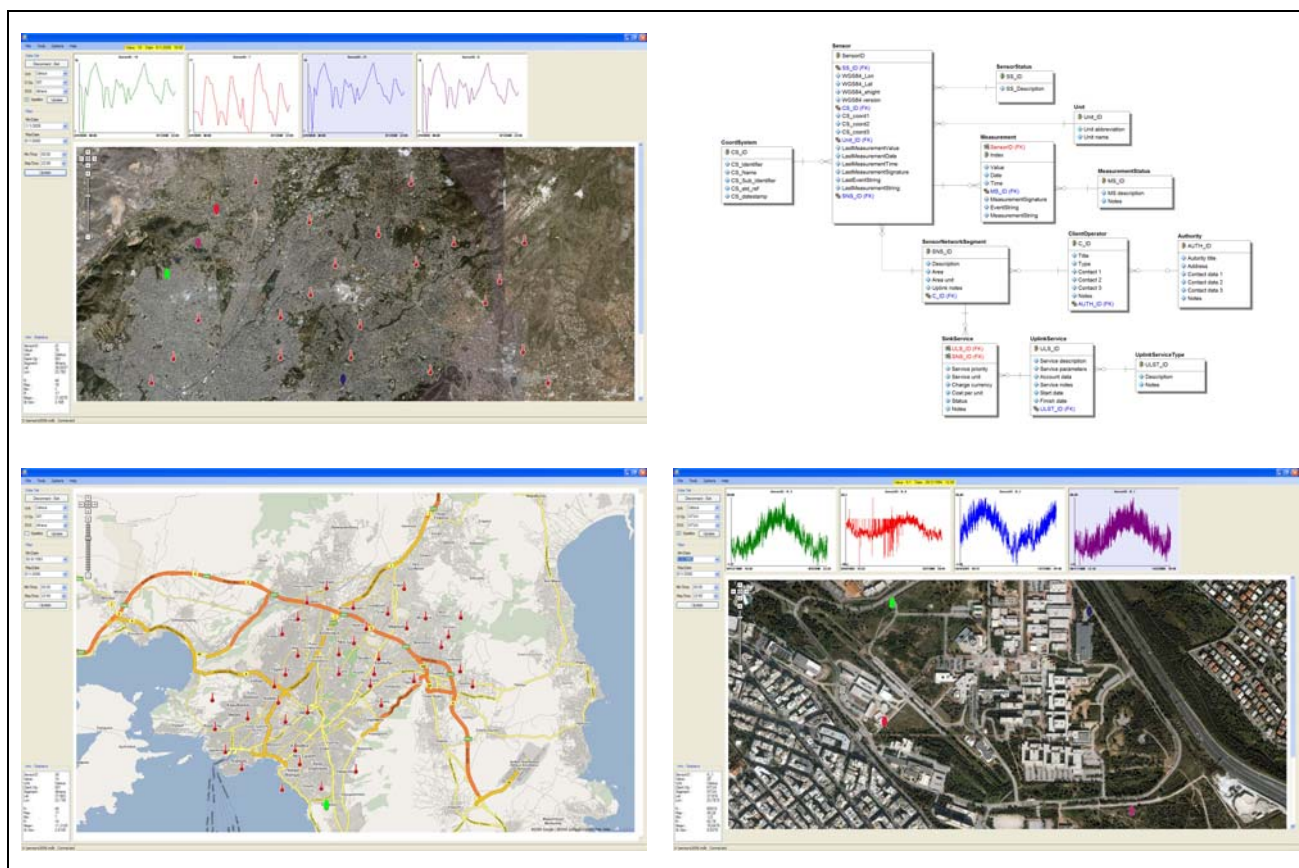




Ανάπτυξη εφαρμογής ανάκτησης και απεικόνισης
πληροφοριών με χωρική διάσταση από προτυποποιημένη
Βάση Δεδομένων σε ελεύθερα υπόβαθρα
με χρήση Google Maps API



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σοφός Ιωάννης

Επιβλέπων Βεσκούκης Βασίλειος, Επ.Καθ.

Αθήνα Μάρτιος 2008

ΠΡΟΛΟΓΟΣ

Η παρούσα εργασία υλοποιείται στα πλαίσια της ολοκλήρωσης των προπτυχιακών μου σπουδών στη Σχολή Αγρονόμων Τοπογράφων Μηχανικών (ΣΑΤΜ) του Εθνικού Μετσόβιου Πολυτεχνείου (ΕΜΠ). Σπουδών που αποτελούν το εφαλτήριο για τη μετέπειτα ενασχόληση μου με την επιστήμη της Τοπογραφίας, όπως αυτή δομείται από τα επιμέρους αντικείμενά της. Έχοντας ευρύτατο πεδίο, δίνει στους συναδέλφους μου μια πληθώρα επιλογών μεταξύ κλασσικής - δορυφορικής τοπογραφίας, οδοποιίας, τηλεπισκόπησης, φωτογραμμετρίας, υδραυλικών, ανάλυσης και οργάνωσης χώρου, δομικών έργων. Ο σύγχρονος Αγρονόμος Τοπογράφος Μηχανικός υπηρετεί μια πολυσυλλεκτική επιστήμη και οφείλει να ανταποκρίνεται στις συνεχώς αυξανόμενες απαιτήσεις, ως ένας “*Homo Universalis*” του χώρου των Μηχανικών.

Παρά τον πολυδιάστατο χαρακτήρα του όμως, ο σύγχρονος Τοπογράφος έχει για θεμέλια δύο βασικούς πυλώνες που ανέκαθεν όριζαν την ταυτότητά του. Τη μέτρηση στο πεδίο του χώρου και την απόδοση αυτής και των παραγώγων της. Έτσι λοιπόν η παρούσα εργασία πραγματεύεται τις μετρήσεις και την απεικόνισή τους με χρήση νέων εργαλείων που διαθέτει η επιστήμη της πληροφορικής. Στο διαρκή αγώνα δρόμου παρακολούθησης των τεχνολογικών εξελίξεων, οφείλουμε να δίνουμε το παρόν καθότι κατά βάση ο ρόλος μας είναι να διαχειριζόμαστε πληροφορίες. Η πληροφορική του χώρου εμβολιάζει τον Τοπογράφο και αυτός με τη σειρά του οφείλει να τη αξιοποιεί και να συμβάλλει στην εξέλιξή της.

Στη πορεία των σπουδών μου μέχρι αυτού του σημείου υπήρξαν πολλοί άνθρωποι που συντέλεσαν στη διαμόρφωση της ταυτότητάς μου ως επιστήμονα και άνθρωπο. Θέλω να ευχαριστήσω αυτούς που προσέφεραν τη γνώση, τις κατευθύνσεις και τις λύσεις, αλλά πολύ περισσότερο αυτούς που έθεσαν τα ερωτήματα, τα ερωτηματικά, τους προβληματισμούς, τα προβλήματα και τα αδιέξοδα. Τον κ. Β. Βεσκούκη, επιβλέποντα καθηγητή, όλο το Διδακτικό σώμα της σχολής μας καθώς και όλους αυτούς που με στήριξαν τα τελευταία χρόνια από το οικογενειακό και φιλικό μου περιβάλλον.



Περιεχόμενα

Πρόλογος	1
Περίληψη	4
Abstract	5
1. Αντικείμενο της εργασίας	6
1.1 Το πρόβλημα της συλλογής και πιστοποίησης μετρήσεων	7
1.2 Το πρόβλημα της αποθήκευσης μετρήσεων	7
1.3 Το πρόβλημα της ανάκτησης και διάθεσης μετρήσεων	8
1.4 Το πρόβλημα της απεικόνισης μετρήσεων	8
1.5 Το περιβάλλον Google Earth/ Google maps (σύντομη αναφορά)	8
1.6 Οριοθέτηση της εργασίας	9
1.6.1 Θέματα με τα οποία θα ασχοληθούμε	10
1.6.2 Θέματα που δεν θα μας απασχολήσουν	11
1.6.3 Η ζητούμενη εμπειρία από τον χρήστη	11
2. Google Maps	12
2.1 Χαρτογραφικά δεδομένα	13
2.2 Στοιχεία απεικόνισης	14
2.3 Google Maps API	15
2.4 Δικαιώματα χρήσης	17
3. Λειτουργική περιγραφή του λογισμικού	18
3.1 Απαιτήσεις σε εξοπλισμό, λογισμικό, δίκτυο κλπ	19
3.2 Ροή εργασιών (workflow) συλλογής δεδομένων, αδειοδότησης, εγκατάστασης λογισμικού, κλπ	20
3.3 Λειτουργίες της εφαρμογής από την πλευρά του χρήστη	21
3.3.1 Σύνδεση με ΒΔ	21
3.3.2 Μορφές απεικόνισης	21
3.3.3 Φίλτρα	22
3.3.4 Γραφήματα	22
3.3.5 Στατιστικά στοιχεία	22

4.	Προγραμματισμός	23
4.1	Περιβάλλον προγραμματισμού	24
4.2	Visual Studio 2005	25
4.3	JavaScript	25
4.4	SQL (Structured Query Language)	26
5.	Αρχιτεκτονική του λογισμικού	27
5.1	Γενική περιγραφή	28
5.2	Δομή της Βάσης Δεδομένων	29
5.3	Database Module	32
5.4	JavaScript Module	34
5.5	Statistics Module	35
5.6	Graphs Module	36
5.7	User Input Module	37
5.8	Αρχεία εφαρμογής	38
5.8.1	Αρχείο Παραμέτρων «Param.txt»	38
5.8.2	Αρχείο διασύνδεσης «GInfo.txt»	39
5.8.3	Αρχείο HTML – JavaScript κώδικα «SJ_file.html»	39
5.8.4	Αρχεία γραφημάτων	40
6.	Τρόπος χρήσης του λογισμικού	42
6.1	Σύνδεση με τη Βάση Δεδομένων	43
6.2	Απεικόνιση αισθητήρων σε χάρτη	44
6.3	Στατιστικά	45
6.4	Γραφήματα	46
6.5	Φίλτρα	47
7.	Δυνατότητες αξιοποίησης του λογισμικού	48
7.1	Πιθανές εφαρμογές	49
	BIBΛΙΟΓΡΑΦΙΑ	51
	Παράρτημα 1. Πηγαίος κώδικας	52

ΠΕΡΙΛΗΨΗ

Αντικείμενο της διπλωματικής εργασίας είναι η απεικόνιση μετρήσεων από προτυποποιημένη Βάση Δεδομένων σε ελεύθερα χαρτογραφικά υπόβαθρα που διαθέτει η υπηρεσία Google Maps, καθώς και η δημιουργία υποστηρικτικών εργαλείων που καθιστούν λειτουργικότερη την απεικόνιση αυτή. Αντιμετωπίζονται ζητήματα όπως η διαχείριση μεγάλου όγκου δεδομένων και η οπτικοποίηση στο πεδίο του χρόνου με χρήση γραφημάτων.

Η απεικόνιση σε ελεύθερα υπόβαθρα μπορεί να δίνεται σε μορφή χάρτη ή δορυφορικής εικόνας και υλοποιείται με χρήση του Google Maps API . Πρόκειται για ένα σύνολο Κλάσεων (Classes) με Ιδιότητες (Properties), Συμβάντα (Events), Μεθόδους (Methods) δομημένα με τρόπο που να κάνουν εφικτή τη πρόσβαση στα δεδομένα και να επιτρέπουν την ανάκτηση αυτών καθώς και τον εμπλουτισμό τους πριν από την απεικόνιση.

Πηγή των μετρήσεων που θα απεικονιστούν, αποτελεί βάση δεδομένων με την οποία επιτυγχάνεται πρόσβαση μέσω Odbc drivers. Έτσι ανοίγει διάλογος επικοινωνίας μέσα από τον οποίο διατυπώνονται SQL ερωτήματα και ανακτώνται τα αποθηκευμένα δεδομένα.

Το ζήτημα του μεγάλου όγκου δεδομένων αντιμετωπίζεται με τη δημιουργία διαδικασιών φιλτραρίσματος του συνόλου των μετρήσεων, αποσκοπώντας σε ένα υποσύνολο αυτών που και πιο εύκολα διαχειρίσιμα είναι αλλά και πιο στοχευμένα προς τις μετρήσεις που μας ενδιαφέρουν.

Η Βάση Δεδομένων διατηρεί ιστορικό των μετρήσεων, συνεπώς αποτελεί ζητούμενο η απεικόνιση όχι μόνο στο χώρο, αλλά και σε συνάρτηση με το χρόνο. Για το σκοπό αυτό αναπτύσσεται διαδικασία δημιουργίας γραφημάτων και πρόσβασης στις πρωτογενείς μετρήσεις.

Η παρούσα, μπορεί να καλύψει ζητούμενες διεργασίες στα πλαίσια εφαρμογών που διαχειρίζονται μετρήσεις. Το πρόγραμμα είναι ανεξάρτητο από το είδος των μετρήσεων και έτσι μπορεί να διαχειριστεί κάθε είδους δεδομένα. Μερικές πιθανές χρήσεις είναι σε κέντρα που παρακολουθούν περιβαλλοντικά, υδρολογικά και οικονομικά μεγέθη, καθώς αποτελούν πληροφορία που είναι άμεσα συνδεδεμένη με το χώρο.

ABSTRACT

The objective of this Thesis is the development of a software application for the depiction of measurements from a Relational Database in free cartographic frameworks that allocates the service of “Google Maps”, as well as the creation of supporting tools that provides the required functionality to this depiction. Issues regarding the management of big volume of data and their visualisation in the field of time are also coped with.

The depiction in open cartographic frameworks can be done in form of map or satellite image and it is implemented by using the Google Maps API, which is a group of Classes of Objects that are characterized of Properties, Events and Methods structured in a way that would make feasible accessing data and allow the recuperation of these as well as their enrichment before the depiction.

The Data Sources of the measurements that will be depicted are accessed through Odbc drivers. This way, the software opens a channel of communication through formulated SQL questions and retrieves the stored data.

The question of big volume of data is faced by using filters to reduce the volume of data, in such way that results in a subset easier to manipulate than the whole set of data that would be available for the object of interest.

The Database contains a time series of measurements, hence it facilitates their depiction not only in the field of space, but in the field of time. In order to present these measurements, a routine has been created that produces functional Graphs .

This thesis aims to help activities in various sections that manage measurements. The program is not depended on the type of measurements and thus it can deal with all kinds of data. Possible users could be centres of environment monitoring, hydrologic information and statistic variables, in their effort to connect their data to spatial information.

ΚΕΦΑΛΑΙΟ 1

Αντικείμενο της εργασίας

1.1 Το πρόβλημα της συλλογής και πιστοποίησης μετρήσεων

Πλήθος φορέων και υπηρεσιών, ασχολείται με τη συλλογή μετρήσεων για μεγέθη του ενδιαφέροντός τους, όπως θερμοκρασία, υγρασία, σεισμικότητα, ακτινοβολία, ατμοσφαιρική πίεση, περιβαλλοντικοί ρύποι κλπ. Οι μετρήσεις γίνονται με χρήση αισθητήρων που τοποθετούνται στη περιοχή ενδιαφέροντος είτε μεμονωμένα, είτε σε ένα σταθμό που καταγράφει ένα σύνολο μεγεθών. Οι αισθητήρες αυτοί λειτουργούν χωρίς συνεχή ανθρώπινη επίβλεψη, συνεπώς πρέπει να εξασφαλίζεται η ορθότητα των μετρήσεων που συλλέγονται για ευνόητους λόγους. Η πιστοποίηση αποτελεί ένα από τα πλέον σημαντικά ζητήματα που απασχολούν τους διαχειριστές του συστήματος καθώς εσφαλμένες μετρήσεις αποτελούν την απαρχή διαδικασιών που θα οδηγήσουν σε προβληματικά αποτελέσματα. Συνεπώς πρέπει να υπάρχουν μηχανισμοί που θα πιστοποιούν την εγκυρότητα στο πρώτο ακόμα επίπεδο συλλογής της πληροφορίας και θα προστατεύουν το σύστημα από σφάλματα όπως λάθη μέτρησης, σφάλματα κατά τη καταγραφή κλπ.

1.2 Το πρόβλημα της αποθήκευσης μετρήσεων

Οι μετρήσεις που λαμβάνονται από τους αισθητήρες, πρέπει κάπως να αποθηκεύονται ώστε να είναι διαθέσιμες ανά πάσα στιγμή προς ανάκτηση και επεξεργασία. Η συχνότητα λήψης μπορεί να είναι έως και αρκετές μετρήσεις το λεπτό άρα αναφερόμαστε σε πολύ μεγάλο όγκο δεδομένων που πρέπει να καταχωρείται σε κάποιο μέσο με τρόπο που να προσφέρει ευκολία στη πρόσβαση και λειτουργικότητα στη διαχείριση. Έτσι χρησιμοποιούνται σχεσιακές βάσεις δεδομένων που ικανοποιούν αυτά τα ζητούμενα. Πέρα από τη φύση του συστήματος που ασχολείται με την αποθήκευση, τίθεται και το ζήτημα της δομής των δεδομένων. Συνήθως κάθε περίπτωση αντιμετωπίζεται ανάλογα με τις ανάγκες που καλείται να ικανοποιήσει χωρίς να λαμβάνει κάποιο πρότυπο δομής δεδομένων. Το γεγονός αυτό έχει ως αποτέλεσμα τον περιορισμό της δυνατότητας χρήσης από άλλους εν δυνάμει χρήστες ή αν μη τι άλλο τη δυσκολία στη διαχείριση. Μια αντιμετώπιση του ζητήματος επιτυγχάνεται με την υιοθέτηση προτύπων που έχουν προταθεί από διεθνείς οργανισμούς που ασχολούνται με τη προτυποποίηση δομών, που όμως δεν γίνεται πάντα αφού συνήθως κάθε φορέας φτιάχνει τη δική του δομή.

