είναι κατάλληλη για συγκεκριμένη χρήση.

Παραδείγματος χάριν, μια προβολή που αντιπροσωπεύει ακριβώς τις μορφές των ηπείρων θα διαστρεβλώσει τα σχετικά μεγέθη τους. Μιας λοιπόν που τα GIS χρησιμοποιούν, είδη κατασκευασμένους χάρτες ως πηγές πληροφοριών, με διαφορετικές προβολές, είναι αναγκαίο να μετατραπούν όλοι σε μια κοινού τύπου προβολή πριν είναι διαθέσιμα για επεξεργασία και περαιτέρω ανάλυση. Η διαδικασία αυτή πραγματοποιείται κυρίως απ' τον Η/Υ με ελαφρά ανάμειξη του ανθρώπου.

• Δομές Δεδομένων

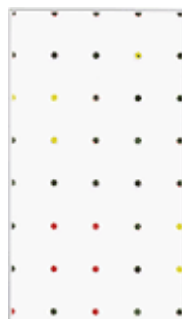
Ένα ακόμη πρόβλημα που προκύπτει με τη συλλογή πληροφοριών από διαφορετικού τύπου πηγές, είναι ο τρόπος με τον οποίο θα αποθηκευθούν τελικά στο σύστημα, ώστε να είναι δυνατή η παράλληλη επεξεργασία τους. Για να καλύψει λοιπόν τις ανάγκες συμβατότητας μεταξύ των πληροφοριών του, το GIS θα πρέπει να έχει δυνατότητες μετατροπής από μία δομή σε άλλη.

Τα δεδομένα από τις ψηφιακές φωτογραφίες των δορυφόρων που έχουν “ερμηνευτεί” (interpreted) απ' το σύστημα σε ένα χάρτη χρήσης του εδάφους, βρίσκεται στο GIS σε μορφή raster. Τα δεδομένα raster αποτελούνται από πίνακες κοινών στοιχείων που περιγράφουν τις αξίες των στοιχείων τους. Ένα παράδειγμα θα ήταν η ταξινόμηση κάλυψης εδάφους.

• Μοντελοποίηση Δεδομένων

Είναι δύσκολο να συσχετίσεις χάρτες υγρασιμότητας με τις ποσότητες βροχοπτώσεων που

καταγράφονται, σε διαφορετικά σημεία, όπως αεροδρόμια, τηλεοπτικού σταθμού και σχολεία. Ένα GIS, ωστόσο μπορεί να απεικονίσει δισδιάστατα και τρισδιάστατα χαρακτηριστικά της γήινης επιφάνειας, του υπεδάφους και της ατμόσφαιρας. Παραδείγματος χάριν, ένα GIS μπορεί γρήγορα να παραγάγει έναν χάρτη με τις γραμμές που δείχνουν τα ποσά βροχοπτώσεων



Σχήμα 2.11 Σημεία με γνωστές ποσότητες βροχοπτώσεων κλιμακωτά από υψηλό(κίτρινο) σε χαμηλότερο(μαύρο) επίπεδο.

Ένας τέτοιος χάρτης μπορεί να θεωρηθεί ως χάρτης περιγράμματος βροχοπτώσεων. Πολλές περίπλοκες μέθοδοι μπορούν να υπολογίσουν προσεγγιστικά τα χαρακτηριστικά των επιφανειών από ένα περιορισμένο αριθμό σημαδεμένων μετρήσεων. Ένας δισδιάστατος χάρτης περιγράμματος που δημιουργείται από τις μετρήσεις των σημείων βροχοπτώσεων, μπορεί να επικαλυφθεί και να αναλυθεί με οποιοδήποτε άλλο χάρτη σε ένα G.I.S. που καλύπτει την ίδια περιοχή.

•Τεχνολογία -Hardware:

Τα G.I.S για να καταφέρουν να αντεπεξέλθουν στις απαιτήσεις που έχουμε και να συλλέξουμε της πληροφορίες που μας χρειάζονται και να της αναλύσουμε πρέπει να διαθέτουν:

Υπολογιστές: Πρέπει να έχουμε υπολογιστές αξιόπιστους και σύγχρονης τεχνολογίας για να μην παρουσιαστούν προβλήματα με την εφαρμογή κατά την διάρκεια λειτουργίας της.

Ασύρματα δίκτυα: Για να έχουμε γρήγορη και έγκυρη ενημέρωση της εφαρμογή με τις πληροφορίες που χρειαζόμαστε πρέπει το δίκτυο που χρησιμοποιούμε να είναι αξιόπιστο και ταχύτητας τουλάχιστον 512 kbps.

Μετρητές: Αυτό το υλικό κομμάτι χρειάζεται μόνο αν η εφαρμογή μας προβλέπει τη μέτρηση ή τον έλεγχο ενός γεωγραφικού σημείου. Οι μετρητές πρέπει να είναι σύγχρονοι έτσι ώστε να συνδέονται ψηφιακά με τον υπολογιστή και θα μεταφέρουν τις πληροφορίες αλλά να είναι ακριβείς και αξιόπιστοι έτσι ώστε να μην μας δίνουν λάθος πληροφορίες.

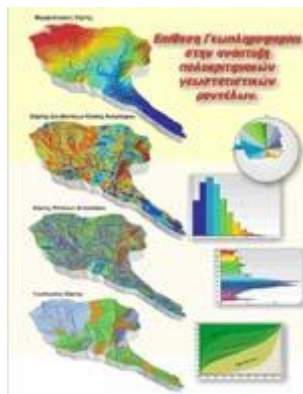
Εξοπλισμός

- Κεντρική μονάδα
- Μονάδες αποθήκευσης
- Περιφερειακές μονάδες εισόδου π.χ. ψηφιοποιητής-σαρωτής
- Περιφερειακές μονάδες εξόδου π.χ. Οθόνη, plotter.

2.5 Κατηγορίες G.I.S.

Όσον αφορά τη δομή τους, τα GIS διακρίνονται σε δύο κατηγορίες: στα VECTOR GIS και στα Raster GIS.

Παραδείγματα της πρώτης κατηγορίας είναι μεταξύ άλλων η απεικόνιση γραμμικών στοιχείων (π.χ. δρόμων), πολυγώνων (π.χ. ιδιοκτησίες), σημείων (π.χ. θέσεις γεωτρήσεων). Παραδείγματα της δεύτερης κατηγορίας είναι η απεικόνιση της πληροφορίας υπό μορφή κελιών (cells) (π.χ. υψόμετρο, απόσταση, αριθμό). Ο συνδυασμός αυτών των δύο κατηγοριών των GIS αποτελεί ένα ισχυρότατο εργαλείο στην ανάπτυξη μοντέλων λήψης απόφασης.



Σχήμα 2.12 VectorGIS



Σχήμα 2.13 Raster GIS

2.5.1 Vector G.I.S.

Η εφαρμογή των Vector-GIS συνίσταται κατά κύριο λόγο στην απεικόνιση της χωρικής πληροφορίας υπό μορφή γραμμικών στοιχείων, όπως σημείων - γραμμών - πολυγώνων.

Οι εφαρμογές των Vector-GIS είναι πολλαπλές, όπως στην Τοπική Αυτοδιοίκηση, στην Ανάπτυξη, στον Σύγχρονο Σχεδιασμό των πόλεων, στον Τουρισμό, στις Συγκοινωνίες, στα Δίκτυα Κοινής Ωφέλειας, στην Εκπαίδευση, στη Δημόσια Πληροφόρηση, στο Κτηματολόγιο, τον Περιβαλλοντικό Σχεδιασμό, μεταξύ άλλων.

Μερικές από τις δυνατότητες των Vector-GIS είναι στους ακόλουθους τομείς:

- Χαρτογραφία.
- Καταγραφή και απεικόνιση των πολεοδομικών συγκροτημάτων των πόλεων
- Καταγραφή και απεικόνιση αστικών, πολιτιστικών, κοινωνικών υποδομών.
- Διαχείριση οδικού και μεταφορικού δικτύου
- Διαχείριση δικτύων κοινής ωφέλειας, όπως δίκτυα ύδρευσης, αποχέτευσης, άρδευσης, κλπ
- Εύρεση συντομότερης διαδρομής μεταξύ δύο σημείων (π.χ. εύρεση κοντινότερης υπηρεσίας).

- Περιβαλλοντικό Σχεδιασμό (απεικόνιση χρήσεων γης και φυτοκάλυψης, μοντελοποίηση υδρογραφικού δικτύου, διαχείριση επιφανειακής υδρολογίας λεκανών).



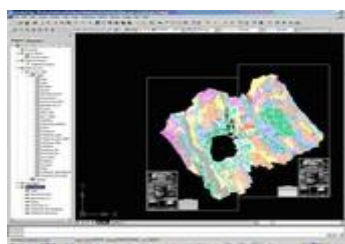
Σχήμα 2.14

2.5.2 Raster G.I.S.

Ιδιαίτερο ενδιαφέρον παρουσιάζει σήμερα η ανάπτυξη μοντέλων σε RASTER-GIS και η εφαρμογή τους στον Περιβαλλοντικό Σχεδιασμό.

Οι πληροφορίες που μας παρέχουν τα RASTER-GIS συνδυαζόμενα με δεδομένα πεδίου χρησιμοποιούνται για την εκπόνηση μελετών, στατιστικών αναλύσεων, γεωτεχνικών, γεωλογικών και περιβαλλοντολογικών μοντέλων με στόχο τον προσδιορισμό επιπτώσεων (impact analysis) καθώς και τη συνεχή παρακολούθηση φαινομένων, τα οποία μεταβάλλονται χωρικά και χρονικά (monitoring).

Τα RASTER-GIS στηρίζονται στην ανάπτυξη πολυκριτηριακών μοντέλων, όπου τα επιμέρους κριτήρια διαβαθμίζονται αρχικά με έναν συντελεστή βαρύτητας και στη συνέχεια από την επάλληλη τοποθέτησή τους και μέσω στατιστικών μεθοδολογιών ασαφούς λογικής (fuzzy analysis) προκύπτει το τελικό μοντέλο.



Σχήμα 2.15



Σχήμα 2.16

Συγκεκριμένα, ο συνδυασμός των Vector-GIS και Raster-GIS καθώς και των Δορυφορικών εικόνων, μπορεί να εφαρμοστεί στους ακόλουθους τομείς:

1. Χωροθετήσεις έργων, όπως:

- Χ.Υ.Τ.Α.

- Φράγματα.
- Κοιμητήρια.

2. Ανάπτυξη μελετών και μοντέλων φυσικού κινδύνου, όπως:

- Μοντέλα μορφολογίας αναγλύφου.
- Μοντέλα Διάβρωσης και παραγωγής φερτών υλικών.
- Μοντέλα Πλημμυρικού Κινδύνου.
- Μοντέλα Κατολισθήσεων, τα οποία στηρίζονται σε γεωλογικά και γεωτεχνικά κριτήρια.
- Μοντέλα Σεισμικής Επικινδυνότητας, όπου υπολογίζεται η χωρική κατανομή της αναμενόμενης εδαφικής επιτάχυνσης, μετάθεσης, ταχύτητας, σεισμικής έντασης,
- Μοντέλα Τρωτότητας υπόγειων υδροφόρων από επιφανειακές εστίες μόλυνσης

3. Ανάπτυξη μοντέλων υδατικού ισοζυγίου, όπου συνδυάζονται μοντέλα επιφανειακής και υπόγειας υδρολογίας

4. Χάραξη οδικού δικτύου περιβαλλοντικός αποδεκτού

2.6 Τα G.I.S στο Διαδίκτυο

Το διαδίκτυο έχει επηρεάσει όλες τις επιστήμες και κάθε ανθρώπινη δραστηριότητα και δεν θα μπορούσε φυσικά να μην εισχωρήσει και στον τομέα των Γεωγραφικών Πληροφοριακών Συστημάτων. Πλέον μπορούμε να εκμεταλλευτούμε τις δυνατότητες των GIS για δικτυακές εφαρμογές και όχι αποκλειστικά για stand-alone εφαρμογές. Μπορούμε να δηλαδή να κάνουμε απομακρυσμένα χρήση υπάρχουσας γεωγραφικής πληροφορίας και να μην είμαστε αναγκασμένοι να συλλέγουμε κάθε φορά εκ των προτέρων ότι γεωγραφική πληροφορία χρειαζόμαστε και να επιβαρυνόμαστε το επιπλέον οικονομικό αλλά και χρονικό κόστος

Η διάθεση γεωγραφικών δεδομένων σε περισσότερες από μια εφαρμογές μέσω διαδικτύου συνεπάγεται στην ταχύτερη και οικονομικότερη υλοποίηση, αλλά παράλληλα έχουμε και περισσότερη προστιθέμενη αξία για τις εφαρμογές μας. Εφόσον τα δεδομένα μας βρίσκονται συγκεντρωμένα σε ένα σημείο, μπορούν εύκολα και άμεσα να ενημερώνονται, και αντίστοιχα οι εφαρμογές μας να είναι πιο ευέλικτες όσον αφορά στην επέκταση και αναβάθμισή τους.

Παράλληλα με τη διαδεδομένη χρήση του διαδικτύου μέσω κινητών τηλεφώνων και φορητών συσκευών, γίνεται εφικτή η εκμετάλλευση των υπηρεσιών των Γεωγραφικών Πληροφοριακών Συστημάτων σε συσκευές με μικρή υπολογιστική δύναμη και δεν περιορίζεται η χρήση τους σε ογκώδεις υπέρ-υπολογιστές. Πλέον μπορεί ο καθένας με χαμηλό κόστος, αλλά κυρίως άμεσα και εύκολα εν κινήσει χρήση των υπηρεσιών των G.I.S, όπως να λάβει οδηγίες πλοήγησης (navigation), ενημέρωση για τα σημεία ενδιαφέροντος γύρω από την περιοχή στην οποία βρίσκεται, παρακολούθηση μετακινούμενων οχημάτων(τηλεματική) κλπ.

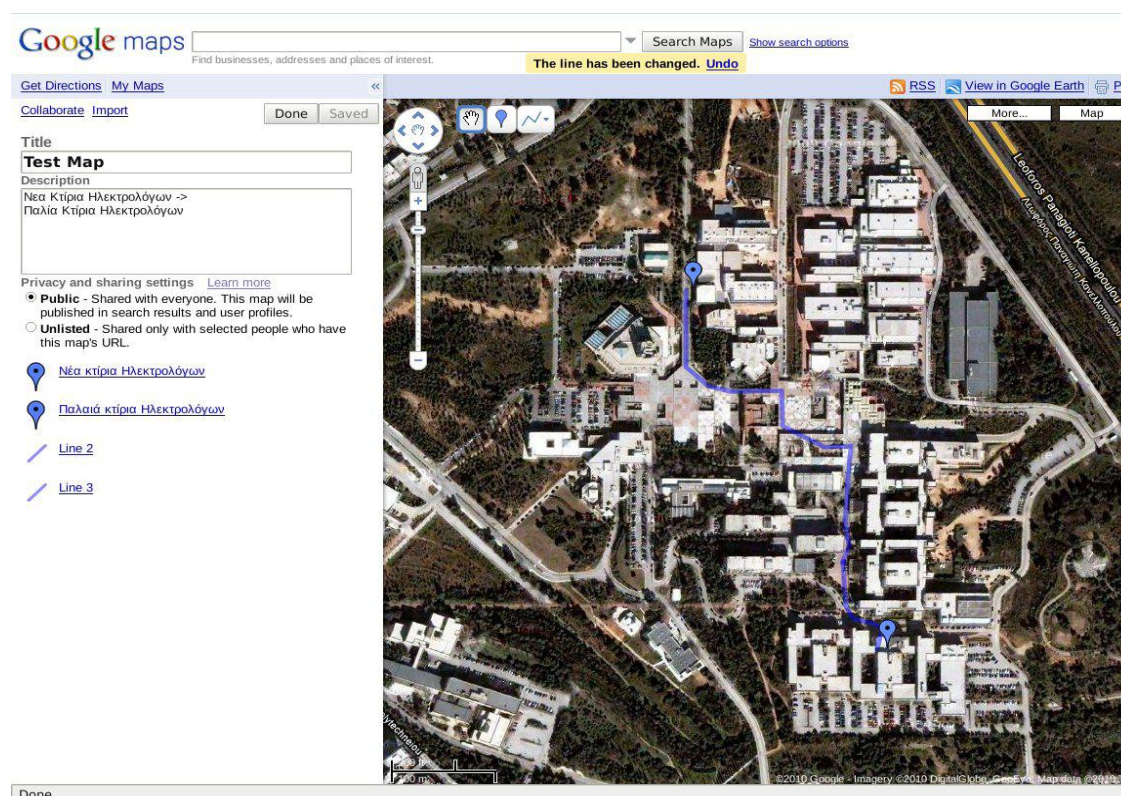
2.7 Google Maps

Το Google Maps είναι μια ελεύθερη υπηρεσία διαδικτυακής χαρτογράφησης (webmapping) που παρέχεται από την Google και τροφοδοτεί με χάρτες υψηλής ανάλυσης πολλές ιστοσελίδες τρίτων μέσω του Google Maps API. Προσφέρει οδικό δίκτυο, δορυφορικές φωτογραφίες, λειτουργία για σχεδιασμό διαδρομών, αλλά και λειτουργία για τον εντοπισμό επιχειρήσεων σε πολλές χώρες σε όλο τον κόσμο.

Με τους χάρτες Google είναι δυνατή η πλοήγηση του χρήστη στο χάρτη με το σύρσιμο του ποντικιού, αλλά και η αναζήτηση κάποιας γεωγραφικής θέσης βάσει κάποιας διεύθυνσης, ενός ταχυδρομικού κώδικα ή κάποιας λεκτικά ορισμένης περιοχής.

Ακόμη, παρέχουν οδηγίες πλοήγησης, παραθέτοντας στο χρήστη μια λίστα με μεμονωμένες κινήσεις για το πώς να φτάσει στον προορισμό του, μαζί με μια εκτίμηση του χρόνου που απαιτείται για αυτή την απόσταση.

Το Google Maps δίνει την δυνατότητα στους χρήστες να δημιουργήσουν τον δικό του Geoblog τοποθετώντας σημεία στο χάρτη με κείμενο και σχεδιάζοντας με γραμμές ο χρήστης την διαδρομή του. Παράδειγμα μια διαδρομής καταχωρημένης στο Google Map GeoBlog ακολουθεί στην εικόνα .



Σχήμα 2.17

Με αυτό το τρόπο μπορούμε εύκολα μέσω της Google να δούμε χάρτες από όλο το κόσμο και να τους αξιοποιήσουμε όπως εμείς θέλουμε . Με το Google maps έχουμε την δυνατότητα να τους χρησιμοποιήσουμε και στα GIS μέσω της εφαρμογής API που θα δούμε στη συνέχεια.

2.8 Application Programming Interface (API)

2.8.1 Τι είναι το API;

Το API αλλιώς και διεπαφή προγραμματισμού εφαρμογών (για συντομία: Διεπαφή ή ΑΠΙ από το αγγλικό API, Application Programming Interface) το γραφικό περιβάλλον (interface) που ένα υπολογιστικό σύστημα, βιβλιοθήκη (library: μια συλλογή από έτοιμα υποπρογράμματα που χρησιμοποιείται για την ανάπτυξη λογισμικού) ή διαδικτυακή εφαρμογή/υπηρεσία (web application) παρέχει, προκειμένου να επιτρέψει να γίνουν προς αυτό αιτήσεις από άλλα προγράμματα και/ή ανταλλαγή δεδομένων.

Πρόκειται για ένα σύνολο από ρουτίνες (routines), δομές δεδομένων (data structures), κλάσσεις αντικειμένων (object classes) και/ή πρωτόκολλα (protocols) που υποστηρίζουν την ανάπτυξη λογισμικού και υπηρεσιών βασισμένων σε μια πρωτογενή (υπολογιστικό σύστημα, βιβλιοθήκη, εφαρμογή).

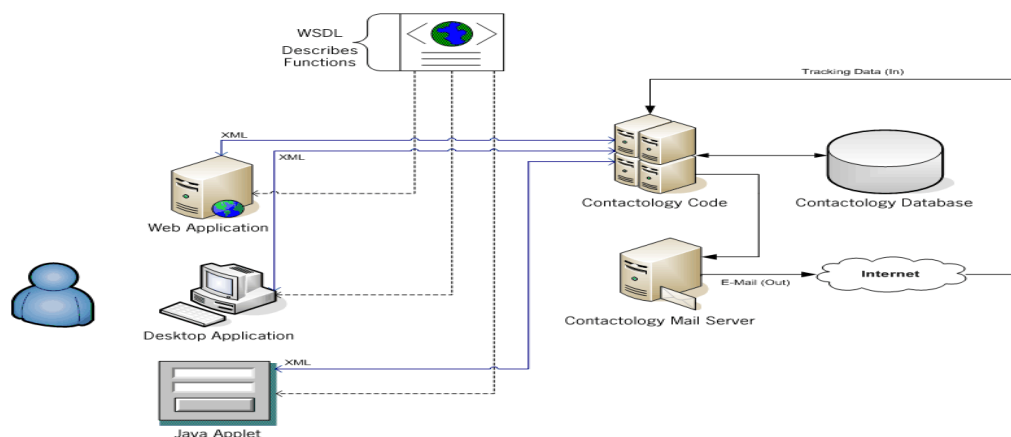
Για να μιλήσουμε με συγκεκριμένα παραδείγματα, είναι ένα κανάλι επικοινωνίας και αλληλεπίδρασης ενός συστήματος (πχ μιας web υπηρεσίας όπως η εφαρμογή on-line κατοχύρωσης domain names της dnHost) με μια εξωτερική πηγή (πχ η ιστοσελίδα ή η web εφαρμογή του Μεταπωλητή της dnHost), μέσω συγκεκριμένων προγραμματιστικών εντολών (που ορίζονται από έναν οδηγό εντολών και χρήσης του API).

2.8.2 Πως λειτουργεί ένα API;

Ένας από τους πρωταρχικούς σκοπούς μιας διεπαφής API, είναι να διατυπώσει το σύνολο των λειτουργιών-υπηρεσιών που μπορεί να παρέχει ένα λειτουργικό σύστημα, μια διαδικτυακή υπηρεσία κλπ σε άλλα προγράμματα χωρίς να γίνεται κάποια αναφορά στον κώδικα που υλοποιεί αυτές τις υπηρεσίες.

Το API απλά ορίζει με ποιές εξωτερικές εντολές θα παρέχει την αμφίδρομη επικοινωνία με την υπηρεσία που θέλει να συνδεθεί προς αυτό χωρίς να αποκαλύπτει τον πηγαίο κώδικά του. Αυτές οι εντολές που καλούν το API, μπορεί να είναι γραμμένες σε συγκεκριμένη προγραμματιστική γλώσσα (Language-dependent API) ή σε οποιαδήποτε γλώσσα (Language-independent), αναλόγως του τύπου του API.

Π.χ. το ταχυδρομείο παρέχει και την υπηρεσία της αποστολής γραμμάτων. Σου λέει τους κανόνες που θα ακολουθήσεις για να κάνεις το αίτημά σου (φόρμα διεύθυνσης παραλαβής, γραμματόσημο, κ.λπ.) αλλά το πώς θα υλοποιηθεί αυτό σου το αίτημα είναι δουλειά ενός ολόκληρου μηχανισμού ανθρώπων υλικού εν πολλοίς αθέατα στον χρήστη της υπηρεσίας. Δηλαδή στο παράδειγμα του ταχυδρομείου η διεπαφή είναι οι υπηρεσίες που παρέχει στους πελάτες και οι οποίες είναι γραμμένες συνήθως σε ένα φυλλάδιο. Το φυλλάδιο αυτό είναι η διεπαφή του ταχυδρομείου προς τους πελάτες.



Σχήμα 2.18

2.9 Google Maps API

Το Google Maps API είναι μια διεπαφή προγραμματισμού εφαρμογών (Application Programming Interface – API) της Google το οποίο δίνει τη δυνατότητα ενσωμάτωσης ενός δυναμικού χάρτη Google σε ιστοσελίδες (web σελίδες) με τη χρήση της javascript. Το API παρέχει έναν αριθμό εργαλείων για το χειρισμό χαρτών και την προσθήκη περιεχομένου σε αυτούς μέσω διαφόρων υπηρεσιών, όπως και στην επίσημη ιστοσελίδα του Google Maps (<http://maps.google.com>), επιτρέποντας έτσι τη δημιουργία εύρωστων εφαρμογών που εκμεταλλεύονται γεωγραφικές πληροφορίες. Το Google Maps API είναι μια δωρεάν υπηρεσία, διαθέσιμη για οποιοδήποτε ιστότοπο (web site) που είναι ελεύθερος για το κοινό. Το μόνο που χρειάζεται είναι ένα κλειδί API (API key) που μπορεί να ζητηθεί εφόσον υπάρχει λογαριασμός στη Google και αυτομάτως συνδέεται με το λογαριασμό. Κάθε κλειδί είναι έγκυρο μόνο για ένα συγκεκριμένο όνομα τομέα. Βασικό στοιχείο οποιασδήποτε Google Maps API εφαρμογής είναι ο ίδιος ο χάρτης, ο οποίος φορτώνεται, αρχικοποιείται με συγκεκριμένες διαστάσεις (ύψος και πλάτος), κέντρο (γεωγραφικό πλάτος, γεωγραφικό μήκος), επίπεδο εστίασης και τύπο (κανονικό, δορυφορικό, υβριδικό). Στη συνέχεια είναι δυνατή η αλληλεπίδραση με το χάρτη μέσω διαφόρων μηχανισμών όπως είναι τα events, τα controls, τα overlays αλλά και συναρτήσεις επεξεργασίας των χαρακτηριστικών του, τα οποία αναλύονται παρακάτω:

2.9.1 Γεγονότα (Events)

Το Google Maps API εμπεριέχει δηλώσεις γεγονότων (events) τα οποία εμφανίζονται μέσω των φυλλομετρητών ιστοσελίδων (web browsers) πάνω σε αντικείμενα του Google Maps. Ας σημειωθεί ωστόσο ότι τα γεγονότα αυτά είναι διαφορετικά από τα standard γεγονότα πάνω στο Μοντέλο Αντικειμένου Εγγράφου (Document Object Model – DOM), το οποίο είναι μια προγραμματιστική διεπαφή που επιτρέπει σε προγράμματα και εκτελέσιμα σενάρια δράσης (scripts) την ενημέρωση του περιεχομένου και του στυλ όλου του εγγράφου με δυναμικό τρόπο. Έτσι το Google Maps API δημιουργεί και διαχειρίζεται δικά του γεγονότα ενώ το Μοντέλο Αντικειμένου Εγγράφου δημιουργεί άλλα γεγονότα και τα επεξεργάζεται σύμφωνα με το μοντέλο αντικειμένου που χρησιμοποιεί ο συγκεκριμένος φυλλομετρητής. Επιπλέον, το Google Maps API έχει το πλεονέκτημα ότι προσφέρει μηχανισμούς που ακούνε (listen) και απαντάνε (respond) στα DOM γεγονότα ανεξάρτητα από τις ιδιαιτερότητες του κάθε

φυλλομετρητή. Παραδείγματα γεγονότων του Google Maps API είναι το κλικ του ποντικιού, το διπλό κλικ, η κίνηση του ποντικιού τα οποία σηματοδοτούν την έναρξη κάποιων διαδικασιών που περιγράφονται στον κώδικα javascript. Εξάλλου η javascript είναι μια scripting γλώσσα που «καθοδηγεί γεγονότα» (“event driven”) εξελισσόμενα στους φυλλομετρητές.

2.9.2 Ρυθμιστές (Controls)

Όπως μπορεί να διαπιστώσει κανείς και στη σελίδα του Google Maps (<http://maps.google.com>) υπάρχουν διάφορα στοιχεία στη διεπιφάνεια με το χρήστη (user interface) που επιτρέπουν την αλληλεπίδρασή του χρήστη με το χάρτη. Τα στοιχεία αυτά είναι γνωστά ως ρυθμιστές (controls) και μπορούν να περιληφθούν στις Google Maps εφαρμογές σε οποιαδήποτε σημείο πάνω στο χάρτη. Τα κυριότερα είναι:

- **GLargeMapControl3D**: Τοποθετεί ένα μεγάλο τρισδιάστατο κουμπί ελέγχου εστίασης και κίνησης του χάρτη, όπως χρησιμοποιείται τώρα στη σελίδα του Google Maps. Εμφανίζεται στην πάνω αριστερή γωνία του χάρτη αυτόματα.
- **GLargeMapControl**: Τοποθετεί ένα μεγάλο απλό κουμπί ελέγχου εστίασης και κίνησης του χάρτη. Εμφανίζεται στην πάνω αριστερή γωνία του χάρτη αυτόματα.
- **GSmallMapControl** : Τοποθετεί ένα μικρό κουμπί ελέγχου εστίασης και κίνησης του χάρτη. Εμφανίζεται στην πάνω αριστερή γωνία του χάρτη αυτόματα.
- **GSmallZoomControl3D**: Τοποθετεί ένα μικρό τρισδιάστατο κουμπιέλεγχου εστίασης (χωρίς έλεγχο κίνησης) του χάρτη. Εμφανίζεται στην πάνω αριστερή γωνία του χάρτη αυτόματα.
- **GSmallZoomControl**: Τοποθετεί ένα μικρό κουμπί ελέγχου εστίασης (χωρίς έλεγχο κίνησης) του χάρτη που χρησιμοποιείται στα μικρά παράθυρα που εμφανίζονται στο χάρτη Google για επεξήγηση των οδικών οδηγιών.
- **GScaleControl**: Τοποθετεί μια κλίμακα με στάδια μεγέθυνσης τουχάρτη.
- **GMapTypeControl**: Τοποθετεί κουμπιά που επιτρέπουν στο χρήστη ναδιαλέγει μεταξύ τύπων του χάρτη (όπως δορυφορικός, εδαφικός κ.ά.)
- **GHierarchicalMapTypeControl**: Τοποθετεί κουμπιά επιλογών (checkboxes) για ιεραρχική προτίμηση των τύπων του χάρτη.
- **GOverviewMapControl**: Τοποθετεί μια πτυσσόμενη επισκόπηση του χάρτη στη γωνία της οθόνης.
- **GNavLabelControl**: Τοποθετεί μια δυναμική ετικέτα που επισημαίνει τη «διεύθυνση» της συγκεκριμένης περιοχής χάρτη ανάλογα με το επίπεδο εστίασης.

Επιπλέον υπάρχει η δυνατότητα δημιουργίας ειδικών ρυθμιστικών στοιχείων από τον προγραμματιστή.