1.3 Το πρόβλημα της ανάκτησης και διάθεσης μετρήσεων

Αφού έχει δημιουργηθεί η βάση που περιέχει τα δεδομένα μας, τίθεται το ζήτημα της διάθεσης αυτών και πρέπει να αντιμετωπιστούν προβλήματα σε δύο επίπεδα. Το πρώτο επίπεδο αφορά τη φυσική διάθεση, καθώς πρέπει να υπάρχει κάποιος τρόπος πρόσβασης στη βάση δεδομένων. Συνήθως πρόκειται για πρόσβαση μέσα από το Internet είτε απευθείας στη βάση, είτε μέσα από κάποιο Service. Το δεύτερο επίπεδο αναφέρεται στη δομή των δεδομένων όπως περιγράφηκε στο προηγούμενο κεφάλαιο. Ο χρήστης για να μπορεί να έχει στη διάθεση του τις μετρήσεις, πρέπει η δομή της βάσης να ακολουθεί κάποιο πρότυπο, κοινά αποδεκτό για το είδος των μετρήσεων, γεγονός που ανάγεται στο προηγούμενο πρόβλημα.

1.4 Το πρόβλημα της απεικόνισης μετρήσεων

Οι μετρήσεις καταγράφονται από αισθητήρες που βρίσκονται στη περιοχή που μας ενδιαφέρει. Σαφώς λοιπόν υπάρχει και χωρική διάσταση στα δεδομένα. Η απεικόνιση είναι πολύ συχνά ζητούμενο διότι απευθύνεται στο περιβάλλον του ανθρώπου, τον χώρο. Βέβαια υπάρχει μια σειρά προαπαιτούμενων για να είναι δυνατή η απεικόνιση, με βασικότερη την ύπαρξη υποβάθρων για την απεικόνιση της χωρικής αναφοράς των μετρήσεων. Κάθε φορέας να αναπτύξει δικό του μηχανισμό απεικόνισης και διάθεσης των προϊόντων, γεγονός πολλές φορές κρίνεται ανέφικτο και αντιοικονομικό.

1.5 Το περιβάλλον Google Earth/Google maps (σύντομη αναφορά)

Η Google διαθέτει ένα εργαλείο που δίνει πρόσβαση σε ελεύθερα υπόβαθρα. Η εφαρμογή Google Earth δίνει τη δυνατότητα επιλογής μεταξύ χαρτών αλλά και δορυφορικών εικόνων. Η κάλυψη είναι παγκόσμια με ανάλυση εικόνας που ποικίλει ανάλογα με την περιοχή και μπορεί να φτάσει μέχρι και 15 cm σε κάποιες περιοχές. Εκτός από την εφαρμογή όμως, διατίθεται και το *Application Program Interface* (API) που αποτελεί εργαλείο για δημιουργία νέων εφαρμογών, προσαρμοσμένων στις ανάγκες του χρήστη, που έχουν πρόσβαση στα ίδια υπόβαθρα.

1.6 Οριοθέτηση της εργασίας

Όπως είδαμε, στους διάφορους φορείς, υπηρεσίες που ασχολούνται με συλλογή, επεξεργασία και διαχείριση δεδομένων, υπάρχουν πληροφορίες οργανωμένες σε βάσεις δεδομένων. Πολλές φορές έχουν και χωρική διάσταση αλλά υπάρχει ελλιπής υποδομή παροχής και παρουσίασης αυτών. Τα προβλήματα που συναντώνται προς μια αντιμετώπιση καθολικής διάθεσης, όταν είναι ανοιχτού - ελεύθερου χαρακτήρα, έχουν ως βάση :

1. Κάθε φορέας θα πρέπει να έχει διαθέσιμα υπόβαθρα.
2. Οι βάσεις δεδομένων έχουν διαφορετική δομή, ακολουθώντας διαφορετικά κάθε φορά πρότυπα.
3. Κάθε φορέας πρέπει να αναπτύξει μηχανισμό διάθεσης.

Εκτός από ανοιχτού χαρακτήρα βάσεις δεδομένων, πλέον υπάρχουν και ανοιχτές πηγές χαρτογραφικού υποβάθρου. Η πλέον διαδεδομένη από αυτές είναι η υπηρεσία Google Maps, η οποία δίνει στους προγραμματιστές τη δυνατότητα να ενσωματώσουν χάρτες στις εφαρμογές τους αφού εμπλουτίσουν πρώτα τα υπόβαθρα με δικά τους δεδομένα. Διαφαίνεται λοιπόν πως είναι ρεαλιστική πλέον μια προσπάθεια αξιοποίησης ευρύτερα και καθολικά των πληροφοριών που δεν είναι λειτουργικά προσβάσιμες σε εν δυνάμει τελικούς χρήστες. Μια τέτοια προσέγγιση πρέπει να στηριχτεί και δομηθεί πάνω σε τρεις βασικούς άξονες :

1. Η οργάνωση των πληροφοριών να γίνεται με κάποιο κοινά ακολουθούμενο πρότυπο. Έτσι θα είναι δυνατή η διαχείριση των πληροφοριών με καθολικότητα.
2. Να αξιοποιηθούν τα διαθέσιμα ελεύθερα υπόβαθρα.
3. Να γίνει διαχείριση των δεδομένων και απαραίτητη επεξεργασία αυτών, ώστε να εμπλουτίσουν τους χάρτες με θεματική πληροφορία.

Ουσιαστικά πρόκειται για ένα Σύστημα Γεωγραφικών Πληροφοριών, του οποίου οι πηγές δεδομένων δεν θα βρίσκονται απαραίτητα τοπικά αλλά και απομακρυσμένα. Τα δεδομένα που πραγματευόμαστε έχουν χωρικό χαρακτήρα και μπορεί να είναι σημειακά, γραμμικά αλλά και επιφανειακά. Φυσικά προστίθεται σε αυτά και η διάσταση του χρόνου.

Στα πλαίσια της διπλωματικής εργασίας θα αναπτυχθεί εφαρμογή που θα ασχοληθεί με τη παραπάνω προσέγγιση. Έτσι ο στόχος της εργασίας αυτής διατυπώνεται ως εξής :

«Ανάπτυξη εφαρμογής ανάκτησης και απεικόνισης πληροφοριών με χωρική διάσταση, από προτυποποιημένη Βάση Δεδομένων σε ελεύθερα υπόβαθρα με χρήση Google Maps API»

Βέβαια δεν θα εξαντληθεί κάθε δυνατότητα καθώς θα απαιτούσε ομάδα εργασίας, χρόνο και προϋπολογισμό που ξεφεύγουν από τα όρια της παρούσης. Έτσι το πρωτόλειο αυτό GIS θα δομηθεί με τους εξής περιορισμούς :

- Η χωρική διάσταση των πληροφοριών θα έχει σημειακό χαρακτήρα.
- Η πηγή της Βάσης Δεδομένων θα ακολουθεί τη δομή της Microsoft Access

1.6.1 Θέματα με τα οποία θα ασχοληθούμε

Στα πλαίσια της εργασίας θα δημιουργηθεί εφαρμογή που θα αντιμετωπίσει το ζήτημα της απεικόνισης μετρήσεων σε ελεύθερα χαρτογραφικά υπόβαθρα. Καλούμαστε να αντιμετωπίσουμε μια σειρά από ζητούμενα.

- Δημιουργούμε σύνδεση με τη βάση δεδομένων ώστε να υπάρχει ροή πληροφοριών προς τη κεντρική εφαρμογή. Ερωτήματα SQL δομούνται και ανακτούν από τη βάση δεδομένων τις ζητούμενες μετρήσεις.
- Οι μετρήσεις που περιέχει η βάση δεδομένων δεν έχουν περιορισμό στο πλήθος. Συνεπώς πρέπει να αναπτυχθεί ένας μηχανισμός διαχείρισης των μετρήσεων. Αυτό επιτυγχάνεται με δημιουργία φίλτρων που περιορίζουν τις ζητούμενες πληροφορίες.
- Τα χαρτογραφικά υπόβαθρα διατίθενται από τη Google με χρήση του Google Maps API. Έτσι θα πρέπει να δομηθεί ο μηχανισμός αλληλεπίδρασης και πρόσβασης στη συγκεκριμένη υπηρεσία.
- Για κάθε αισθητήρα στη βάση έχουμε ιστορικό μετρήσεων. Αυτές οι μετρήσεις τοποθετούνται σε γράφημα που φτιάχνει η εφαρμογή με ταυτόχρονη δυνατότητα ανάκτησης μετρήσεων με χρήση του γραφήματος.

1.6.2 Θέματα που δεν θα μας απασχολήσουν

Θέματα που συνδέονται με το ευρύτερο αντικείμενο αλλά δεν θα μας απασχολήσουν είναι :

- Διαδικασία συλλογής μετρήσεων από τους αισθητήρες, καθώς και οι διαδικασίες πιστοποίησης – αποθήκευσης των μετρήσεων.
- Προτυποποίηση δομής δεδομένων που ακολουθεί η βάση δεδομένων. Δεν θα μας απασχολήσουν ζητήματα που πραγματεύονται τα πρότυπα δεδομένων και μετρήσεων.

1.6.3 Η ζητούμενη εμπειρία από τον χρήστη

Ο χρήστης που θα έρθει σε επαφή με την εφαρμογή μπορεί να είναι εξειδικευμένος πάνω στο αντικείμενο των μετρήσεων αλλά ενδέχεται να είναι και απλός χρήστης. Συνεπώς το πρώτο ζητούμενο είναι η εύκολη χρήση του προγράμματος. Οι λειτουργίες που επιτελεί δεν πρέπει να προβληματίζουν το χρήστη. Έτσι η ευχρηστία είναι ένα από τα βασικά ζητούμενα, καθώς ο μέσος χρήστης δεν ζητάμε να έχει εξειδικευμένες γνώσεις. Αυτό που μας απασχολεί είναι να μπορεί να αντιληφθεί τη χωρική κατανομή των μεγεθών που εξετάζει. Η χαρτογραφική απόδοση προσφέρει δυνατότητα αλληλεπίδρασης, με λειτουργίες επιλογής των αισθητήρων. Στη συνέχεια η πρόσβαση στο σύνολο των μετρήσεων για κάθε αισθητήρα επιτυγχάνεται και με την εξαγωγή στατιστικών μεγεθών που περιγράφουν το επιλεγμένο σύνολο μετρήσεων αλλά και με τη δυνατότητα γραφικής παράστασης που προσδίδεται. Έτσι μπορούμε να επιτύχουμε μια πλοήγηση στα πρωτογενή δεδομένα με εργαλεία που δεν παραπέμπουν στη «σκληρή» μορφή των μετρήσεων και των πληροφοριών που περιέχει η βάση μας, αλλά με στατιστικά μεγέθη και γραφικές παραστάσεις που είναι φιλικότερες και πιο ευνόητες από το μέσο χρήστη.

ΚΕΦΑΛΑΙΟ 2

Google Maps

2.1 Χαρτογραφικά δεδομένα

Η πηγή των χαρτογραφικών δεδομένων είναι οι χάρτες και οι εικόνες που διαθέτει η GOOGLE μέσα από την υπηρεσία Google Maps. Διατίθενται σε τρεις μορφές.

1. Χάρτες



(εικόνα 2.1)

2. Φωτογραφίες



(εικόνα 2.2)

3. Υβριδική μορφή (Φωτογραφίες με ενσωματωμένα χαρτογραφικά στοιχεία)



(εικόνα 2.3)

2.2 Στοιχεία απεικόνισης

Τα διαθέσιμα δεδομένα χρησιμοποιούν το Παγκόσμιο Γεωδαιτικό Σύστημα (WGS 84) με απεικόνιση σε Ορθή Μερκατορική προβολή. Πρόκειται για το σύστημα αναφοράς που χρησιμοποιείται παγκοσμίως. Πρέπει να αναφέρουμε ότι το ίδιο σύστημα είναι αυτό που χρησιμοποιεί εγγενώς το GPS σύστημα με το CA κώδικα. Παρατίθενται τα στοιχεία του Ελλειψοειδούς.

- $a = 6378137,0m$
- $b = 6356752.314m$
- $\frac{1}{f} = 298,257223563$

Οι εικόνες είναι δορυφορικές λήψεις ή αεροφωτογραφίες. Η διακριτική τους ικανότητα ποικίλει από περιοχή σε περιοχή με μέγεθος pixel που είναι από δεκάδες μέτρα μέχρι ~1 μέτρο για τις δορυφορικές λήψεις και μέχρι λίγα εκατοστά για τις αεροφωτογραφίες. Η λήψη ανάλογα με την περιοχή μπορεί να είναι χρονικά ενήμερη, συνήθως όμως οι εικόνες είναι αρκετά χρόνια παλαιότερες. Η Ελλάδα απεικονίζεται μόνο σε δορυφορικές λήψεις που χρονικά τοποθετούνται 2002 – 2003.

- Βασική ανάλυση
 - Η.Π.Α.: 15 m (μερικές πολιτείες έχουν 1 m)
 - Γερμανία, Ελβετία, Δανία, Ολλανδία, Αγγλία, Ανδόρα, Λουξεμβούργο, Λιχτενστάιν, San Marino, Βατικανό: 1 m
 - Παγκόσμια: Γενικά 15 m (μερικές περιοχές όπως η Ανταρκτική έχουν χαμηλή ανάλυση), αλλά εξαρτάται από την ποιότητα της δορυφορικής εικόνας/αεροφωτογραφίας.
- Υψηλή ανάλυση
 - Η.Π.Α.: 1 m, 0.6 m, 0.3 m, 0.15 m (εξαιρετικά σπάνια π.χ. Cambridge και Google Campus)
 - Ευρώπη : 0.3 m, 0.15 m (e.g. Βερολίνο, Ζυρίχη, Αμβούργο)
- Χρόνος: Οι ημερομηνίες των εικόνων ποικίλουν. Με τη μεγέθυνση των εικόνων μπορεί να αλλάξει και η ημερομηνία. Οι περισσότερες εικόνες αστικών περιοχών είναι από το 2004. Οι περισσότερες εικόνες των Η.Π.Α. είναι ενημερωμένες.

2.3 Google Maps API

Η πρόσβαση στα δεδομένα που διαθέτει η GOOGLE γίνεται μέσω του Google Maps *Application Program Interface* (API). Πρόκειται για ένα σύνολο Κλάσεων (Classes) με Ιδιότητες (Properties), Συμβάντα (Events), Μεθόδους (Methods) δομημένα με τρόπο που να κάνουν εφικτή τη πρόσβαση στα δεδομένα και να επιτρέπουν την ανάκτηση αυτών καθώς και τον εμπλουτισμό τους πριν από την απεικόνισή. Οι βασικές κλάσεις αναφέρονται στον πίνακα 2.1.

ΚΛΑΣΗ	ΠΕΡΙΓΡΑΦΗ
class GMap2	Απαραίτητη κλάση για τη δημιουργία ενός χάρτη. Αποτελεί τη κεντρική κλάση στο API. Όλα τα υπόλοιπα λειτουργούν προσθετικά σε αυτή τη κλάση.
class GMapOptions	Αναπαιστικά προαιρετικές παραμέτρους για την κλάση GMap2.
class GKeyboardHandler	Χρησιμοποιείται για να αποδώσουμε λειτουργίες στα πλήκτρα.
class GInfoWindow	Διαχειρίζεται πληροφορίες που απεικονίζονται στο χάρτη.
class GMarker	Αντιπροσωπεύει επίσημανση θέσης στο χάρτη. Αποτελείται από ένα σημείο (Point) Που ορίζει τη γεωγραφική του θέση και ένα εικονίδιο (Icon). Μέσα από το GMarker καλείται το GInfoWindow για απεικόνιση πρόσθετων πληροφοριών
class GPolyline	Αναφέρεται σε πολυγωνική γραμμή που απεικονίζεται στον χάρτη.
class GPolygon	Αναφέρεται σε κλειστό πολύγωνο που απεικονίζεται στον χάρτη, με πρόσθετες ιδιότητες όπως γέμισμα και διαφάνεια.
class GIcon	Προσδιορίζει το εικονίδιο που χρησιμοποιεί το GMarker.
class GPoint	Αντιπροσωπεύει ένα σημείο του χάρτη με τις συντεταγμένες του σε pixel.
class GLatLng	Αντιπροσωπεύει ένα σημείο του χάρτη με τις γεωγραφικές του συντεταγμένες (μήκος, πλάτος).
class GBounds	Ορίζει μια ορθογωνική περιοχή στο χάρτη με συντεταγμένες σε pixel.

ΚΛΑΣΗ	ΠΕΡΙΓΡΑΦΗ
class GLatLngBounds	Ορίζει μια ορθογωνική περιοχή στο χάρτη με γεωγραφικές συντεταγμένες (μήκος, πλάτος).
class GMapType	Τύπος του χάρτη που θα απεικονιστεί

(πίνακας 2.4)

Το API έχει δημιουργηθεί για χρήση με τη γλώσσα προγραμματισμού JavaScript. Εναλλακτικά προτείνεται η μεθοδολογία χρήσης PHP ή ASP Script που να φτιάχνει κώδικα HTML με εντολές JavaScript. Μια παραλλαγή αυτής της μεθόδου έχει αναπτυχθεί στη παρούσα εργασία. Μέσα από το κεντρικό περιβάλλον της εφαρμογής, δημιουργείται κατά το χρόνο εκτέλεσης (at runtime) κώδικας HTML με εντολές JavaScript που χρησιμοποιούν το Google Maps API.

2.4 Δικαιώματα χρήσης

Προκειμένου να έχει πρόσβαση ο χρήστης στα δεδομένα, πρέπει να προμηθευτεί από τη Google ένα αλφαριθμητικό κωδικό 86 ψηφίων (API key). Στην ηλεκτρονική διεύθυνση <http://code.google.com/apis/maps/signup.html> αφού συμπληρωθεί η φόρμα με τα απαραίτητα στοιχεία, ολοκληρώνεται η εγγραφή στην υπηρεσία και αποστέλλεται με email ο κωδικός. Μέρος της διαδικασίας εγγραφής αποτελεί η αποδοχή των όρων χρήσης (Google Maps API Terms of Service). Εκεί αναφέρεται ότι :

- Το API μπορεί να χρησιμοποιηθεί μόνο για προβολή του υλικού και για καμιά άλλη χρήση.
- Δεν επιτρέπεται η αντιγραφή, διανομή, αλλοίωση ή άλλη χρήση.
- Δεν επιτρέπεται η χρήση για real time πλοήγηση
- Δεν επιτρέπεται η χρήση για διαχείριση οχημάτων.
- Δεν επιτρέπεται η χρήση για σύστημα διαχείρισης στόλου ή άλλη παρόμοια εφαρμογή.

Το API μπορεί να χρησιμοποιηθεί μόνο για υπηρεσίες που είναι γενικά προσβάσιμες στους καταναλωτές χωρίς χρέωση. Δεν μπορεί να χρησιμοποιηθεί για υπηρεσίες που απαιτούν συνδρομή ή άλλου είδους περιορισμένη πρόσβαση.

ΚΕΦΑΛΑΙΟ 3

Λειτουργική περιγραφή του λογισμικού

3.1 Απαιτήσεις σε εξοπλισμό, λογισμικό, δίκτυο κλπ

Οι απαιτήσεις σε εξοπλισμό δεν είναι ιδιαίτερες. Η εφαρμογή λειτουργεί ικανοποιητικά σε υπολογιστή με τα εξής τεχνικά χαρακτηριστικά :

Μνήμη RAM :	512 MB
CPU :	2000 MHz Athlon Xp
Κάρτα γραφικών :	128 MB μνήμη

Η χρήση της υπηρεσίας Google Maps απαιτεί την ύπαρξη δικτύου για πρόσβαση στο Internet. Σύνδεση ταχύτητας 512 kbps κρίνεται ικανοποιητική για τις λειτουργίες αλληλεπίδρασης με τη Google για χρήση των υποβάθρων.

Η εφαρμογή εκτός από απαιτήσεις σε εξοπλισμό έχει κάποιες απαιτήσεις σε λογισμικό. Διακρίνεται σε τρία τμήματα και κάθε ένα από αυτά καθορίζει και μια απαίτηση.

- Google Maps API. Το API για να λειτουργήσει σωστά θέλει εγκατεστημένη μια πρόσφατη έκδοση του Internet Explorer. Η έκδοση 7 είναι πλήρως λειτουργική και συμβατή με την εφαρμογή.
- Λειτουργία κυρίως λογισμικού. Η εφαρμογή δημιουργήθηκε με χρήση του Visual Studio 2005. Για να μπορεί να εκτελεστεί χρειάζεται να εγκατασταθεί το Microsoft .NET Framework 2.0.
- Βάση δεδομένων. Ως μορφή έχει χρησιμοποιηθεί η μορφή της Microsoft Access. Η ύπαρξη αυτής όμως κρίνεται πολύ περιοριστική. Γι' αυτό χρησιμοποιήθηκε ένας τρόπος σύνδεσης που αξιοποιεί τους Odbc Drivers. Για να λειτουργήσει η εφαρμογή πρέπει να εγκατασταθεί το Microsoft ODBC .NET Data Provider. Το ODBC .NET Data Provider είναι μια πρόσθετη μονάδα στο Microsoft .NET Framework που παρέχει πρόσβαση στους οδηγούς Open Database Connectivity (ODBC).

3.2 Ροή εργασιών (workflow) συλλογής δεδομένων, αδειοδότησης, εγκατάστασης λογισμικού, κλπ

Προκειμένου να υλοποιηθεί το σύστημα και να είναι πλήρως λειτουργικό, πρέπει μια σειρά από ενέργειες να υλοποιηθούν τα διάφορα στάδια του.

- Συλλογή, αποθήκευση, διάθεση μετρήσεων.



Αφού στηθεί το δίκτυο των αισθητήρων, αρχίζει η συλλογή των μετρήσεων. Κάθε μέτρηση που έρχεται καταχωρείται στη Βάση Δεδομένων, ενημερώνοντας τα κατάλληλα πεδία. Ενημερώνεται το πεδίο της τελευταίας μέτρησης του αισθητήρα καθώς και τα στοιχεία του αισθητήρα από τον οποίο έρχεται η μέτρηση όπως πχ. Ο τομέας (Segment) στο οποίο ανήκει. Στη φάση αυτή τίθενται τα ζητήματα πιστοποίησης των καταγραφόμενων δεδομένων. Αφού δημιουργηθεί η βάση και έχει συμπληρωθεί με μετρήσεις, πρέπει να γίνει δυνατή η πρόσβαση σε αυτή απ'τον τελικό χρήστη. Συνήθως ανοίγει το αρχείο της βάσης για πρόσβαση μέσω Web.

- Απόκτηση κλειδιού χρήσης για την υπηρεσία Google Maps.

Για να χρησιμοποιηθούν τα υπόβαθρα που διαθέτει η υπηρεσία Google Maps πρέπει να προμηθευτεί ο χρήστης ένα κωδικό πρόσβασης. Στη διεύθυνση :

<http://code.google.com/apis/maps/signup.html>

Αφού συμπληρώσει ο χρήστης τα ζητούμενα στοιχεία, προμηθεύεται τον κωδικό που χρειάζεται για τη χρήση του Google Maps API.

- Εγκατάσταση προαπαιτούμενων στοιχείων.

Για να λειτουργήσει η εφαρμογή πρέπει να έχουμε προεγκαταστήσει τα απαραίτητα προγράμματα για τη λειτουργία της, όπως αυτά περιγράφονται στην ενότητα 3.1 Απαιτήσεις σε εξοπλισμό, λογισμικό, δίκτυο κλπ.

3.3 Λειτουργίες της εφαρμογής από την πλευρά του χρήστη

Η εφαρμογή δημιουργείται για αποτελέσει ένα εργαλείο στα χέρια του χρήστη, με το οποίο θα μπορεί να αντιληφθεί τη χωρική διάσταση των μετρήσεων αλλά και να αποδώσει τη διακύμανσή τους με χρήση στατιστικών στοιχείων και διαγραμμάτων. Η διαδικασία αυτή εμπεριέχει και κάποιες βοηθητικές εργασίες (αλληλεπίδραση με ΒΔ, φίλτρα δεδομένων). Στη συνέχεια γίνεται αναφορά στις λειτουργίες που πρέπει να επιτελεί η εφαρμογή.

3.3.1 Σύνδεση με ΒΔ

Η Βάση Δεδομένων αποτελεί την πηγή των πληροφοριών για την εφαρμογή. Συνεπώς πρέπει σε πρώτη φάση να υλοποιείται η σύνδεση με αυτή ώστε να είναι εφικτή η διατύπωση SQL ερωτημάτων προς αυτή. Στη διαδικασία αυτή ο χρήστης καλείται να προσδιορίσει τη θέση της βάσης, καθώς και τα στοιχεία πρόσβασης (UserName, Password). Αφού υλοποιηθεί η σύνδεση είναι πλέον δυνατή η ροή πληροφοριών από τη ΒΔ προς την εφαρμογή, ώστε να μπορεί να γίνει επεξεργασία και παρουσίαση των μετρήσεων στις μορφές που υποστηρίζονται.

3.3.2 Μορφές απεικόνισης

Η απεικόνιση των μετρήσεων στο πεδίο του χώρου θέλουμε να προσφέρεται με δύο τρόπου. Απεικόνιση σε χαρτογραφικό υπόβαθρο αλλά και σε δορυφορικές εικόνες. Στους χάρτες έχουμε φιλτραρισμένη πληροφορία ώστε να μην ασχολούμαστε με περιττές πληροφορίες παρά μόνο με τα χωρικά στοιχεία (οδούς, τοπωνυμία κλπ). Πολλές φορές όμως η εξαγωγή συμπερασμάτων για ένα φαινόμενο απαιτεί την εποπτεία της περιοχής

με χρήση δορυφορικών εικόνων. Έτσι η εφαρμογή θα υποστηρίζει και τα δύο είδη υποβάθρων ώστε να προσδίδεται η μέγιστη δυνατή λειτουργικότητα. Επιπρόσθετα θα δίνεται η δυνατότητα προβολής στοιχείων για τους αισθητήρες αφού γίνει επιλογή του εικονιδίου τους.

3.3.3 Φίλτρα

Οι μετρήσεις που περιέχουν οι ΒΔ συνήθως είναι πολύ μεγάλου πλήθους. Το γεγονός αυτό καθιστά δύσκολη τη διαχείρισή τους. Ακόμα είναι γεγονός ότι ο χρήστης δεν επιθυμεί το σύνολο των μετρήσεων αλλά ένα υποσύνολό τους. Έτσι λοιπόν η εφαρμογή πρέπει να δίνει τη δυνατότητα επιλεκτικής απεικόνισης μέσω ενός μηχανισμού κριτηρίων που θα υλοποιείται από φίλτρα χρονικού ενδιαφέροντος. Ο χρήστης θέλουμε να μπορεί να περιορίσει τον όγκο των μετρήσεων που διαχειρίζεται ώστε να είναι πιο ευέλικτη η απεικόνιση των μετρήσεων.

3.3.4 Γραφήματα

Εκτός από τη χωρική απεικόνιση των μετρήσεων, ζητούμενο είναι και η απεικόνιση τους σε σχέση με το χρόνο. Η ΒΔ έχει καταχωρημένο το ιστορικό των μετρήσεων που έχει καταγράψει κάθε αισθητήρας. Η εφαρμογή θα αναλάβει να δημιουργεί γραφήματα των χρονοσειρών που έχουμε στη διάθεσή μας. Τα γραφήματα αυτά θα δίνουν τη δυνατότητα, μετά από τοποθέτηση του δείκτη του ποντικιού στην περιοχή τους, να απεικονίζεται η μέτρηση ώστε να έχουμε και απευθείας πρόσβαση στα πρωτογενή δεδομένα.

3.3.5 Στατιστικά στοιχεία

Για κάθε αισθητήρα όπως έχει αναφερθεί, διατηρείται στη βάση δεδομένων το ιστορικό των μετρήσεων που έχει καταγράψει. Επειδή αναλύουμε ένα σύνολο που ουσιαστικά αποτελεί δείγμα ενός πληθυσμού, κρίνεται σκόπιμη η εξαγωγή στατιστικών μεγεθών από το σύνολο των μετρήσεων. Η εφαρμογή, αφού επιλεγεί ο αισθητήρας ενδιαφέροντος, θα υπολογίζει και απεικονίζει στατιστικά στοιχεία όπως μέγιστο, ελάχιστο, μέση τιμή, τυπικό σφάλμα, εύρος.

ΚΕΦΑΛΑΙΟ 4

Προγραμματισμός

4.1 Περιβάλλον προγραμματισμού

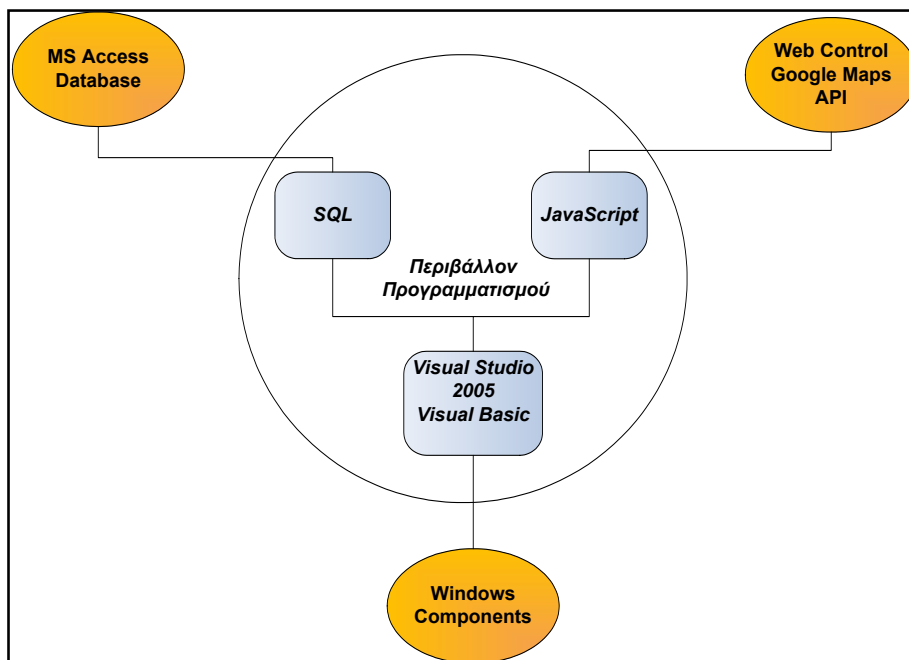
Το περιβάλλον ανάπτυξης της εφαρμογής επιλέχθηκε έτσι ώστε να ικανοποιεί τα παρακάτω κριτήρια :

- Συνδεσιμότητα σε Βάση Δεδομένων
- Χρήση του Google maps API (JavaScript)
- Απεικόνιση Web Components
- Δυνατότητες Γραφικής απόδοσης

Ύστερα από αξιολόγηση των κριτηρίων και απαιτήσεων αποφασίστηκε το εξής μοντέλο εργασίας. Η κύρια πλατφόρμα της εφαρμογής στήνεται σε Visual Studio 2005 παραμετροποιημένη σε Visual Basic. Σε χρόνο εκτέλεσης (runtime) δημιουργείται :

1. Κώδικας JavaScript που χρησιμοποιείται σε Web Control για την απεικόνιση
2. SQL ερωτήματα που απευθύνονται στη βάση δεδομένων

Άρα πρόκειται για μια μορφή προγραμματισμού που χρησιμοποιεί τρεις «Γλώσσες επικοινωνίας» με τον υπολογιστή.



(σχήμα 4.1)

4.2 Visual Studio 2005

Πρόκειται για σύνολο από εργαλεία προγραμματισμού. Έχει αναπτυχθεί από τη Microsoft και η τρέχουσα έκδοση είναι η Visual Studio 2008. Χρησιμοποιείται για ανάπτυξη εφαρμογών Console, Windows form, web, web services, που εκτελούνται σε πλατφόρμες Microsoft windows, windows mobile, .Net framework, .Net compact framework και Microsoft Silverlight. Υποστηρίζει τις γλώσσες :

- C/C++
- VB.NET
- C#
- F#
- Python

Ακόμα υποστηρίζει XML/HTML/JavaScript και CSS. Η επιλογή έγινε γιατί καλύπτει βασικές απαιτήσεις.

- Επιτρέπει τη σύνδεση με βάσεις δεδομένων. Μάλιστα η σύνδεση μπορεί να επιτευχθεί με πολλαπλούς τρόπους και από διάφορες πηγές. Στη παρούσα εργασία έγινε σύνδεση με Odbc driver που διαχειρίζεται βάσεις Ms Access.
- Διαθέτει εργαλεία απεικόνισης και μπορεί να διαχειριστεί γραφικό περιβάλλον.
- Μπορεί να φιλοξενήσει Web περιεχόμενο με χρήση αντικειμένων που δίνουν τη δυνατότητα απεικόνισης Html σελίδων.

4.3 JavaScript

Η JavaScript επελέχθη για δύο λόγους. Ο ένας είναι ότι δουλεύει σε web περιβάλλον και ως εκ τούτου έχει όλα τα πλεονεκτήματα των HTML σελίδων. Ο κύριος λόγος όμως είναι το ότι προσδιορίζεται από το Google maps API ως το περιβάλλον στο οποίο είναι στημένες οι κλάσεις που θα χρησιμοποιήσουμε.