2.9.3 Overlays

Τα overlays είναι αντικείμενα πάνω στο χάρτη που αντιστοιχούν σε συγκεκριμένες συντεταγμένες και γι' αυτό μετακινούνται ανάλογα όταν μετακινείται ή αλλάζει εστίαση ο χάρτης. Χρησιμοποιούνται για το σχεδιασμό και την επίδειξη σημείων, γραμμών ή ολόκληρων περιοχών. Το Google Maps API διαθέτει πολλά είδη overlays, τα οποία αναλύονται παρακάτω:

- **Σημεία**

Αναπαριστώνται με τη χρήση δεικτών (markers) πάνω στο χάρτη και συχνά προβάλλουν μια σχετική εικόνα στο σημείο που επιδεικνύουν. Οι markers είναι σχεδιασμένοι να είναι διαδραστικοί, δηλαδή επάνω τους πραγματοποιούνται γεγονότα, όπως ένα κλικ ή δεξί κλικ, και συχνά προβάλλεται παράλληλα με αυτούς ένα παράθυρο πληροφοριών για το σημείο.

Μια επιπλέον ιδιότητα τους είναι ότι μπορεί να είναι συρόμενοι (draggable) από ένα σημείο σε άλλο στο χάρτη. Επίσης δυνατή είναι η επεξεργασία και αλλαγή της εικόνας (icon) που αντιστοιχεί σε κάθε δείκτη. Η εικόνα αποτελείται από μια εικόνα στο προσκήνιο και μια άλλη ως σκιά τοποθετημένες σε συγκεκριμένες θέσεις μεταξύ τους. Παρότι οι δείκτες έχουν μεγάλη χρησιμότητα, η προσθήκη πολλών από αυτούς σε ένα χάρτη Google μπορεί να έχει αρνητική επίδραση στην ταχύτητα απεικόνισης του χάρτη (map rendering) ειδικά σε συγκεκριμένα επίπεδα εστίασης. Λύση σε αυτό δίνει ο Διαχειριστής Δεικτών (Marker Manager), ο οποίος έχει την ικανότητα να παρακολουθεί ποιοί δείκτες είναι ορατοί σε ποιά επίπεδα εστίασης και να στέλνει μόνο τους απαραίτητους στο χάρτη για σχεδιαστικούς σκοπούς.

Δηλαδή προσθέτει ή αφαιρεί δυναμικά δείκτες από το χάρτη με αποτέλεσμα την αύξηση της ταχύτητας απεικόνισης του χάρτη και της μείωσης των άχρηστων οπτικών στοιχείων.

- **Γραμμές**

Αναπαριστώνται με τη χρήση των polylines (που ουσιαστικά είναι μια ακολουθία συνδεδεμένων σημείων) πάνω στο χάρτη, για τις οποίες μπορεί να καθοριστεί χρώμα, πάχος και επίπεδο διαφάνειας. Χωρίζονται σε τρεις κατηγορίες: τις γραμμικές (drawing), τις γεωδαισικές (geodesic) και τις κωδικοποιημένες (encoded). Οι γραμμικές polylines εμφανίζονται ως ευθύγραμμα τμήματα στο χάρτη και για την απεικόνιση τους το Google Maps API εκμεταλλεύεται πιθανές σχεδιαστικές ικανότητες του φυλλομετρητή, αλλιώς γίνεται αίτηση για μια εικόνα polyline από τους εξυπηρετητές (servers) της Google. Οι γεωδαισικές polylines εμφανίζονται ως καμπυλωτές γραμμές λαμβάνοντας υπόψιν τους την καμπυλότητα της γης. Οι κωδικοποιημένες polylines είναι συνήθως μεγαλύτερες και πιο περίπλοκες από τις υπόλοιπες και αποτελούνται από πολλά ευθύγραμμα τμήματα που συνδέονται με τυχαίο τρόπο μεταξύ τους. Γι' αυτό το λόγο απαιτούν περισσότερη μνήμη και χρόνο για να απεικονισθούν. Έχουν όμως το πλεονέκτημα ότι μπορούν να καθορίζουν πόσο λεπτομερής θα είναι μια polyline σε ένα δοσμένο επίπεδο εστίασης καθώς δίνουν τη δυνατότητα να αγνοούνται τμήματα της γραμμής σε συγκεκριμένες ομάδες επιπέδων εστίασης.

- **Περιοχές**

Αναπαριστώνται με τη χρήση είτε πολυγώνων (polygons) πάνω στο χάρτη, αν η περιοχή έχει τη μορφή αυθαίρετου σχήματος, είτε με τη χρήση ground overlays, αν η περιοχή έχει τη

μορφή ορθογώνιου σχήματος. Τα πολύγωνα είναι παρόμοια με τις polylines αφού και αυτά είναι μια ακολουθία σημείων, προσθέτοντας ότι σχηματίζουν κύκλο και μπορούν να πάρουν οποιαδήποτε μορφή. Τα ground overlays είναι συχνά χρήσιμα για περιοχές που τα άκρα τους στοχεύουν κάθετα σε σημεία στο χάρτη.

- **Ο ίδιος ο χάρτης**

Αναπαρίσταται με τη χρήση του tile overlay, κάτι το οποίο μπορεί να τροποποιηθεί μέσω του Google Maps API. Τα διαθέσιμα tiles δεν καλύπτουν όλες τις περιοχές σε όλα τα επίπεδα εστίασης. Για παράδειγμα πολλές περιοχές στον Ειρηνικό Ωκεανό δεν εμφανίζονται σε υψηλά επίπεδα εστίασης, ενώ το Μανχάταν προσφέρει πολύ λεπτομερείς εικόνες στον δορυφορικό τύπο χάρτη. Αξίζει να σημειωθεί ότι στο μικρότερο επίπεδο εστίασης (επίπεδο 0), ένα tile αναπαριστά ολόκληρη τη γη. Αν N είναι το επίπεδο εστίασης, τότε το Google Maps διαιρεί το tile ολόκληρης της γης σε 4N tiles και προβάλλει το κατάλληλο.

- **Το παράθυρο πληροφοριών (info window)**

Αποτελεί ένα ειδικό overlay. Προστίθεται στο χάρτη αυτόματα και σε κάθε χάρτη αντιστοιχεί ένα μόνο αντικείμενο παραθύρου πληροφοριών.

2.9.4 Προσφερόμενες υπηρεσίες του Google Maps API

Πρόσθετες υπηρεσίες τις οποίες προσφέρει το Google Maps API είναι η μετατροπή πραγματικών διευθύνσεων ανά τον πλανήτη σε γεωγραφικές συντεταγμένες (geocoding), κάτι το οποίο είναι χρήσιμο στην τοποθέτηση δεικτών, αλλά και το αντίστροφο (reverse geocoding). Επιπλέον το Google Maps API είναι εξοπλισμένο με κρύφη μνήμη (cache memory) από τη μεριά του πελάτη (client), η οποία αποθηκεύει διάφορες απαντήσεις σε μετατροπές διευθύνσεων, επιτρέποντας ταχύτερη εξυπηρέτηση σε περίπτωση που ζητηθεί η ίδια μετατροπή διεύθυνσης στο μέλλον. Μια άλλη υπηρεσία είναι η άποψη από δρόμο (street view), η οποία παρέχει πανοραμική θέα 360ο περιοχών για τις οποίες υπάρχουν διαδραστικές εικόνες. Κάθε οπτική της εικόνας χαρακτηρίζεται από την τοποθεσία και τον προσανατολισμό και είναι μοναδική. Επιπλέον υπηρεσία είναι η προσθήκη του Google Earth στις Maps API εφαρμογές, το οποίο προβάλλει τον χάρτη Google τρισδιάστατα.

Η μπάρα αναζήτησης Google είναι άλλο ένα χρήσιμο εργαλείο με το οποίο αναζητούνται πόλεις, πανεπιστήμια, επιχειρήσεις κ.ά. Η μπάρα αναζήτησης είναι φιλική προς το χρήστη και συμπεριλαμβάνει στα αποτελέσματα της διαφημίσεις με σκοπό το οικονομικό κέρδος αν προστεθεί στις Google Maps εφαρμογές. Δίνεται η δυνατότητα επιπλέον να περιληφθεί στην εφαρμογή ο υπολογισμός οδηγίων για τη μετάβαση από ένα μέρος σε ένα άλλο, χρησιμοποιώντας μια ποικιλία μέσων μεταφοράς).

Σύνοψη

Τα **G.I.S** είναι πληροφοριακά συστήματα που εξαπλώθηκαν παρά πόλη, όλο και περισσότεροι χρησιμοποιούν τη τεχνολογία αυτή. Μια τεχνολογία που δίνει λύση σε πολλά προβλήματα και διευκόλυνε τη ζωή πολλών ανθρώπων. Η ψηφιακή χαρτογράφηση διευκόλυνε πολλούς τομείς να ανατηθούν και να δημιουργήσουν νέες υπηρεσίες.

Το **Google maps** είναι μια εφαρμογή της Google που με την ψηφιακή χαρτογράφηση που έγινε σε όλο τον πλανήτη γη έδωσε την δυνατότητα της περιήγηση σε αυτών από μια οθόνη υπολογιστή και την δυνατότητα να εξελιχθούν υπηρεσίες βασισμένες σε αυτή την δωρεάν υπηρεσία. Μια τέτια υπηρεσία είναι και αυτή που θα προσπαθήσαμε να φτιάξουμε.

Τα **API** είναι διεπαφές που χρησιμοποιούνται σε διάφορες εφαρμογές και διευκολύνουν την λειτουργία τους στο interface τους. Το Google maps api χρησιμοποιείτε στους χάρτες της Google και δίνει τη δυνατότητα εισαγωγής ενός δυναμικού χάρτη στις σελίδες της Google.

Κεφάλαιο 3

3. Μεθοδολογία ολοκλήρωσης της εφαρμογής

3.1 Εισαγωγή

Στο κεφάλαιο αυτό θα αναφερθούμε στη μεθοδολογία που ακολουθήσαμε προκειμένου να ολοκληρώσουμε την Πτυχιακή Εργασία. Αρχικά πραγματοποιήθηκε μια μελέτη αγοράς ώστε να καταγραφεί η διείσδυση των GIS στην αγορά και στη συνείδηση του γενικού αστικού πληθυσμού. Μέσω ερωτηματολογίου έγινε η καταγραφή και εντοπίστηκαν τυχόν κοινωνικά εμπόδια για την εισαγωγή ως προϊόντος μιας εφαρμογής παρακολούθησης της οικίας.

Στη συνέχεια, χρησιμοποιώντας τις απαντήσεις στα ερωτηματολόγια αλλά και λαμβάνοντας υπόψη το πεδίο δραστηριοποίησης των περισσότερων επιχειρήσεων στην Ελλάδα, δημιουργήσαμε τις προδιαγραφές μιας δυναμικής εφαρμογής επίβλεψης και ανίχνευσης ζημιάς σε μια οικία. Στόχος είναι κατά την μεθοδολογία ανάπτυξης να καταγραφούν οι διαδικασίες και οι εξαρτήσεις μεταξύ των πόρων και των χρηστών.

Σε αυτό το κεφάλαιο θα παρουσιαστεί μόνο ότι πραγματοποιήθηκε σχετικά με την έρευνα αγοράς, μιας και η μεθοδολογία – ροή ανάπτυξης του λογισμικού είναι αρκετά εκτενής και θα δημιουργούσε δυσαναλογία στο κείμενο. Επομένως, επιλέχθηκε να παρουσιαστεί αυτόνομα στο αμέσως επόμενο κεφάλαιο.

3.2 Μελέτη αγοράς

3.2.1 Ποιες συγκεκριμένες εταιρίες παρέχουν λύσεις GIS

Εταιρείες, συμπεριλαμβανομένης της Metropolitan Life και Chase Manhattan Bank βασίζονται σε λογισμικό Γεωγραφικών Συστημάτων Πληροφοριών για να τους βοηθήσουν να βελτιώσουν τη λειτουργική αρτιότητα και την αποδοτικότητα.

3.2.2 Ποιές ανάγκες εξυπηρετούν

- Ασφάλειες

Πολλές ασφαλιστικές εταιρείες έχουν κάνει το GIS κεντρικό στοιχείο των δραστηριοτήτων τους. Το χρησιμοποιούν για την οπτικοποίηση, ανάλυση, και διανομή του κινδύνου. Εταιρείες όπως CHUBB το χρησιμοποιούν για τη διαχείριση του κινδύνου του χαρτοφυλακίου.



Σχήμα 3.1

- **Logistics Εφοδιαστική**

Εφοδιαστική διαχείριση απαιτεί τον προγραμματισμό των δραστηριοτήτων του στόλου διανομής του, τοποθεσίες διαδρομής, και χρονοδιαγράμματα. Η τηλεφωνική εταιρεία της Αργεντινής, Grupo Telefónica, χρησιμοποιεί την εφοδιαστική GIS για να δρομολόγησει λύση και να αποδειχθεί ότι έχει το δικαίωμα να αυξήσει τον αριθμό των καθημερινών παραδόσεων χωρίς την αύξηση των πόρων.



Σχήμα 3.2

- **Media**

Τα GIS χρησιμοποιείται από τα μέσα για τα πάντα, από την ανάλυση της κυκλοφορίας και την προσέλκυση διαφημιστών στη δημιουργία των χαρτών που χρησιμοποιούνται για το ίδιο το υλικό. Η κοινωνική χρήση λογισμικού GIS για να δημιουργήσουν ακριβείς χάρτες γρήγορα για περιοδικά, εφημερίδες, και online υπηρεσίες ειδήσεων. Οι GIS χάρτες μπορούν να βοηθήσουν τα ΜΜΕ να τηρούν ενήμερο το κοινό σχετικά με το κλείσιμο των δρόμων ή ανοίγματα και άλλες υπηρεσίες έκτακτης ανάγκης, .



Σχήμα 3.3

- **Real Estate Ακίνητα**

Όπως βλέπουμε στο σχήμα 4.4 πολλές εταιρίες επενδύσεων ακινήτων, κτηματομεσιτικά γραφεία και άλλες πολλές εξαρτώνται σε μεγάλο βαθμό από την ηλεκτρονική χαρτογράφηση. Πολλοί μεσίτες έχουν βρει μεγάλη επιτυχία στη χρήση του Διαδικτύου για την αγορά ακινήτων. Εταιρείες όπως η MTA Realty Advisors, Inc με τη χρήση GIS σε εμπορικά ακίνητα, ενώ στο REALTOR.com χρησιμοποιεί GIS για να φέρει χάρτες σε απευθείας σύνδεση με 1,2 εκατομμύρια ανθρώπους για ένα νέο σπίτι.



Σχήμα 3.4

- **Retail Business Retail Business**

Ως εκ τούτου, διευθυντές επιχειρήσεων, στρατηγικοί μάρκετινγκ, οι οικονομικοί αναλυτές, προγραμματιστές και επαγγελματίες βασίζονται όλο και περισσότερο στα GIS για την οργάνωση, ανάλυση, και την παρουσίαση των στοιχείων των επιχειρήσεων τους. Εταιρείες, όπως η Sears έχουν αποθηκεύσει εκατομμύρια δολάρια από τη διαχείριση των παραδόσεων με GIS. Γυμναστήριο Χρυσό για Επιχειρήσεις χρησιμοποιεί GIS να μελετήσει νέες τοποθεσίες. Οι μικρότερες εταιρείες, όπως η Ultra θαλάσσια καγιάκ της Santa Cruz, California, βρείτε χρήσιμες και οικονομικά αποδοτικές τις GIS υπηρεσίες για τη δημιουργία ελκυστικών και ακριβείς χάρτες για διαφημιστικό υλικό.



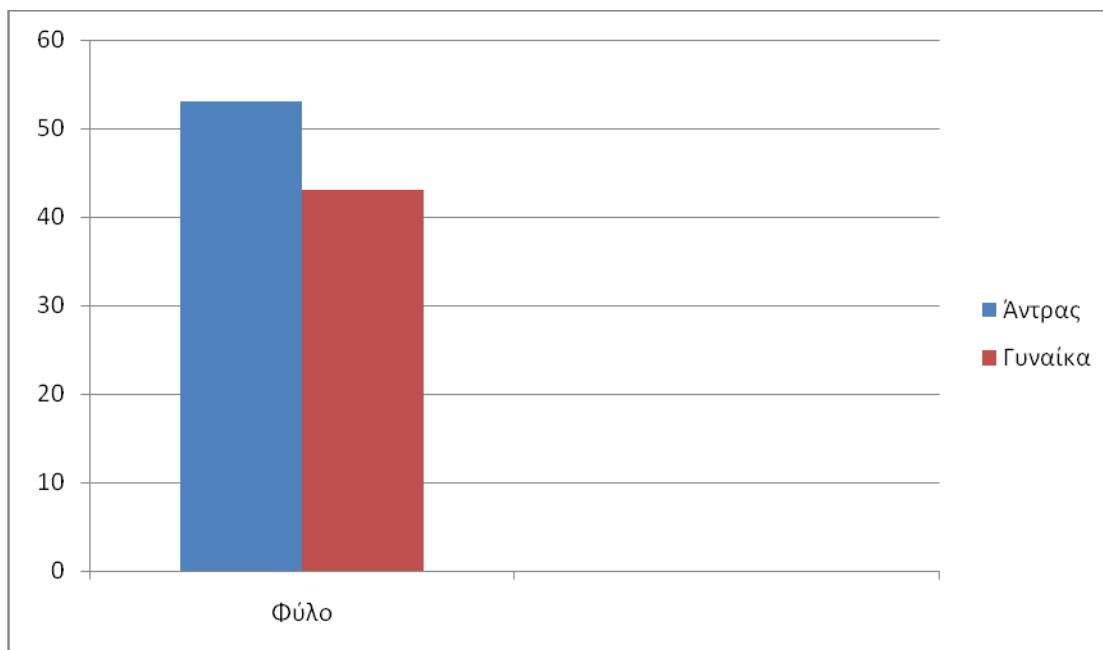
Σχήμα 3.5

3.3 Έρευνα με τη μέθοδο του ερωτηματολογίου - Συμπεράσματα

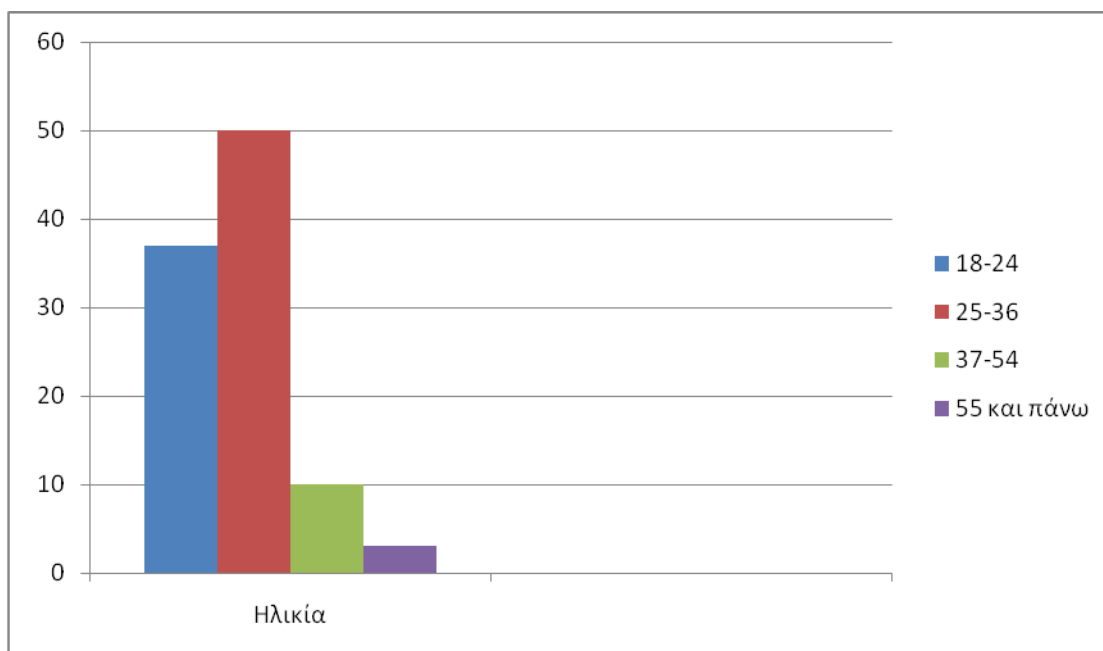
Όπως αναφέρθηκε, προκειμένου να καταγραφεί η διείσδυση της τεχνολογίας GIS σε μια ευρύτερη αγορά, αυτή της οικίας, ζητήθηκε από ένα πλήθος ατόμων να συμπληρώσουν ένα ερωτηματολόγιο. Μέσω αυτού του ερωτηματολογίου, θα ήταν δυνατή η καταγραφή της δυναμικής της αγοράς αλλά και οι πιθανές φοβίες (κυρίως λόγω κοινωνικών ταμπού) που μπορούν να οδηγήσουν σε αποτυχία ένα τέτοιο εγχείρημα, δηλαδή την ανάπτυξη της εφαρμογής στην οποία αναφέρεται η Πτυχιακή Εργασία.

Ο ερωτηματολόγιο μπορεί να βρεθεί στο Παράρτημα Α της Πτυχιακής Εργασίας. Παρακάτω, παραθέτουμε την ανάλυση, ανά ερώτημα, των αποκρίσεων όσων συμπληρώσαν το ερωτηματολόγιο. Αν και το δείγμα ήταν μικρό (30 άτομα) είναι αρκετά ενδεικτικό της τάσης της αγοράς. Το ηλικιακό εύρος καλύπτει όλες σχεδόν τις ηλικίες από 18 και πάνω.

Ερώτηση 1

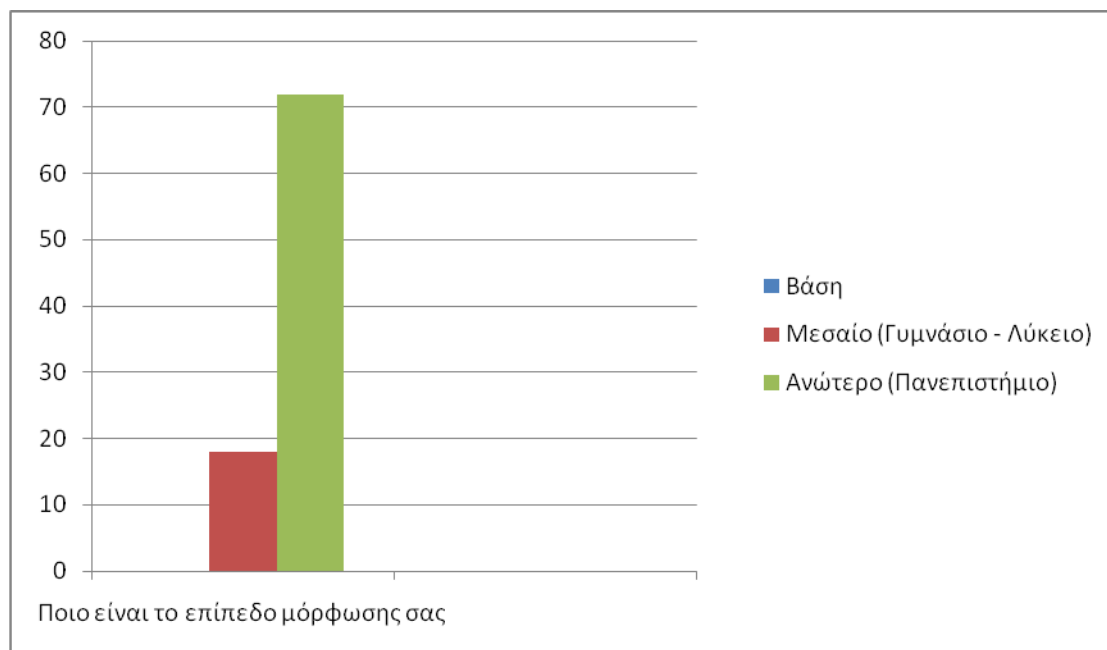


Ερώτηση 2



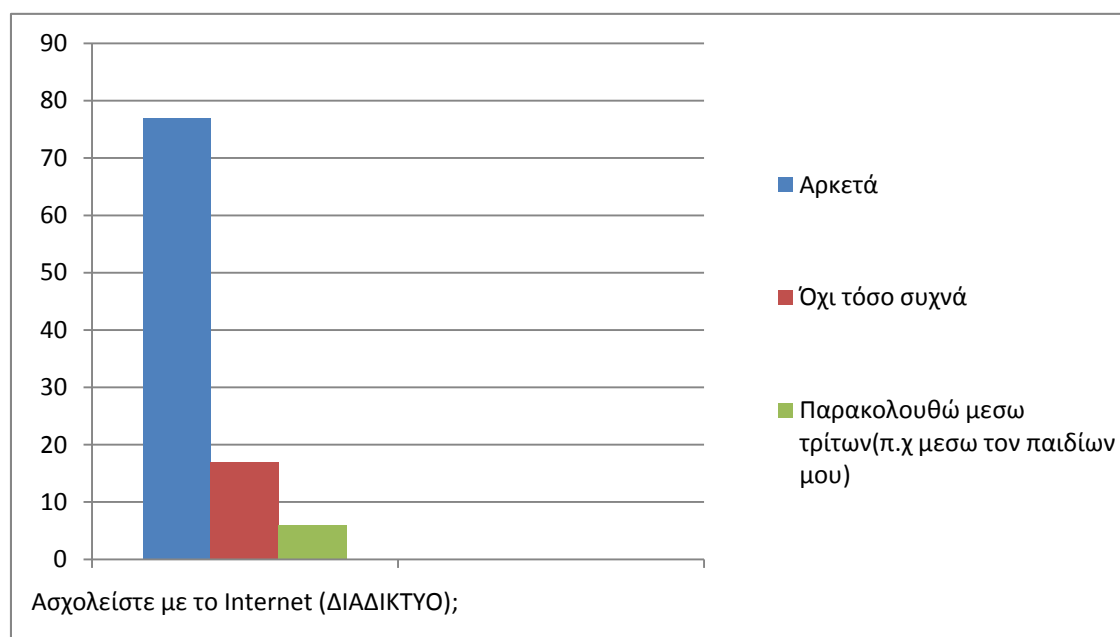
Οι πλειοψηφία των ερωτηθέντων είναι μεταξύ 18-36. Αυτό θα μας βοηθήσει να δούμε πώς αντιμετωπίζει η νέα γενιά τη συγκεκριμένη τεχνολογία και την εφαρμογή που σκοπεύουμε να κάνουμε.

Ερώτηση 3



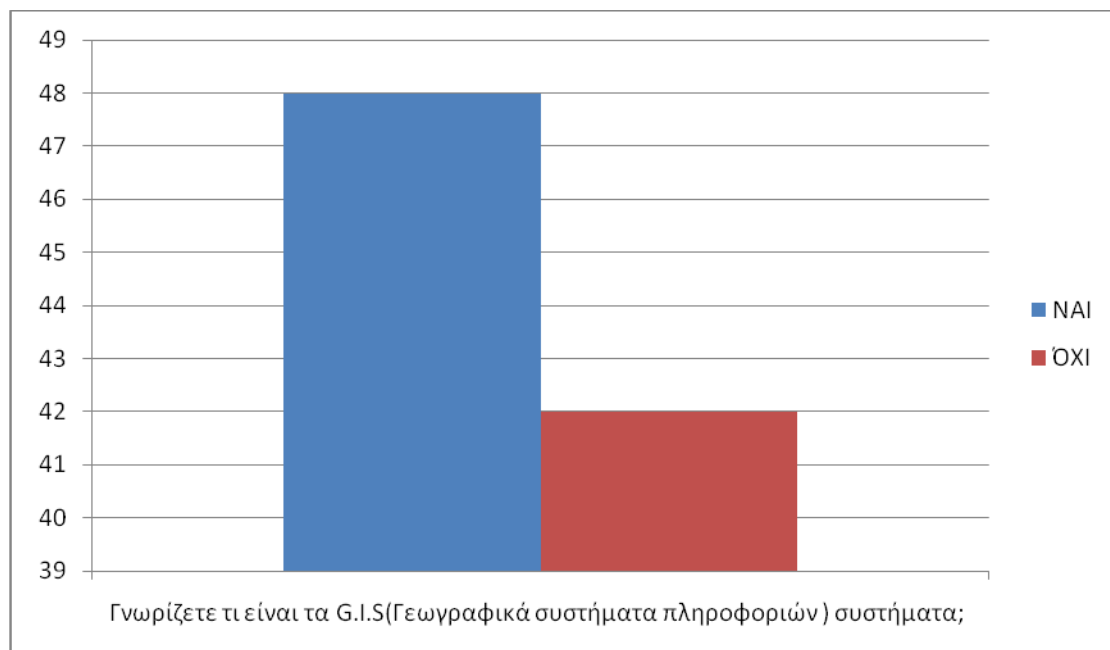
Πέραν του 70% των ερωτηθέντων έχουν το ανώτερο επίπεδο μόρφωσης.

Ερώτηση 4



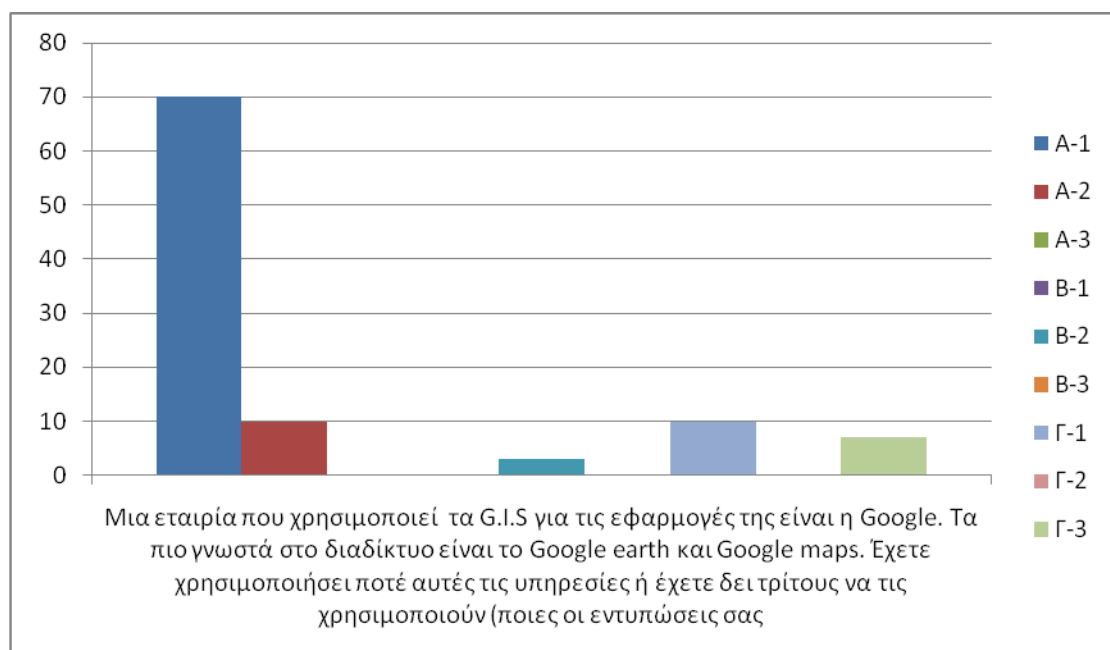
Το διαδίκτυο έχει μπει για τα καλά στη ζωή μας, οι περισσότεροι ερωτηθέντες ασχολούνται σε μεγάλο βαθμό με αυτό.