Είναι γλώσσα προγραμματισμού η οποία έχει σαν σκοπό την παραγωγή δυναμικού περιεχομένου σε ιστοσελίδες. Έχει τις ρίζες της στην ECMAScript της οποίας ουσιαστικά αποτελεί επέκταση με μερικές πρόσθετες δυνατότητες. Όπως και η PHP, η Javascript έχει βασιστεί όσον αφορά τον τρόπο σύνταξης του κώδικά της στη γλώσσα προγραμματισμού C, με την οποία παρουσιάζει πολλές ομοιότητες. Όμως ενώ η PHP είναι μια server side γλώσσα προγραμματισμού, η Javascript είναι client side. Αυτό σημαίνει ότι η επεξεργασία του κώδικα Javascript και η παραγωγή του τελικού

περιχομένου HTML δεν πραγματοποιείται στον server, αλλά στο πρόγραμμα περιήγησης των επισκεπτών. Αυτή η διαφορά έχει και πλεονεκτήματα και μειονεκτήματα για καθεμιά από τις δύο γλώσσες. Συγκεκριμένα, η Javascript δεν έχει καμία απαίτηση από πλευράς δυνατοτήτων του server για να εκτελεστεί (επεξεργαστική ισχύ, συμβατό λογισμικό διακομιστή), αλλά βασίζεται στις δυνατότητες του browser των επισκεπτών. Επίσης μπορεί να ενσωματωθεί σε στατικές σελίδες HTML. Παρόλα αυτά, οι δυνατότητές της είναι σημαντικά μικρότερες από αυτές της PHP και δεν παρέχει συνδεσιμότητα με βάσεις δεδομένων. Η Javascript δεν θα πρέπει να συγχέεται με τη Java, που είναι διαφορετική γλώσσα προγραμματισμού και με διαφορετικές εφαρμογές. Τονίζεται ότι ο σωστός τρόπος γραφής της είναι "Javascript" και όχι 'Java script' σαν δύο λέξεις, όπως λανθασμένα γράφεται ορισμένες φορές.

4.4 SQL (Structured Query Language)

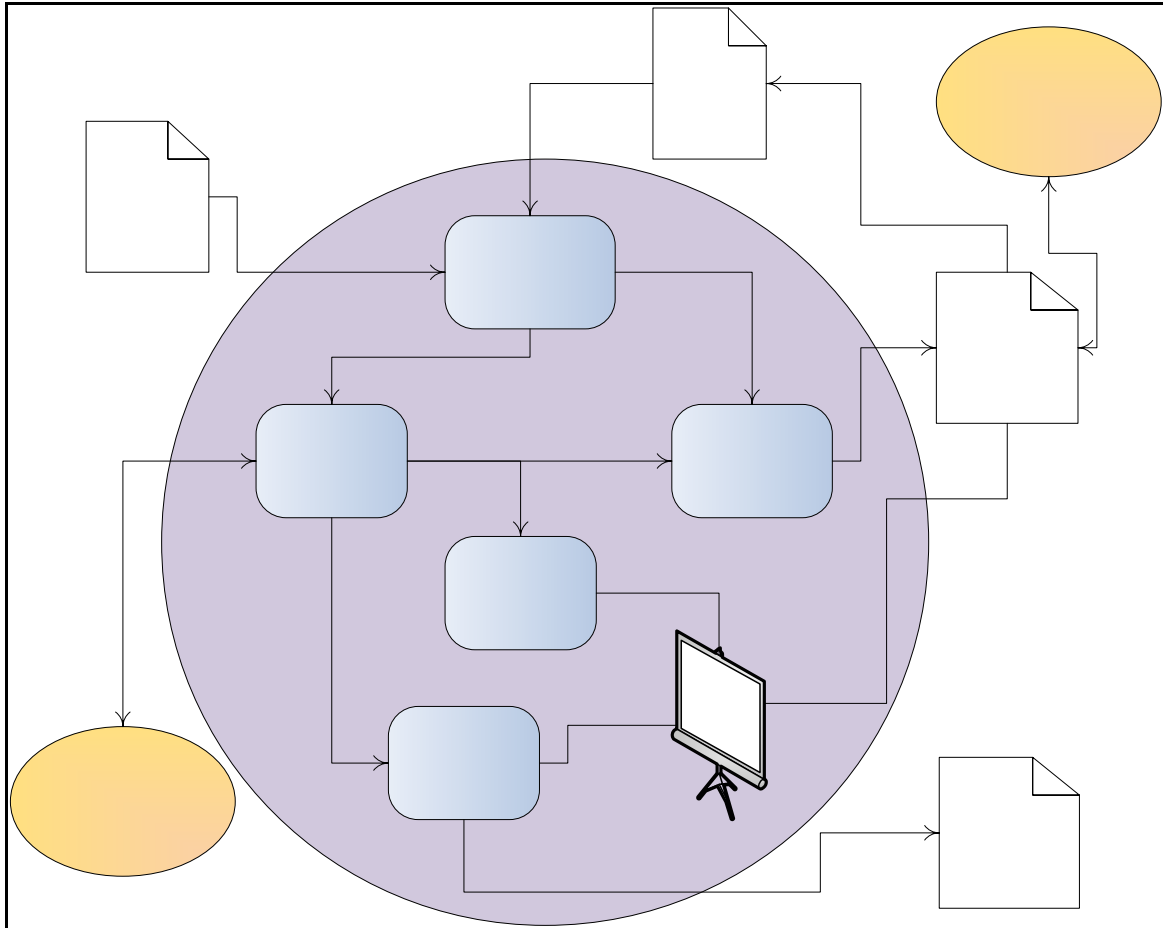
Πρόκειται για γλώσσα βάσεων δεδομένων σχεδιασμένη για ανάκτηση και διαχείριση δεδομένων σε Συστήματα Διαχείρισης Σχεσιακών Βάσεων Δεδομένων (RDBMS). Η SQL είναι αλληλεπιδραστική και προγραμματιστική γλώσσα. Ο πυρήνας της είναι διαμορφωμένος με εντολές Γραμμής Εντολών (Command Line) που επιτρέπουν την ανάκτηση, εγγραφή, ενημέρωση και διαγραφή δεδομένων. Υποστηρίζεται από όλα τα συστήματα διαχείρισης βάσεων δεδομένων (RDBMS) και έτσι προσφέρει συμβατότητα με κάθε υλοποίηση βάσης ανεξαρτήτου πλατφόρμας. Στην εργασία αποτελεί το μέσο επικοινωνίας της εφαρμογής με τη βάση δεδομένων ώστε να έχουμε πρόσβαση στις μετρήσεις.

ΚΕΦΑΛΑΙΟ 5

Αρχιτεκτονική Λογισμικού

5.1 Γενική περιγραφή

Η εφαρμογή έχει δομηθεί από πέντε βασικές Λειτουργικές Μονάδες (Modules)



(σχήμα 5.1)

- | | |
|----------------------|---|
| 1. Database Module | Διαχειρίζεται την αλληλεπίδραση με τη Βάση Δεδομένων. |
| 2. JavaScript Module | Δημιουργεί τον κώδικα JavaScript που διαχειρίζεται το Google Maps API για την απεικόνιση των πληροφοριών. |
| 3. Statistics Module | Αναλύει τα δεδομένα και εξάγει στατιστικά στοιχεία |
| 4. Graphs Module | Δημιουργεί τα διαγράμματα και τα γραφικά απεικόνισης |
| 5. User Input Module | - Διαχειρίζεται τα στοιχεία και τις παραμέτρους που εισάγει ο Χρήστης
- Ελέγχει τη διασύνδεση Κυρίας εφαρμογής και Web Component |

5.2 Δομή της ΒΔ

Η βάση δεδομένων που θα χρησιμοποιηθεί, πρέπει να μας δίνει τα εξής στοιχεία :

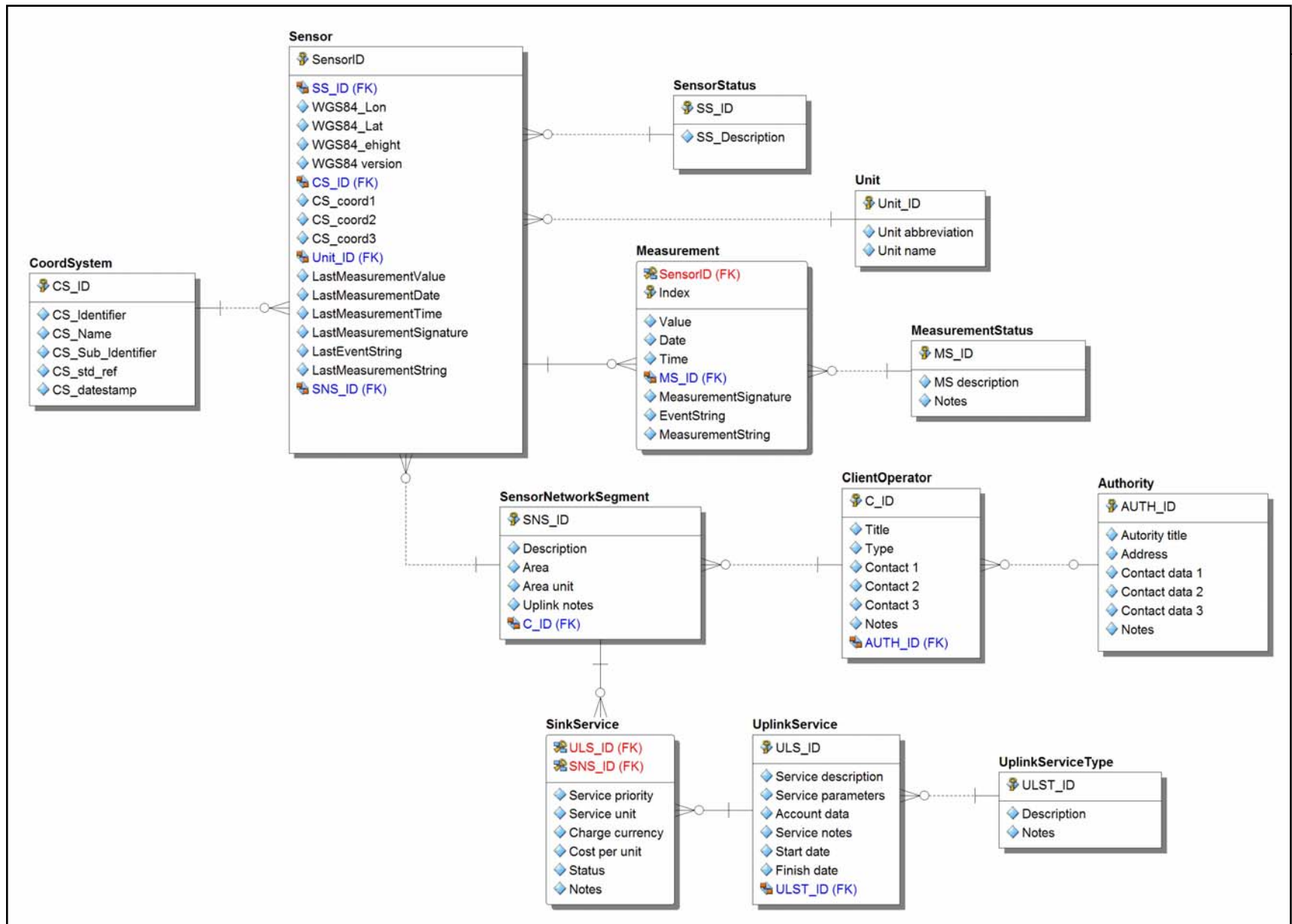
Για κάθε αισθητήρα

- Για κάθε αισθητήρα τη θέση του σε γεωγραφικές συντεταγμένες (φ,λ) που αναφέρονται στο WGS 84'.
- Την τελευταία μέτρηση του κάθε αισθητήρα.
- Πληροφορίες για το είδος της μέτρησης (μονάδες)
- Εικονίδιο απεικόνισης.
- Client operator (διαχειριστής) στο οποίο ανήκει
- Segment (τομέας) στον οποίο ανήκει

Για κάθε μέτρηση

- Τον αισθητήρα στον οποίο αναφέρεται
- Την τιμή της μέτρησης
- Το χρόνο που υλοποιήθηκε

Για να ανακτήσουμε τα απαραίτητα στοιχεία, εκτελούμε τα κατάλληλα SQL ερωτήματα, που απευθύνονται στη βάση. Για να είναι εφικτό αυτό, πρέπει να γνωρίζουμε τη δομή της. Η βάση δεδομένων που χρησιμοποιήθηκε έχει σχεδιαστεί από τον Βεσκούκη Β. Ακολουθεί μια σύντομη αναφορά στη δομή της.



(σχήμα 5.2)

Στο διάγραμμα βλέπουμε ότι η βάση αποτελείται από 12 Οντότητες – πίνακες. Ακολουθεί συνοπτική περιγραφή των πινάκων που χρησιμοποιεί η εφαρμογή κατά τη λειτουργία της (πίνακας 5.3).

ΠΙΝΑΚΑΣ	ΠΕΡΙΓΡΑΦΗ
CoordSystem	Αναφέρεται στο σύστημα συντεταγμένων της θέσης των αισθητήρων. Το βασικό σύστημα είναι το WGS 84' αλλά έχουμε τη δυνατότητα χρήσης ενός εναλλακτικού συστήματος που καταγράφεται σε αυτόν τον πίνακα.
Sensor	Περιέχει πληροφορίες για τους αισθητήρες <ul style="list-style-type: none">• Θέση σε δύο συστήματα (WGS 84' – Εναλλακτικό σύστημα)• Τελευταία μέτρηση
Unit	Μονάδα μέτρησης
Measurement	Περιέχει πληροφορίες για τις μετρήσεις <ul style="list-style-type: none">• Μέτρηση• Χρονική στιγμή τελευταίας μέτρησης
SensorNetworkSegment	Περιέχει πληροφορίες για τους τομείς στους οποίους είναι ομαδοποιημένοι οι αισθητήρες <ul style="list-style-type: none">• Περιγραφή• Εμβαδό

(πίνακας 5.3)

5.3 Database Module

Αυτή η λειτουργική μονάδα αναλαμβάνει την επικοινωνία με τη Βάση Δεδομένων. Συγκεκριμένα αναλαμβάνει να συνδεθεί με τη Βάση Δεδομένων, να διατυπώσει τα απαραίτητα ερωτήματα, και να πάρει τις πληροφορίες που κάθε φορά απαιτεί η εφαρμογή.

Παραδοχές :

- Ύπαρξη Βάσης Δεδομένων MS Access (τοπικά ή απομακρυσμένα) στην οποία έχουμε πρόσβαση.
- Γνώση Username και Password της Βάσης

Input :

- Γενική δομή Ερωτημάτων όπως αυτά προσδιορίζονται στο αρχείο “Param.txt”.
- Παράμετροι αναζήτησης για κάθε ερώτημα (Unit, Client Operator, Segment, SensorID, Date, Time)

Processing :

- Διαμόρφωση τελικών Ερωτημάτων (Queries)
- Αποστολή στο Σύστημα Διαχείρισης Βάσης Δεδομένων (DBMS MS Access)
- Παραλαβή πληροφοριών
- Οργάνωση πληροφοριών σε πίνακες

Output :

- Πίνακες με καταχωρημένες πληροφορίες

Νέες παραδοχές :

- Ύπαρξη πινάκων που έχουν καταχωρημένες τις προαπαιτούμενες πληροφορίες για τη λειτουργία των άλλων λειτουργικών μονάδων της εφαρμογής.

Συγκεκριμένα οι λειτουργίες που κάνει το Database module είναι :

- Σύνδεση με τη Βάση Δεδομένων. Απαραίτητες προϋποθέσεις είναι :
 1. Πρόσβαση στη θέση της Βάσης.
 2. Username και Password
- Ανάκτηση των βασικών περιεχομένων της Βάσης :
 1. Units
 2. Operation Clients
 3. Segments
 4. Μέγιστες και ελάχιστες τιμές για τα δεδομένα του χρόνου
- Αφού επιλεγεί το Segment που μας ενδιαφέρει, το module αναλαμβάνει να ανακτήσει πληροφορίες για τους αισθητήρες που περιέχει όπως :
 1. Τη τελευταία μέτρηση
 2. Τη θέση τους
 3. Την ημερομηνία
 4. Το εικονίδιο απεικόνισης
- Ανάκτηση των μετρήσεων για ένα συγκεκριμένο αισθητήρα στο προσδιορισμένο διάστημα και για συγκεκριμένη περίοδο της ημέρας. Π.χ Από 12/1/1990 έως 25/2/1992 για τις μεσημεριανές ώρες 12:00 – 14:00.

Τα απαραίτητα δεδομένα για τη δομή των ερωτημάτων είναι στο αρχείο “Param.txt”. Δεν είναι κλειστή στο χρήστη ώστε να προσδίδεται ευελιξία.

Το αρχείο αυτό είναι αρχείο ASCII το οποίο περιέχει πληροφορίες για:

1. Το Google Maps API Key
2. Τη δομή των ερωτημάτων που απευθύνονται στη βάση δεδομένων
3. Θέση της Βάσης Δεδομένων που χρησιμοποιήθηκε τελευταία φορά και UserID – Password.

5.4 JavaScript Module

Αυτή η λειτουργική μονάδα αναλαμβάνει τη δημιουργία του HTML αρχείου (SJ_file.html), που περιέχει και τον κώδικα JavaScript που απευθύνεται στο Google Maps API. Αποτελεί πολύ σημαντικό κομμάτι της εφαρμογής γιατί ουσιαστικά είναι αυτό που κάνει της σύνδεση με το χαρτογραφικό μέρος της απεικόνισης. Αφού ολοκληρωθεί η διαδικασία απεικονίζεται σε ένα Web Control. Ακόμα μέσα από τη διαδικασία αυτή υλοποιείται η διασύνδεση μεταξύ κυρίας εφαρμογής και JavaScript κώδικα. Αυτό επιτυγχάνεται με τη δημιουργία ενός ενδιάμεσου αρχείου (Ginfo.txt) το οποίο δημιουργείται από το Web περιβάλλον και περιέχει πληροφορίες για τον αισθητήρα που έχει επιλεγεί (index, UserID, εικονίδιο επίσημανσης).

Παραδοχές :

- Ύπαρξη πινάκων που έχουν καταχωρημένες τις προαπαιτούμενες πληροφορίες για τη δημιουργία του αρχείου.
- Σύνδεση στο Internet
- Κλειδί του Google API που επιτρέπει τη χρήση της υπηρεσίας

Input :

- Κλειδί του Google API που επιτρέπει τη χρήση της υπηρεσίας.
- Πίνακες που έχουν καταχωρημένες τις προαπαιτούμενες πληροφορίες για τη δημιουργία του αρχείου. (UserID, γεωγραφικές συντεταγμένες των αισθητήρων, δεδομένα που θα απεικονίσουμε, θέση εικονιδίων).

Processing :

- Δημιουργία αρχείου SJ_file.html που θα απεικονιστεί στο Web Control.
- Δημιουργία GInfo.txt μέσα από κώδικα JavaScript.

Output :

- Αρχείο HTML που απεικονίζει τους αισθητήρες μας και τις πληροφορίες που μας ενδιαφέρουν.
- Αρχείο GInfo.txt.

Νέες παραδοχές :

- Ύπαρξη αρχείου GInfo.txt.
- Δεδομένα που έχουν απεικονιστεί σε χάρτη στο Web Control.

5.5 Statistics Module

Αυτή η λειτουργική μονάδα επεξεργάζεται τα δεδομένα του αισθητήρα που έχουμε επιλέξει, και εξάγει στατιστικά στοιχεία. Τα στατιστικά αυτά είναι :

1. Μέγιστη - ελάχιστη τιμή
2. Εύρος
3. Μέση τιμή
4. Τυπικό σφάλμα

Παραδοχές :

- Ύπαρξη πινάκων που έχουν καταχωρημένες τις μετρήσεις κάθε αισθητήρα.
- Αρχείο GInfo.txt.

Input :

- Πίνακες που έχουν καταχωρημένες τις μετρήσεις κάθε αισθητήρα.
- SensorID του επιλεγμένου αισθητήρα μέσα από το αρχείο GInfo.txt.

Processing :

- Έλεγχος για μέγιστη – ελάχιστη τιμή.
- Υπολογισμός μέσης τιμής
- Υπολογισμός εύρους
- Υπολογισμός τυπικής απόκλισης

$$\tilde{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

$$R = x_{\max} - x_{\min}$$

$$StDev = \sqrt{\frac{\sum_{i=1}^n (x_i - \tilde{x})^2}{n - 1}}$$

Output :

- Υπολογισμένες τιμές
- String που περιέχει τις τιμές αυτές και απεικονίζεται σε κουτί κειμένου της εφαρμογής.

5.6 Graphs Module

Το Graphs Module αναλαμβάνει να δημιουργήσει τα γραφήματα που απεικονίζουν σειρές μετρήσεων για το ζητούμενο αισθητήρα. Ακόμα ενσωματώνει ένα μηχανισμό απεικόνισης τιμών των χρονοσειρών.

Παραδοχές :

- Ύπαρξη πινάκων που έχουν καταχωρημένες τις μετρήσεις κάθε αισθητήρα.
- Επιλογή αισθητήρα.

Input :

- Πίνακες που έχουν καταχωρημένες τις μετρήσεις κάθε αισθητήρα.
- SensorID του επιλεγμένου αισθητήρα μέσα από το αρχείο GInfo.txt.

Processing :

- Δημιουργία αρχείου γραφήματος χαμηλής ανάλυσης (1024x768) για απεικόνιση στο κεντρικό παράθυρο της εφαρμογής, μαζί με το χάρτη.
- Δημιουργία αρχείου γραφήματος υψηλής ανάλυσης (4096x3072) για λεπτομερέστερη παρουσίαση των πληροφοριών σε ξεχωριστή φόρμα.
- Δημιουργία βοηθητικών πινάκων που περιέχουν ζεύγη τιμών για παρουσίαση στο χρήστη με ένδειξη του mouse cursor.

Output :

- Αρχεία Graph_i.bmp (1024x768)
- Αρχεία GraphMax_i.bmp (4096x3072)
- Πίνακες τιμών

5.7 User Input Module

Το Input Module διαχειρίζεται τα δεδομένα εισόδου και τις παραμέτρους που χρησιμοποιούν οι υπόλοιπες λειτουργικές μονάδες. Υλοποιείται μέσα από :

1. Τα combo επιλογής
2. Τα φίλτρα δεδομένων (χρονικά)
3. Το αρχείο GInfo.txt

Input :

- Στοιχεία για τον επιλεγμένο αισθητήρα.
- Επιλογές χρήστη μέσα από τα φίλτρα

Output :

- Παράμετροι ερωτημάτων προς
 1. Database Module
 2. Graph Module
 3. Statistic Module

5.8 Αρχεία εφαρμογής

Η εφαρμογή δημιουργεί για εσωτερική της χρήση μια σειρά βοηθητικών αρχείων τα οποία τοποθετούνται στο φάκελο που βρίσκεται η εφαρμογή. Στη συνέχεια γίνεται παρουσίαση αυτών και ανάλυση της λειτουργικότητάς τους.

5.8.1 Αρχείο Παραμέτρων «Param.txt»

Το αρχείο αυτό είναι αρχείο ASCII το οποίο περιέχει πληροφορίες για:

1. Το Google Maps API Key
2. Τη δομή των ερωτημάτων που απευθύνονται στη βάση δεδομένων
3. Θέση της Βάσης Δεδομένων που χρησιμοποιήθηκε τελευταία φορά και UserID – Password.

Περιγραφή	Περιεχόμενο αρχείου “Param.txt”
Google Maps API key	<i>ABQIAAAA6JpYJL4LxSfJDQA-0sLkORSq- X9PqyFLZq81FnDkijIOv8KrGBTivr_sBKal4uRuhPWAntMZdH3agw</i>
SQL ερώτημα 1	<i>SELECT SNM_Sensor.LastMeasurementValue, SNM_Sensor.WGS84_Lat, SNM_Sensor.WGS84_Lon, SNM_ClientOperator.Title, SNM_SensorNetworkSegment.Description, SNM_Unit.[Unit name], SNM_Sensor.SensorID, SNM_Sensor.LastMeasurementDate, SNM_Sensor.LastMeasurementTime, SNM_Unit.Icon_Functional, SNM_Unit.Icon_NotFunctional FROM SNM_Unit INNER JOIN (SNM_ClientOperator INNER JOIN SNM_SensorNetworkSegment ON SNM_ClientOperator.C_ID = SNM_SensorNetworkSegment.C_ID) INNER JOIN SNM_Sensor ON SNM_SensorNetworkSegment.SNS_ID = SNM_Sensor.SNS_ID) ON SNM_Unit.Unit_ID = SNM_Sensor.Unit_ID;</i>
SQL ερώτημα 2	<i>SELECT SNM_Sensor.SensorID, SNM_Sensor.WGS84_Lat, SNM_Sensor.WGS84_Lon, SNM_Measurement.Value, SNM_Measurement.Index, SNM_SensorNetworkSegment.Description, SNM_Unit.[Unit name], SNM_Measurement.Date, SNM_Measurement.Time FROM SNM_Unit INNER JOIN (SNM_SensorNetworkSegment INNER JOIN (SNM_Sensor INNER JOIN SNM_Measurement ON SNM_Sensor.SensorID = SNM_Measurement.SensorID) ON SNM_SensorNetworkSegment.SNS_ID = SNM_Sensor.SNS_ID) ON SNM_Unit.Unit_ID = SNM_Sensor.Unit_ID WHERE SNM_Measurement.Date>=##*DATE*## AND SNM_Measurement.DATE<=##*DATE*## AND SNM_Measurement.SensorID=**SensorID** AND SNM_Measurement.Time>=##*TIME*## AND SNM_Measurement.Time<=##*TIME*##;</i>
Τελευταία χρησιμοποιούμενη ΒΔ	<i>C:\TEST.mdb;Admin;;</i>

(πίνακας 5.4)

5.8.2 Αρχείο διασύνδεσης JavaScript – κύριας εφαρμογής «GInfo.txt»

Το αρχείο αυτό είναι αρχείο ASCII το οποίο υλοποιεί τη διασύνδεση του κώδικα JavaScript με το περιβάλλον της κύριας εφαρμογής. Όταν ο χρήστης επιλέγει έναν αισθητήρα, στο αρχείο αυτό καταγράφονται πληροφορίες για:

1. Τον επιλεγμένο αισθητήρα (Index, SensorID)
2. Το εικονίδιο απεικόνισης της επιλογής στο Web Control.

Η κύρια εφαρμογή (Visual Basic κώδικας) χρησιμοποιεί ένα αντικείμενο χρόνου (timer control) και ελέγχει περιοδικά τις καταχωρήσεις στο αρχείο αυτό. Έτσι μπορεί να έχει γνώση της ενέργειας επιλογής που κάνει ο χρήστης μέσα από το Web Control και να προβεί στις απαραίτητες ενέργειες.

Selected Sensors (Index;SensorID;ColorUrl)

2;S_1; <http://users.ntua.gr/rs03015/icons/Green.png>

6;S_7; <http://users.ntua.gr/rs03015/icons/Green.png>

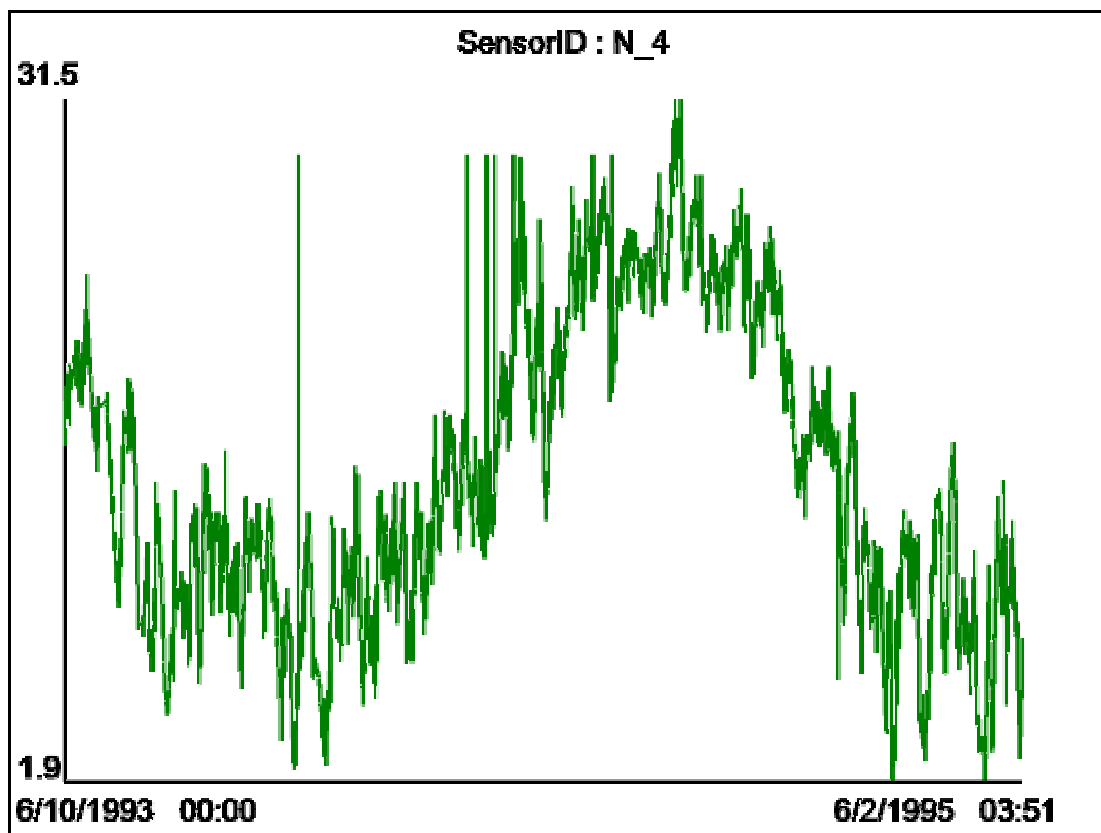
3;S_23; <http://users.ntua.gr/rs03015/icons/Green.png>

5.8.3 Αρχείο HTML – JavaScript κώδικα «SJ file.html»

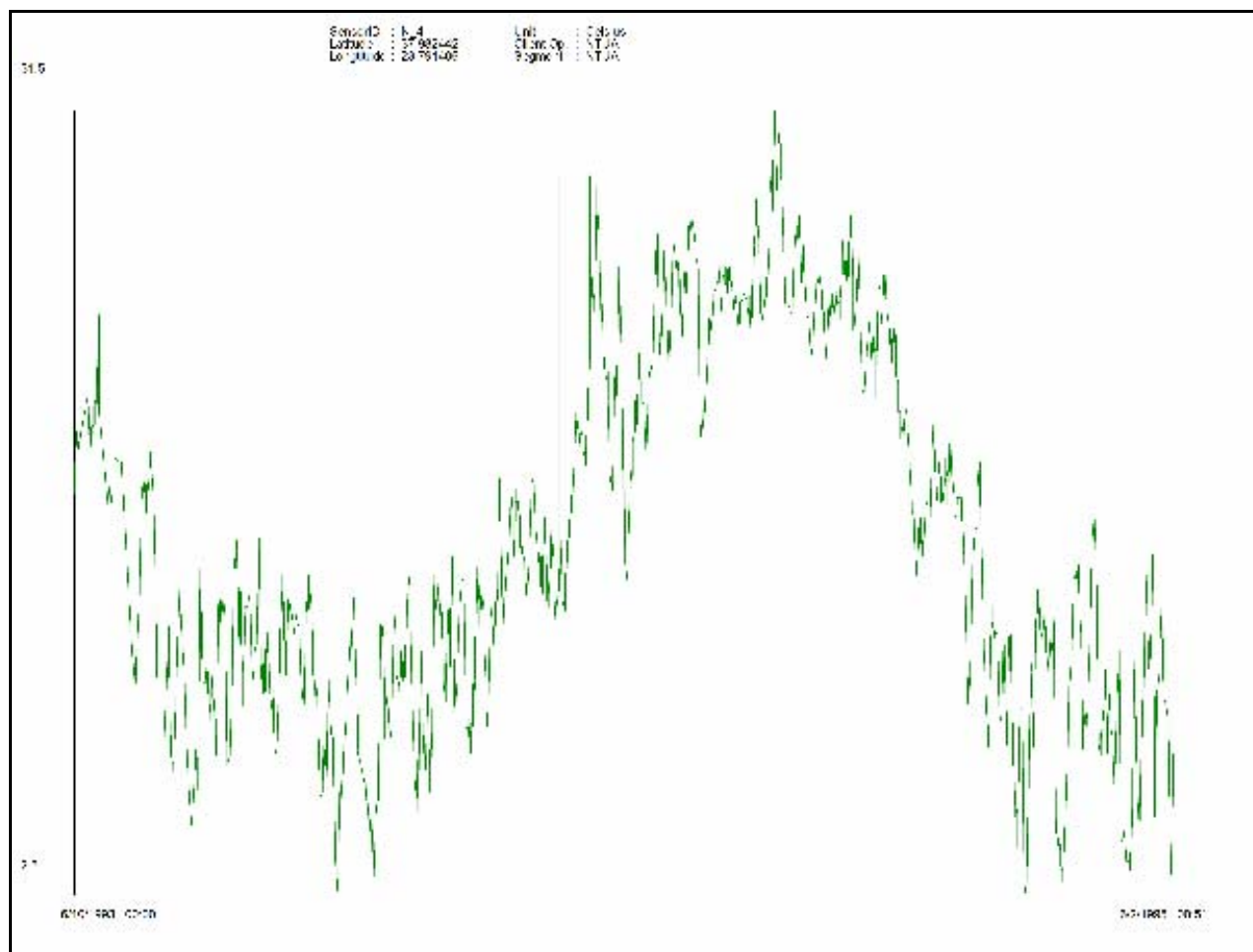
Το αρχείο αυτό είναι HTML αρχείο με κώδικα JavaScript. Περιέχει τις εντολές που απευθύνονται στο Google Maps API και ουσιαστικά υλοποιεί το κομμάτι της χαρτογραφικής απόδοσης. Ενσωματώνει πληροφορίες για τη θέση των αισθητήρων, τα στοιχεία που ζητάμε να απεικονιστούν, τα εικονίδια που θα χρησιμοποιηθούν και λοιπές παραμέτρους και πληροφορίες απεικόνισης.

5.8.4 Αρχεία γραφημάτων «Graph_i.bmp» και «GraphMax_i.bmp»

Τα αρχεία αυτά έχουν τα διαγράμματα των χρονοσειρών σε μορφή bitmap (.bmp). Είναι δύο ομάδων με διαφορετική ανάλυση. Έχουμε τα αρχεία χαμηλής ανάλυσης (1024x768) «Graph_i.bmp» τα οποία χρησιμοποιούνται από την εφαρμογή για την παρουσίαση των μετρήσεων παράλληλα με την χαρτογραφική τους απεικόνιση, και τα αρχεία υψηλής ανάλυσης (4096x3072) που υλοποιούν λεπτομερέστερη απόδοση.