Ερώτηση 5



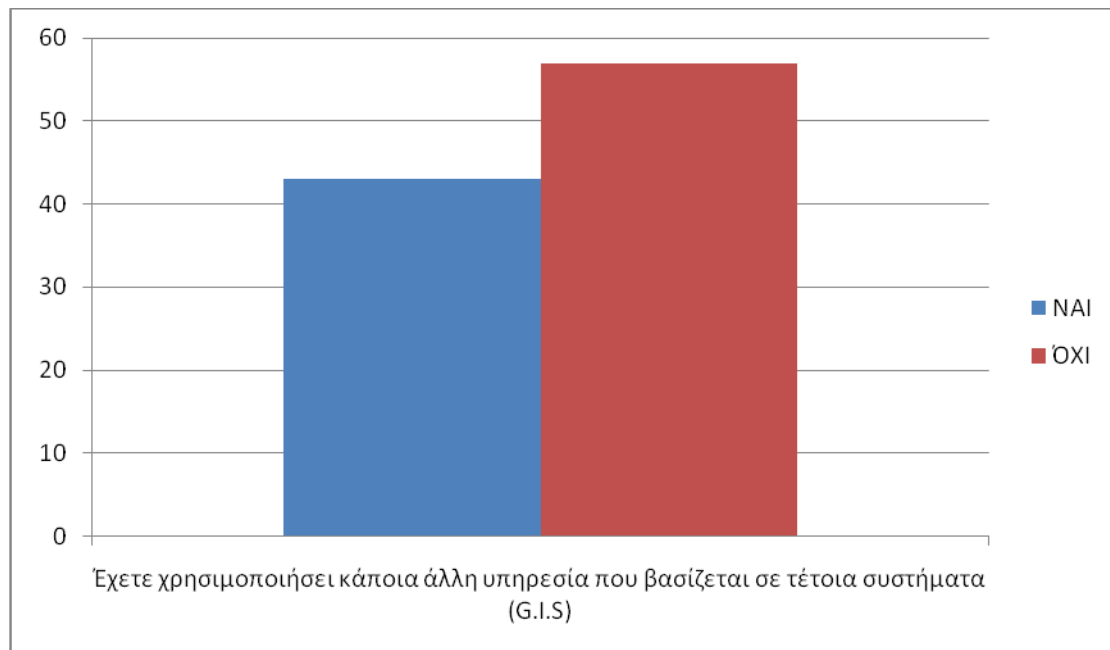
Ένα μεγάλο ποσοστό των ερωτηθέντων που φτενή σχεδόν το 50% ξέρουν τη είναι τα G.I.S

Ερώτηση 6



Οι περισσότεροι έχουν ασχοληθεί σε την εφαρμογή της Google το Google earth Google και maps. Τα βρίσκουν πολύ ενδιαφέρον. Η Google κατάφερε να κεντρίσει το ενδιαφέρον των χρηστών με αυτές τις εφαρμογές.

Ερώτηση 7



Φαίνεται πως οι περισσότεροι χρήστες ότι η μόνη υπηρεσία που χρησιμοποιούν με G.I.S ενώ ένα μεγάλο ποσοστό χρησιμοποίησε και άλλες υπηρεσίες.

Η ερώτηση περιείχε και μια τρίτη επιλογή

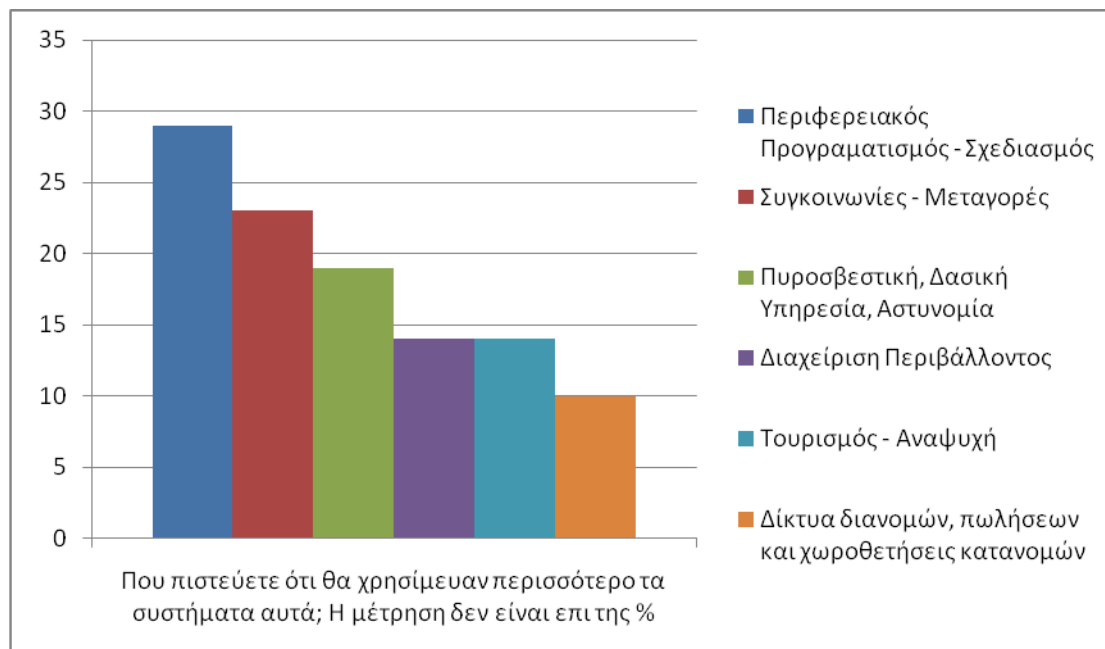
Αν ΝΑΙ ποια;(ΜΟΝΟ ΟΝΟΜΑΣΤΙΚΑ);

οι πιο δημοφιλείς απάντησης ήταν :

- GPS
- Arc maps
- Δίκτυα σεισμογράφων

Το GPS είναι μακράν η πιο δημοφιλής υπηρεσία μετά τις υπηρεσίες τις Google.

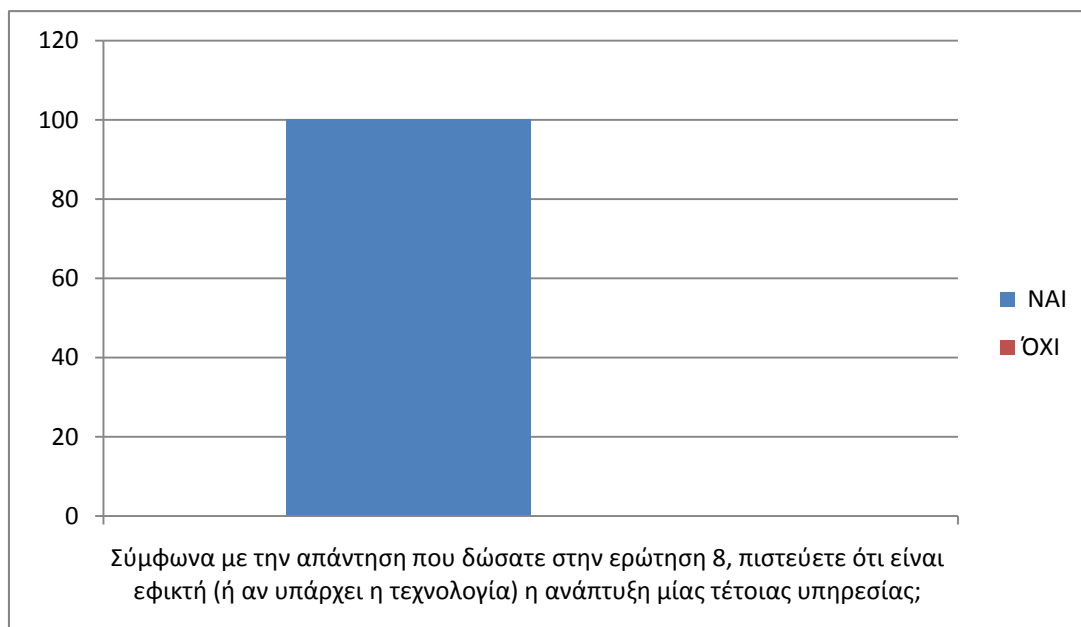
Ερώτηση 8



Τις περισσότερες ψήφους έχουν πάρει υπηρεσίες που σχετίζονται με την λειτουργικότητα (βλέπε περιφερικός προγραμματισμός – σχεδιασμό , συγκοινωνίες – μεταφορές) και την ασφάλεια (βλέπε πυροσβεστική, δασική Υπηρεσία, Αστυνομία , Διαχείριση Περιβάλλοντος). Αυτό δείχνει που πρέπει να κινηθεί η εφαρμογή για να γίνει αποδεχτεί από την αγορά.

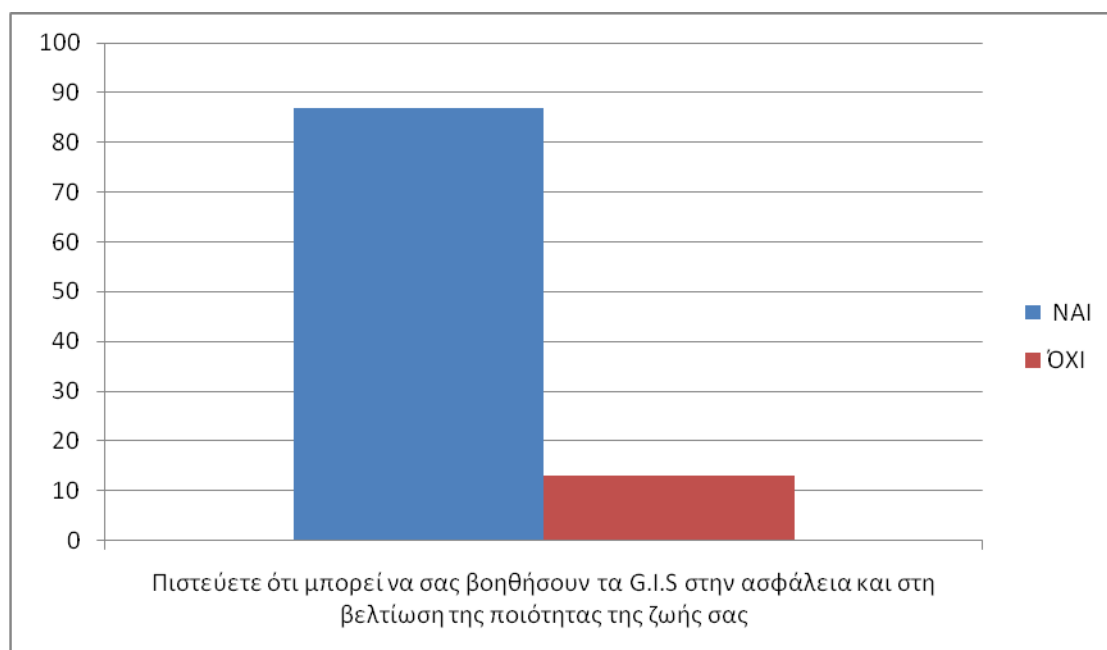
*Ο κάθε ερωτούμενος είχε να σημειώσει όσα συστήματα ήθελε.

Ερώτηση 9



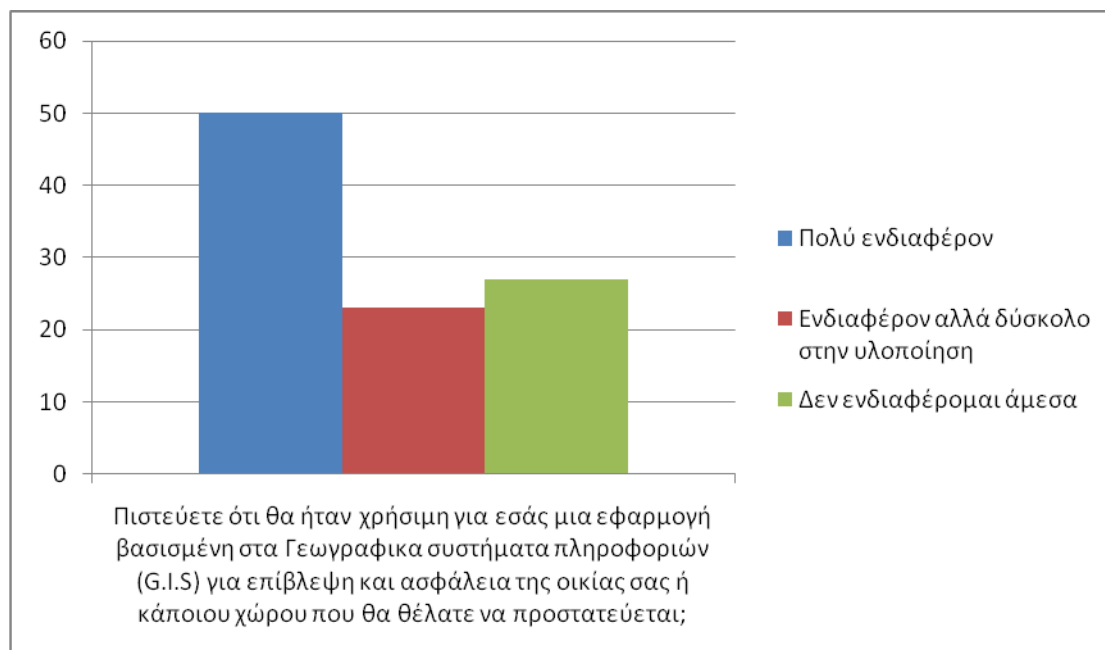
Όλοι πιστεύουν ότι είναι δυνατό να υλοποιηθούν οι εφαρμογές που επέλεξαν, ή υπάρχουν ήδη.

Ερώτηση 10



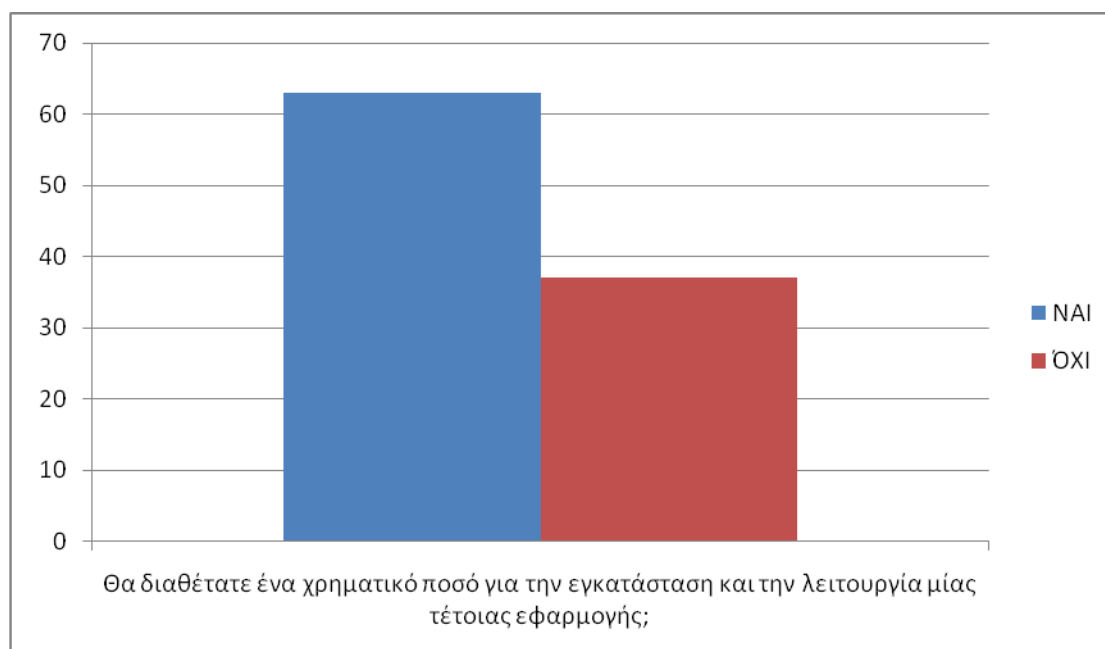
Πολύ ενθαρρυντική για την εφαρμογή μας η απάντηση στην πιο πανό ερώτηση, δείχνει την εμπιστοσύνη του κόσμου στα G.I.S

Ερώτηση 11



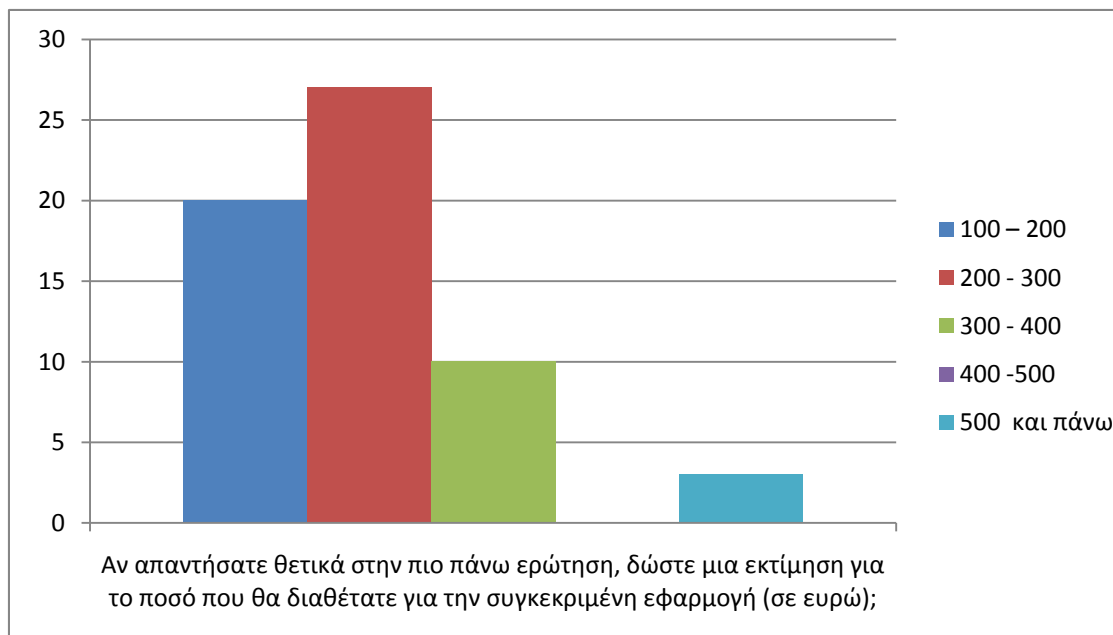
Ένα μεγάλο ποσοστό (50%) βρίσκει πολύ ενδιαφέρον η εφαρμογή μας. Ενώ ένα ποσοστό την βρίσκει ενδιαφέρον αλλά δεν πιστεύει στην υλοποίηση της.

Ερώτηση 12



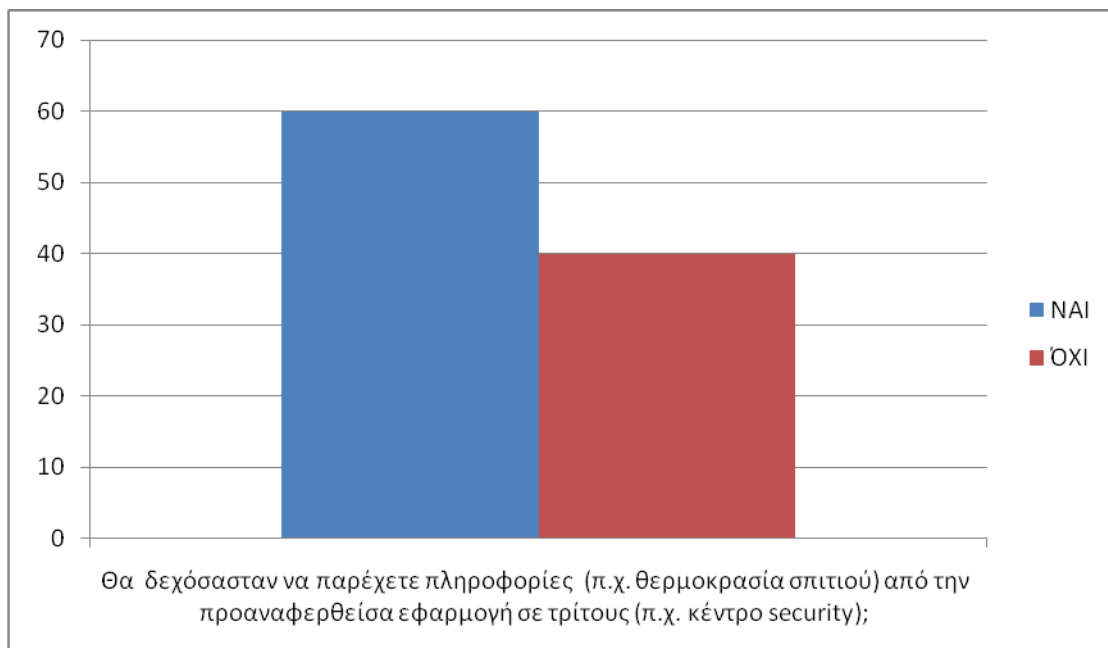
Είναι ενθαρρυντική ότι ένα μεγάλο ποσοστό θα έδινε ένα χρήματα.

Ερώτηση 13



Οι περισσότεροι ερωτηθέντες είναι διατίθενται να δώσουν ένα ποσό μεταξύ 200 – 300 ευρώ, ενώ το άμεσος επόμενο ποσό είναι 100 – 200. Σε πρώτο στάδιο κρίνετε ικανοποιητική η πρόθεση για το προϊόν

Ερώτηση 14



Ενθαρρυντικό είναι και το ποσοστό που θα δεχόταν να παρέχει συγκεκριμένες πληροφορίες, αυτό δείχνει ότι το αγοραστικό κοινό δεν φοβάται να δώσει πληροφορίες από την οικία του

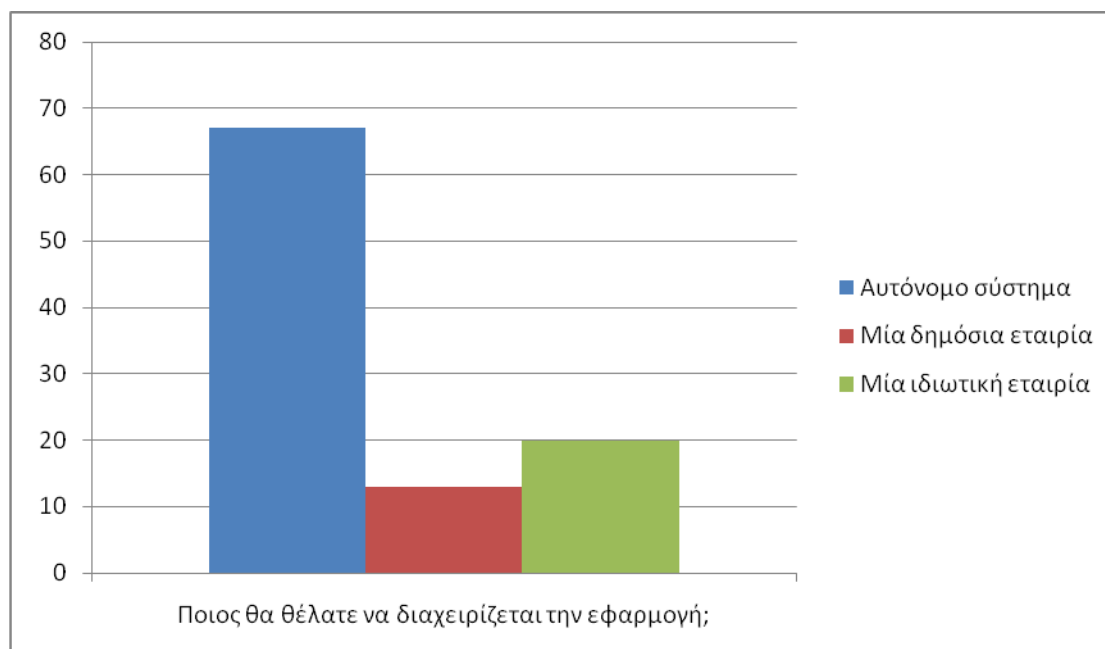
Ερώτηση 15

. Τι είδους και ποιές πληροφορίες θα δεχόσασταν να παρέχετε;

- Περιβάλλοντος
- Θερμοκρασίας- υγρασίας
- Κίνησης

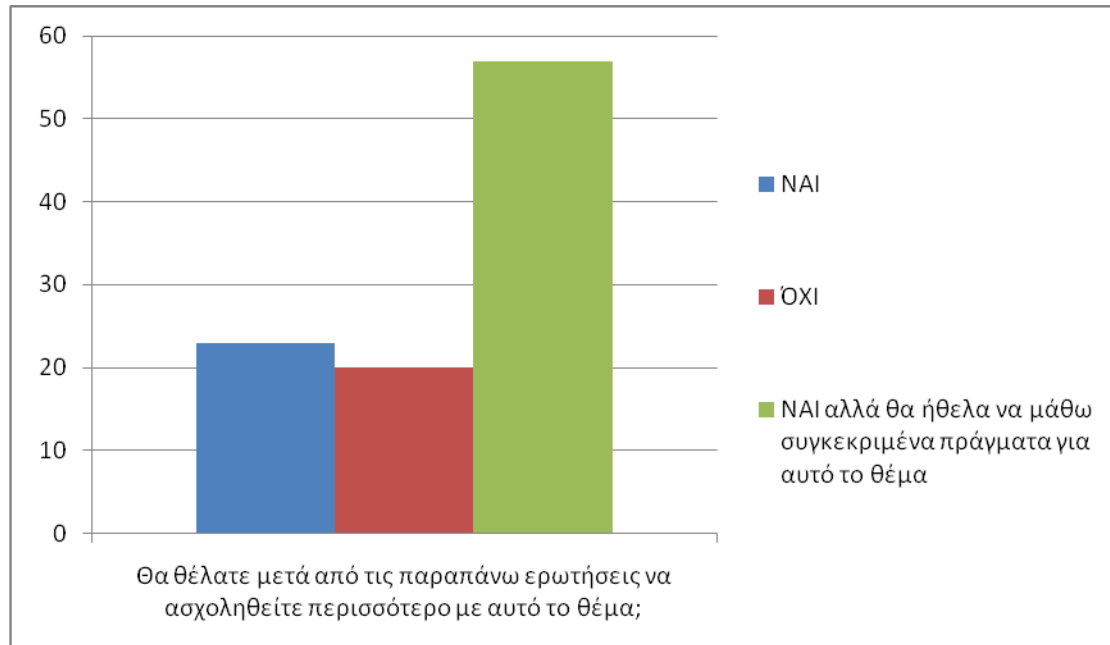
Οι περισσότεροι δέχονται να παρέχουν πληροφορίες περιβάλλοντος (π.χ. θερμοκρασίας – υγρασίας) και κίνησης. Αυτό είναι καλό για την εφαρμογή μας γιατί ο πρωταρχικός μας στόχος είναι να παρέχονται αυτές οι πληροφορίες.

Ερώτηση 16



Οι περισσότεροι προτιμούν μία αυτόνομη την εφαρμογή, αυτό μας δείχνει ότι δεν θέλουν κάποιον τρίτο να διαχωρίζει τις πληροφορίες.

Ερώτηση 17



Ενθαρρυντικό το ότι ένα μεγάλο ποσοστό θέλει περαιτέρω ενημέρωση για την εφαρμογή μας και τον ενδιαφέρει.

Συμπεράσματα Ερωτηματολογίου

Τα συμπεράσματα που βγάλαμε από το ερωτηματολόγιο θα μας φανούν πολύ χρήσιμα για τις προδιαγραφές που θα ορίσουμε στην εφαρμογή.

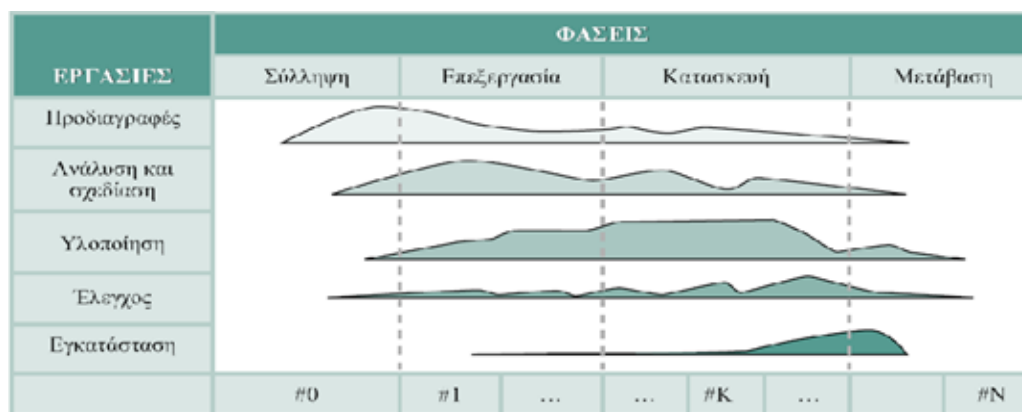
Τα σημαντικότερα στοιχεία που παίρνουμε από τα αποτελέσματα είναι ότι:

- Το διαδίκτυο έχει μπει για τα καλά στη ζωή μας.
- Η χρήση των G.I.S είναι αρκετά διαδεδομένη, είτε μέσω του Google είτε μέσω άλλων υπηρεσιών όπως το GPS.
- Η ασφάλεια των χώρων είναι στις προτεραιότητες του κόσμου
- Είναι διαφεντεμένη να δώσουν ένα ικανοποιητικό ποσό για να χρησιμοποιήσουν την εφαρμογή.
- Δεν είναι διστακτική να δώσουν πληροφορίες για το περιβάλλον της οικίας τους.
- Έδειξαν ενδιαφέρον για την εφαρμογή και θέλουν να μάθουν περισσότερα.

3.4 Μεθοδολογία υλοποίησης

Στόχος μας είναι η ολοκλήρωση του έργου: «**Ανάπτυξη μιας εφαρμογής επίβλεψης και ανίχνευσης ζημιών σε μια οικία με τη χρήση εργαλείων της Google**» που στο εξής θα αποκαλούμε: ASDDH (application for surveillance and detection of damage to homes).

Για την ολοκλήρωση του έργου θα ακολουθήσουμε την μεθοδολογία της ενοποιημένης προσέγγισης (unified process) ανάπτυξης λογισμικού. Ακριβώς για αυτό τον λόγο θα εξετάσουμε τέσσερις φάσεις ανάπτυξης όπως φαίνεται και από το παρακάτω σχήμα.



Σχήμα 3.6 Μια απεικόνιση της αναλογίας των εργασιών ανάπτυξης λογισμικού κατά της γενικές φάσεις της ενοποιημένης προσέγγισης.

Φάση σύλληψης: Σε αυτό το στάδιο θα εξετάσουμε πως λειτουργούν τα G.I.S , η υπηρεσία Google Maps και Google Api που θα χρησιμοποιήσουμε για ανάπτυξη του έργου. Καθώς και μία μελέτη αγοράς όπου θα δούμε πώς κινητή η αγορά με τα G.IS και την άποψη για την εφαρμογή που θέλουμε να φτιάξουμε μέσω ερωτηματολογίου Με βάση τα συμπεράσματα που θα βγάλουμε από την έρευνα μας θα καταλείψουμε στην προδιαγραφές και θα μας οδηγήσουν στον σχεδιασμό της εφαρμογής.

Φάση επεξεργασίας: Κατά τη φάση επεξεργασίας θα γίνει εκτίμηση των προδιαγραφών και απαιτήσεων της εφαρμογής μας και θα δημιουργηθούν όλα τα σχεδιαγράμματα ανάλυσης και σχεδίασης του. Εκτιμάτε πως θα στηριχθούμε στη μεθοδολογία UML αναπτύσσονται όλα τα σχεδιαγράμματα που η εφαρμογή απαιτεί. Στο τέλος αυτής της φάσης θα έχουμε δημιουργήσει ένα σύνολο κειμένων και μοντέλων παράστασης της εφαρμογής.

Φάση Κατασκευής : Σε αυτό το στάδιο θα περάσουμε στην υλοποίηση της εφαρμογής στο τεχνικό και προγραμματιστικό κομμάτι. Στο τέλος αυτής της φάσης θα έχουμε κατασκευή, εγκατάσταση και δοκιμή της εφαρμογής σε ένα τοπικό server, η οποία θα πλησιάσει κατά πολύ την τελική έκδοση του έργου.

Φάση Μετάβαση: Στη φάση αυτή θα γίνει εγκατάσταση και έλεγχος του λογισμικού στο τελικό σύστημα(server) ώστε να επαληθεύσουμε την ικανότητα των απαιτήσεων που καθορίσαμε και κατά συνέπεια να οδηγηθούμε στην διόρθωση και τροποποίηση τυχών μη επιθυμητών αποτελεσμάτων που θα μας οδήγηση στην τελική έκδοση της εφαρμογής μας. Στο τέλος αυτής της φάσης θα έχουμε την 1^η έκδοση της εφαρμογής μας.

Κεφάλαιο 4

4. Κύκλος Ανάπτυξης λογισμικού

4.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα ασχοληθούμε με το κύκλο ανάπτυξης του λογισμικού που έχουμε σκοπό να φτιάξουμε. Θα παρουσιάσουμε το μοντέλο ζωής του λογισμικού, τη διαδικασία ανάπτυξης, τη μεθοδολογία και τα εργαλεία. Δηλαδή της φάσεις από τις οποίες θα περάσει η εφαρμογή μας.



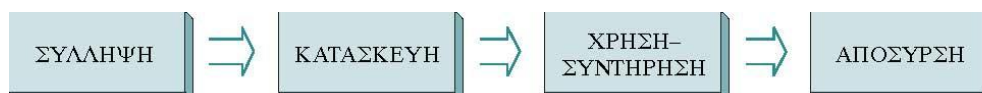
Σχήμα 4.1 Σχέσεις εννοιών στην ανάπτυξη του λογισμικού

4.2 Έννοιες για την ανάπτυξη λογισμικού

4.2.1 Μοντέλα κύκλου ζωής

Αυτό το σημείο της ανάπτυξης του λογισμικού είναι πολύ σημαντικό και συνάμα η βάση για υλοποιήσουμε την εφαρμογή μας. Ένα Μοντέλο Κύκλου Ζωής Λογισμικού είναι μια περιγραφή των δραστηριοτήτων και των επιμέρους φάσεων από τις οποίες διέρχεται μια εφαρμογή λογισμικού από τη σύλληψη μέχρι την απόσυρσή της, καθώς και των εργασιών που λαμβάνουν χώρα σε καθεμιά από τις φάσεις αυτές.

Κάθε εφαρμογή λογισμικού, από τη σύλληψη μέχρι την απόσυρσή της, διέρχεται από διάφορες φάσεις, σε καθεμιά εκ των οποίων πρέπει να γίνονται ορισμένες εργασίες ώστε να επιτυγχάνεται το επιθυμητό αποτέλεσμα. Σε μακροσκοπικό επίπεδο οι πολύ γενικές φάσεις είναι: σύλληψη, κατασκευή, χρήση / συντήρηση και απόσυρση και, όπως είναι εύκολα αντιληπτό, λαμβάνουν χώρα με τη σειρά αυτή. Μια εικόνα των γενικών αυτών φάσεων φαίνεται στο 4.2, που ακολουθεί:

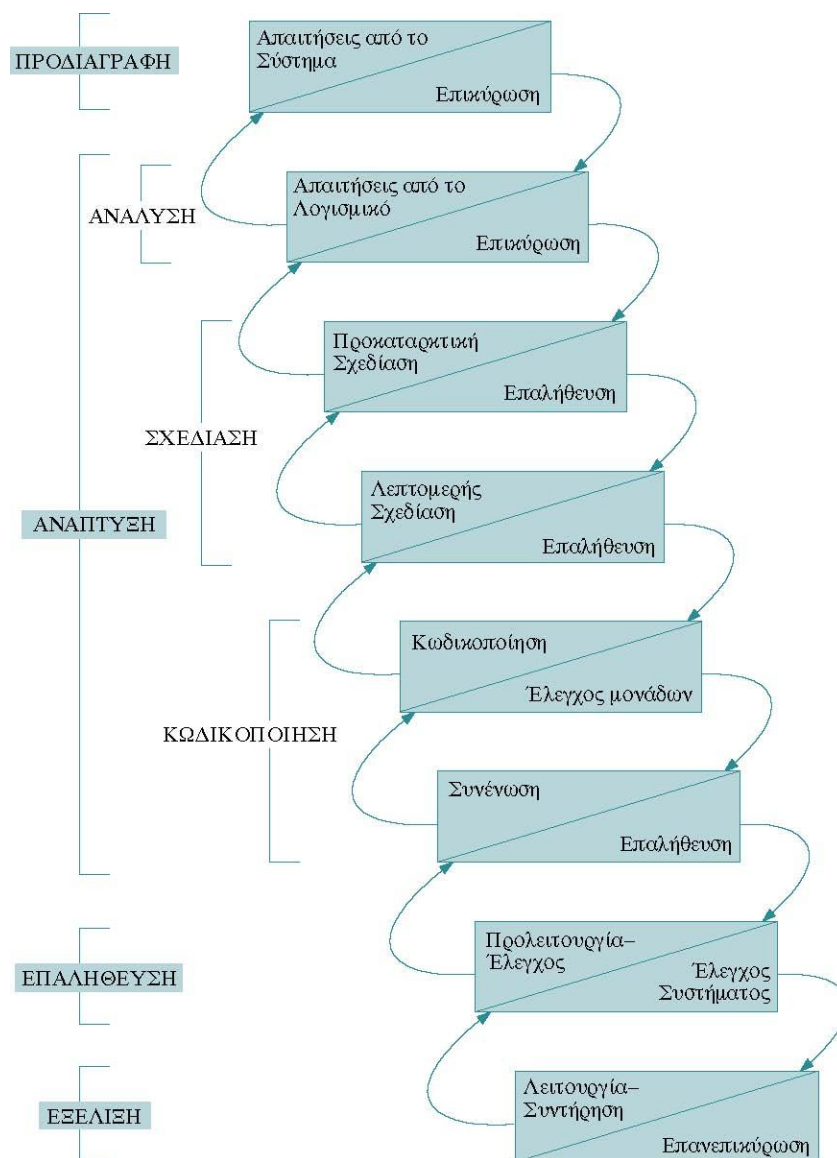


Σχήμα 4.2 Γενικές φάσεις του κύκλου ζωής του λογισμικού

Υπάρχουν πολλά μοντέλα, εμείς θα αναφερθούμε σε μερικά από αυτά και θα καταλήξουμε σε αυτό που μας εξυπηρετεί καλύτερα.

Το Μοντέλο του Καταρράκτη

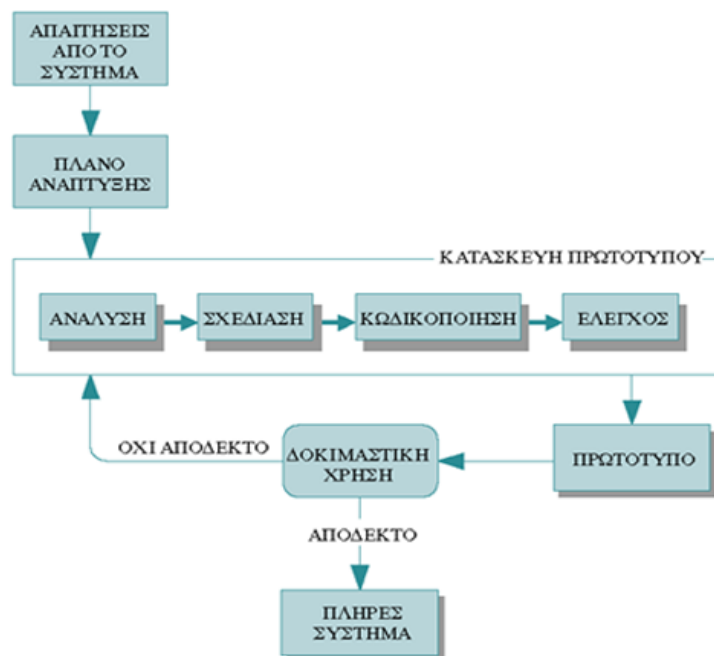
Ένα από τα πιο διαδεδομένα μοντέλα κύκλου ζωής είναι αυτό του καταρράκτη (Waterfall), το οποίο φαίνεται στο Σχήμα 4.3. Η κεντρική ιδέα του μοντέλου του καταρράκτη είναι ότι το σύστημα λογισμικού αναπτύσσεται περνώντας ολόκληρο από διαδοχικές επιμέρους φάσεις, καθεμία από τις οποίες θεωρείται περατωμένη με την παραγωγή ορισμένων συστατικών λογισμικού. Κάθε επιμέρους φάση ολοκληρώνεται με μια εργασία επαλήθευσης / επικύρωσης των προϊόντων της, κατά την οποία αποφασίζεται η μετάβαση ή όχι στην επόμενη. Το λογισμικό εμφανίζεται πλήρες, δηλαδή με όλα τα λειτουργικά του χαρακτηριστικά, από την επιμέρους φάση της συνένωσης και μετά. Χαρακτηριστικό του μοντέλου του καταρράκτη είναι ότι, για να ξεκινήσει μια φάση πρέπει να έχει ολοκληρωθεί πλήρως η προηγούμενη. Η ανάπτυξη με τον τρόπο αυτό χαρακτηρίζεται ακολουθιακή, διότι οι επιμέρους φάσεις από τις οποίες διέρχεται είναι διακριτές και ακολουθούν η μία την άλλη.



Σχήμα 4.3 Το Μοντέλο του Καταρράκτη

Το Μοντέλο Πρωτοτυποποίησης

Η κεντρική ιδέα του μοντέλου πρωτοτυποποίησης (prototyping model) είναι η ανάπτυξη του λογισμικού όχι εξολοκλήρου, αλλά σε τμήματα, που ονομάζονται «πρωτότυπα». Οι διαδικασίες ανάπτυξης επαναλαμβάνονται για ένα τμήμα του συστήματος κάθε φορά και, για το λόγο αυτό, το μοντέλο χαρακτηρίζεται ως επαναληπτικό. Κάθε πρωτότυπο περιλαμβάνει τις βασικές από τις λειτουργίες που προορίζεται να εκτελεί το λογισμικό και τίθεται σε δοκιμασία από τον πελάτη. Από εκεί συλλέγονται παρατηρήσεις και η διαδικασία κατασκευής νέου πρωτοτύπου επαναλαμβάνεται μέχρις ότου ένα πρωτότυπο να ικανοποιεί τις απαιτήσεις, δηλαδή να εκτελεί τις επιθυμητές λειτουργίες του λογισμικού με τρόπο ικανοποιητικό και να γίνεται αποδεκτό από τον πελάτη (Σχήμα 4.4). Από το σημείο αυτό και μετά μπορούν να προστεθούν και οι υπόλοιπες λειτουργίες, ώστε το λογισμικό να ολοκληρωθεί. Ένα σημαντικό πλεονέκτημα του μοντέλου αυτού είναι η δυνατότητα απόκτησης άποψης για την εφαρμογή λογισμικού νωρίτερα απ. ότι στο μοντέλο του καταρράκτη. Αυτό μπορεί να γλιτώσει την ανάπτυξη από καθυστερήσεις (και συνεπαγόμενα κόστη) ή ακόμη και από ολική αποτυχία, τα οποία θα επέρχονταν, αν ο κατασκευαστής αναγκαζόταν να οπισθοδρομήσει την ανάπτυξη, ενώ αυτή είχε προχωρήσει πολύ. Παράλληλα, ιδιαίτερη σημασία αποκτά η διοίκηση του έργου, η οποία πρέπει να εξασφαλίζει την υλοποιησιμότητα του πρωτοτύπου και την εύκολη τροποποίησή του. Κάθε κατασκευή πρωτοτύπου μπορεί να θεωρηθεί ως ένα μικρό έργο λογισμικού το οποίο κατασκευάζεται με διαδικασίες που μπορούν να ακολουθούν άλλα μοντέλα κύκλου ζωής, όπως αυτό του καταρράκτη. Με βάση τις παραπάνω παρατηρήσεις, το μοντέλο πρωτοτυποποίησης χρησιμοποιείται στην ανάπτυξη εφαρμογών λογισμικού για τις απαιτήσεις από τις οποίες δεν υπάρχει βεβαιότητα στην αρχή της ανάπτυξης, οπότε δεν μπορούν να συμφωνηθούν και να παγιωθούν. Τέτοιες είναι εφαρμογές που κατασκευάζονται για πρώτη φορά ή που είναι στενά εξαρτημένες από τον πελάτη, χωρίς να υπάρχει αποδεκτό προηγούμενο παράδειγμα. Ωστόσο, το μέγεθος των εφαρμογών αυτών δεν μπορεί να είναι ιδιαίτερα μεγάλο, διότι ο χρόνος ανάπτυξης κάθε πρωτοτύπου μεγαλώνει και η απαιτούμενη ευελιξία μειώνεται.



Σχήμα 4.4 Το Μοντέλο Πρωτοτυποποίησης

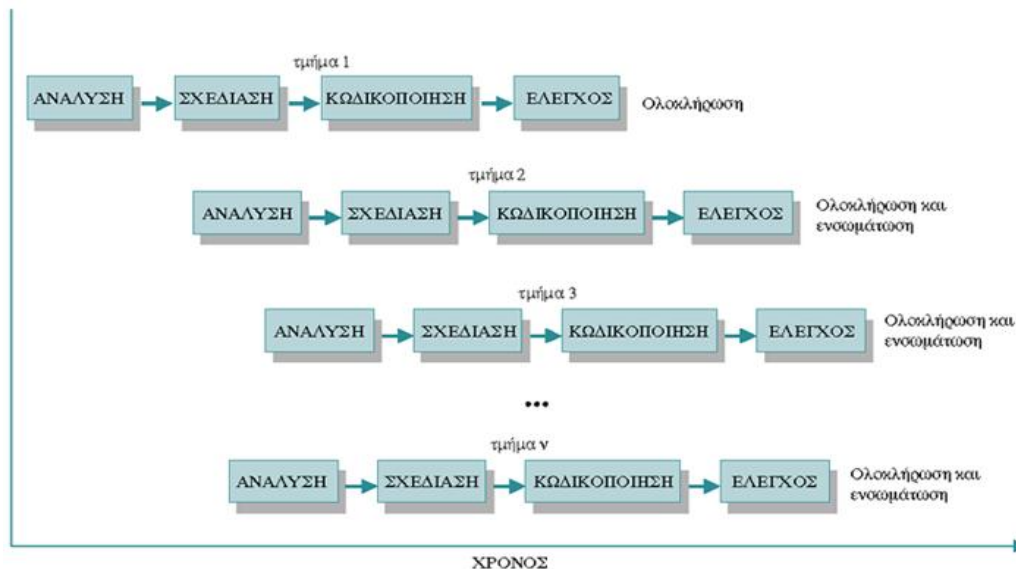
Το Μοντέλο της Λειτουργικής Επαύξεσης

Το μοντέλο της λειτουργικής επαύξεσης (incremental model) συνδυάζει την ακολουθιακή ανάπτυξη του μοντέλου του καταρράκτη με την τμηματική ανάπτυξη του μοντέλου της πρωτοτυποποίησης. Κεντρική ιδέα είναι η κατάτμηση του υπό κατασκευή λογισμικού σε τμήματα που αναπτύσσονται ανεξάρτητα, ακολουθώντας το καθένα ακολουθιακή ανάπτυξη σύμφωνα με το μοντέλο του καταρράκτη, όπως φαίνεται στο Σχήμα 4.6. Κατά την αρχική φάση ανάλυσης και σχεδίασης αποφασίζονται τα τμήματα στα οποία θα κατατμηθεί η εφαρμογή, η ανάπτυξη των οποίων γίνεται στη συνέχεια ανεξάρτητα και παράλληλα. Όταν ολοκληρώνεται η ανάπτυξη κάθε τμήματος, αυτό ενσωματώνεται στο σύνολο της εφαρμογής, διαδικασία η οποία δικαιολογεί και τον τίτλο «λειτουργική επαύξηση».

Πλεονεκτήματα της ιδέας είναι η δυνατότητα παράλληλης ανάπτυξης, η οποία τελικά καταλαμβάνει μικρότερο χρόνο, καθώς και ο διαδοχικός εμπλουτισμός των λειτουργικών χαρακτηριστικών του λογισμικού. Τα βασικά μειονεκτήματα του μοντέλου είναι τα ακόλουθα:

- Η αρχική κατάτμηση και γενική σχεδίαση του συστήματος αποκτά ιδιαίτερη βαρύτητα. Σφάλματα σε αυτή μπορούν να έχουν σημαντικές επιπτώσεις στο λογισμικό που θα κατασκευαστεί στη συνέχεια.
- Σε περίπτωση μεταβολής των λειτουργικών απαιτήσεων κατά τη χρήση του ημιτελούς συστήματος, μπορεί η αρχιτεκτονική αυτού να μεταβληθεί σε βαθμό που να κλονιστεί η ανάπτυξη των υπόλοιπων τμημάτων αυτού.

Το μοντέλο της λειτουργικής επαύξεσης χρησιμοποιείται στην ανάπτυξη μεγάλων εφαρμογών λογισμικού για τις οποίες ισχύουν οι απαιτήσεις του μοντέλου του καταρράκτη, δηλαδή σαφής γνώση και μικρή ή καθόλου μεταβλητότητα των απαιτήσεων κατά την ανάπτυξη.



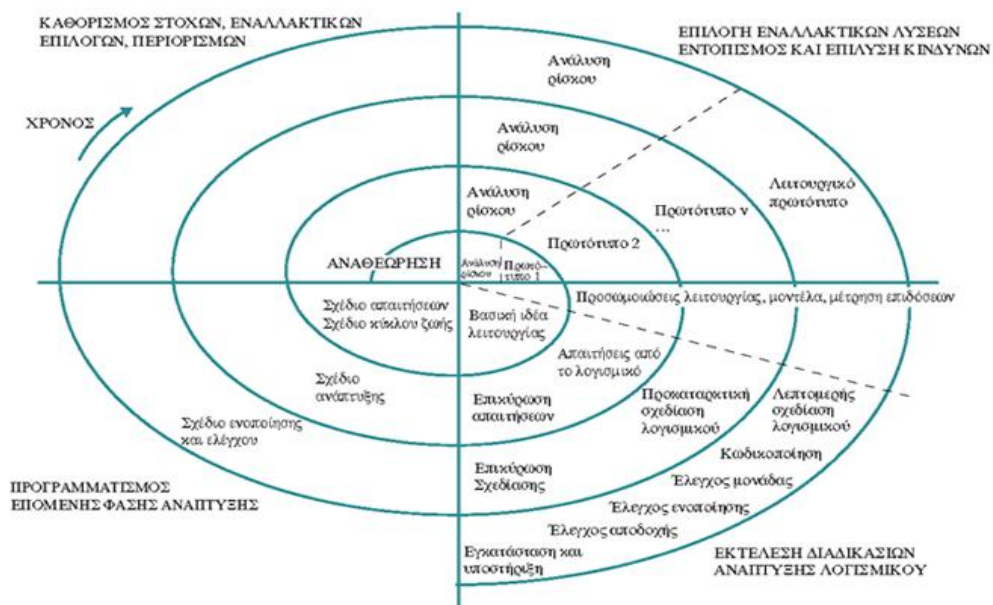
Σχήμα 4.5 Το Μοντέλο της Λειτουργικής Επαύξεσης

Το Σπειροειδές Μοντέλο

Τα μοντέλα κύκλου ζωής που παρουσιάστηκαν μέχρι τώρα αποτελούν παραλλαγές της βασικής ιδέας του μοντέλου του καταρράκτη. Η ανάπτυξη παραμένει, επί της ουσίας, μια ακολουθιακή διαδικασία, η οποία εφαρμόζεται είτε σε ολόκληρο είτε σε ένα μέρος του συστήματος. Απ' ότι φαίνεται, δεν είναι η σύλληψη των διαδικασιών ανάπτυξης λογισμικού που διαφοροποιεί τα μοντέλα κύκλου ζωής, αλλά η διάταξή τους. Στο μοντέλο της προτυποποίησης, καθώς και σε αυτό της λειτουργικής επαύξεσης, η κατάτμηση είναι λίγο ως πολύ αυθαίρετη. Το ρίσκο δεν αποτιμάται, με αποτέλεσμα κάθε οπισθοδρόμηση ή ανατροπή να κοστίζει σε χρόνο και σε οικονομικούς όρους, συχνά δε σε συνολική αποτυχία των έργων.

Από την άλλη, η μετά πειθαρχίας αποδοχή των αυστηρών φάσεων που προτείνονται από το μοντέλο του καταρράκτη δεν είναι εφικτό να ακολουθείται σε όλες τις περιπτώσεις και από όλους τους κατασκευαστές, με αποτέλεσμα η ανάπτυξη λογισμικού είτε να γίνεται άναρχα, με βάση τη διαίσθηση των κατασκευαστών, είτε να είναι μια δαπανηρή και στρυφνή διαδικασία, στην οποία «πρέπει» να ακολουθηθούν κάποια συγκεκριμένα βήματα, ανεξάρτητα από τις εκάστοτε συνθήκες. Απάντηση στα παραπάνω έρχεται να δώσει το σπειροειδές μοντέλο κύκλου ζωής, το οποίο πήρε το όνομά του από την απεικόνιση σε διάγραμμα, όπως φαίνεται στο Σχήμα 4.6. Πρόκειται για μια γενίκευση των μοντέλων της λειτουργικής επαύξεσης και της προτυποποίησης, με σημαντικά νέα στοιχεία:

- Οι φάσεις και οι διαδικασίες ανάπτυξης λογισμικού δεν είναι προκαθορισμένες από το μοντέλο, αλλά εξειδικεύονται στο χώρο της εφαρμογής του.
- Η ανάπτυξη ολόκληρου του συστήματος χωρίζεται σε πολλούς κύκλους, σε καθέναν από τους οποίους προστίθενται νέα λειτουργικά χαρακτηριστικά στο σύστημα.
- Πριν από την έναρξη κάθε κύκλου γίνεται μια μελέτη σκοπιμότητας και ανάλυση κινδύνων, από την οποία προκύπτουν, αφενός, οι συγκεκριμένες εργασίες που θα εκτελεστούν μέσα στον κύκλο, αφετέρου, η ίδια η εφικτότητα εκτέλεσης του κύκλου αυτού.



Σχήμα 4.6 Το Σπειροειδές Μοντέλο

Όπως φαίνεται στο Σχήμα 4.6, στο σπειροειδές μοντέλο διακρίνονται τέσσερις κατηγορίες εργασιών: προσδιορισμός στόχων, εντοπισμός και επίλυση κινδύνων, εκτέλεση διαδικασιών ανάπτυξης και επαλήθευση, καθώς και εργασίες προγραμματισμού.

- Κατά τον προσδιορισμό στόχων καθορίζονται τα αντικείμενα εργασιών κάθε επανάληψης, καταγράφονται οι περιορισμοί επί του προϊόντος, αλλά και επί της διαδικασίας για την οποία κατασκευάζεται ένα αναλυτικό πλάνο διοίκησης. Επίσης, καταγράφονται οι κίνδυνοι που εμπεριέχει η διαδικασία και οι εναλλακτικές λύσεις, όπου υπάρχουν.
- Κατά τις εργασίες επίλυσης κινδύνων αναλύονται οι κίνδυνοι που έχουν καταγραφεί και αποτιμάται κάθε εναλλακτική λύση. Στο σημείο αυτό λαμβάνονται αποφάσεις για τη συνέχιση ή όχι της ανάπτυξης, για το μοντέλο που θα ακολουθηθεί στη συγκεκριμένη επανάληψη, για την κατασκευή ή όχι πρωτοτύπου κ.ά.
- Ακολουθεί η εκτέλεση των βημάτων της διαδικασίας ανάπτυξης λογισμικού που έχει επιλεγεί για το τμήμα εκείνο του συστήματος που αφορά η τρέχουσα επανάληψη.
- Τέλος, μετά την επαλήθευση των αποτελεσμάτων . ενδιάμεσων προϊόντων λογισμικού γίνεται προγραμματισμός της συνέχισης της ανάπτυξης.

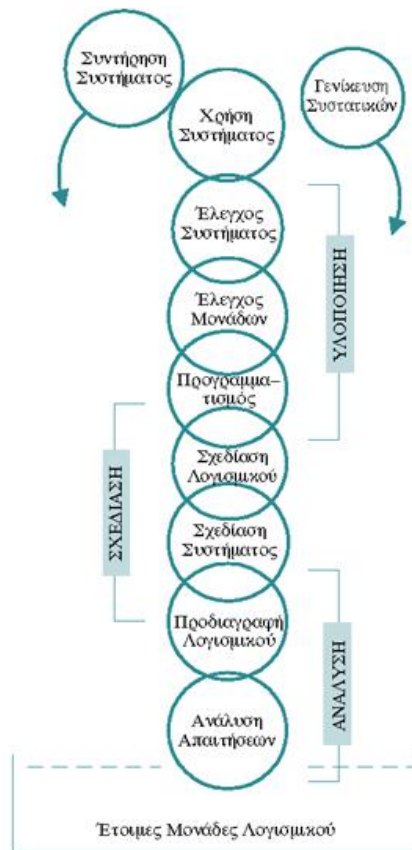
Το σπειροειδές μοντέλο δεν καθορίζει εκ των προτέρων ποιες ακριβώς είναι οι εργασίες ανάπτυξης λογισμικού που πρέπει να γίνουν ούτε σε ποια έκταση του συστήματος αυτές θα εφαρμοστούν. Διαφορετικές διαδικασίες ανάπτυξης μπορεί να επιλεγούν για διαφορετικά τμήματα του λογισμικού. Αυτό που προτείνει είναι ότι ο καθορισμός των λεπτομερειών υλοποίησης πρέπει να γίνεται συνεχώς κατά την ανάπτυξη (και όχι μία φορά, όπως συμβαίνει με τα μοντέλα κύκλου ζωής που αναφέρθηκαν μέχρι τώρα) με ευθύνη και με τεκμηρίωση από πλευράς του ίδιου του κατασκευαστή.

Η εφαρμογή του σπειροειδούς μοντέλου στην πράξη δεν είναι πάντα εύκολη υπόθεση. Εισάγονται νέες εργασίες, που δεν ανήκουν καθαρά στις εργασίες ανάπτυξης λογισμικού, αλλά αφορούν την τεκμηρίωση της σκοπιμότητας και τον τμηματικό προγραμματισμό της ανάπτυξης. Οι εργασίες αυτές επιφέρουν ασφαλώς κάποιο κόστος, το οποίο όμως μπορεί να αποσβεστεί από τον έγκαιρο εντοπισμό προβλημάτων και την αποφυγή πιθανού ναυαγίου, κάτι που έχει συμβεί σε αρκετές περιπτώσεις.

Το Μοντέλο του Πίδακα

Αρκετά μοντέλα κύκλου ζωής που έχουν προταθεί αποτελούν παραλλαγές αυτών που αναφέρθηκαν, τα χαρακτηριστικά των οποίων υποβάλλονται από τις μεθοδολογίες ανάπτυξης. Οι πρώτες προσεγγίσεις του θέματος με βάση την αντικειμενοστραφή (object-oriented) τεχνολογία διαφοροποίησαν το παραπάνω σχήμα βασιζόμενες σε δύο ιδιαίτερα γνωρίσματά της: πρώτον, ότι οι έννοιες «ανάλυση .-σχεδίαση- κωδικοποίηση» έρχονται στο αντικειμενοστραφές παράδειγμα πολύ πιο κοντά και, δεύτερον, ότι το αποτέλεσμα κάθε διαδικασίας κατασκευής λογισμικού είναι όχι μόνο ένα σύστημα, αλλά και επαναχρησιμοποιήσιμες μονάδες, οι οποίες μπορούν να χρησιμοποιηθούν από τις πρώτες φάσεις της ανάπτυξης μελλοντικών συστημάτων. Με τον τρόπο αυτό προέκυψε το μοντέλο του πίδακα (fountain model), που φαίνεται στο Σχήμα 4.8. Κατά την ανάπτυξη παρατηρούνται επικάλυψεις των φάσεων «ανάλυση .-σχεδίαση - κωδικοποίηση», οι οποίες φαίνονται με την επικάλυψη των κύκλων στο σχήμα. Κατά το τέλος της ανάπτυξης, ορισμένα από τα συστατικά λογισμικού που έχουν παραχθεί ενσωματώνονται σε μια «δεξαμενή» συστατικών και διατίθενται για να χρησιμοποιηθούν στην ανάπτυξη και νέων συστημάτων. Η ιδέα του μοντέλου κύκλου ζωής του πίδακα τονίζει περισσότερο τα επιθυμητά χαρακτηριστικά της

μεθοδολογίας κατασκευής του λογισμικού σύμφωνα με την αντικειμενοστραφή λογική, ήταν δε αρκετά επίκαιρη κατά την έκρηξη ενδιαφέροντος για την αντικειμενοστραφή τεχνολογία στα τέλη της δεκαετίας του .80 και στις αρχές της δεκαετίας του .90.