(εικόνα 5.5)



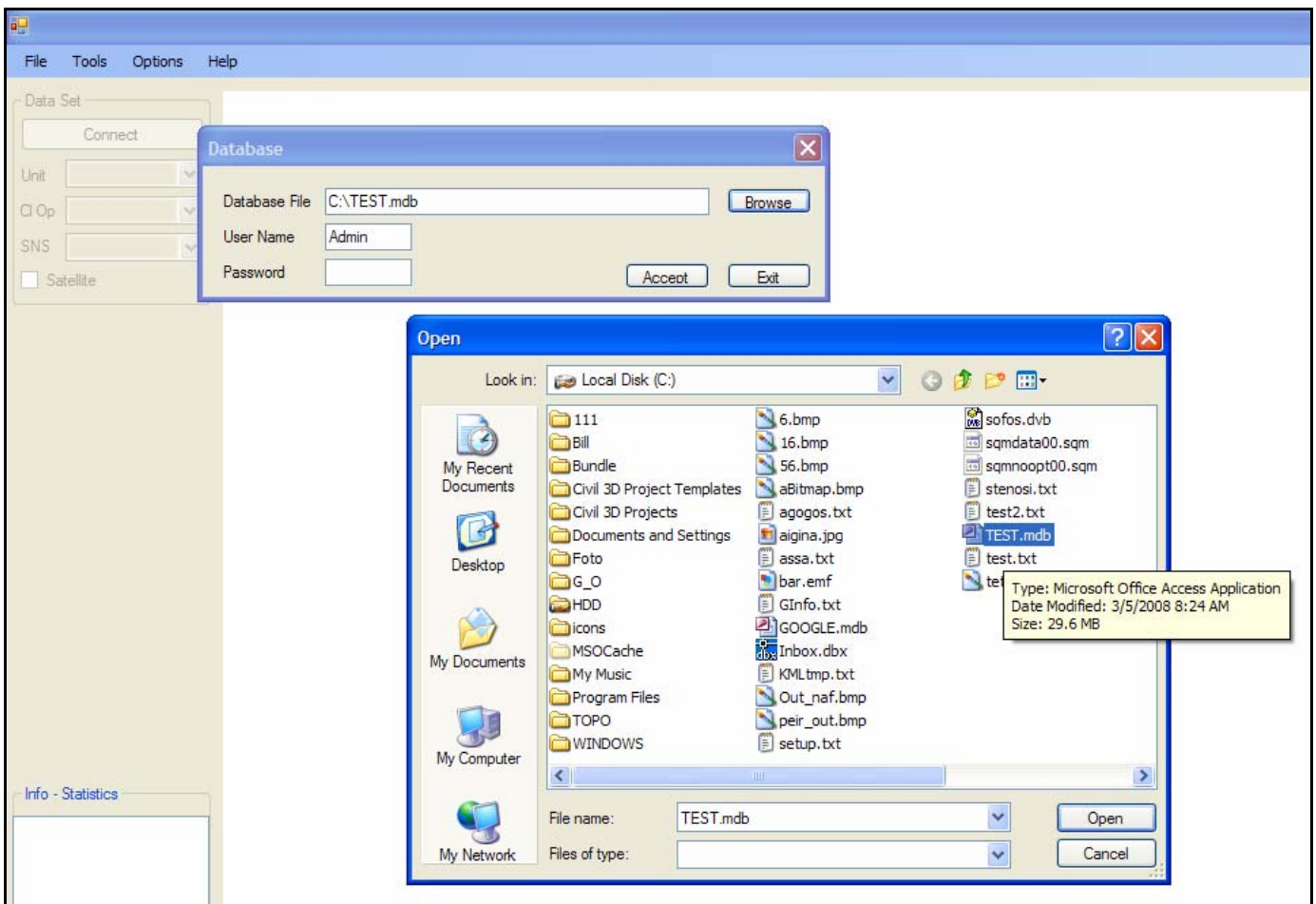
(εικόνα 5.6)

ΚΕΦΑΛΑΙΟ 6

Τρόπος χρήσης του λογισμικού

6.1 Σύνδεση με τη Βάση Δεδομένων

Η πρώτη ενέργεια που πρέπει να κάνει ο χρήστης είναι να συνδεθεί με τη βάση δεδομένων. Για να γίνει αυτό από το Menu File, επιλέγει Open Database. Έτσι εμφανίζεται το παρακάτω περιβάλλον όπου καλείται να επιλέξει το αρχείο της Βάσης καθώς και να εισάγει το User Name και Password που του δίνουν δικαιώματα πρόσβασης.

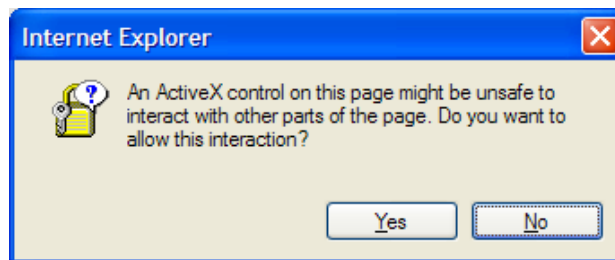


(εικόνα 6.1)

Αφού ολοκληρωθεί η διαδικασία, ενεργοποιείται το κουμπί Connect. Πατώντας το ολοκληρώνεται η διαδικασία σύνδεσης με τη Βάση δεδομένων.

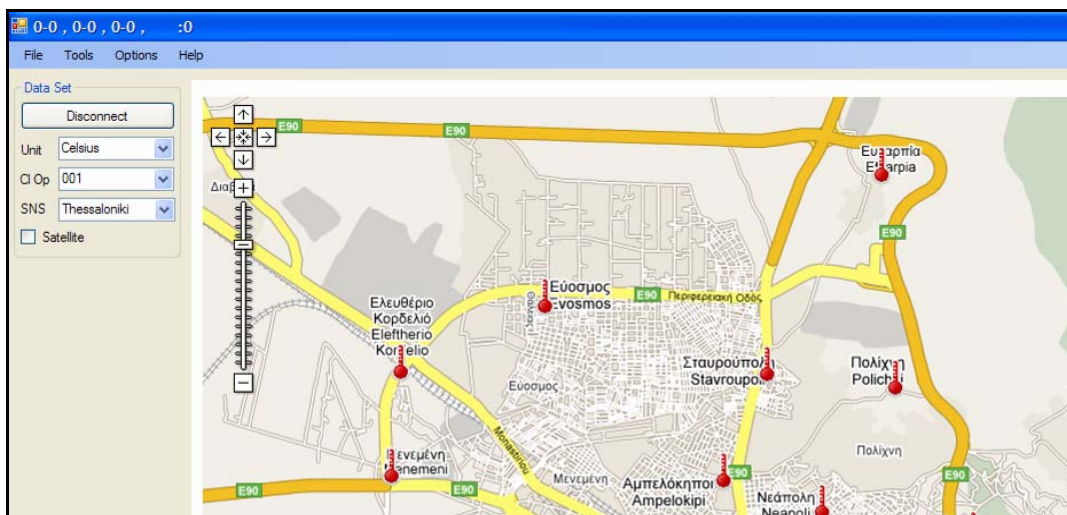
6.2 Απεικόνιση Αισθητήρων σε χάρτη

Αφού έχει ολοκληρωθεί η διαδικασία σύνδεσης με τη Βάση δεδομένων, κάτω αριστερά εμφανίζεται η ένδειξη που μας πληροφορεί για την επιτυχή σύνδεση. Στη συνέχεια έχουμε τρία Combo επιλογής για να προσδιορίσουμε τις μετρήσεις που θέλουμε να απεικονίσουμε. Από το πρώτο Combo επιλέγουμε το είδος των μετρήσεων (Unit) που θέλουμε να απεικονίσουμε. Στη δεύτερη επιλογή έχουμε το Client Operator (CI Op) και στη Τρίτη το Segment (SNS). Αρχικά εμφανίζεται ένα προειδοποιητικό μήνυμα.



(εικόνα 6.2)

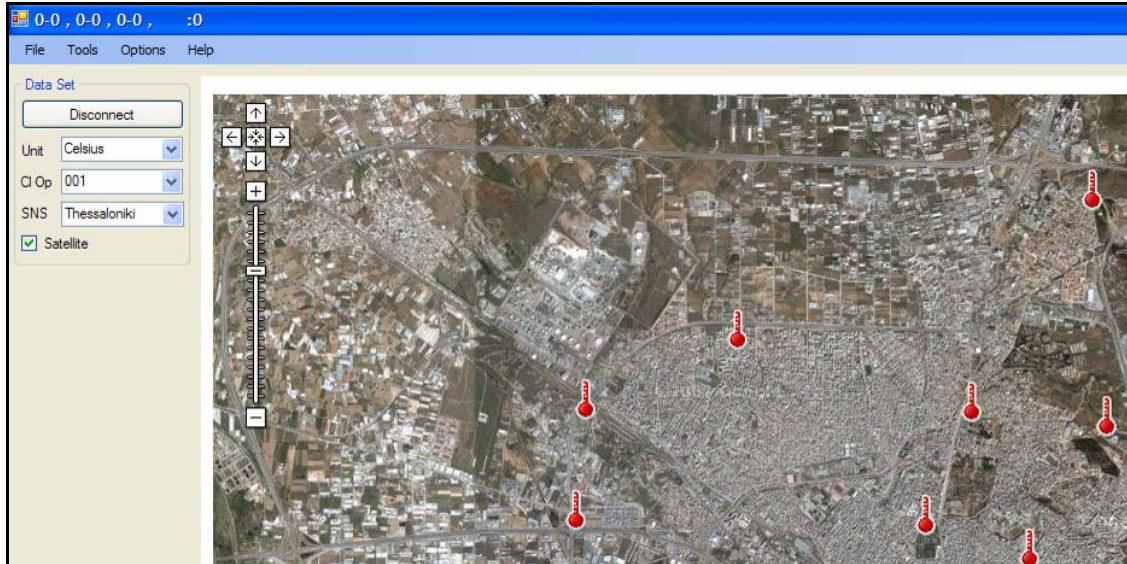
Αυτό γίνεται γιατί η JavaScript καλεί ActiveX Objects και ο Browser ενημερώνει το χρήστη για την επικείμενη πρόσβαση στο σύστημα αρχείων, επειδή η JavaScript θα δημιουργήσει ένα αρχείο. Το αρχείο που δημιουργείται είναι το «GInfo.txt» που υλοποιεί τη σύνδεση JavaScript και Visual Basic. Αφού πατήσουμε OK έχουμε το χάρτη με την απεικόνιση.



(εικόνα 6.3)

Εφαρμογή απεικόνισης μετρήσεων σε Google Maps

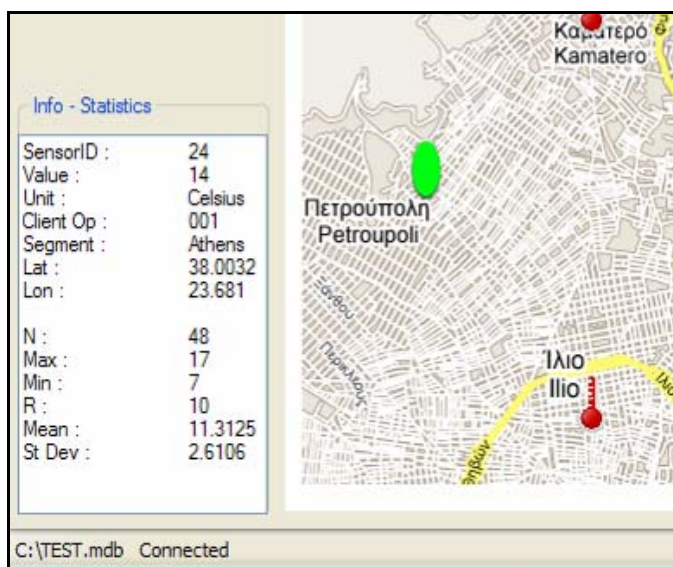
Αν επιλέξουμε το CheckBox με την ένδειξη Satellite, εμφανίζεται ως υπόβαθρο η δορυφορική εικόνα.



(εικόνα 6.4)

6.3 Στατιστικά

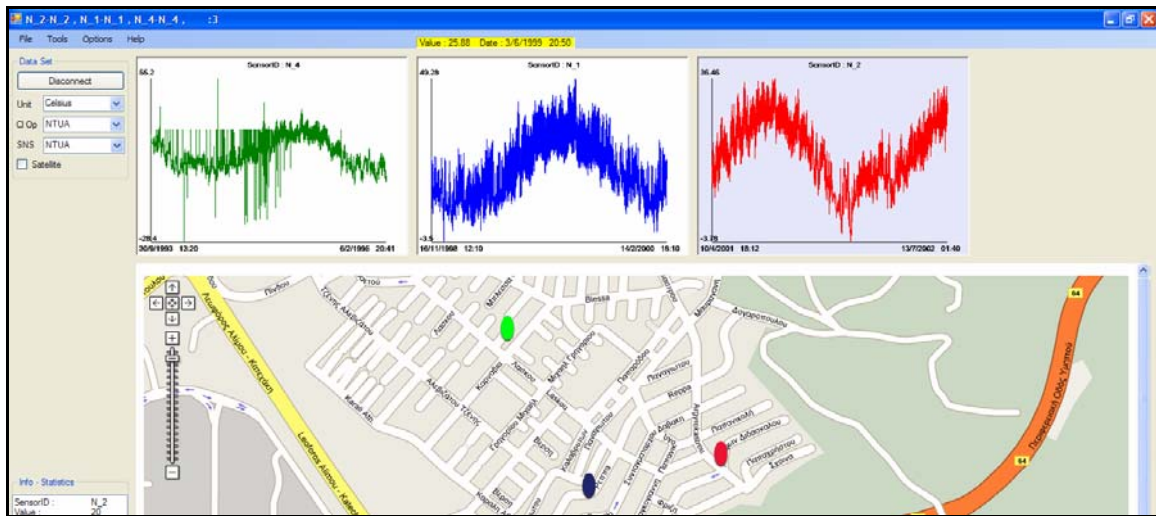
Επιλέγοντας έναν αισθητήρα, αλλάζει το εικονίδιο του σε έλλειψη. Το χρώμα της έλλειψης αντιστοιχεί στο χρώμα του γραφήματος. Αφού επιλέξουμε έναν αισθητήρα, στο πλαίσιο κάτω αριστερά εμφανίζονται στατιστικά στοιχεία για τη σειρά μετρήσεων που υπάρχουν στη βάση.



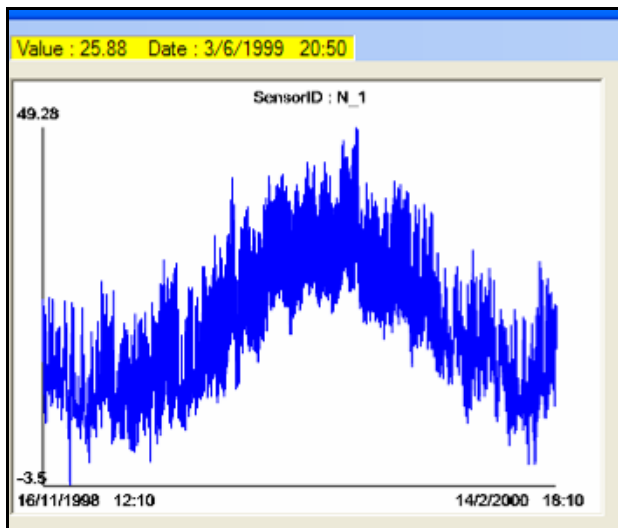
(εικόνα 6.5)

6.4 Γραφήματα

Η εφαρμογή μας δίνει τη δυνατότητα να δούμε σε μορφή γραφήματος τις διαθέσιμες μετρήσεις. Από το Menu Tools, επιλέγουμε Graph και περιορίζεται η διαθέσιμη επιφάνεια του χάρτη ώστε να εμφανιστούν τα διαγράμματα. Πηγαίνοντας το mouse cursor σε κάποια θέση του διαγράμματος, βλέπουμε τη μέτρηση και το χρόνο που την πήραμε.