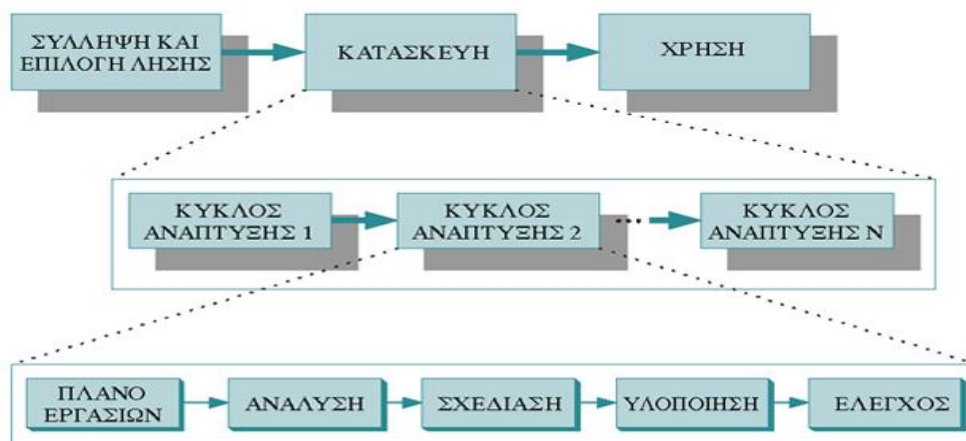


Σχήμα 4.7 Το Μοντέλο του Πίδακα

Σύγχρονα Μοντέλα Κύκλου Ζωής

Μεταγενέστερα μοντέλα κύκλου ζωής λογισμικού προσπαθούν να δώσουν μια γενική κατεύθυνση εφαρμογής των υπάρχουσών ιδεών, αφήνοντας σημαντικούς βαθμούς ελευθερίας στον κατασκευαστή που τα ακολουθεί. Αυτό είναι ιδιαίτερα επιθυμητό, διότι η αυστηρή πειθαρχία που επιχειρήθηκε να εισαχθεί τα πρώτα χρόνια της έκρηξης της χρήσης του λογισμικού δε συμβάδιζε με την ωριμότητα σκέψης που διέθετε η τεχνική κοινότητα την εποχή εκείνη ούτε και μπορούσε να παρακολουθήσει τους υψηλούς ρυθμούς εξέλιξεων στο χώρο της πληροφορικής. Χαρακτηριστικό είναι ότι συχνά ένα πολύ μεγάλο μέρος ογκωδέστατων παραδοτέων (σχεδίων, προδιαγραφών κτλ.) δεν ήταν παρά λευκές σελίδες με τη μόνη ένδειξη «this page has been intentionally left blank», οι οποίες όμως ήταν υποχρεωτικό να υπάρχουν, σύμφωνα με το ακολουθούμενο μοντέλο ανάπτυξης. Η πειθαρχία αυτή τελικά δεν οδήγησε στην κατασκευή λογισμικού αναμενόμενης ποιότητας. Μια περιγραφή ενός σύγχρονου μοντέλου κύκλου ζωής λογισμικού περιέχει μόνο γενικές κατευθύνσεις, οι οποίες εξειδικεύονται εκάστοτε περιβάλλον ανάπτυξης, πρόβλημα κτλ.

Επίσης, δεν είναι άρρηκτα συνδεδεμένο με κάποια μεθοδολογία ανάπτυξης λογισμικού, αλλά μπορεί να εξειδικευτεί για την πρακτική τού κάθε κατασκευαστή. Ένα τέτοιο μοντέλο φαίνεται στο Σχήμα 4.8 και μπορεί να χαρακτηριστεί ως απόγονος πολλών από τα μοντέλα που προαναφέρθηκαν.



Σχήμα 4.8 Σύγχρονα Μοντέλα Κύκλου Ζωής

Το γενικό πλαίσιο του μοντέλου αυτού περιλαμβάνει τις φάσεις σύλληψης, κατασκευής και λειτουργίας. Καθεμιά από αυτές αναλύεται σε επιμέρους εργασίες, σύμφωνα με τα χαρακτηριστικά του εκάστοτε περιβάλλοντος. Ιδιαίτερα η γενική φάση της κατασκευής αναλύεται σε «κύκλους ανάπτυξης», καθένας εκ των οποίων προσθέτει νέα χαρακτηριστικά και λειτουργίες στο υπό κατασκευή λογισμικό. Τα επιμέρους βήματα μέσα σε κάθε κύκλο ανάπτυξης μοιάζουν με τα βήματα του μοντέλου του καταρράκτη, μόνο που δεν εφαρμόζονται για ολόκληρο το σύστημα, αλλά για το μικρό μέρος του που κατασκευάζεται στον εν λόγω κύκλο, όπως στο μοντέλο της πρωτοτυποποίησης. Για την εκκίνηση κάθε κύκλου ανάπτυξης μπορεί να έχει προηγηθεί ανάλυση ρίσκου και σκοπιμότητας, όπως στο σπειροειδές μοντέλο. Ζητήματα όπως αλληλουχία των ενεργειών, ακριβής καθορισμός των κύκλων ανάπτυξης κ.ά. αφήνονται στη διακριτική ευχέρεια του κάθε κατασκευαστή, από τον οποίο και καθορίζονται σύμφωνα με τις ιδιαιτερότητες κάθε περίπτωσης. Ένα πραγματικό τέτοιο πλαίσιο ανάπτυξης προτείνεται από τη μεθοδολογία Rational Unified Process, η οποία είναι το προϊόν σύγκλισης των επικρατέστερων αντικειμενοστραφών μεθοδολογιών ανάπτυξης λογισμικού.

4.3 Επιλογή μοντέλου κύκλου ζωής

Μετά από την παρουσίαση όλων των γνωστών μοντέλων και των τρόπων λειτουργίας τους αποφασίσαμε ότι το πιο ευνοϊκό για την υλοποίηση της εφαρμογής μας είναι το μοντέλο του καταρράκτη. Πιστεύουμε ότι ο τρόπος που περνά από στάδιο σε στάδιο θα μας βοηθήσει ιδιαίτερα.

Επιλέξαμε το μοντέλο του καταρράκτη γιατί:

- είναι ιδανικό για μικρού ή μεσαίου μεγέθους εφαρμογές,
- οι μεταβολές των απαιτήσεων είναι ανεπιθύμητες (οι απαιτήσεις μας είναι συγκεκριμένες και πιθανές αλλαγές θα είναι ελάχιστες)

- το σύστημα λογιστικού αναπτύσσεται περνώντας ολόκληρο από διαδοχικές επιμέρους φάσεις , όπου η κάθε μία θεωρείται υπεύθυνη για την παράγωγη ορισμένων συστατικών του λογισμικού , επίσης κάθε φάση επαληθεύετε πριν προχωρήσουμε στην επόμενη (έτσι προλαμβάνουμε τυχών λάθη)
- το λογισμικό εμφανίζετε πλήρες(με όλα τα λειτουργικά του χαρακτηριστικά) από τη φάση της σύναψης και μετά, με όλα τα τεχνικά του χαρακτηριστικά

Κεφάλαιο 5

5. Προδιαγραφές Συστήματος

5.1 Εισαγωγή

Στο κεφάλαιο αυτό θα ασχοληθούμε με τις προδιαγραφές του συστήματος. Θα καθορίσουμε τις απαιτήσεις που ζητάμε από την εφαρμογή και τον τρόπο λειτουργίας που θα επιδιώξουμε να έχει. Θα δήξουμε με πια σειρά θα εκτελούνται και που θα εξυπηρετούν οι διεργασίες της εφαρμογής. Θα αναλύσουμε εκτενώς την εφαρμογή για να έχουμε μια καλύτερη εικόνα.

5.2 Προδιαγραφές Ανάπτυξης

5.2.1 Απαίτησης συστήματος

Θέλουμε ένα σύστημα που να συλλέγει πληροφορίες από τους αισθητήρες από κάθε σημείο που τους έχουμε τοποθέτηση. Στη συνέχεια να εμφανίζει στο χάρτη τα σημεία που έχουμε τους αισθητήρες . Επιλέγοντας το σημείο που θέλουμε να ελιξουμε να μας εμφανίζει τις πληροφορίες που έχει συλλέξει. Ακόμη θέλουμε ανεξάρτητα με το αν ο χρήστης είναι συνδεδεμένος με το σύστημα, να γίνεται συνεχής έλεγχος στις πληροφορίες που παρέχουν οι αισθητήρες και αν κρίνει το σύστημα με τις προϋπόθεσης που θα του δίνουμε ότι υπάρχει κίνδυνος να ενημερώνει άμεσα κάποιον αρμόδιο (π.χ. πυροσβεστική).

5.2.2 Προδιαγραφές των απαιτήσεων από το λογισμικό.

1. Εισαγωγή

1.1 Ταυτότητα του εγγράφου

ASDDH

1.2 Σκοπός

Ο σκοπός της εφαρμογής αυτής είναι η επίβλεψη και ανίχνευση ζημιών σε μια οικία με τη χρήση εργαλείων της Google

1.3 Εμβέλεια

Θα υπάρχει δυνατότητα πρόσβασης από παντού, μόνο με τη χρήση internet

1.4. Ορισμοί, ακρωνύμια, συντομογραφίες

ASDDH = application for surveillance and detection of damage to homes

1.5 Πηγές αναφορών

1.6 Περίληψη

2. Γενική περιγραφή του λογισμικού

2.1 Στίγμα

2.2 Προοπτική

Να γίνει μία εφαρμογή που να προσφέρει σμηνουργία σε αυτούς που την χρησιμοποιούν ότι τα σημεία που ελέγχει σε περίπτωση επικίνδυνων συνθηκών θα υπάρξει ενημέρωση για να αποφευχθούν οι ζημιές.

2.3 Γενικές λειτουργίες του λογισμικού

Το λογισμικό θα έχει την δυνατότητα να ελέγχει την κατάσταση σε μια οικία, και να ενημερώνει για τοίχων κινδύνους.

2.4 Χαρακτηριστικά χρηστών

Θα υπάρχουν 2 είδη χρηστών, ο απλός χρήστης που θα έχει την δυνατότητα να ενημερώνετε μόνο για την οικία του. Και ο προνομιούχος χρήστης θα έχει την δυνατότητα να επιβλέπει όλες τις οικίες που έχει τη δυνατότητα η εφαρμογή να δώσει στοιχεία επίσης θα έχει την δυνατότητα προσθήκης καινούργιας οικίας (σημείου).

2.5 Περιορισμοί

Δεν μπορεί ο απλός χρήστης να πάρει πληροφορίες πέραν από αυτές που τον αφορούν.

Οι πληροφορίες που θα συλλέγονται θα είναι συγκριμένες και με έγκριση του ιδιοκτήτη.

Ο προνομιούχος χρήστης δεν θα μπορεί να διαγράφη κανένα σημείο μέσω του λογισμικού

2.6 Παραδοχές και εξαρτήσεις

Για να λειτουργήσει η εφαρμογή είναι απαραίτητη η σύνδεση με το internet και η χρήση οποιουδήποτε browser. Η οικία που θα συλλέγονται οι πληροφορίες θα πρέπει να υπάρχει σύνδεση στο internet.

3. Ειδικές απαιτήσεις

3.1 Απαιτήσεις εξωτερικών διαπροσωπειών(interface)

Απαραίτητη η χρήση internet.

3.1.1 Διαπροσωπείες χρήστη

Το περιβάλλον που ο χρήστης θα χρησιμοποιεί την εφαρμογή είναι περιβάλλον διαδικτύου.

3.1.2 Διαπροσωπείες υλικού

3.1.3 Διαπροσωπείες λογισμικού

3.1.4 Διαπροσωπείες επικοινωνιών

3.2 Λειτουργικές απαιτήσεις

Σύνδεση στο internet, και χρήση browser.

3.2.1 Τρόπος λειτουργίας 1

Θα λειτουργεί μόνο όταν κάποιος πληκτρολόγησε την διεύθυνση στο browser και στη συνέχεια τον σωστό κωδικό εκεί που θα του ζητηθεί.

3.2.1.1 Λειτουργική απαίτηση 1.1

Προβολή των πληροφοριών. Η εφαρμογή ζητά τον κωδικό πρόσβασης, αν ο κωδικός είναι σωστός τότε ανάλογα με την κατηγορία χρήστη που ανήκει ο χρήστης έχει την ανάλογη δυνατότητα στο λογισμικό. Αν είναι προνομιούχος χρήστης έχει την δυνατότητα παρακολούθησης όλων των σημείων σε όλους τους χάρτες. Αντίθετα αν είναι απλός χρήστης έχει την δυνατότητα παρακολούθησης μόνο του σημείου που τον αφορά.

Εισαγωγή: κωδικό πρόσβασης και πληροφοριών από τους αισθητήρες, Επεξεργασία: επαλήθευση του κωδικού. Έξοδος: το χάρτη με τα σημεία που έχει την εξουσιοδότηση ο χρήστης να ενημερώνετε.

3.2.1.2 Λειτουργική απαίτηση 1.2

Εισαγωγή καινούργιων σημείων στο χάρτη μόνο από το προνομιούχο χρήστη.

Είσοδος: Όνομα σημείου, Διεύθυνση, Συντεταγμένες, Τύπος οικίας. Επεξεργασία: έλεγχος αν έχουν συμπληρωθεί όλα τα στοιχεία. Έξοδος: καινούργια σημείο καταχωρημένο στη Βάση Δεδομένων και στο χάρτη

3.2.1.3 Λειτουργικές απαιτήσεις 1.3

Αυτή η λειτουργία τρέχει ανεξάρτητα με το αν ο χρήστης τρέχει την εφαρμογή. Ανάλογα με της συνθήκες ασφαλείας που ορίζουμε στο λογισμικό αν με κάποια από τις πληροφορίες που λαμβάνουμε από τους αισθητήρες κρίνεται ότι υπάρχει κίνδυνος στην οικία ενημερώνει κάποιον αρμόδιο που έχουμε προεπιλέξει, αν επιδεινωθεί ο κίνδυνος ή να ξεπεραστεί τότε ξανά ενημερώνει. Επίσης ανεξάρτητα αν υπάρχει κίνδυνος η όχι κάθε 6 ώρες θα ενημερώνετε ο αρμόδιος ότι όλα είναι καλά. Είσοδος: πληροφορίες από τους αισθητήρες, συνθήκες κατά τις οποίες πρέπει να θεωρείτε ότι υπάρχει κίνδυνος και το e-mail που θα σταλούν τα μηνύματα. Επεξεργασία: σύγκριση των πληροφοριών με της συνθήκες που ορίσαμε. Έξοδος: ένα ενημερωτικό e-mail για την κατάσταση της οικίας

3.3 Απαιτήσεις επιδόσεων

Απαιτούμε γρήγορη και σωστή ενημέρωση της εφαρμογής από τους αισθητήρες και άμεση προβολή των πληροφοριών. Επίσης συνεχή έλεγχο των πληροφοριών και άμεση ενημέρωση με e-mail τους υπεύθυνους.

3.4 Περιορισμοί σχεδίασης

Η εφαρμογή πρέπει να είναι συνέχεια συνδεδεμένη με το internet. Πρέπει να τρέχει γρήγορα για να μην υπάρχει χρόνοκαυστήρηση.

3.4.1 Περιορισμοί από το υλικό

Περιορισμοί στο υλικό της εφαρμογής δεν υπάρχουν. Η δυσκολία που εντοπίσαμε κατά τη διάρκεια της σχολιασμός είναι στην εύρεση αξιόπιστων και φτηνών αισθητήρων.

3.4.2 Συμμόρφωση με πρότυπα

Πρέπει να είναι σύμφωνα με τα πρότυπα κατασκευής ιστοσελίδων (web 3) για να τρέχει σε όλους τους browser.

3.5 Χαρακτηριστικά του λογισμικού

Γρήγορο και εύκολο στη χρήση.

3.5.1 Αξιοπιστία

Πρέπει να είναι παρά πόλη αξιόπιστο.

3.5.2 Διαθεσιμότητα

Διαθεσιμότητα σε όσους έχουν κάνει αίτηση εγκατάστασης τον αισθητήρων στο χώρο τους.

3.5.3 Ασφάλεια

Να μην υπάρχει δυνατότητα προβολής πληροφοριών σε χρήστη που δεν είναι εξουσιοδοτημένος.

3.5.4 Χαρακτηριστικά συντήρησης

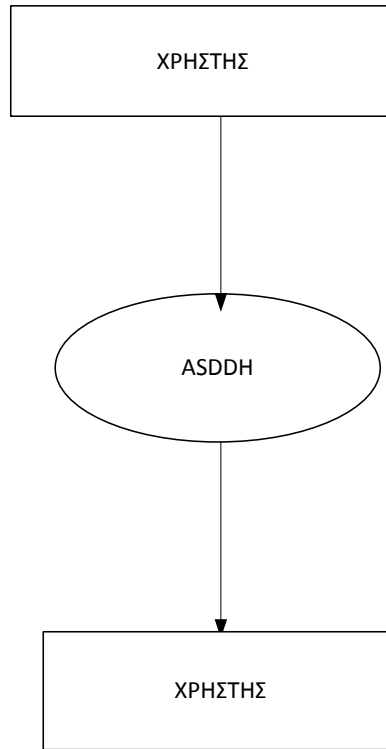
Συνεχής αναβάθμιση του κώδικα για να είναι σύμφωνα με τα πρότυπα του internet . Επίσης η συνεχής αναβάθμιση τις ασφάλειας της εφαρμογής ώστε να αποφεύγονται (όσο είναι δυνατόν) η παρενόχλησης του λογισμικού από εισβολής .

3.5.5 Μεταφερισιμότητα

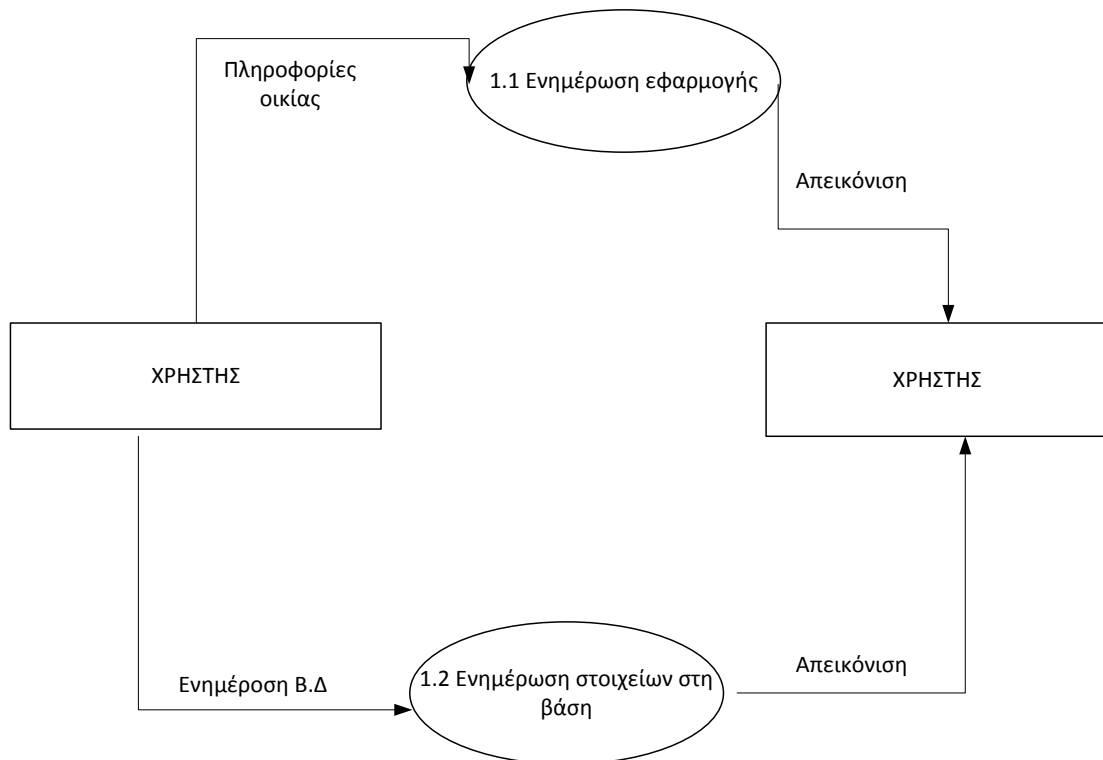
Δυνατότητα να το λειτουργήσει από οποιανδήποτε συσκευή υποστηρίζει σύνδεση στο internet.

5.3 Σχεδιαγράμματα

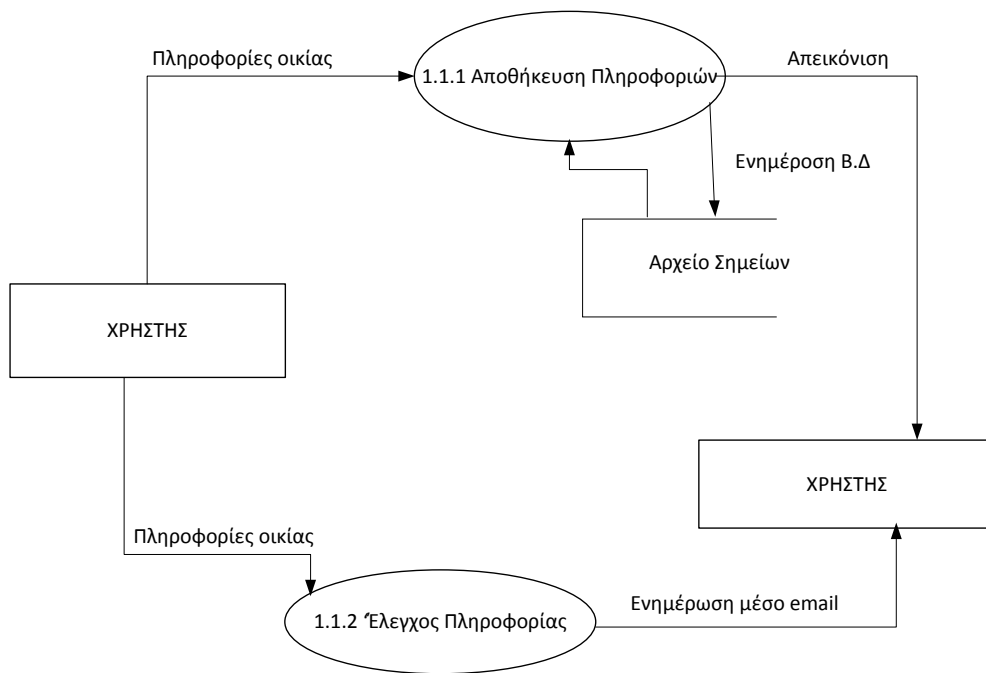
5.3.1 Διαγράμματα ροής δεδομένων



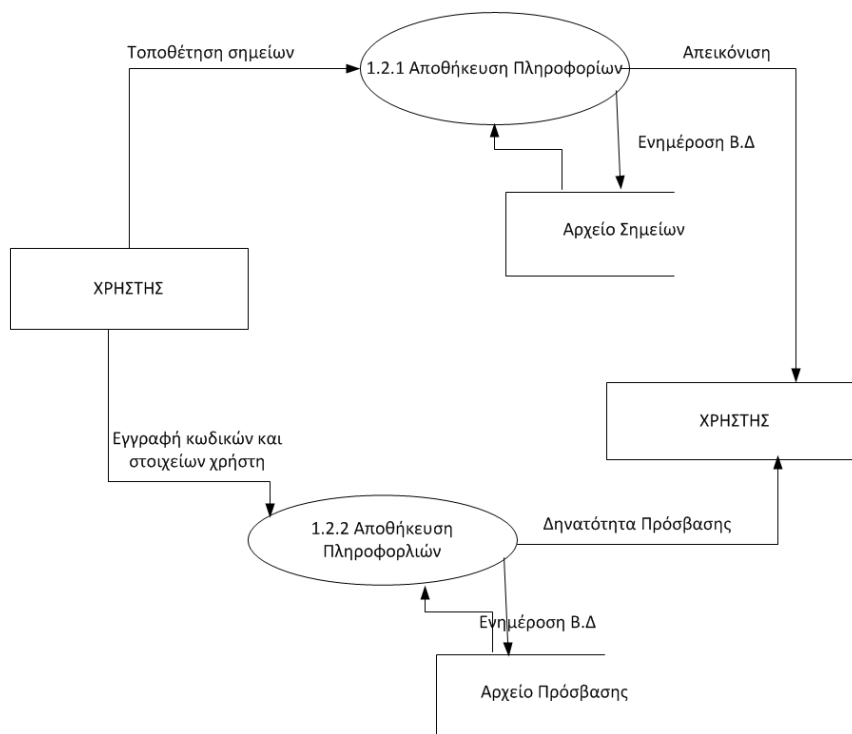
Σχήμα 5.1 Διάγραμμα ροής δεδομένων 1 επίπεδο



Σχήμα 5.2 Διάγραμμα ροής δεδομένων 2 επίπεδο

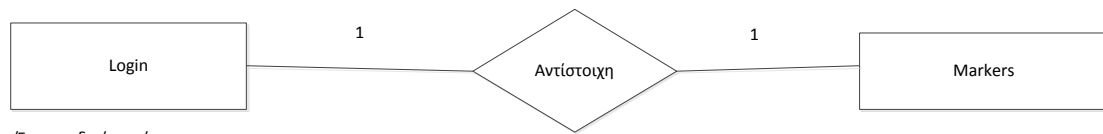


Σχήμα 5.3 Διάγραμμα ροής δεδομένων 3 επίπεδο 1.1



Σχήμα 5.4 Διάγραμμα ροής δεδομένων 3 επίπεδο 1.2

5.3.2 Διάγραμμα οντοτήτων συσχετίσεων



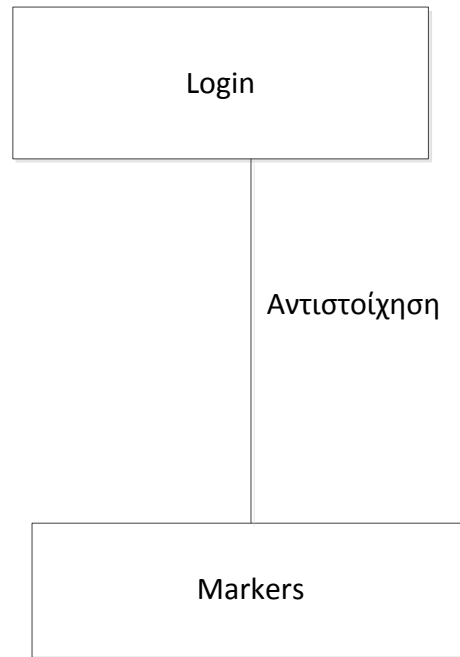
Ένας κωδικός από το πίνακα Login αντίστοιχη σε μια εγγραφή από το πίνακα Markers



Ένα ID Από τον πίνακα Markers αντίστοιχη σε μια εγγραφή στον πίνακα Login

Ο μόνος που έχει πρόσβαση σε όλα τα στοιχεία του πίνακα Markers είναι ο administrator

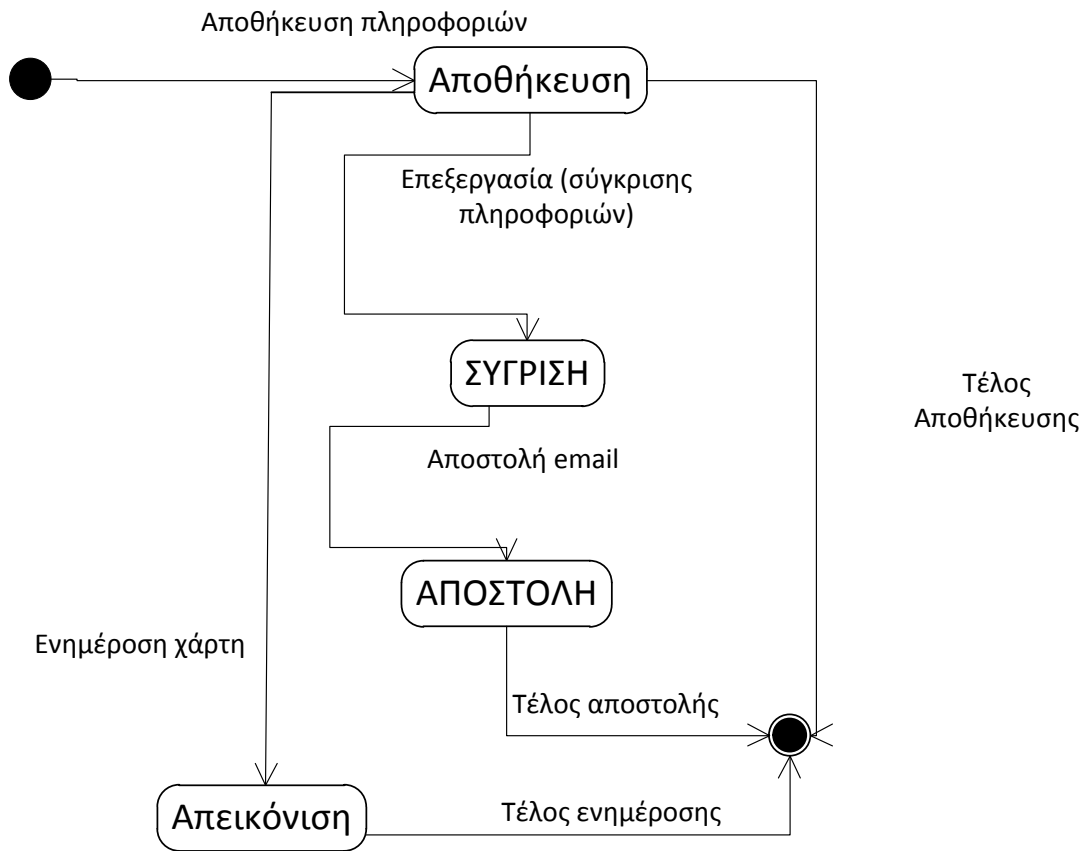
Σχήμα 5.5 Διάγραμμα οντοτήτων συσχετίσεων 1



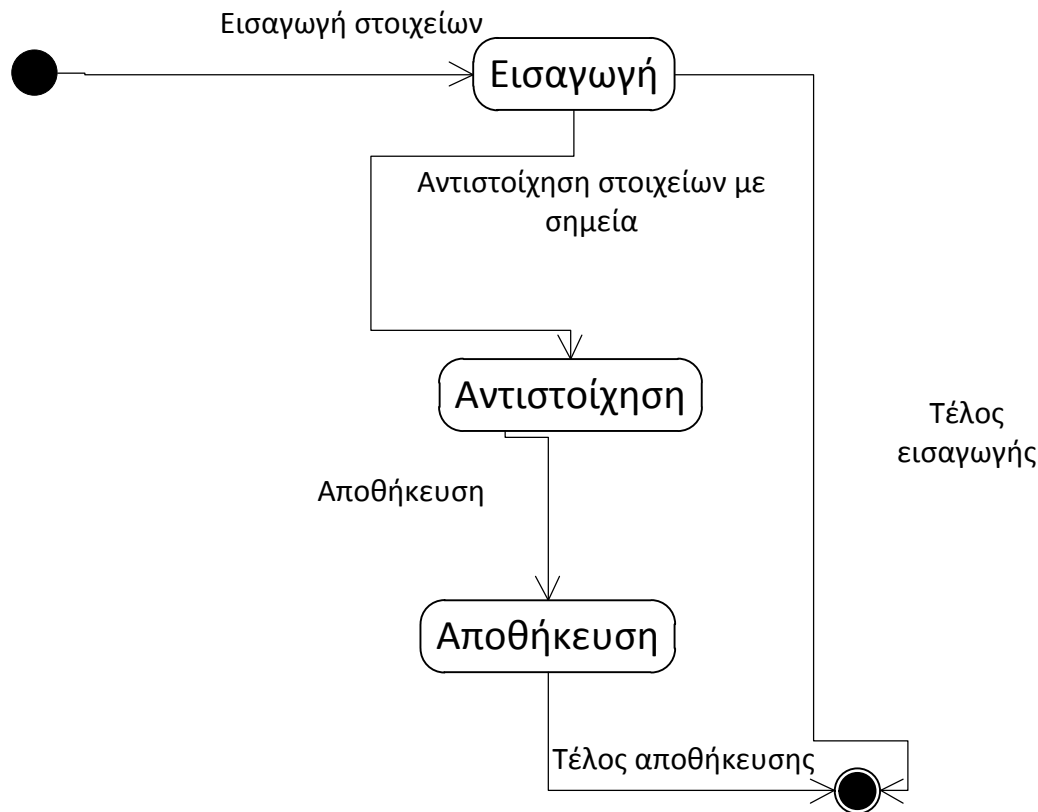
Ο μόνος που έχει πρόσβαση σε όλα τα στοιχεία του πίνακα Markers είναι ο administrator

Σχήμα 5.6 Διάγραμμα οντοτήτων συσχετίσεων 2

5.3.4 Διάγραμμα μετάβασης καταστάσεων



Σχήμα 5.7 Διάγραμμα μετάβασης καταστάσεων 1



Σχήμα 5.8 Διάγραμμα μετάβασης καταστάσεων 2

5.4 Λεξιικό δεδομένων

Πληροφορίες οικίας είναι οι πληροφορίες που λαμβάνονται από τους αισθητήρες που είναι εγκατεστημένοι στο χώρο που ελέγχουμε. Αποθηκεύονται στη Β.Δ μας (itgooglemaps) στον πίνακα markers. Εντάχτηκα αναφέρουμε ότι στη παρούσα εφαρμογή υφίσταται πεδίο για την θερμοκρασία και την υγρασία.