(εικόνα 6.6)



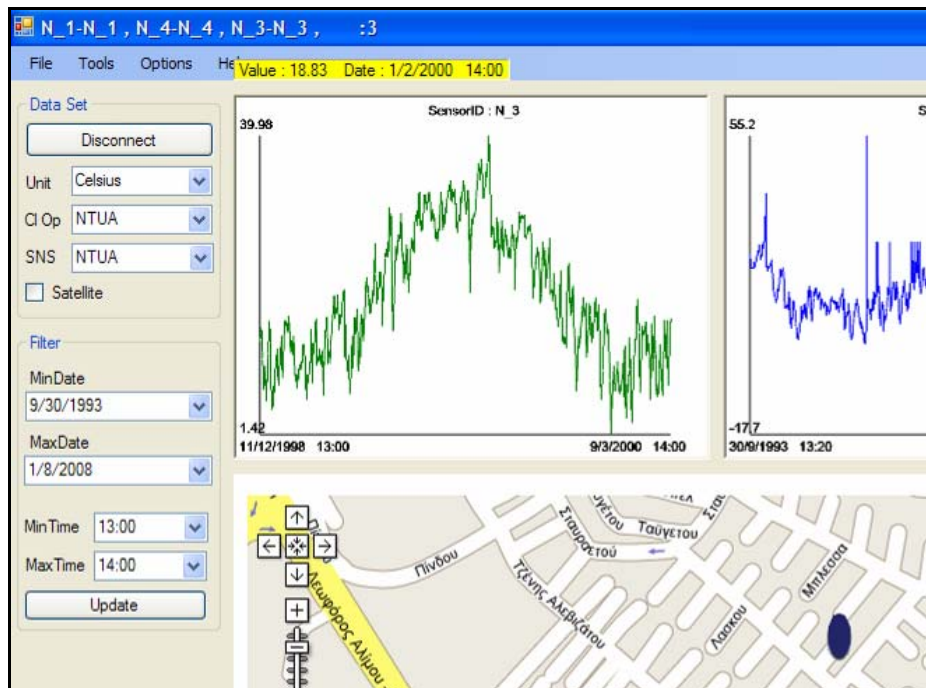
(εικόνα 6.7)

6.5 Φίλτρα

Αρκετές φορές οι μετρήσεις που αναφέρονται στον επιλεγμένο αισθητήρα είναι πολύ πυκνές ή αναφέρονται σε μεγάλο χρονικό διάστημα. Έτσι προκύπτει η ανάγκη φιλτραρίσματος της διαθέσιμης πληροφορίας σε δύο επίπεδα :

1. Περιορισμός της χρονικής περιόδου στην οποία αναφέρονται οι μετρήσεις που ζητάμε να πάρουμε.
2. Περιορισμός σε κάποιο μέρος της ημέρας.

Για παράδειγμα, μπορεί να ζητάμε τις μεσημεριανές (13:00 – 14:00) μετρήσεις για μια χρονική περίοδο. Η εφαρμογή δίνει αυτή τη δυνατότητα μέσα από τα φίλτρα. Από το Menu Tools επιλέγουμε Filter και έτσι έχουμε πρόσβαση στα φίλτρα.



(εικόνα 6.8)

ΚΕΦΑΛΑΙΟ 7

Δυνατότητες αξιοποίησης του λογισμικού

7.1 Πιθανές εφαρμογές

Η δημιουργία μιας λογισμικής μονάδας, έχει νόημα όταν καλύπτει κάποιες ανάγκες ή εν δυνάμει ανάγκες. Έτσι λοιπόν και η παρούσα, μπορεί να καλύψει ζητούμενες διεργασίες στα πλαίσια εφαρμογών που διαχειρίζονται μετρήσεις. Το πρόγραμμα είναι ανεξάρτητο από το είδος των μετρήσεων και έτσι μπορεί να διαχειριστεί κάθε είδους δεδομένα. Συνεπώς προσφέρεται ευελιξία και δυνατότητα χρήσης που περιορίζεται μόνο από την ύπαρξη της κατάλληλης βάσης δεδομένων. Οι εφαρμογές που μπορούν να αξιοποιήσουν το λογισμικό μας, διακρίνονται σε δύο βασικές κατηγορίες.

1. Εφαρμογές για εσωτερική χρήση σε εργαστήρια τα οποία θέλουν να οπτικοποιήσουν τα διαθέσιμα δεδομένα τόσο όσο αφορά τη γεωγραφική τους αναφορά, αλλά και τη συμπεριφορά των μετρήσεων στο πεδίο του χρόνου, καθώς με τα γραφήματα μπορούμε να έχουμε εικόνα των χρονοσειρών που διαμορφώνουν οι μετρήσεις. Πιθανές εφαρμογές μπορούν να έχουν κατεύθυνση προς:

- Παρακολούθηση περιβαλλοντικών μεγεθών (επίπεδα ατμοσφαιρικών ρύπων) σε περιοχές ειδικού ενδιαφέροντος. Μπορούμε να παρακολουθούμε μετρήσεις και να έχουμε εικόνα σε πραγματικό χρόνο για την κατάσταση που επικρατεί στην περιοχή.
- Παρακολούθηση θερμοκρασιών σε περιοχές υψηλού κινδύνου για ανίχνευση πιθανής εκδήλωσης πυρκαγιάς. Με τα γραφήματα μπορούμε να έχουμε εικόνα για την περιοδικότητα των θερμοκρασιών και τη συμπεριφορά ανάλογα με την εποχή, ώστε κέντρα προστασίας να μπορούν εκτιμήσουν κατάσταση που χρήζει άμεσης επέμβασης.
- Παρακολούθηση βροχόπτωσης και συνεκτίμηση με το ύψος ροής ποταμών για εκτίμηση ενδεχόμενης υπερχειλίσης.
- Απεικόνιση οικονομικών μεγεθών σε επίπεδο ΔΟΥ με σκοπό τη σύνδεση δεικτών με την χωρική κατανομή της οικονομικής δραστηριότητας ανά τομέα παραγωγής. Με τη χρήση των γραφημάτων μπορεί να εκτιμηθεί η διαμορφωμένη τάση των οικονομικών δραστηριοτήτων ανά περιοχή.
- Απεικόνιση μεγεθών και δεικτών αμιγώς οικονομικού χαρακτήρα αλλά και δεικτών παραγωγής, ως ένα εργαλείο που συμβάλλει στη λήψη αποφάσεων στα πλαίσια άσκησης χωροταξίας.

- Παρακολούθηση σεισμικής δραστηριότητας όπου υπάρχουν σταθμοί καταγραφής.
- Χρήση από φορείς που ασχολούνται με ανάλυση στατιστικών μεγεθών, ως ένα εργαλείο άμεσης οπτικοποίησης των μεγεθών που πραγματεύονται.

2. Η δεύτερη κατηγορία εφαρμογών που μπορεί να χρησιμοποιήσει το λογισμικό, αναφέρεται σε υπηρεσίες που προσφέρονται σε χρήστες τρίτους προς τους φορείς που έχουν και διαχειρίζονται τα δεδομένα. Είτε πρόκειται για περιορισμένη πρόσβαση σε συγκεκριμένες ομάδες (ερευνητές, συνεργαζόμενα εργαστήρια), είτε για ανοιχτά προς το ευρύ κοινό δεδομένα, αποτελεί μια δυνατότητα εύκολης πρόσβασης από τον χρήστη χωρίς να είναι απαραίτητο τα έχει μετρήσει ή υπόβαθρα στην κατοχή του. Ουσιαστικά πρόκειται για προσφορά υπηρεσίας που έχει το πλεονέκτημα ότι ακολουθεί μια συγκεκριμένη δομή, και δεν χρειάζεται να έχουμε διαφορετικό περιβάλλον σε κάθε περίπτωση. Έτσι μπορεί ο χρήστης να έχει πρόσβαση με ένα τρόπο σε πολλές πηγές πληροφοριών. Με αυτό τον τρόπο μπορούμε να επιτύχουμε και οικονομία στο στήσιμο ενός μηχανισμού, που αφού θα απευθύνεται σε μεγαλύτερο κοινό χωρίς επιπλέον κόστος, θα είναι αποδοτικότερο. Χρήση κοινών δεδομένων από συνεργαζόμενα εργαστήρια συνεπάγεται μειωμένο κόστος.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- “*Beginning JavaScript With DOM Scripting And Ajax - From Novice To Professional (2006)*”, Christian Heilmann, Apress
- “*SQL Practicalguide for developers*”, Michael D. Donahoo - Gregory J Speegle , Elsevier
- “<http://code.google.com/apis/maps/documentation/reference.html>”
- “http://www.developer.com/java/web/article.php/10935_3528381_1”

ΠΑΡΑΡΤΗΜΑ 1

Πηγαίος Κώδικας

Εφαρμογή απεικόνισης μετρήσεων σε Google Maps

```
Imports System.IO
```

```
Module GENERAL
```

```
Public ProcIndex As Integer  
Public LastProcTime As Date ' krataei tin teleytaia dimiourgia graph  
Public MaxImageFile As String
```

```
Public DataFile As String  
Public Table As String  
Public fCol As String  
Public lCol As String  
Public xCol As String
```

```
Public DataS As DataSet  
Public DataS2 As DataSet
```

```
Public DB_File As String  
Public DB_UserID As String  
Public DB_Password As String  
Public DB_Q1 As String  
Public DB_Q2 As String  
Public GInfo_File As String
```

```
Public MinDate As String  
Public MaxDate As String
```

```
Public G_Key As String
```

```
Public C_f As Double 'Current f  
Public C_l As Double 'Current l  
Public F_f As Double 'Final f  
Public F_l As Double 'Final l  
Public S_f As Double 'Start f  
Public S_l As Double 'Start l
```

```
Public Last_f As Double 'Last f  
Public Last_l As Double 'Last l
```

```
Public Zoom_Level As Integer  
Public ShowBubble As Boolean
```

```
Public S_path As String 'System path
```

```
Public txtData() As TextBox  
Public chkData() As CheckBox
```

```
Public tmpSID() As String 'SID ton apeiknizomenon
```

```
Public Point(,) As Double
```

```
Public P_Client_SNS_Unit(,) As String      ' SensorID,
ClientOperator,Segment,Unit, Date, Time ton points, icon on, icon off
Public SNS_N As Integer
Public SNS() As String
Public Unit_N As Integer
Public Unit(,) As String      'unit, icon path on, icon path off
Public Client_N As Integer
Public Client() As String
Public Coors(,) As Double

Public CurIconON As String
Public CurIconOFF As String
Public Chart_N As Integer      'arithmos diagrammaton pou xorane
Public ChartBox() As PictureBox      'array picture box charts
Public ChartBoxFile() As String
Public CSensorID() As String      'krataei ta Sensord ton charts
Public CurSensorChart As Integer      'trexon arithmos diagrammatos
Public Chart_Stat(,) As Double      'xmax, ymax, xmix, ymin, mean, s0,N

Public Chart_Label(,) As String

Public Chart_Maximized As Boolean

Public Measurements(,) As String
Public Measurements_N As Long

Public Chart_par(,) As Double

Public Graph_Flag As Boolean

Public Sub SaveParam()
Dim fs As New FileStream(S_path & "Param.txt", FileMode.Create,
FileAccess.Write)
    Dim s As New StreamWriter(fs)
    s.WriteLine(G_Key)
    s.WriteLine(DB_Q1)
    s.WriteLine(DB_Q2)
    s.WriteLine(DB_File & ";" & DB_UserID & ";" & DB_Password & ";")
    s.Close()
End Sub

Public Sub SaveData(ByVal Filename As String)

    Dim fs As New FileStream(S_path & Filename, FileMode.Create,
FileAccess.Write)
    Dim s As New StreamWriter(fs)
    s.WriteLine("DataFile : " & DataFile)
    s.WriteLine("MaxImageFile : " & MaxImageFile)
    s.WriteLine("Table : " & Table)
    s.WriteLine("fCol : " & fCol)
    s.WriteLine("lCol : " & lCol)
    s.WriteLine("xCol : " & xCol)
    s.WriteLine("DB_File : " & DB_File)
    s.WriteLine("DB_UserID : " & DB_UserID)
    s.WriteLine("DB_Password : " & DB_Password)
```

```
s.WriteLine("DB_Q1 : " & DB_Q1)
s.WriteLine("DB_Q2 : " & DB_Q2)
s.WriteLine("GInfo_File : " & GInfo_File)
s.WriteLine("MinDate : " & MinDate)
s.WriteLine("MaxDate : " & MaxDate)
s.WriteLine("C_f : " & C_f)
s.WriteLine("C_l : " & C_l)
s.WriteLine("F_f : " & F_f)
s.WriteLine("F_l : " & F_l)
s.WriteLine("S_f : " & S_f)
s.WriteLine("S_l : " & S_l)
s.WriteLine("Last_f : " & Last_f)
s.WriteLine("Last_l : " & Last_l)
s.WriteLine("Zoom_Level : " & Zoom_Level)
s.WriteLine("ShowBubble : " & ShowBubble)
s.WriteLine("S_path : " & S_path)
s.WriteLine("SNS_N : " & SNS_N)
s.WriteLine("Unit_N : " & Unit_N)
s.WriteLine("Client_N : " & Client_N)
s.WriteLine("CurIconON : " & CurIconON)
s.WriteLine("CurIconOFF : " & CurIconOFF)
s.WriteLine("Chart_N : " & Chart_N)
s.WriteLine("CurSensorChart : " & CurSensorChart)
s.WriteLine("Chart_Maximized : " & Chart_Maximized)
s.WriteLine("Measurements_N : " & Measurements_N)
s.WriteLine("Graph_Flag : " & Graph_Flag)
s.WriteLine("S_path : " & S_path)

s.WriteLine()
Dim i As Integer
' For i = 1 To UBound(tmpSID)
' s.WriteLine("tmpSID(" & i & " : " & tmpSID(i))
' Next

For i = 1 To Chart_N
    s.WriteLine("CSensorID(" & i & ") : " & CSensorID(i))
Next

For i = 1 To Chart_N
    s.WriteLine("ChartBoxFile(" & i & ") : " & ChartBoxFile(i))
Next

s.Close()
```

End Sub

Εφαρμογή απεικόνισης μετρήσεων σε Google Maps

```
Public Sub PutStat(ByVal Chart_index As Integer, ByVal Sensor_index As Integer)

    Dim t1 As String
    t1 = ""
    t1 = t1 & "SensorID :" & vbTab & P_Client_SNS_Unit(0,
Sensor_index) & vbCrLf
    t1 = t1 & "Value :" & vbTab & vbTab & Point(1, Sensor_index) &
vbCrLf
    t1 = t1 & "Unit :" & vbTab & vbTab & P_Client_SNS_Unit(3,
Sensor_index) & vbCrLf
    t1 = t1 & "Client Op :" & vbTab & P_Client_SNS_Unit(1,
Sensor_index) & vbCrLf
    t1 = t1 & "Segment :" & vbTab & P_Client_SNS_Unit(2,
Sensor_index) & vbCrLf
    t1 = t1 & "Lat :" & vbTab & vbTab & Math.Round(Point(2,
Sensor_index), 4) & vbCrLf
    t1 = t1 & "Lon :" & vbTab & vbTab & Math.Round(Point(3,
Sensor_index), 4) & vbCrLf
    t1 = t1 & vbCrLf
    t1 = t1 & "N :" & vbTab & vbTab & Chart_Stat(7,
Chart_index).ToString & vbCrLf
    t1 = t1 & "Max :" & vbTab & vbTab & Chart_Stat(2,
Chart_index).ToString & vbCrLf
    t1 = t1 & "Min :" & vbTab & vbTab & Chart_Stat(4,
Chart_index).ToString & vbCrLf
    t1 = t1 & "R :" & vbTab & vbTab & Chart_Stat(2, Chart_index) -
Chart_Stat(4, Chart_index) & vbCrLf
    t1 = t1 & "Mean :" & vbTab & vbTab & Math.Round(Chart_Stat(5,
Chart_index), 4) & vbCrLf
    t1 = t1 & "St Dev :" & vbTab & vbTab & Math.Round(Chart_Stat(6,
Chart_index), 4) & vbCrLf

    'MsgBox(t1)
    Form1.TextBox1.Text = t1

End Sub

Public Sub StartUp()

    ShowBubble = False
    S_path = Application.ExecutablePath
    S_path = Mid(S_path, 1, Len(S_path) - 15)
    GInfo_File = S_path & "GInfo.txt"

    Dim tmp As String

    FileOpen(1, S_path & "Param.txt", OpenMode.Input)
    G_Key = LineInput(1)
    DB_Q1 = LineInput(1)
    DB_Q2 = LineInput(1)
    tmp = LineInput(1)
```