Συνθήκες έλεγχου ή έλεγχος πληροφορίας είναι ο έλεγχος που γίνεται για να δούμε αν οι πληροφορίες που έρχονται από το σημείο που ελέγχουμε είναι στα όρια που θέσαμε και δεν υπάρχει οποιοσδήποτε κίνδυνος.

Απεικόνιση είναι το αποτέλεσμα από μία διεργασία της εφαρμογής που μετά από την αποθήκευση των πληροφοριών εμφανίζονται στο χάρτη της περιοχής που και στο σημείο που προήλθαν.

Αρχείο Σημείων είναι ο πίνακας **markers** όπου εκεί αποθηκεύονται όλες οι πληροφορίες για το κάθε σημείο που παρακολουθείτε.

Αρχείο Πρόσβασης είναι ο πίνακας **login** όπου αποθηκεύονται τα στοιχεία του χρήστη και σε πιο σημείο τοθ πίνακα markers αντιστοιχη. Στο πίνακα αυτό καταχωρούνται και οι κωδικό του κάθε χρηστή.

Κεφάλαιο 6

6. Σχεδίαση Εφαρμογής

6.1 Εισαγωγή

Στο κεφάλαιο αυτό θα δήξουμε πως σχεδιάσαμε την εφαρμογή. Από πια στάδια πέρασε και για ποιο λόγο αποφασίσαμε να πράξουμε σε συγκεκριμένο τρόπο (π.χ. στο κώδικα).

Για να υπαρχή σωστή και δομημένη σχεδίαση μιας διαδικτυακής εφαρμογής πρέπει να τηρούνται κάποια πρότυπα. Έτσι από τα μαθήματα του ηλεκτρονικού εμπορίου που κάναμε στη σχολή και με την εμπειρία που έχουμε από την πρακτική μας άσκηση προσπαθήσαμε να υλοποιήσουμε την εφαρμογή με αυτά που μάθαμε, αλλά και περνώντας ιδέες από άλλες διαδικτυακές εφαρμογές.

Για να μπορούμε να δουλέψουμε σε επαγγελματικά επίπεδα και να έχουμε όσο το δυνατόν πιο γρήγορη και σωστή σχεδίαση και λειτουργία τις εφαρμογής πρέπει:

- Να κατανοήσουμε τις απαιτήσεις λογισμικού
- Να σχεδιάσουμε το interface της εφαρμογής και το λογότυπο της εφαρμογής
- Να μετατρέψουμε την διεπαφή σε κώδικα και επιλογή του κατάλληλου κώδικα για την βέλτιστη λειτουργία του λογισμικού.

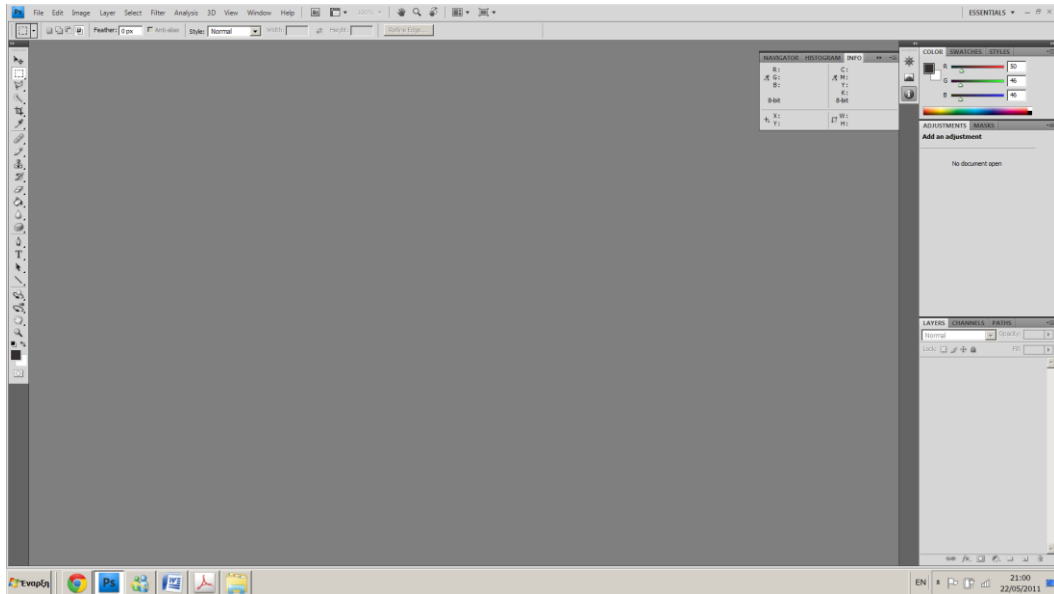
6.2 Κατανόηση των απαιτήσεων

Στο σημείο αυτό πρέπει να αναλύσουμε τις απαιτήσεις του λογισμικού που παίρνουμε από το ερωτηματολόγιο έτσι ώστε να μπορούμε να τις τοποθετήσουμε στη σχεδίαση .

- Αυτόνομο σύστημα
- Να έχει την δυνατότητα συλλογής πληροφοριών από την οικία (π.χ. θερμοκρασίας – υγρασίας).
- Να παρέχει ασφάλεια.
- Να είναι άμεση και ακριβής η ενημέρωση και σε πραγματικό χρόνο.

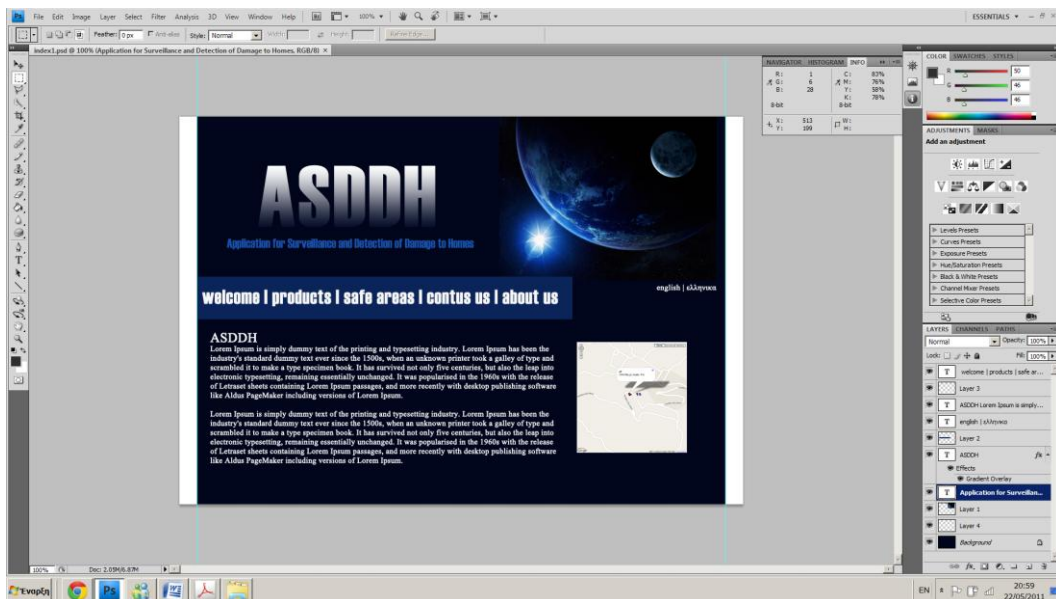
6.3 Σχεδίαση interface

Σχεδιάσαμε ένα interface για την εφαρμογή που να είναι εύχρηστο, ομοιόμορφο και να είναι λειτουργικό σε όλους τους browser. Για να γίνει πιο εύκολη η σχεδίαση θα χρησιμοποιήσαμε το πρόγραμμα Adobe Photoshop.

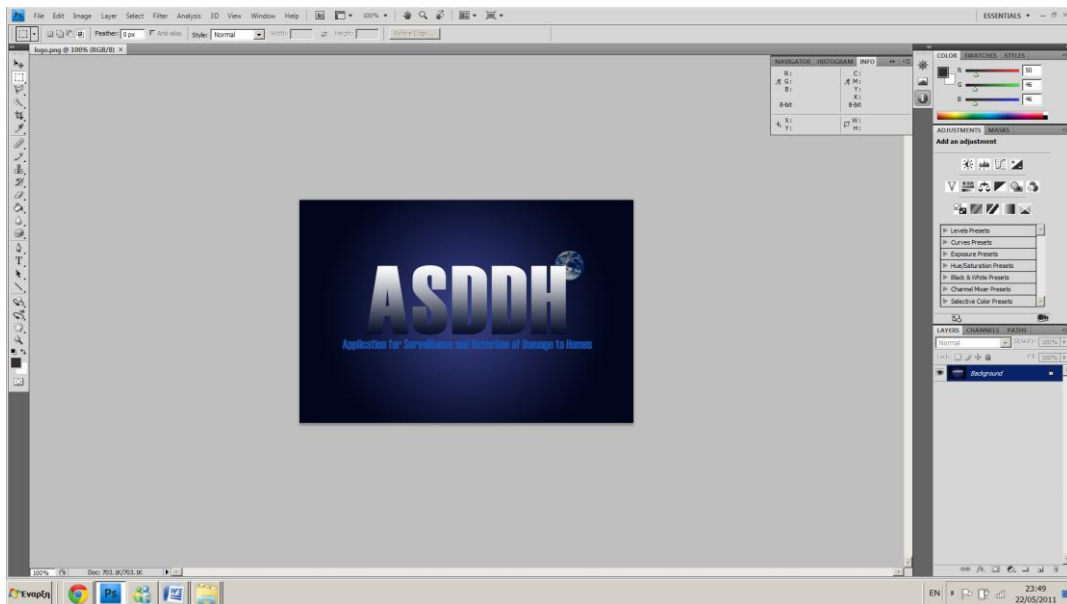


Σχήμα 6.1 Adobe Photoshop

- Πρώτα βάλουμε το όνομα της εφαρμογής ώστε να ξέρη ο χρήστης ότι βρίσκεται στην ιστοσελίδα μας.
- Επιλέξαμε ένα οριζόντιο μενού γιατί πιστεύουμε ότι είναι πιο λειτουργικό και αρκετά συνηθισμένο σε πολλές ιστοσελίδες. Επίσης βάλουμε 5 υπομενού σε αυτό ώστε οι χρήστες να έχουν αρκετές επιλογές στην εφαρμογή.
- Κρίναμε αναγκαίο να βάλουμε 2 επιλογές γλώσσας (ελληνικά, αγγλικά) έτσι ώστε να εξυπηρετούνται όλοι οι χρήστες.
- Στη μέση της ιστοσελίδας θα εμφανίζονται ότι αντίστοιχη στο κάθε υπομενού (π.χ. κείμενο, φόρμα επικοινωνίας)



Σχήμα 6.2 Template



Σχήμα 6.3 Λογότυπο

6.4 Μετατροπή του Template σε κώδικα

Το σημαντικότερο κομμάτι της σχεδίασης είναι ο κώδικας που θα γράψουμε για να υλοποιήσουμε την εφαρμογή που έχουμε σκοπό να κάνουμε. Το πώς και με ποια λογική θα χρησιμοποιήσουμε το κώδικα θα καθορίσει την λειτουργικότητα και βιωσιμότητα της εφαρμογής. Θα χρησιμοποιήσουμε τις γλώσσες προγραμματισμού html που είναι μια καθαρά διαδικτυακή γλώσσα. Και την γλώσσα προγραμματισμού php που μας δίνει την δυνατότητα να έχουμε μία δυναμική ιστοσελίδα με την δυνατότητα να αντλούμε στοιχεία από τη βάση δεδομένων.

Η λογική με την όποια υλοποιήσαμε την εφαρμογή είναι :

Έχουμε μια κύρια σελίδα που ανάλογα με την επιλογή στο μενού καλείτε η αντίστοιχη σελίδα στο κεντρικό σημείο της ιστοσελίδας. Αλλά και ανάλογα με την επιλογή της γλώσσας εμφανίζετε το κείμενο που αντίστοιχη στη κάθε γλωσσά και στο επιλεγμένο υπομενού. Επίσης αν δεν υπάρχει προεπιλογή υπομενού ή γλώσσας τότε έχουμε αυτόματη προεπιλογή την αγγλική γλώσσα και την σελίδα _index.php.

Δηλώνουμε τις προεπιλογές μας. Δηλαδή αν δεν έχουμε επιλέξει υπομενού αυτόματα καλή την σελίδα που έχει menu_id =1 που αντιστοιχεί στην σελίδα _index.php και στο υπομενού welcome. Επίσης γίνεται αυτόματη προεπιλογή της γλώσσας με το να ορίσουμε το lang_id=1 που αντιστοιχεί στην αγγλική γλώσσα, αν δεν έχει γίνει επιλογή από τον χρήστη

```
if (!isset($_REQUEST[menu_id])){$_REQUEST[menu_id]=1;}  
if (!isset($_REQUEST[lang_id])) {$_REQUEST[lang_id]=1;}.
```

.....

Εδώ ορίζουμε στο μενού κάθε λέξη που γράφουμε σε ποια σελίδα αντιστοιχεί (π.χ. products αντιστοιχεί στο menu_id=2) και σε ανάλογα με το τη lang_id που είναι επιλεγμένο καλή το κείμενο που αντιστοιχεί στη συγκεκριμένη γλώσσα

```
<a href="index.php?menu_id=1&lang_id=<?php echo "$_REQUEST[lang_id]"; ?>"
class="menu_link"> welcome </a><span class="sep">|</span>
```

```
<a href="index.php?menu_id=2&lang_id=<?php echo "$_REQUEST[lang_id]"; ?>"
class="menu_link">products</a><span class="sep">|</span>
```

```
<a href="index.php?menu_id=3&lang_id=<?php echo "$_REQUEST[lang_id]"; ?>"
class="menu_link">safe areas</a><span class="sep">|</span>
```

```
<a href="index.php?menu_id=4&lang_id=<?php echo "$_REQUEST[lang_id]"; ?>"
class="menu_link">contact us</a><span class="sep">|</span>
```

```
<a href="index.php?menu_id=5&lang_id=<?php echo "$_REQUEST[lang_id]"; ?>"
class="menu_link">about us</a>
```

.....

Ορίζουμε της γλώσσες. Με το lang_id=1 ορίζουμε την λέξη English και με το lang_id=2 την λέξη ελληνικά. Έτσι ο χρήστης με την επιλογή μιας από της 2 λέξεις επιλέγει τη γλώσσα που θέλει να είναι τα κείμενα της εφαρμογής

```
<td width="269" align="right" ><a href="index.php?menu_id=<?php echo
"$_REQUEST[menu_id]"; ?>&lang_id=1" class="menu1">english</a> |<a
href="index.php?menu_id=<?php echo "$_REQUEST[menu_id]"; ?>&lang_id=2"
class="menu1"> ελληνικά </a></td>
```

.....

Σε αυτό το σημείο του κώδικα κάνουμε ένα έλεγχο, ελέγχουμε την τιμή που πήρε το menu_id που επιλέχθηκε από το χρηστη στο μενου. Αν το menu_id είναι 1 καλή την σελίδα _index.php, αν το menu_id είναι 2 καλή την σελίδα _products.php κ.τλ.

```
if($_REQUEST[menu_id]==1){include('_index.php');}
```

```
if($_REQUEST[menu_id]==2){include('_products.php');}
```

```
if($_REQUEST[menu_id]==3){include('login.php');}
```

```
if($_REQUEST[menu_id]==4){include('_contact.php');}
```

```
if($_REQUEST[menu_id]==5){include('_about.php');}
```

```
if($_REQUEST[menu_id]==31){include('_areas.php');}
```

```
if($_REQUEST[menu_id]==32){include('_insert.php');}
```

```
if($_REQUEST[menu_id]==34){include('lefkada.php');}
```

```
if($_REQUEST[menu_id]==35){include('lefkada_user.php');}
```

```
if($_REQUEST[menu_id]==41){include('email.php');}
```

Κώδικας 6.1 Αρχική σελίδα

Ο κώδικας της αρχικής και βασικής σελίδας index.php όπου δηλώνουμε στο μενού που αντιστοιχεί το και υπομενού. Επίσης φαίνετε το κάλεσμα των σελίδων που αντιστοιχούν στο κάθε υπομενού και την αλλαγή της γλώσσας. Αλλά και την αυτόματη προεπιλογή σελίδας και γλώσσας πριν γίνει επιλογή από το χρήστη

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<link href="defaul.css" rel="stylesheet" type="text/css" />
```

```
<table width="953" border="0">
```

```
<tr>
```

```
<td width="762" align="left" valign="top">
```

```
<?php if ($_REQUEST[lang_id]==1)
```

```
    echo "The application is still at an experimental stage. In theory and in relation to  
    market research and buying mood will be able to monitor and collect information related to  
    home environment (eg temperature, humidity, gas). With the ability to upgrade later so give  
    other information to be requested. <br><br>";
```

```
else
```

```
echo "Η εφαρμογή είναι ακόμη σε πειραματικό στάδιο. Θεωρητικά και σε σχέση με την έρευνα  
αγοράς και την αγοραστική διάθεση θα έχει την δυνατότητα τις παρακολούθησης και τη  
συλλογής πληροφοριών που σχετίζονται με το περιβάλλον του σπιτιού τους (π.χ.  
θερμοκρασία, υγρασία, φυσικό αέριο). Με την δυνατότητα μετέπειτα αναβάθμισης να παρέχει  
άλλα στοιχεία που θα ζητηθούν.<br><br>";
```

```
?></td>
```

```
<td width="175"> </td>
```

```
</tr>
```

```
</table>
```

Κώδικας 6.2 Σελίδα που καλείτε στη κεντρική σελίδα

Ένα παράδειγμα κειμένου που καλείτε από την κεντρική σελίδα είναι το ποιο πάνω. Είναι η σελίδα _products.php και καλείτε όταν ο χρήστης επιλέξει στο μενού την επιλογή products και ανάλογα με τη γλώσσα έχει επιλέξει ο χρήστης κάνη ένα έλεγχο στο lang_id και ανάλογα με την τιμή εμφανίζει το κείμενο. Όπως φαίνετε στο κώδικα το κείμενο το έχω τοποθετήσει σε ένα πίνακα 2 στηλών γιατί να έχω την δυνατότητα να βάλω κάποια φωτογραφία ή ένα flash αρχείο με τα προϊόντα που θέλω να δήξω στο χρήστη. Τα α έβαλα σε πίνακα για να είναι στοιχισμένα και ομοιόμορφα.

Ένα σημαντικό κομμάτι της εφαρμογής είναι το σημείο εισόδου στον χάρτη για να έχει επαφή ο χρήστης ενημέρωση για την οικία που τον αφορά. Επίσης πρέπει να έχει πρόσβαση και ένας χρήστης 'administrator' που θα έχει την δυνατότητα να παρακολουθεί όλα τα σημεία του χάρτη, αλλά και να καταχώρηση καινούργια σημεία.

Σε αυτό το σημείο δηλώνουμε τους κωδικούς που έχουν δικαίωμα να χρήσης της εφαρμογής. Αν είναι διαχειριστής ο χρήστης πέρνη την τιμή 1 στη μεταβλητη logged, αν είναι απλός χρήστης πέρνη την τιμή 2.

```
<?php if (($_POST[username]=='admin') && ($_POST[password]=='adm1n')) $logged=1;
if (($_POST[username]=='user') && ($_POST[password]=='us3r'))
$logged=2;
else if (($_POST[username]=='user') && ($_POST[password]=='us2r'))
$logged=2;
else if (($_POST[username]=='user') && ($_POST[password]=='us1r'))
$logged=2;
?>
```

Η φόρμα που εισάγουμε username και password, με το \$_POST[username] και \$_POST[password] παίρνουμε τα στοιχεία για να τα ελέγξουμε αν είναι σωστά.

```
<form name="login" method="post">
<input name="logged" type="hidden" value="1"/>
    Username :<br>
    <input name="username" type="text" size="12" maxlength="8" value="<?php
echo"$_POST[username]"; ?>"/><br>
    Password : <br>
    <input name="password" type="password" size="12" maxlength="8" value="<?php
echo"$_POST[password]"; ?>"/><br>
    <input type="submit" value="Login" class="input1"/>
</form>
```

Ελέγχουμε την τιμή που πήρε η μεταβλητή `logged` αν είναι 0 ζητά να εισάγουμε User Name και Password, αν είναι 1 ελέγχει αν είναι σωστά τα στοιχεία και είναι πηγαίνει στην σελίδα που αντίστοιχη στο χρήστη διαχειριστή αν δεν είναι σωστά εμφανίζεται μήνυμα λάθους. Αν είναι 2 το `logged` ελέγχει αν είναι σωστά τα στοιχεία του απλό χρήστη, αν είναι μεταφέρετε σε άλλη σελίδα όπου αντιστοιχίζετε το password του με το σημείο που τον αφορά για να εμφανιστή στο χάρτη, αν δεν είναι σωστά τα στοιχεία τότε εμφανίζει μήνυμα λάθους.

```

if ($logged==0)
    echo "enter User Name and Password";

if ($logged==1)
{
    if (($_POST[username]=='admin') && ($_POST[password]=='adm1n'))
        echo "<meta http-equiv='refresh' content='0;URL=index.php?menu_id=31'>";
    else
        echo "<font color='#CC0033'>Invalid Username and/or Password!</font><br>Try again!";
}

if ($logged==2)
{
    if (($_POST[username]=='user') && ($_POST[password]=='us3r'))
        echo "<meta http-equiv='refresh' content='0;URL=index.php?menu_id=35&p=$_REQUEST[password]'>";
        else if (($_POST[username]=='user') && ($_POST[password]=='us2r'))
            echo "<meta http-equiv='refresh' content='0;URL=index.php?menu_id=35&p=$_REQUEST[password]'>";

            else if (($_POST[username]=='user') && ($_POST[password]=='us1r'))
                echo "<meta http-equiv='refresh' content='0;URL=index.php?menu_id=35&p=$_REQUEST[password]'>";
                else
                    echo "<font color='#CC0033'>Invalid Username and/or Password!</font><br>Try again!";
}
    
```

Κώδικας 6.3 Φόρα εισόδου στο χάρτη (login)

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-7"> <meta http-equiv="refresh" content="60"> <!-- Εδώ ανανεώνουμε την σελίδα κάθε 60 δευτερόλεπτα για να ανανεώνονται και οι πληροφορίες -->
```

.....

Στο ποίο κάτω script τοποθετούμε την διεύθυνση του χάρτη που μας δόθηκε από το Google map. Στο σημείο key βάζουμε το κλειδί , κάθε ιστοσελίδα έχει δικό της κλειδί και δεν μπορεί να το χρησιμοποιήσει άλλη σελίδα

```
<script
src="http://maps.google.com/maps?file=api&v=2&sensor=false;key=ABQIAAAAxQ
_2SfoEMFa7I4G7VdOylRTIEJiiQkmuaLMEhh89rn5lh2bPKhSi7H09sAhet7QIDd09gbAKs6xql
g" type="text/javascript"></script>
```

.....

Εδώ ορίζονται οι διαστάσεις του πίνακα

```
<script type="text/javascript">
```

```
//<![CDATA[
```

```
var iconBlue = new GIcon();
```

```
iconBlue.image = 'http://labs.google.com/ridefinder/images/mm_20_blue.png';
```

```
iconBlue.shadow = 'http://labs.google.com/ridefinder/images/mm_20_shadow.png';
```

```
iconBlue.iconSize = new GSize(12, 20);
```

```
iconBlue.shadowSize = new GSize(22, 20);
```

```
iconBlue.iconAnchor = new GPoint(6, 20);
```

```
iconBlue.infoWindowAnchor = new GPoint(5, 1);
```

```
var iconRed = new GIcon();
```

```
iconRed.image = 'http://labs.google.com/ridefinder/images/mm_20_red.png';
```

```
iconRed.shadow = 'http://labs.google.com/ridefinder/images/mm_20_shadow.png';
```

```
iconRed.iconSize = new GSize(12, 20);
```

```
iconRed.shadowSize = new GSize(22, 20);
```

```
iconRed.iconAnchor = new GPoint(6, 20);
```

```
iconRed.infoWindowAnchor = new GPoint(5, 1);
```

```
var customIcons = [];  
  
customIcons["c1"] = iconBlue;  
  
customIcons["c2"] = iconRed;  
  
function load() {  
    if (GBrowserIsCompatible()) {  
        var map = new GMap2(document.getElementById("map"));  
        map.addControl(new GSmallMapControl());  
        map.addControl(new GMapTypeControl());  
        map.setCenter(new GLatLng(38.710000, 20.670000), 11);  
    }  
}
```

.....

Στο σημείο αυτό καλούμε ένα αρχείο php. Εμείς έχουμε 2 πανομοιότυπα php αρχεία, 1 για τους απλούς χρήστες (phpsqlajax_genxml_user.php?) που εκεί γίνεται η αντιστοίχιση του κωδικού τους με το σημείο που τους αφορά και τες πληροφορίες που αφορούν το σημείο τους. Αυτό για να γίνει πρέπει να μεταφερθεί και ο κώδικας για τον έλεγχο(<?php echo "\$_REQUEST[p]"; ?>). Αν ο χρήστης είναι διαχωριστής τότε καλείτε phpsqlajax_genxml.php. Να αναφέρουμε εδώ ότι εδώ καλήτε και το xml αρχείο (το συννεφάκι με τις πληροφορίες).

```
GDownloadUrl("phpsqlajax_genxml_user.php?p=<?php echo "$_REQUEST[p]"; ?>",  
function(data) {  
    var xml = GXml.parse(data);  
  
    var markers = xml.documentElement.getElementsByTagName("marker");  
  
    for (var i = 0; i < markers.length; i++) {  
        var name = markers[i].getAttribute("name");  
        var address = markers[i].getAttribute("address");  
        var temp = markers[i].getAttribute("temp");  
        var humidity = markers[i].getAttribute("humidity");  
  
        var type = markers[i].getAttribute("type");  
        var point = new GLatLng(parseFloat(markers[i].getAttribute("lat")),  
            parseFloat(markers[i].getAttribute("lng")));  
        var marker = createMarker(point, name, address, type, temp, humidity);  
        map.addOverlay(marker);  
    }  
}
```

```

function createMarker(point, name, address, type, temp, humidity) {
    var marker = new GMarker(point, customIcons[type]);
    var html = "<b>" + name + "</b> <br/>" + address + "<b> <br/>temp: " + temp + "oC" +
"<b> <br/>humidity: " + humidity ;
    GEvent.addListener(marker, 'click', function() {
        marker.openInfoWindowHtml(html);
    });
    return marker;
}
//]]>
</script>

```

.....

Ποίο κάτω ορίζουμε το style τις σελίδας μέσω το css αρχείου όπου εκεί ορίζουμε τη γραμματοσειρά κτλ

```

<link href="defaul.css" rel="stylesheet" type="text/css" />
<link href="*.css" rel="stylesheet" type="text/css" />
<link href="defaul.css" rel="stylesheet" type="text/css" />

```

Κώδικας6.4 Σελίδα lefkada_user.php

Επικοινωνία με τη βάση δεδομένων για να πάρουμε τα στοιχεία που χρειαζόμαστε

```

<?php
require("dbconnect.php");

```

Δημιουργία του xml αρχείου

```

function parseToXML($htmlStr)
{

```



```
$xmlStr=str_replace('<','&lt;',$htmlStr);  
$xmlStr=str_replace('>','&gt;',$xmlStr);  
$xmlStr=str_replace('"','&quot;',$xmlStr);  
$xmlStr=str_replace("'",'&apos;',$xmlStr);  
$xmlStr=str_replace("&","&amp;",$xmlStr);  
return $xmlStr;  
}
```

```
$connection=mysql_connect ('localhost', $username, $password);  
if (!$connection) {  
    die('Not connected : ' . mysql_error());  
}  
  
$db_selected = mysql_select_db($database, $connection);  
if (!$db_selected) {  
    die ('Can\'t use db : ' . mysql_error());  
}
```

.....

Ερώτηση στη βάση για να μας εμφανίσει τα στοιχεία που αντιστοιχούν στο χρήστη με τον αντίστοιχο κωδικό. Αν δεν υπάρχουν στοιχεία τότε εμφανίζει μήνυμα λάθους. Αν υπάρχουν στοιχεία τότε τα εμφανίζει και τα τοποθετεί στις μεταβλητές του xml αρχείου.

```
$query = "SELECT * FROM markers WHERE pass='$_REQUEST[p]";  
$result = mysql_query($query);  
if (!$result) {  
    die('Invalid query: ' . mysql_error());  
}  
  
header("Content-type: text/xml");
```

```

// Start XML file, echo parent node

echo '<markers>';

// Iterate through the rows, printing XML nodes for each

while ($row = @mysql_fetch_assoc($result)){

    // ADD TO XML DOCUMENT NODE

    echo '<marker ' . parseToXML($row['name']) . " " . ' ';
    echo 'name="' . parseToXML($row['name']) . " " . ' ';
    echo 'address="' . parseToXML($row['address']) . " " . ' ';
    echo 'lat="' . $row['lat'] . " " . ' ';
    echo 'lng="' . $row['lng'] . " " . ' ';
    echo 'type="' . $row['type'] . " " . ' ';
    echo 'temp="' . $row['temp'] . " " . ' ';
    echo 'humidity="' . $row['humidity'] . " " . ' ';

    echo '>';

}// End XML file

```

Κώδικας 6.5 phpsqlajax_genxml_user.php

Το ίδιο συμβαίνει και στο αρχείο phpsqlajax_genxml.php που είναι για τον διαχειριστή χωρίς τον έλεγχο του κωδικού πρόσβασης.

Επικοινωνία με τη βάση για να μπορέσουμε να καταχωρήσουμε τα στοιχεία.

```
<?php require("dbconnect.php");
```

Εντολή εισαγωγής των δεδομένων που πήραμε από τη φόρμα από τη σελίδα_inster.php που συμπλήρωσε ο διαχειριστής.

```

    $q2=mysql_query("INSERT INTO markers (name, address, lat, lng, type, ) VALUES
    ('$_POST[name]','$_POST[address]','$_POST[lat]','$_POST[lng]','$_POST[type]' )");

```

Κώδικας 6.6 Εισαγωγή στοιχείων στη βάση και στο πίνακα markers

Φόρμα επικοινωνίας Εισάγει ο χρήστης τα στοιχεία του και αν είναι σωστά γραμμένο (συντακτικά) το email του τότε επικοινωνεί με τη σελίδα email.php (index.php?menu_id=41).

```
<script src="SpryValidationTextField.js" type="text/javascript"></script>
<link href="SpryValidationTextField.css" rel="stylesheet" type="text/css" />
<p><br />
<em>email: panayiotou_p@yahoo.com</em></p>
<form name="contact" method="post" action="index.php?menu_id=41">
Name- Surname:<br />
<input name="name" type="text" size="40" maxlength="60" />
<br /> <br />
Email :
<br />
<span id="sprytextfield1">
<label>
</label>
<span class="textfieldRequiredMsg">A value is required.</span></span>
<input type="text" name="email"/><br>
<br />Comments:
<br>
<textarea name="message" id="textarea" cols="45" rows="5"></textarea>
<br /><br />
<input type="reset" name="reset" value="reset" class="button" /> <input type="submit"
name="send" value="send" class="button" />
</form>
<script type="text/javascript">
<!--var sprytextfield1 = new Spry.Widget.ValidationTextField("sprytextfield1");
/--> </script>
```

Κώδικας 6.7 Φόρμα επικοινωνίας

Αν υπάρχει περιεχόμενο στη μεταβλητή message τότε επικοινωνεί με τη σελίδα postman.php για να γίνει η αποστολή των στοιχείων του χρήστη. Και μετά εμφανίζει ένα μήνυμα ότι έχει πάει το email και με link δίνει την δυνατότητα στο χρήστη να μεταβεί στην πρώτη σελίδα.

```
<?php
require_once "postman.php";
if ($_REQUEST[message]!=NULL)
{

postman($_REQUEST[email],
        'panayiotou_p@yahoo.com',
        'epikoinonia apo website',
        $_REQUEST[message]',
        $_REQUEST[message]);
}
?>
<link href="defaul.css" rel="stylesheet" type="text/css" />

<br /><p align="center" class="main_text">email has sent <br>
<a href="index.php" class="main_text"><em>go back to home page</em></a>
```

Κώδικας 6.8 email.php

Ένα πολύ σημαντικό κομμάτι της εφαρμογής αφού εδώ γίνεται αυτόματος έλεγχος των σημείων της βάσης κάθε 1 λεπτό και αν υπάρχει κίνδυνος σύμφωνα με τις συνθήκες που θέσαμε στέλνεται ένα email σε κάποιον αρμόδιο που έχουμε ορίσει εμείς. Επίσης κάθε 6 ώρες στέλνεται 1 ενημερωτικό email ανεξάρτητα αν υπάρχει κίνδυνος ή όχι. Πρέπει να πουλούμε εδο ότι για να λειτουργήσει ο συγκεκριμένος κώδικας πρέπει τα modus που θα χρησιμοποιήσουμε στις οικίες να τα ρυθμίσουμε έτσι ώστε αν υπάρχει μεταβολή της θερμοκρασίας στα επίπεδα που ορίζουμε εμείς (π.χ. κάτω από 40 ή πάνω από 40 και κάτω από 50 ή πάνω από 50) τότε αυτόματα θα αλλάζει το πεδίο m_status του πίνακα markers από 0 σε 1, αυτό θα γίνεται επίσης κάθε 6 ώρες για το ενημερωτικό email.php.

Ο συγκεκριμένος κώδικας τρέχει ανεξάρτητα αν είναι σε λειτουργία η εφαρμογή. Αυτό γίνεται μέσα από ρυθμίσεις του server που χρησιμοποιεί η εφαρμογή.

Σύνδεση με τη βάση για πάρουμε τες πληροφορίες για τον έλεγχο και με το classes.php γία να στείλουμε το ενημερωτικό email.

```
<?php
```

```
require_once "classes.php";
```

```
require_once "dbconnect.php";
```

.....

Ερώτηση στο πίνακα markers και ενημέρωση των μεταβλήτων με τα πεδία που μας αφορούν.

```
$q_Read_Temperature = mysql_query ("select * from markers");
```

```
while ($data=mysql_fetch_array($q_Read_Temperature))
```

```
{
```

```
    $id=$data[id];
```

```
    $temp=$data[temp];
```

```
    $m_status=$data[m_status];
```

```
    $name=$data[name];
```

```
    $address=$data[address];
```

.....

Έλεγχος σε τις θερμοκρασίας για να δούμε ποιο μήνυμα θα στείλουμε. Αυτό εξαρτάτε τη τιμή έχει η μεταβήτη message.

```
if ($temp<40) $message=0; else
```

```
    if (($temp>=40) && ($temp<50)) $message=1; else
```

```
    if ($temp>=50) $message=2;
```

```
    if ($m_status==1)
```

```
    {
```

```
        if ($message==0) postman('panayiotou_p@yahoo.com',  
'panayiotou.p86@gmail.com', 'Everything is ok', 'Temperature in station $id name: $name ,  
address: $address is back to normal. station name: $name , station address: $address ',  
'Temperature in station $id is back to normal. station name: $name , station address:  
$address ');
```

```

postman('panayiotou_p@yahoo.com', 'panayiotou_p1@hotmail.com', 'Everything is ok',
'Temperature in station $id name: $name , address: $address is back to normal. station
name: $name , station address: $address ', 'Temperature in station $id is back to normal.
station name: $name , station address: $address ');

        if ($message==1) postman('panayiotou_p@yahoo.com',
'panayiotou.p86@gmail.com', 'Warning', 'Temperature in station $id is above 40 degrees!
station name: $name , station address: $address ', 'Temperature in station $id is above 40
degrees! station name: $name , station address: $address ');

postman('panayiotou_p@yahoo.com', 'panayiotou_p1@hotmail.com', 'Warning', 'Temperature
in station $id is above 40 degrees! station name: $name , station address: $address ',
'Temperature in station $id is above 40 degrees! station name: $name , station address:
$address ');

        if ($message==2) postman('panayiotou_p@yahoo.com',
'panayiotou.p86@gmail.com', 'Danger', 'Temperature in station $id is above 50 degrees.
station name: $name , station address: $address ', 'Temperature in station $id is above 50
degrees. station name: $name , station address: $address ')

postman('panayiotou_p@yahoo.com', 'panayiotou_p1@hotmail.com', 'Danger', 'Temperature
in station $id is above 50 degrees. station name: $name , station address: $address ',
'Temperature in station $id is above 50 degrees. station name: $name , station address:
$address '    );

Μηδενίζουμε τη μεταβλητή m_status για να μην γίνετε συνέχεια ο έλεγχος και η
αποστολή του email.

$q_reset_m_status = mysql_query ("update markers set m_status='0' where s_id=$s_id");

        }}

?>

<meta http-equiv="Refresh" content="60;url=warning.php"> <!--Κάθε 1 λεπτό γίνετε
ανανέωση της σελίδας -->

```

Κώδικας 6.9 warring.php

Για να μπορούμε να έχουμε επικοινωνία με τη βάση, δηλαδή να παίρνουμε και να εισάγουμε στοιχεία πρέπει να γίνει μια σύνδεση. Αυτό γίνεται με το ποιο κάτω τρόπο, βάζοντας τα στοιχεία που χρειάζονται.

```
<?php
```

```
$dbhost="localhost";  
$username="root";  
$password="";  
$database="itgooglemaps";  
  
?>
```

Κώδικας 6.10 dbconnect.php

Για να έχουμε μία ομοιόμορφη ιστοσελίδα και χωρίς να γράφουμε περιττό κώδικα με το αρχείο CSS αυτό γίνετε ποιο εύκολο με το να δηλώνουμε μια κλάση για κάθε διαφορετική ομάδα της ιστοσελίδας (π.χ. για το κείμενο τις ιστοσελίδας χρησιμοποιούμε τη κλάση main_text

```
@charset "utf-8";  
  
.main_text  
{  
    font-size:18px;  
    color:#FFF;  
    text-decoration: none;  
    font-style: normal;  
    font-family: Arial, Helvetica, sans-serif;  
    text-transform: none;  
    text-decoration:none;  
}  
  
.map  
{font: Tahoma, Geneva, sans-serif;  
font-size: 18px;  
color:#000;  
}
```

```
.menu1
{
    font-family: "Times New Roman", Times, serif;
    font-size: 14px;
    color: #f8f8f8;
    padding-right: 4px;
    text-decoration: none;
}
```

```
.menu_link
{
    font-family: Haettenschweiler;
    font-size: 36px;
    color: #f8f8f8;
    text-decoration:none;
padding-right: 4px;
    padding-left: 4px;
    padding-top: 2px;
    padding-bottom: 2px;
}
```

```
.menu_link_active
{
    font-family: Haettenschweiler;
    font-size: 36px;
    color: #f8f8f8;
    padding-right: 4px;
    padding-left: 4px;
    padding-top: 2px;
    padding-bottom: 2px;
```



```
}  
.sep  
{  
    font-family: Haettenschweiler;  
    font-size: 36px;  
    color: #f8f8f8;  
    padding-right: 5px;  
    padding-left: 5px;  
}  
.link {  
    font-size:18px;  
    color: #FFF;  
    text-decoration:none;  
    font-style: normal;  
    font-family: Arial, Helvetica, sans-serif;  
    text-transform: none;  
}  
  
.main_text1  
  
{  
    font-size:18px;  
    color:#FFF;  
    text-decoration: underline;  
    font-style: normal;  
    font-family: Arial, Helvetica, sans-serif;  
    text-transform: none;  
}  
  
.button  
{color:#FFF;
```

```
background:#006;
```

```
border:#006;
```

```
font-size:16px;
```

```
font-family:Arial, Helvetica, sans-serif;
```

Κώδικας 6.11 Css αρχείο

Κεφάλαιο 7

7. Αξιολόγηση Εφαρμογής

7.1 Εισαγωγή

Στο κεφάλαιο της αξιολόγησης της εφαρμογής θα εξετάσουμε αν η εφαρμογή εκπληρώνει τις απαιτήσεις που θέσαμε, αν είναι λειτουργική ποια τα πλεονεκτήματα και ποιες οι αδυναμίες της.

7.2 Εκπλήρωση απαιτήσεων

Οι εφαρμογή πιστεύουμε εκπληρώνει κατά πολύ τις απαιτήσεις που θέσαμε. Δηλαδή:

- Είναι αυτόνομη αλλά και με δυνατότητα από το χρήστη να δει από μόνος του τη συμβαίνει στην οικία του.
- Είναι γρήγορη και σε πραγματικό χρόνο.
- Βοηθά στην πρόληψη ζημιών σε μια οικία.
- Παρέχει μια αίσθηση ασφάλειας στους χρήστες της.
- Συλλέγει και επεξεργάζεται πληροφορίες περιβάλλοντος μια οικίας (θερμοκρασίας, υγρασίας), με τη δυνατότητα με μικρές αλλαγές του κώδικα να έχει την δυνατότητα και για άλλες πληροφορίες.

7.3 Λειτουργικότητα της εφαρμογής

Η λειτουργικότητα της εφαρμογής είναι σε πολύ καλά επίπεδα. Μετά το τέλος της εφαρμογής ζητήσαμε από κόσμο που δεν έχει σχέση με τέτοιου είδους εφαρμογές να τη χρησιμοποιήσει και να μας πει τις απόψεις του. Τα σχολεία που λάβαμε ήταν πολύ ευχάριστα αφού όλοι το βρήκαν εύκολο στη χρήση και ότι εξυπηρετούσε τις ανάγκες τους. Να σημειώσουμε ότι η δοκιμή από τους χρήστες έγινε εκτός από Η/Υ (φορητούς και επιτραπέζιους) και σε smart phone αρκετή επιτυχία.

7.4 Πλεονεκτήματα

- Αρκετά γρήγορη εφαρμογή.
- Παρέχει ενημέρωση και χωρίς να λειτουργεί ο χρήστης την εφαρμογή.
- Δυνατότητα αναβάθμισης.

7.5 Μειονεκτήματα

Όπως όλες οι εφαρμογές έχουν και μειονεκτήματα έτσι και η δική μας εφαρμογή έχει μερικά. Είναι κάλο να επισημάνουμε τα μειονεκτήματα της εφαρμογής έτσι ώστε στο μέλλον να προσπαθήσουμε να την βελτιώσουμε.

- Το πρώτο που θέτουμε ως μειονέκτημα είναι η ασφάλεια που παρέχει στους χρήστες που εισάγουν username και password για να μπορέσουν να ενημερωθούν για την οικία τους. Δεν μας ικανοποιεί τα μέτρα ασφάλειας για προφύλαξη του κωδικού του χρήστη και των πληροφοριών που ανακτά η εφαρμογή από τη βάση δεδομένων.
- Η εφαρμογή είναι μία διαδικτυακή εφαρμογή. Χωρίς το διαδίκτυο δεν είναι δυνατή η λειτουργία της και από πλευράς ενημέρωσης των χρηστών αλλά και από πλευράς ενημέρωσης της βάσης από τους αισθητήρες, αφού χλωρής διαδίκτυο δεν επικοινωνούν.
- Το τελευταίο σημείο που θέλουμε να θίξουμε δεν αφορά άμεσα την εφαρμογή (σαν λογισμικό σύστημα), αλλά για να λειτουργήσει σωστά η εφαρμογή πρέπει να υπάρχουν και οι αισθητήρες και τα modus. Στην ουσία είναι το hardware κομμάτι της εφαρμογής αφού εκείνα συλλέγουν και επεξεργάζονται σε πρώτο βαθμό τις πληροφορίες της μετατρέπουν σε ψηφιακή μορφή και τις αποθηκεύουν στην βάση. Στην διάρκεια της ενασχόλησης μας με την συγκεκριμένη εργασία διαπιστώσαμε ότι η εύρεση αξιόπιστου συμβατού και φτηνού υλικού είναι πολύ δύσκολη, και η άμεση αποστολή τους από τις εταιρίες ακόμα πιο δύσκολη.

Πιστεύουμε ωστόσο σε μία ενδεχόμενη επαγγελματική χρήση της εφαρμογής ότι όλα τα μειονεκτήματα που θέσαμε αν δεν ξεπεραστούν πλήρως θα μειωθούν στο ελάχιστο.

7.6 Συμπέρασμα

Τα συμπεράσματα που βγάζουμε από το κεφάλαιο της αξιολόγησης είναι ότι η εφαρμογή είναι σε πολύ καλό επίπεδο και τα μειονεκτήματα που εντοπίσαμε δεν είναι απροσπέλαστα. Εν κατακλείδι συμπεραίνουμε ότι η εφαρμογή μας καλύπτει σε ένα μεγάλο ποσοστό τον αρχικό μας στόχο.

Κεφάλαιο 8

8. Συμπεράσματα

Μετά από πολλούς μήνες εργασίας, άγχους και γκρίνιας φτάσαμε στο τέλος τις πτυχιακής μας εργασίας. Μιας εργασίας που χωρίζετε σε 2 μέρη, το θεωρητικό μέρος που εκεί μελετήσαμε και καταγράψαμε στοιχεία που μας βοήθησαν να πάρουμε συγκεκριμένες πληροφορίες για το θέμα της πτυχιικής μας. Μάθαμε αρκετά πράγματα για τα G.I.S, το Google maps και τα APIs.

Επίσης κάναμε μια έρευνα αγοράς όπου μας βοήθησε προς τα πού να κινηθούμε για τις απαιτήσεις που θα βάζαμε στην εφαρμογή και προς τα πού κινητέ η αγοράς με βάση τα G.I.S.

Μετά ασχοληθήκαμε με τη σχεδίαση και την υλοποίηση της εφαρμογής, με πολλές σπαζοκεφαλιές και ατελείωτες γραμμές κώδικα καταφέραμε να φτιάξουμε μία εφαρμογή αρκετά αξιόπιστη και χρήσιμη. Μπορεί η εφαρμογή να μην είναι μια καινούργια ανακάλυψη στο χώρο της πληροφορικής αφού υπάρχουν αρκετές πανομοιότυπες αλλά πιστεύουμε ότι έχει την προοπτική να γίνει μια αρκετά καλή εφαρμογή και να συναγωνίζεται με τις άλλες.

Να πούμε επίσης ότι η τεχνολογία των G.I.S έχει μεγάλη ανάπτυξη και νομίζουμε ότι μπορούν να γίνουν πολλές εφαρμογές για να εξυπηρετήσουν τους ανθρώπους. Επίσης είναι αδιανόητο να μην αναφερθούμε στην Google που με το Google maps και Google Earth έβαλε ένα ολόκληρο κόσμο στο σπίτι μας και τη δυνατότητα που δύνη με τον ανοικτό κώδικα που παρέχει την ευκαιρία σε ανθρώπους που έχουν την όρεξη και τον χρόνο να «παίξουν» μαζί του και να δημιουργήσουν ωραία πράγματα

Πηγές

Βιβλιογραφία

ΒΑΣΙΛΕΙΟΣ ΒΕΣΚΟΥΚΗΣ : Τεχνολογία Λογισμικού Ι (Πάτρα 2000)
Τασόπουλος Αναστάσιος : Πληροφορικά Συστήματα

Ιστοσελίδες

<http://domainresellers.gr/what-is-api.php>

<http://delivorias.me>

<http://www.websys.gr>

<http://www.hellascams.gr>

Πτυχιακές – εργασίες – Σημειώσεις

Σοφός Ιωάννης: Ανάπτυξη εφαρμογής ανάκτησης και απεικόνισης πληροφοριών με χωρική διάσταση από προτυποποιημένη Βάση Δεδομένων σε ελεύθερα υπόβαθρα με χρήση Google Maps API (Αθήνα 2008)

Δωροθέα Αυφαντοπούλου : Διαχρονική Παρακολούθηση Δασών με την Δορυφορική Τεχνολογία. (Αθήνα 2005)

Α.Τασόπουλος–Α. Μιμής : Προγράμματα Η/Υ στη Γεωγραφική Ανάλυση.

Δρ. Βασιλόπουλος Αντρέας, Χαρτίδου Κωνσταντίνα, Τσιλιγιάνη Λαμπρινή : Γεωγραφικά Συστήματα Πληροφοριών

Web GIS Εφαρμογές Πολεοδομίας Δήμου Ρεθύμνου(2008)

Κ.Κουτσόπουλος, Ν Ευελπίδου, Α Βασιλόπουλος: ΓΣΠ Χρηση του MapsInfo Prof.

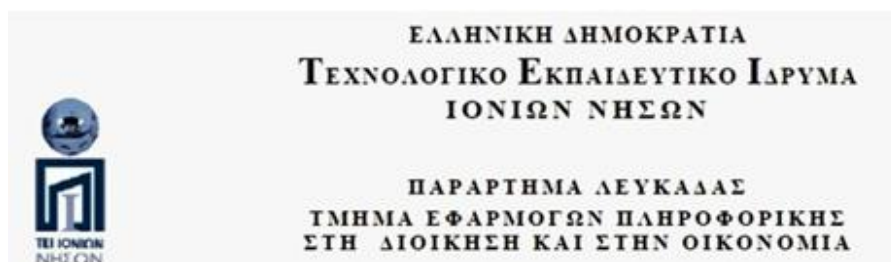
Αργυριάδης Χρήστος Γεωγραφικά Συστήματα Πληροφοριών (2002)

Μαρία Κολωνιάρη , Ειρήνη Χρυσάδακου : Διεπαφές (API – Application Program Interface) (ΑΘΗΝΑ 2007-2008)

Παραρτήματα

Παράρτημα Α

Ερωτηματολόγιο



Υπεύθυνος καθηγητής

Δρ. Αθανάσιος Κακαρούντας

ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ

Αξιότιμε κύριε / Αξιότιμη κυρία

Το ερωτηματολόγιο αυτό έχει σκοπό να κάνει μια έρευνα κατά πόσο οι εφαρμογές που χρησιμοποιούν Γεωγραφικά Συστήματα Πληροφοριών (G.I.S) είναι γνωστές στο ευρύ κοινό. Επίσης ερευνά κατά πόσο μια εφαρμογή ασφάλειας μιας οικίας ή ενός κτιρίου θα έχει απήχηση στην αγορά.

Επίσης, στόχο έχει να εισάγει με απλές ερωτήσεις, το κοινό στις έννοιες των GIS και να αποδείξει ότι κάποιος μπορεί εύκολα να τις κατανοήσει ανεξαρτήτως φύλου, ηλικίας και επιπέδου μορφώσεως.

Υπεύθυνοι της έρευνας

Παπαδόπουλος Νικόλαος

Παναγιώτου Παναγιώτης

ΔΗΜΟΓΡΑΦΙΚΑ ΣΤΟΙΧΕΙΑ

1. Φύλο

α. Άνδρας

β. Γυναίκα

2. Ηλικία

α. 18-24

β. 25-36

γ. 37-54

δ. 55 και άνω

3. Ποιο είναι το επίπεδο μόρφωσης σας;

α. Βασικό (Δημοτικό)

β. Μεσαίο (Γυμνάσιο - Λύκειο)

γ. Ανώτερο (Πανεπιστήμιο)

4. Ασχολείστε με το INTERNET(ΔΙΑΔΙΚΤΥΟ);

α. Αρκετά

β. Όχι τόσο συχνά

γ. Παρακολουθώ μέσω τρίτων(π.χ. Μέσω των παιδιών μου)

ΕΞΟΙΚΕΙΩΣΗ ΜΕ ΤΑ GIS

5. Γνωρίζετε τι είναι τα G.I.S(Γεωγραφικά συστήματα πληροφοριών) συστήματα;

α. ΝΑΙ

β. ΟΧΙ

(Αν απαντήσατε **ΟΧΙ** παρακαλούμε να ανατρέξετε στην τελευταία σελίδα που σας δίνεται μια γρήγορη εξήγηση για τα γεωγραφικά συστήματα πληροφοριών. Αυτό θα σας βοηθήσει να συνεχίσετε το ερωτηματολόγιο. Αν απαντήσατε **ΝΑΙ** μπορείτε να πάτε κατευθείαν στην 6 ερώτηση).

6. Μια εταιρία που χρησιμοποιεί τα G.I.S για τις εφαρμογές της είναι η Google. Τα πιο γνωστά στο διαδίκτυο είναι το Google earth και Google maps. Έχετε χρησιμοποιήσει ποτέ αυτές τις υπηρεσίες ή έχετε δει τρίτους να τις χρησιμοποιούν (ποιες οι εντυπώσεις σας σε αυτή την περίπτωση);

	1.Πολύ ενδιαφέρον	2.Όχι και τόσο ενδιαφέρον	3.Δεν ξέρω δεν έχω γνώμη	ΑΠΑΝΤΗΣΕΙΣ
α. ΝΑΙ έχω χρησιμοποιήσει	A-1	A-2	A-3	
β. ΟΧΙ δεν έχω χρησιμοποιήσει	B-1	B-2	B-3	
γ. Έχω δει από τρίτους να τα χρησιμοποιούν	Γ-1	Γ-2	Γ-3	

7. Έχετε χρησιμοποιήσει κάποια άλλη υπηρεσία που βασίζεται σε τέτοια συστήματα (G.I.S);

α. ΝΑΙ

β. ΟΧΙ

γ. Αν ΝΑΙ ποια;(ΜΟΝΟ ΟΝΟΜΑΣΤΙΚΑ).....

8. Που πιστεύετε ότι θα χρησίμευαν περισσότερο τα συστήματα αυτά. (Η απάντηση αυτού το ερωτήματος είναι συνδεδεμένη άμεσα με τις δυο παρακάτω ερωτήσεις)

- | | |
|--|--------------------------|
| Περιφερειακός Προγ/σμός – Σχεδιασμός | <input type="checkbox"/> |
| Αστικός Προγραμματισμός – Σχεδιασμός | <input type="checkbox"/> |
| Συγκοινωνίες – Μεταφορές | <input type="checkbox"/> |
| Τεχνική υποδομή | <input type="checkbox"/> |
| Διαχείριση Περιβάλλοντος | <input type="checkbox"/> |
| Φορολογία | <input type="checkbox"/> |
| Τηλεπικοινωνίες | <input type="checkbox"/> |
| Εκπαίδευση και Υγεία – Κοινωνική Πρόνοια | <input type="checkbox"/> |
| Πυροσβεστική, Δασική Υπηρεσία, Αστυνομία | <input type="checkbox"/> |
| Τουρισμός – Αναψυχή | <input type="checkbox"/> |
| Ανάλυση Αγοράς | <input type="checkbox"/> |
| Αγορά Εργασίας | <input type="checkbox"/> |
| Δίκτυα διανομών, πωλήσεων και χωροθετήσεις κατανομών | <input type="checkbox"/> |

9. Σύμφωνα με την απάντηση που δώσατε στην ερώτηση 8, πιστεύετε ότι είναι εφικτή (ή αν υπάρχει η τεχνολογία) η ανάπτυξη μίας τέτοιας υπηρεσίας;

α. ΝΑΙ

β. ΟΧΙ

10. Πιστεύετε ότι μπορεί να σας βοηθήσουν τα GIS στην ασφάλεια και στη βελτίωση της ποιότητας της ζωής σας;

α. ΝΑΙ

β. ΟΧΙ

ΕΡΕΥΝΑ ΑΓΟΡΑΣ

11. Πιστεύετε ότι θα ήταν χρήσιμη για εσάς μια εφαρμογή βασισμένη στα Γεωγραφικά συστήματα πληροφοριών(G.I.S) για επίβλεψη και ασφάλεια της οικίας σας ή κάποιου χώρου που θα θέλατε να προστατεύεται;

α. Πολύ ενδιαφέρον.

β. Ενδιαφέρον αλλά δύσκολο στην υλοποίηση.

γ. Δεν ενδιαφέρομαι άμεσα.

12. Θα διαθέτατε ένα χρηματικό ποσό για την εγκατάσταση και την λειτουργία μίας τέτοιας εφαρμογής;

α. ΝΑΙ

β. ΟΧΙ

13. Αν απαντήσατε θετικά στην πιο πάνω ερώτηση, δώστε μια εκτίμηση για το ποσό που θα διαθέτατε για την συγκεκριμένη εφαρμογή (σε ευρώ);

α.100 – 200

β. 200 – 300

γ.300 – 400

δ. 400 – 500

ε. Πάνω από 500

14. Θα δεχόσασταν να παρέχετε πληροφορίες (π.χ. θερμοκρασία σπιτιού) από την προαναφερθείσα εφαρμογή σε τρίτους (π.χ. κέντρο security);

α. ΝΑΙ

β. ΟΧΙ

15. Τι είδους και ποιές πληροφορίες θα δεχόσασταν να παρέχετε;

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

16. Ποιος θα θέλατε να διαχειρίζεται την εφαρμογή;

- α. Αυτόνομο σύστημα
- β. Μία δημόσια εταιρία
- γ. Μία ιδιωτική εταιρία

17. Θα θέλατε μετά από τις παραπάνω ερωτήσεις να ασχοληθείτε περισσότερο με αυτό το θέμα;

- α. ΝΑΙ
- β. ΟΧΙ
- γ. ΝΑΙ αλλά θα ήθελα να μάθω συγκεκριμένα πράγματα για αυτό το θέμα.

Τι είναι τα Γεωγραφικά Συστήματα Πληροφοριών (G.IS):

Τα Γεωγραφικά Συστήματα Πληροφοριών είναι πληροφοριακά συστήματα (Information Systems) που παρέχουν την δυνατότητα συλλογής, διαχείρισης, αποθήκευσης, επεξεργασίας, ανάλυσης και οπτικοποίησης, σε ψηφιακό περιβάλλον, των δεδομένων που σχετίζονται με ένα συγκεκριμένο χώρο.

Παράρτημα Β

Κώδικας

Index.php

```
<?php
if (!isset($_REQUEST[menu_id])){$_REQUEST[menu_id]=1;}
if (!isset($_REQUEST[lang_id])) {$_REQUEST[lang_id]=1;}

?>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<link href="defaul.css" rel="stylesheet" type="text/css" />

<title>Application for Surveillance and Detection of Damage to Houses</title>
</head>

<body bgcolor="#01061c" class="main_text">

<table width="960" border="0" align="center">
<tr>
<td colspan="3"><a href="index.php"></a></td>
</tr>
<tr>
<td width="682" align="left" bgcolor="#0b255b">
<a href="index.php?menu_id=1&lang_id=<?php echo $_REQUEST[lang_id]; ?>"
class="menu_link"> welcome </a><span class="sep">|</span>
<a href="index.php?menu_id=2&lang_id=<?php echo $_REQUEST[lang_id]; ?>"
class="menu_link">products</a><span class="sep">|</span>
```

```

<a href="index.php?menu_id=3&lang_id=<?php echo "$_REQUEST[lang_id]"; ?>"
class="menu_link">safe areas</a><span class="sep">|</span>

<a href="index.php?menu_id=4&lang_id=<?php echo "$_REQUEST[lang_id]"; ?>"
class="menu_link">contact us</a><span class="sep">|</span>

<a href="index.php?menu_id=5&lang_id=<?php echo "$_REQUEST[lang_id]"; ?>"
class="menu_link">about us</a>

</td>

<td width="269" align="right" ><a href="index.php?menu_id=<?php echo
"$_REQUEST[menu_id]"; ?>&lang_id=1" class="menu1">english</a> |<a
href="index.php?menu_id=<?php echo "$_REQUEST[menu_id]"; ?>&lang_id=2"
class="menu1"> ελληνικά </a></td>

<td width="1" align="left">&nbsp;</td>

</tr>

<tr>

<td colspan="4" class="main_text">

<?php

if($_REQUEST[menu_id]==1){include('_index.php');}

if($_REQUEST[menu_id]==2){include('_products.php');}

if($_REQUEST[menu_id]==3){include('login.php');}

if($_REQUEST[menu_id]==4){include('_contact.php');}

if($_REQUEST[menu_id]==5){include('_about.php');}

if($_REQUEST[menu_id]==31){include('_areas.php');}

if($_REQUEST[menu_id]==32){include('_insert.php');}

if($_REQUEST[menu_id]==34){include('lefkada.php');}

if($_REQUEST[menu_id]==35){include('lefkada_user.php');}

if($_REQUEST[menu_id]==41){include('email.php');}

```

```
    ?>

    </td>

</tr>

<tr>

    <td colspan="2" bgcolor="#0b255b" class="menu1" align="right" border="0"> copyrigh
    2011 | asddh </td>

</tr>

</table>

</body>

</html>
```

dbconnect.php

```
<?php

$dbhost="localhost";

$username="root";

$password="";

$database="itgooglemaps";

?>
```

defaul.css

```
@charset "utf-8";

.main_text

{

    font-size:18px;

    color:#FFF;

    text-decoration: none;

    font-style: normal;

    font-family: Arial, Helvetica, sans-serif;

    text-transform: none;

    text-decoration:none;

}

.map

{font: Tahoma, Geneva, sans-serif;

font-size: 18px;

color:#000;

}

.menu1

{

    font-family: "Times New Roman", Times, serif;

    font-size: 14px;

    color: #f8f8f8;

    padding-right: 4px;

    text-decoration: none;
```

```
}  
  
.menu_link  
{  
    font-family: Haettenschweiler;  
    font-size: 36px;  
    color: #f8f8f8;  
    text-decoration:none;  
padding-right: 4px;  
    padding-left: 4px;  
    padding-top: 2px;  
    padding-bottom: 2px;  
}  
  
.menu_link_active  
{  
    font-family: Haettenschweiler;  
    font-size: 36px;  
    color: #f8f8f8;  
padding-right: 4px;  
padding-left: 4px;  
padding-top: 2px;  
padding-bottom: 2px;  
}
```

```
.sep  
  
{  
  
    font-family: Haettenschweiler;  
  
    font-size: 36px;  
  
    color: #f8f8f8;  
  
    padding-right: 5px;  
  
    padding-left: 5px;  
  
}  
  
.link {  
  
    font-size:18px;  
  
    color: #FFF;  
  
    text-decoration:none;  
  
    font-style: normal;  
  
    font-family: Arial, Helvetica, sans-serif;  
  
    text-transform: none;  
  
}  
  
.main_text1  
  
{  
  
    font-size:18px;  
  
    color:#FFF;  
  
    text-decoration: underline;  
  
    font-style: normal;  
  
    font-family: Arial, Helvetica, sans-serif;  
  
    text-transform: none;  
  
}  
  
.button
```

```
{color:#FFF;  
background:#006;  
border:#006;  
font-size:16px;  
font-family:Arial, Helvetica, sans-serif;
```

index.php

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
  
<link href="defaul.css" rel="stylesheet" type="text/css" />  
  
<?php if ($_REQUEST[lang_id]==1)  
    echo "  
Welcome to our website and ASDDH! <br><br>  
  
The ASDDH is an application that was created during our study of the graduation work at the  
TEI of Ionian Islands in the Department of Applied Informatics in Management and Economics  
in Lefkada. The appetite and the desire to deal with the computer science gave us the idea to  
create a service that will have direct application and service in the world.<br>  
  
We dealt with the tools offered by Google and we have developed an application for  
monitoring and preventing damage to a house.  
  
The application informs the map and signs authorized to oversee the placement of sensors in  
the house will pass information (eg temperature and humidity). This vortex the user can visit  
this page by entering the code corresponds will be able to provide information about  
becoming a home. At the same time, if the application based on the information is taken from  
the sensors to the risk, then tell a responsible way e-mail (eg firedepartment). <br>  
  
For more information please contact us through our contact form ";  
  
else echo "  
Καλώς ήρθατε στην ιστοσελίδα μας και στο ASDDH!  
  
<br><br>  
  
Το ASDDH είναι μια εφαρμογή που δημιουργήθηκε κατά τη διάρκεια της μελέτης της  
πτυχιακής μας εργασίας στο Τ. Ε. Ι. Ιονίων Νήσων στο τμήμα Εφαρμογών της Πληροφορικής  
στη Διοίκηση και στην Οικονομία στη Λευκάδα. Η όρεξη και το μεράκι να ασχοληθούμε με την  
επιστήμη της πληροφορικής μας έδωσε την ιδέα για τη δημιουργία μια υπηρεσίας που θα
```

έχει άμεση εφαρμογή και εξυπηρέτηση στον κόσμο.

Ασχοληθήκαμε με τα εργαλεία που προσφέρει η Google και έχουμε αναπτύξει μια εφαρμογή για την παρακολούθηση και την πρόληψη των ζημιών σε μια οικία.

Η εφαρμογή ενημερώνει τον χάρτη και τα σημεία που έχει εξουσιοδοτηθεί να επιβλέπει με τις τοποθετήσεις αισθητήρων στην κατοικία που θα περνούν πληροφορίες (π.χ. θερμοκρασία και υγρασία) . Αυτό δίνει την δυνατότητα στο χρήστη να επισκεφθεί αυτή τη σελίδα και με το να εισάγετε τον κωδικό που του αντιστοιχεί θα έχει την δυνατότητα να ενημερώνετε για το τη γίνετε στο σπίτι του. Την ίδια στιγμή, αν κριθεί από την εφαρμογή με βάση τις πληροφορίες έχει πάρει από τους αισθητήρες ότι υπάρχει κίνδυνος τότε ενημερώνετε κάποιος αρμοδίων μέσω e-mail (π.χ. πυροσβεστική).

Για περισσότερες πληροφορίες μην διστάσετε να επικοινωνήσετε μαζί μας μέσω της φόρμας επικοινωνίας,

";

?>

_products.php

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<link href="defaul.css" rel="stylesheet" type="text/css" />
```

```
<table width="953" border="0">
```

```
<tr>
```

```
<td width="762" align="left" valign="top">
```

```
<?php if ($_REQUEST[lang_id]==1)
```

```
    echo "The application is still at an experimental stage. In theory and in relation to
    market research and buying mood will be able to monitor and collect information related to
    home environment (eg temperature, humidity, gas). With the ability to upgrade later so give
    other information to be requested. <br><br>";
```

```
else
```

```
    echo "Η εφαρμογή είναι ακόμη σε πειραματικό στάδιο. Θεωρητικά και σε σχέση με την έρευνα
    αγοράς και την αγοραστική διάθεση θα έχει την δυνατότητα τις παρακολούθησης και τη
    συλλογής πληροφοριών που σχετίζονται με το περιβάλλον του σπιτιού τους (π.χ.
    θερμοκρασία, υγρασία, φυσικό αέριο). Με την δυνατότητα μετέπειτα αναβάθμισης να παρέχει
    άλλα στοιχεία που θα ζητηθούν.<br><br>";
```



```
?></td>

  <td width="175"> </td>

</tr>

</table>
```

login.php

```
<?php if (($_POST[username]=='admin') && ($_POST[password]=='adm1n')) $logged=1;
if (($_POST[username]=='user') && ($_POST[password]=='us3r'))
$logged=2;
else if (($_POST[username]=='user') && ($_POST[password]=='us2r'))
$logged=2;
else if (($_POST[username]=='user') && ($_POST[password]=='us1r'))
$logged=2;

?>

<form name="login" method="post">

<input name="logged" type="hidden" value="1"/>

    Username :<br>

    <input name="username" type="text" size="12" maxlength="8" value="<?php
echo"$_POST[username]"; ?>"/><br>

    Password : <br>

    <input name="password" type="password" size="12" maxlength="8" value="<?php
echo"$_POST[password]"; ?>"/><br>

<input type="submit" value="Login" class="input1"/>
```

```
</form>

<?php

if ($logged==0)

    echo "enter User Name and Password";

if ($logged==1)
{
    if (($_POST[username]=='admin') && ($_POST[password]=='adm1n'))
        echo "<meta http-equiv='refresh' content='0;URL=index.php?menu_id=31'>";

    else
        echo "<font color='#CC0033'>Invalid Username and/or Password!</font><br>Try
again!";
}

if ($logged==2)
{
if (($_POST[username]=='user') && ($_POST[password]=='us3r'))

    echo "<meta http-equiv='refresh'
content='0;URL=index.php?menu_id=35&p=$_REQUEST[password]'>";

    else if (($_POST[username]=='user') && ($_POST[password]=='us2r'))

        echo "<meta http-equiv='refresh'
content='0;URL=index.php?menu_id=35&p=$_REQUEST[password]'>";

    else if (($_POST[username]=='user') && ($_POST[password]=='us1r'))

        echo "<meta http-equiv='refresh'
content='0;URL=index.php?menu_id=35&p=$_REQUEST[password]'>";
```

```
else
    echo "<font color='#CC0033'>Invalid Username and/or Password!</font><br>Try
again!";
}
?>
```

_areas.php

```
<link href="defaul.css" rel="stylesheet" type="text/css" />
<table width="400" border="0" align="center">
  <tr>
    <td class="main_text1">Safe Areas </td>
    <td class="main_text1">Insert in database</td>
  </tr>
  <tr>
    <td class=" "><a href="index.php?menu_id=34" class="link">lefkada</a></td>
    <td class=" main_text"><a href="index.php?menu_id=32" class="link">Insert
Location</a></td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
```

```
</tr>
</table>
```

_insert.php

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-7" />
<link href="default.css" rel="stylesheet" type="text/css" />
<link href="defaul.css" rel="stylesheet" type="text/css" />
<form name="insert" method="post" action="dbase.php">
<br />
<br>
Insert location:<br /><br />
Name:<br />
<input type="text" name="name"><br />
Address:<br />
<input type="text" name="address">
<br />
Coordinates Lat:<br />
<input type="text" name="lat">
<br />
Coordinates Lng:<br />
<input type="text" name="lng">
<br />
Type (c1,c2): <br />
<input type="text" name="type">
<br />
<input name="action" type="hidden" value="3"/>
</label>
```

```
<br /><br /><label>

  <input class="button" type="submit" value="insert" />

</label>

</form>
```

dbase.php

```
<?php require("dbconnect.php");

      $q2=mysql_query("INSERT INTO markers (name, address, lat, lng, type, ) VALUES
('$ _POST[name]','$ _POST[address]','$ _POST[lat]','$ _POST[lng]','$ _POST[type]' )");

?>

<html>
<head>
<title>dbase</title>
<meta http-equiv="refresh" content="10; URL=index.php?menu_id=31" />
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-7" />
</head>
<body>

</body>
</html>
```

lefkada.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-7"> <meta http-
equiv="refresh" content="60"> <!-- Here we set the refresh interval of this page to 60 seconds
(1 minutes) -->
<link href="text.css" rel="stylesheet" type="text/css">
<title>Google Maps</title>

<script
src="http://maps.google.com/maps?file=api&v=2&sensor=false;key=ABQIAAAAxQ
_2SfoEMFa714G7VdOyIRTIEJiiQkmuaLMEhh89rn5lh2bPKhSi7H09sAhet7QIDd09gbAKs6xql
g" type="text/javascript"></script>

<script type="text/javascript">

  <![CDATA[

var iconBlue = new GIcon();
iconBlue.image = 'http://labs.google.com/ridefinder/images/mm_20_blue.png';
iconBlue.shadow = 'http://labs.google.com/ridefinder/images/mm_20_shadow.png';
iconBlue.iconSize = new GSize(12, 20);
iconBlue.shadowSize = new GSize(22, 20);
iconBlue.iconAnchor = new GPoint(6, 20);
iconBlue.infoWindowAnchor = new GPoint(5, 1);

var iconRed = new GIcon();
iconRed.image = 'http://labs.google.com/ridefinder/images/mm_20_red.png';
iconRed.shadow = 'http://labs.google.com/ridefinder/images/mm_20_shadow.png';
iconRed.iconSize = new GSize(12, 20);
iconRed.shadowSize = new GSize(22, 20);
iconRed.iconAnchor = new GPoint(6, 20);
iconRed.infoWindowAnchor = new GPoint(5, 1);
```

```
var customIcons = [];  
  
customIcons["c1"] = iconBlue;  
  
customIcons["c2"] = iconRed;  
  
function load() {  
  if (GBrowserIsCompatible()) {  
    var map = new GMap2(document.getElementById("map"));  
    map.addControl(new GSmallMapControl());  
    map.addControl(new GMapTypeControl());  
    map.setCenter(new GLatLng(38.710000, 20.670000), 11);  
  
    // Change this depending on the name of your PHP file  
    GDownloadUrl("phpsqlajax_genxml.php", function(data) {  
      var xml = GXml.parse(data);  
      var markers = xml.documentElement.getElementsByTagName("marker");  
      for (var i = 0; i < markers.length; i++) {  
        var name = markers[i].getAttribute("name");  
        var address = markers[i].getAttribute("address");  
        var temp = markers[i].getAttribute("temp");  
        var humidity = markers[i].getAttribute("humidity");  
        var type = markers[i].getAttribute("type");  
        var point = new GLatLng(parseFloat(markers[i].getAttribute("lat")),  
                                parseFloat(markers[i].getAttribute("lng")));  
        var marker = createMarker(point, name, address, type, temp, humidity);  
        map.addOverlay(marker);  
  
      }  
    });  
  }  
}
```

```
}

function createMarker(point, name, address, type, temp, humidity) {

    var marker = new GMarker(point, customIcons[type]);

    var html = "<b>" + name + "</b> <br/>" + address + "</b> <br/>temp: " + temp + "oC" +
"</b> <br/>humidity: " + humidity ;

    GEvent.addListener(marker, 'click', function() {

        marker.openInfoWindowHtml(html);

    });

    return marker;

}

//]]>
</script>

<link href="defaul.css" rel="stylesheet" type="text/css" />
</head>

<body onload="load()" onunload="GUnload()">

    <div class="map" id="map" style="width: 700px; height: 700px"></div>

    <a href="index.php?menu_id=31" class="main_text">back</a>

</body>
</html>
```

phpsqlajax_genxml.php

```
<?php
require("dbconnect.php");
```



```
function parseToXML($htmlStr)
{
$xmlStr=str_replace('<','&lt;',$htmlStr);
$xmlStr=str_replace('>','&gt;',$xmlStr);
$xmlStr=str_replace('"','&quot;',$xmlStr);
$xmlStr=str_replace("'",'&apos;',$xmlStr);
$xmlStr=str_replace('&','&amp;',$xmlStr);
return $xmlStr;
}

// Opens a connection to a mySQL server
$connection=mysql_connect ('localhost', $username, $password);
if (!$connection) {
    die('Not connected : ' . mysql_error());
}

// Set the active mySQL database
$db_selected = mysql_select_db($database, $connection);
if (!$db_selected) {
    die ('Can\'t use db : ' . mysql_error());
}

// Select all the rows in the markers table
$query = "SELECT * FROM markers";
$result = mysql_query($query);
if (!$result) {
    die('Invalid query: ' . mysql_error());
}
```

```
header("Content-type: text/xml");

// Start XML file, echo parent node
echo '<markers>';

// Iterate through the rows, printing XML nodes for each
while ($row = @mysql_fetch_assoc($result)){
    // ADD TO XML DOCUMENT NODE
    echo '<marker ' ;
    echo 'name="' . parseToXML($row['name']) . "' '";
    echo 'address="' . parseToXML($row['address']) . "' '";
    echo 'lat="' . $row['lat'] . "' '";
    echo 'lng="' . $row['lng'] . "' '";
    echo 'type="' . $row['type'] . "' '";
    echo 'temp="' . $row['temp'] . "' '";
    echo 'humidity="' . $row['humidity'] . "' '";
    echo '/>';
}

// End XML file
echo '</markers>';

?>
```

lefkada_user.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-7"> <meta http-equiv="refresh" content="60"> <!-- Here we set the refresh interval of this page to 60 seconds (1 minutes) -->
```

```
<title>Google Maps</title>
```

```
<script  
src="http://maps.google.com/maps?file=api&v=2&sensor=false;key=ABQIAAAAxQ  
_2SfoEMFa7l4G7VdOyIRTIEJiiQkmuaLMEhh89rn5lh2bPKhSi7H09sAhet7QIDd09gbAKs6xql  
g" type="text/javascript"></script>
```

```
<script type="text/javascript">
```

```
  <![CDATA[
```

```
    var iconBlue = new GIcon();
```

```
    iconBlue.image = 'http://labs.google.com/ridefinder/images/mm_20_blue.png';
```

```
    iconBlue.shadow = 'http://labs.google.com/ridefinder/images/mm_20_shadow.png';
```

```
    iconBlue.iconSize = new GSize(12, 20);
```

```
    iconBlue.shadowSize = new GSize(22, 20);
```

```
    iconBlue.iconAnchor = new GPoint(6, 20);
```

```
    iconBlue.infoWindowAnchor = new GPoint(5, 1);
```

```
    var iconRed = new GIcon();
```

```
    iconRed.image = 'http://labs.google.com/ridefinder/images/mm_20_red.png';
```

```
    iconRed.shadow = 'http://labs.google.com/ridefinder/images/mm_20_shadow.png';
```

```
    iconRed.iconSize = new GSize(12, 20);
```

```
    iconRed.shadowSize = new GSize(22, 20);
```

```
    iconRed.iconAnchor = new GPoint(6, 20);
```

```
    iconRed.infoWindowAnchor = new GPoint(5, 1);
```

```
    var customIcons = [];
```

```
    customIcons["c1"] = iconBlue;
```

```
    customIcons["c2"] = iconRed;
```

```
function load() {  
    if (GBrowserIsCompatible()) {  
        var map = new GMap2(document.getElementById("map"));  
        map.addControl(new GSmallMapControl());  
        map.addControl(new GMapTypeControl());  
        map.setCenter(new GLatLng(38.710000, 20.670000), 11);  
  
        // Change this depending on the name of your PHP file  
        GDownloadUrl("phpsqlajax_genxml_user.php?p=<?php echo "$_REQUEST[p]"; ?>",  
function(data) {  
    var xml = GXml.parse(data);  
    var markers = xml.documentElement.getElementsByTagName("marker");  
    for (var i = 0; i < markers.length; i++) {  
        var name = markers[i].getAttribute("name");  
        var address = markers[i].getAttribute("address");  
        var temp = markers[i].getAttribute("temp");  
        var humidity = markers[i].getAttribute("humidity");  
        var type = markers[i].getAttribute("type");  
        var point = new GLatLng(parseFloat(markers[i].getAttribute("lat")),  
        parseFloat(markers[i].getAttribute("lng")));  
        var marker = createMarker(point, name, address, type, temp, humidity);  
        map.addOverlay(marker);  
  
    }  
});  
}  
}
```

```
function createMarker(point, name, address, type, temp, humidity) {  
    var marker = new GMarker(point, customIcons[type]);  
    var html = "<b>" + name + "</b> <br/>" + address + "<b> <br/>temp: " + temp + "oC" +  
    "</b> <br/>humidity: " + humidity ;  
    GEvent.addListener(marker, 'click', function() {  
        marker.openInfoWindowHtml(html);  
    });  
    return marker;  
}  
//]]>  
</script>  
  
<link href="defaul.css" rel="stylesheet" type="text/css" />  
<link href="*.css" rel="stylesheet" type="text/css" />  
<link href="defaul.css" rel="stylesheet" type="text/css" />  
</head>  
  
<body onload="load()" onunload="GUnload()">  
    <div class=" map" id="map" style="width: 700px; height: 700px"></div>  
</body>  
</html>
```

phpsqlajax_genxml_user.php

```
<?php  
require("dbconnect.php");  
  
function parseToXML($htmlStr)
```

```
{
$xmlStr=str_replace('<','&lt;',$htmlStr);
$xmlStr=str_replace('>','&gt;',$xmlStr);
$xmlStr=str_replace('"','&quot;',$xmlStr);
$xmlStr=str_replace("'",'&apos;',$xmlStr);
$xmlStr=str_replace('&','&amp;',$xmlStr);
return $xmlStr;
}

// Opens a connection to a mySQL server
$connection=mysql_connect ('localhost', $username, $password);
if (!$connection) {
    die('Not connected : ' . mysql_error());
}

// Set the active mySQL database
$db_selected = mysql_select_db($database, $connection);
if (!$db_selected) {
    die ('Can\'t use db : ' . mysql_error());
}

// Select all the rows in the markers table
$query = "SELECT * FROM markers WHERE pass='$_REQUEST[p]";
$result = mysql_query($query);
if (!$result) {
    die('Invalid query: ' . mysql_error());
}

header("Content-type: text/xml");
```

```
// Start XML file, echo parent node
echo '<markers>';

// Iterate through the rows, printing XML nodes for each
while ($row = @mysql_fetch_assoc($result)){
    // ADD TO XML DOCUMENT NODE
    echo '<marker ' ;
    echo 'name="' . parseToXML($row['name']) . "' ' ;
    echo 'address="' . parseToXML($row['address']) . "' ' ;
    echo 'lat="' . $row['lat'] . "' ' ;
    echo 'lng="' . $row['lng'] . "' ' ;
    echo 'type="' . $row['type'] . "' ' ;
    echo 'temp="' . $row['temp'] . "' ' ;
    echo 'humidity="' . $row['humidity'] . "' ' ;
    echo '/>';
}

// End XML file
echo '</markers>';

?>
```

_contact.php

```
<link href="css/default.css" rel="stylesheet" type="text/css">

<script src="SpryValidationTextField.js" type="text/javascript"></script>
<link href="SpryValidationTextField.css" rel="stylesheet" type="text/css" />
<p><br />
```

```
<em>email: panayiotou_p@yahoo.com</em></p>
<form name="contact" method="post" action="index.php?menu_id=41">
  Name- Surname:
  <br />
  <input name="name" type="text" size="40" maxlength="60" />
  <br /> <br />
  Email :
  <br />
  <span id="sprytextfield1">
  <label>
  </label>
  <span class="textfieldRequiredMsg">A value is required.</span></span>
  <input type="text" name="email"/><br>
  <br />Comments:
  <br>
  <textarea name="message" id="textarea" cols="45" rows="5"></textarea>
  <br />
  <br />
  <input type="reset" name="reset" value="reset" class="button" /> <input type="submit"
name="send" value="send" class="button" />
</form>
<script type="text/javascript">
<!--
var sprytextfield1 = new Spry.Widget.ValidationTextField("sprytextfield1");
//-->
</script>
```


email.php

```
<?php

require_once "postman.php";

if ($_REQUEST[message]!=NULL)
{

postman($_REQUEST[email],
        'panayiotou_p@yahoo.com',
        'epikoinonia apo website',
        $_REQUEST[message],
        $_REQUEST[message]);

}

?>

<link href="defaul.css" rel="stylesheet" type="text/css" />

<br /><p align="center" class="main_text">email has sent <br>
<a href="index.php" class="main_text"><em>go back to home page</em></a>
```

postman.php

```
<?php

function postman ($sender,$recipient,$sub,$tmes,$hmes)
{
```

```
$email = $recipient;

$mime_boundary = "----postman----".md5(time());

$to = $email;

$subject = $sub;

$headers = "From: $sender\n";
$headers .= "MIME-Version: 1.0\n";
$headers .= "Content-Type: multipart/alternative; boundary=\"$mime_boundary\n";

$message = "--$mime_boundary\n";
$message .= "Content-Type: text/plain; charset=iso-8859-7\n";
$message .= "Content-Transfer-Encoding: 8bit\n\n";
$message .= "$tmes \n";
$message .= "--$mime_boundary\n";
$message .= "Content-Type: text/html; charset=iso-8859-7\n";
$message .= "Content-Transfer-Encoding: 8bit\n\n";

$message .= "name : $_REQUEST[name] <br>\n";
$message .= "email : $_REQUEST[email] <br>\n";
$message .= "message : $_REQUEST[message] <br>\n";

$message .= "--$mime_boundary--\n\n";

mail( $to, $subject, $message, $headers );

return;

}
```

```
function bcontainer ($width,$height)
{
}
?>
```

_about.php

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link href="defaul.css" rel="stylesheet" type="text/css" />

<?php if ($_REQUEST[lang_id]==1)
echo "
We are 2 students at the TEI of Ionian Islands of Lefkada in computer applications in
management and economics.<br><br>
Panayiotis Panayiotou from Nicosia, Cyprus.<br><br>
e-mail: panayiotou_p@yahoo.com<br><br>
Nikos Papadopoulos Exaplatanios from the prefecture of Pella.<br>
e-mail: niklakis@yahoo.gr<br><br> ";

else
echo "
Είμαστε 2 φοιτητές του TEI Ιονίων Νήσων του τμήματος Εφαρμογών της Πληροφορικής στην
διοίκηση και στην οικονομία στη Λευκάδα.<br><br>
Παναγιώτης Παναγιώτου από τη Λευκωσία, Κύπρος.<br>
e-mail: panayiotou_p@yahoo.com<br><br>
Νίκος Παπαδόπουλος Εξαπλατάνου από το Νομό Πέλλας.<br>
e-mail: niklakis@yahoo.gr<br><br>";
?>
```

warning.php

```
<?php
require_once "classes.php";
require_once "dbconnect.php";

$q_Read_Temperature = mysql_query ("select * from markers");
while ($data=mysql_fetch_array($q_Read_Temperature))
{
    $id=$data[id];
    $temp=$data[temp];
    $m_status=$data[m_status];
    $name=$data[name];
    $address=$data[address];

    if ($temp<40) $message=0; else

    if (($temp>=40) && ($temp<50)) $message=1; else

    if ($temp>=50) $message=2;

    if ($m_status==1)
    {
        if ($message==0) postman('panayiotou_p@yahoo.com',
'panayiotou.p86@gmail.com', 'Everything is ok', 'Temperature in station $id name: $name ,
address: $address is back to normal. station name: $name , station address: $address ',
'Temperature in station $id is back to normal. station name: $name , station address:
$address ');

postman('panayiotou_p@yahoo.com', 'panayiotou_p1@hotmail.com', 'Everything is ok',
'Temperature in station $id name: $name , address: $address is back to normal. station
```

```
name: $name , station address: $address ', 'Temperature in station $id is back to normal.
station name: $name , station address: $address ');

        if ($message==1) postman('panayiotou_p@yahoo.com',
'panayiotou.p86@gmail.com', 'Warning', 'Temperature in station $id is above 40 degrees!
station name: $name , station address: $address ', 'Temperature in station $id is above 40
degrees! station name: $name , station address: $address ');

postman('panayiotou_p@yahoo.com', 'panayiotou_p1@hotmail.com', 'Warning', 'Temperature
in station $id is above 40 degrees! station name: $name , station address: $address ',
'Temperature in station $id is above 40 degrees! station name: $name , station address:
$address ');

        if ($message==2) postman('panayiotou_p@yahoo.com',
'panayiotou.p86@gmail.com', 'Danger', 'Temperature in station $id is above 50 degrees.
station name: $name , station address: $address ', 'Temperature in station $id is above 50
degrees. station name: $name , station address: $address ');

postman('panayiotou_p@yahoo.com', 'panayiotou_p1@hotmail.com', 'Danger', 'Temperature
in station $id is above 50 degrees. station name: $name , station address: $address ',
'Temperature in station $id is above 50 degrees. station name: $name , station address:
$address '      );

        $q_reset_m_status = mysql_query ("update markers set m_status='0' where
s_id=$s_id");
    }

}

?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta http-equiv="Refresh" content="60;url=warning.php"> <!-- Here we set the refresh
```

interval of this page to 60 seconds (1 minutes) -->

```
<title>Untitled Document</title>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

classes.php

```
<?php
```

```
function postman ($sender,$recipient,$sub,$tmes,$hmes)
```

```
{
```

```
    $email = $recipient;
```

```
    $mime_boundary = "----postman----".md5(time());
```

```
    $to = $email;
```

```
    $subject = $sub;
```

```
    $headers = "From: $sender\n";
```

```
    $headers .= "MIME-Version: 1.0\n";
```

```
    $headers .= "Content-Type: multipart/alternative; boundary=\"\$mime_boundary\"\n";
```

```
    $message = "--$mime_boundary\n";
```

```
    $message .= "Content-Type: text/plain; charset=iso-8859-7\n";
```

```
    $message .= "Content-Transfer-Encoding: 8bit\n\n";
```

```
$message .= "$tmes \n";  
  
$message .= "--$mime_boundary\n";  
  
$message .= "Content-Type: text/html; charset=iso-8859-7\n";  
  
$message .= "Content-Transfer-Encoding: 8bit\n\n";  
  
$message .= "$hmes \n";  
  
$message .= "--$mime_boundary--\n\n";  
  
mail( $to, $subject, $message, $headers );  
  
return;  
  
}  
  
?>
```