Εφαρμογή απεικόνισης μετρήσεων σε Google Maps

```
DB_File = Mid(tmp, 1, InStr(1, tmp, ";") - 1)
tmp = Mid(tmp, InStr(1, tmp, ";") + 1, Len(tmp) - InStr(1, tmp,
";"))
DB_UserID = Mid(tmp, 1, InStr(1, tmp, ";") - 1)
DB_Password = Mid(tmp, InStr(1, tmp, ";") + 1, Len(tmp) -
InStr(1, tmp, ";") - 1)

FileClose(1)

Form1.WindowState = FormWindowState.Maximized
Form1.WebBrowser1.Left = 160
Form1.WebBrowser1.Top = Form1.MenuStrip1.Bottom + 10
Form1.WebBrowser1.Height = Form1.StatusStrip1.Top -
Form1.MenuStrip1.Bottom - 20
Form1.WebBrowser1.Width = Form1.Right - 160 - 15

Dim CH As Double = 4 * (300 - 10 - Form1.MenuStrip1.Bottom -
10) / 3 + 10
Chart_N = Math.Floor(Form1.WebBrowser1.Width / CH)

Data_Reset()

Form1.GroupBox1.Left = 5
Form1.GroupBox1.Top = Form1.WebBrowser1.Top
Form1.GroupBox3.Left = 5
Form1.GroupBox3.Top = Form1.GroupBox1.Bottom + 10
Form1.GroupBox4.Left = -250
Form1.GroupBox4.Top = Form1.GroupBox3.Top
Form1.GroupBox2.Left = 5
Form1.GroupBox2.Top = Form1.StatusStrip1.Bottom - 30 -
Form1.GroupBox2.Height

Form1.GroupBox3.Visible = False

Form1.PictureBox1.Top = -500
Form1.Label6.Top = -500

Dim i, j As Integer
For i = 0 To 23
    For j = 0 To 59 Step 15
        Form1.ComboBox6.Items.Add(Add0to1Dig(CStr(i)) & ":" &
Add0to1Dig(CStr(j)))
        Form1.ComboBox7.Items.Add(Add0to1Dig(CStr(i)) & ":" &
Add0to1Dig(CStr(j)))
    Next
Next
Form1.ComboBox6.Items.Add("23:59")
Form1.ComboBox7.Items.Add("23:59")
Form1.ComboBox6.Text = "00:00"
Form1.ComboBox7.Text = "23:59"

Graph_Flag = False
End Sub
```

```
Public Sub Data_Reset()  
  
    DB_Connected = False  
  
    ReDim Point(4, 1)  
    ReDim Coors(4, 1)  
    ReDim Measurements(9, 1)  
    Measurements_N = 0  
    Point(0, 0) = 0  
    Coors(0, 0) = 0  
    Chart_Maximized = False  
  
    ReDim P_Client_SNS_Unit(7, 1)  
    SNS_N = 0  
    ReDim SNS(1)  
    Unit_N = 0  
    ReDim Unit(3, 1)  
    Client_N = 0  
    ReDim Client(1)  
    ReDim Coors(3, 1)  
  
    CurSensorChart = -1  
  
    ReDim ChartBox(Chart_N)  
    ReDim ChartBoxFile(Chart_N)  
    ReDim CSensorID(Chart_N)  
  
    ReDim Chart_Stat(7, Chart_N)  
  
  
End Sub  
  
  
Public Function ReplaceStr(ByVal StrIn As String, ByVal strOld As  
String, ByVal StrNew As String) As String  
  
    Dim i As Integer  
    Dim OUT, tmp1, tmp2 As String  
    OUT = StrIn  
    i = InStr(1, OUT, strOld)  
    Do While i > 0  
        tmp1 = Mid(OUT, 1, i - 1)  
        tmp2 = Mid(OUT, i + Len(strOld), Len(OUT) - i - Len(strOld)  
+ 1)  
        OUT = tmp1 & StrNew & tmp2  
        i = InStr(i + Len(StrNew), OUT, strOld)  
  
    Loop  
  
    ReplaceStr = OUT  
  
End Function
```

```
Public Function Add0to1Dig(ByVal Instr As String) As String
    If Len(Instr) = 1 Then
        Add0to1Dig = "0" & Instr
    Else
        Add0to1Dig = Instr
    End If
End Function

Public Function cDStr(ByVal input As Double) As String

    Dim out As String
    out = CStr(input)
    Dim i1, i2 As Integer
    i1 = Len(out)
    i2 = Instr(out, ",")
    If i2 > 0 Then
        cDStr = Mid(out, 1, i2 - 1) & "." & Mid(out, i2 + 1, i1 -
i2)
    Else
        cDStr = out
    End If

End Function

End Module
```

```
Imports System.IO

Imports System.Runtime.InteropServices
Imports mshtml
Imports System.Drawing
Imports System.Drawing.Imaging
Imports System

Module JSCRIPT

    Public Function CreateJScript3(ByVal sat As Boolean, ByVal Point(,)
    As Double, ByVal P_Client_SNS_Unit(,) As String, ByVal Client As
    String, ByVal SNS As String, ByVal Unit As String, ByVal Coors(,) As
    Double, ByVal chart_N As Integer, ByVal IconONFile As String, ByVal
    IconOFFFile As String) As String

        Form1.Timer2.Enabled = False
        System.IO.File.Delete(S_path & "GInfo.txt")

        Form1.savedIndex += 1
        SaveData(Form1.savedIndex & ".txt")

        Measurements_N = 0
        Dim ff, jj As Integer
        For ff = 1 To chart_N
            ChartBox(ff).Image.Dispose()
            ChartBox(ff).Visible = False
        Next

        For ff = 1 To chart_N
            For jj = 1 To 1024
                Chart_Label(jj, ff) = ""
            Next
        Next

        For ff = 1 To chart_N

            ChartBoxFile(ff) = S_path & "null_" & ff & ".bmp"
            ChartBox(ff).Image = Image.FromFile(ChartBoxFile(ff))
            ChartBox(ff).SizeMode = PictureBoxSizeMode.StretchImage
            ChartBox(ff).BackColor = Color.Empty
            ChartBox(ff).Visible = True

        End For

    End Function

End Module
```

```
Next

CurSensorChart = -1

ReDim CSensorID(chart_N)

Dim fs As New FileStream(S_path & "SJ_file.htm",
FileMode.Create, FileAccess.Write)
Dim s As New StreamWriter(fs)

Dim minf, minl, maxf, maxl As Double
Dim i As Integer
minf = 500
minl = 500
maxf = -500
maxl = -500

For i = 1 To Point(0, 0)
    If P_Client_SNS_Unit(1, i) = Client And
P_Client_SNS_Unit(2, i) = SNS And P_Client_SNS_Unit(3, i) = Unit Then
        If Point(2, i) > maxf Then maxf = Point(2, i)
        If Point(2, i) < minf Then minf = Point(2, i)
        If Point(3, i) > maxl Then maxl = Point(3, i)
        If Point(3, i) < minl Then minl = Point(3, i)
    End If
Next

Dim Jout As String
Jout = ""

Jout = Jout & "<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN""
Jout = Jout & vbCrLf
Jout = Jout & "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd" > "
Jout = Jout & vbCrLf
Jout = Jout & "<html xmlns=""http://www.w3.org/1999/xhtml"">"
Jout = Jout & vbCrLf
Jout = Jout & "<head>"
Jout = Jout & vbCrLf
Jout = Jout & "<meta http-equiv=""content-type""
content=""text/html; charset=utf-8""/>"
Jout = Jout & vbCrLf
Jout = Jout & "<title>SOFOS IOANNHS</title>"
Jout = Jout & vbCrLf
Jout = Jout & "<script
src=""http://maps.google.com/maps?file=api&v=2&key="" & G_Key
& """"
Jout = Jout & vbCrLf
Jout = Jout & "type=""text/javascript""></script>"
Jout = Jout & vbCrLf
Jout = Jout & "<script type=""text/javascript"">"
Jout = Jout & vbCrLf
```

```
Jout = Jout & "//<![CDATA["
Jout = Jout & vbCrLf
Jout = Jout & "function load() {"
Jout = Jout & vbCrLf
Jout = Jout & "if (GBrowserIsCompatible()) {"
Jout = Jout & vbCrLf
Jout = Jout & "var map = new
GMap2(document.getElementById("map"));
Jout = Jout & vbCrLf
Jout = Jout & "var markers = new Array();" & vbCrLf
Jout = Jout & vbCrLf & vbCrLf
Jout = Jout & "var bounds = new GLatLngBounds();"
Jout = Jout & vbCrLf
Jout = Jout & "bounds.extend(new GLatLng(" & cDStr(minf) & ", "
& cDStr(minl) & "));"
Jout = Jout & vbCrLf
Jout = Jout & "bounds.extend(new GLatLng(" & cDStr(maxf) & ", "
& cDStr(maxl) & "));"
Jout = Jout & vbCrLf
Jout = Jout & vbCrLf
Jout = Jout &
"map.setCenter(bounds.getCenter(),map.getBoundsZoomLevel(bounds));"
Jout = Jout & vbCrLf
Jout = Jout & "map.addControl(new GLargeMapControl());"
Jout = Jout & vbCrLf & vbCrLf

Dim Ginfo2 As String
Ginfo2 = ReplaceStr(GInfo_File, "\", "\\")

Jout = Jout & "function WriteFile()" & vbCrLf
Jout = Jout & "var fso = new
ActiveXObject("Scripting.FileSystemObject");" & vbCrLf
Jout = Jout & "var s = fso.CreateTextFile("" & Ginfo2 & "",
true);" & vbCrLf
Jout = Jout & "s.WriteLine("Selected Sensors
(Index;SensorID;ColorUrl)");" & vbCrLf
Jout = Jout & "var k=0;" & vbCrLf
Jout = Jout & "for (k=1;k<=" & CStr(chart_N) & ";k++){" &
vbCrLf
Jout = Jout &
"s.WriteLine(CurSel[k]+";"+SelectedSensorID[CurSel[k]]+";"+Colors[S
elColors[k]]);}" & vbCrLf
Jout = Jout & "s.Close();" & vbCrLf
Jout = Jout & "}"
Jout = Jout & vbCrLf

Jout = Jout & "function RefreshIcons()" & vbCrLf
Jout = Jout & "for (k=1;k<=index;k++){" & vbCrLf
Jout = Jout &
"markers[k].setImage("http://users.ntua.gr/rs03015/icons/null.png");"
" & vbCrLf
```

```
Jout = Jout & "for (k=1;k<=" & CStr(chart_N) & ";k++){" &
vbCrLf
Jout = Jout & "if
(CurSel[k]>0){markers[CurSel[k]].setImage(Colors[SelColors[k]]);}" &
vbCrLf
Jout = Jout & vbCrLf

Jout = Jout & "var baseIconON = new GIcon();" & vbCrLf
Jout = Jout & "baseIconON.shadow = "" & IconONFile & "";" &
vbCrLf
Jout = Jout & "baseIconON.image = "" & IconONFile & "";" &
vbCrLf
Jout = Jout & "baseIconON.iconSize = new GSize(35/2, 70/2);" &
vbCrLf
Jout = Jout & "baseIconON.shadowSize = new GSize(35/2, 70/2);"
& vbCrLf
& vbCrLf
Jout = Jout & "baseIconON.iconAnchor = new GPoint(9/2, 34/2);"
& vbCrLf
Jout = Jout & "baseIconON.infoWindowAnchor = new GPoint(9/2,
2/2);" & vbCrLf
Jout = Jout & "baseIconON.infoShadowAnchor = new GPoint(18/2,
25/2);" & vbCrLf
Jout = Jout & "var FinalIconON = new GIcon(baseIconON);" &
vbCrLf
Jout = Jout & "markerOptionsON = { icon:FinalIconON };"
& vbCrLf
Jout = Jout & vbCrLf

Jout = Jout & vbCrLf
Jout = Jout & "var Colors = new Array();" & vbCrLf

Jout = Jout & "Colors[1] = "icons\\Green.png";" & vbCrLf
Jout = Jout & "Colors[3] = "icons\\Blue.png";" & vbCrLf
Jout = Jout & "Colors[2] = "icons\\Red.png";" & vbCrLf
Jout = Jout & "Colors[6] = "icons\\Yellow.png";" & vbCrLf
Jout = Jout & "Colors[5] = "icons\\Brown.png";" & vbCrLf
Jout = Jout & "Colors[4] = "icons\\Purple.png";" & vbCrLf
Jout = Jout & "Colors[7] = "icons\\Black.png";" & vbCrLf
Jout = Jout & "Colors[8] = "icons\\Galazio.png";" & vbCrLf
Jout = Jout & vbCrLf
Jout = Jout & "var SelColors = new Array();" & vbCrLf
Jout = Jout & "var CurColor = 0;" & vbCrLf
Jout = Jout & vbCrLf
Jout = Jout & "var CurSel = new Array();" & vbCrLf
Jout = Jout & "var i=0;" & vbCrLf
```



```
Jout = Jout & "for (i=1;i<=" & CStr(chart_N) & ";i++){" &
vbCrLf
Jout = Jout & "CurSel[i]=0;" & vbCrLf
Jout = Jout & "SelColors[i]=0;}" & vbCrLf
Jout = Jout & vbCrLf

Jout = Jout & "var Selected = new Array();" & vbCrLf
Jout = Jout & "var SelectedSensorID = new Array();" & vbCrLf
Jout = Jout & "var index = 0;" & vbCrLf & vbCrLf
Jout = Jout & vbCrLf
Jout = Jout & "SelectedSensorID[0] = 0;" & vbCrLf & vbCrLf

Jout = Jout & "WriteFile();" & vbCrLf & vbCrLf

Jout = Jout & "function createInfoMarkerON(point, address,
indexx) {"
Jout = Jout & vbCrLf
    Jout = Jout & "var marker = new GMarker(point,
markerOptionsON);"
Jout = Jout & vbCrLf
Jout = Jout & "GEvent.addListener(marker, "click", "
Jout = Jout & vbCrLf
Jout = Jout & "function() {"
Jout = Jout & vbCrLf

Jout = Jout &
"if(address!="") {marker.openInfoWindowHtml(address);}"
Jout = Jout & vbCrLf
Jout = Jout & "var same=1;" & vbCrLf      'metavliti pou exetazei
an exei epilogi
Jout = Jout & "var k=0;" & vbCrLf
Jout = Jout & "for (k=" & CStr(chart_N) & ";k>0;k--){" & vbCrLf
Jout = Jout & vbTab & "if (CurSel[k]==indexx){same=0}" &
vbCrLf
Jout = Jout & vbCrLf
Jout = Jout & "if (same==1){" & vbCrLf
Jout = Jout & "for (k=" & CStr(chart_N) & ";k>=2;k--){" &
vbCrLf
Jout = Jout & vbTab & "CurSel[k] = CurSel[k-1];" & vbCrLf
Jout = Jout & vbTab & "SelColors[k] = SelColors[k-1];}" &
vbCrLf
Jout = Jout & "if (CurColor==" & CStr(chart_N) & "){" & vbCrLf
Jout = Jout & vbTab & "CurColor=1;}" & vbCrLf
Jout = Jout & vbTab & "else {CurColor+=1;}" & vbCrLf & vbCrLf
Jout = Jout & "SelColors[1]=CurColor;" & vbCrLf
Jout = Jout & "CurSel[1]=indexx;" & vbCrLf & vbCrLf
Jout = Jout & "RefreshIcons();" & vbCrLf

Jout = Jout & "WriteFile();" & vbCrLf
Jout = Jout & vbCrLf
Jout = Jout & "}"
Jout = Jout & vbCrLf
Jout = Jout & ");"
Jout = Jout & vbCrLf
Jout = Jout & "return marker;"
Jout = Jout & vbCrLf
```

```
Jout = Jout & "}"
Jout = Jout & vbCrLf & vbCrLf
Jout = Jout & vbCrLf

Jout = Jout & "function createInfoMarkerOFF(point, address,
indexx) {"
Jout = Jout & vbCrLf
Jout = Jout & "var marker = new GMarker(point,
markerOptionsOFF);"
Jout = Jout & vbCrLf
Jout = Jout & "GEvent.addListener(marker, "click", "
Jout = Jout & vbCrLf
Jout = Jout & "function() {"
Jout = Jout & vbCrLf

Jout = Jout &
"if(address!="") {marker.openInfoWindowHtml(address);}"
Jout = Jout & vbCrLf
Jout = Jout & "var k=0;" & vbCrLf
Jout = Jout & "for (k=" & CStr(chart_N) & ";k>=2;k--){" &
vbCrLf
Jout = Jout & "CurSel[k] = CurSel[k-1];" & vbCrLf
Jout = Jout & "SelColors[k] = SelColors[k-1];}" & vbCrLf
Jout = Jout & "if (CurColor==" & CStr(chart_N) & "){" & vbCrLf
Jout = Jout & "CurColor=1;}" & vbCrLf
Jout = Jout & "else {CurColor+=1;}" & vbCrLf & vbCrLf
Jout = Jout & "SelColors[1]=CurColor;" & vbCrLf
Jout = Jout & "CurSel[1]=indexx;" & vbCrLf & vbCrLf

Jout = Jout & "RefreshIcons();" & vbCrLf

Jout = Jout & "WriteFile();" & vbCrLf
Jout = Jout & vbCrLf
Jout = Jout & "}"
Jout = Jout & vbCrLf
Jout = Jout & ");"
Jout = Jout & vbCrLf
Jout = Jout & "return marker;"
Jout = Jout & vbCrLf
Jout = Jout & "}"
Jout = Jout & vbCrLf & vbCrLf

s.WriteLine(Jout)
Jout = ""

Dim PolNum As Integer
Dim CurPolNum As Integer
PolNum = Coors(3, Coors(0, 0))
CurPolNum = 0
```

```
For i = 1 To Coors(0, 0)
    Form1.Text = i & "/" & Coors(0, 0)
    If CurPolNum = Coors(3, i) Then
        Jout = Jout & "," & vbCrLf & "new
GLatLng(" & CStr(Coors(1, i)) & ", " & CStr(Coors(2, i)) & ") "
    Else
        If CurPolNum <> 0 Then
            Jout = Jout & vbCrLf & "], 'ff0000', 1);"
            Jout = Jout & vbCrLf
            Jout = Jout & "map.addOverlay(polyline_" &
CStr(CurPolNum) & ");"
            Jout = Jout & vbCrLf & vbCrLf
        End If
        CurPolNum = CurPolNum + 1
        Jout = Jout & "var polyline_" & CStr(CurPolNum) & " =
new GPolyline([" & vbCrLf
            Jout = Jout & "new GLatLng(" & CStr(Coors(1, i)) & ", "
& CStr(Coors(2, i)) & ") "
        End If

        If i = Coors(0, 0) Then
            Jout = Jout & vbCrLf & "], 'ff0000', 1);"
            Jout = Jout & vbCrLf
            Jout = Jout & "map.addOverlay(polyline_" &
CStr(CurPolNum) & ");"
            Jout = Jout & vbCrLf & vbCrLf
        End If
        s.WriteLine(Jout)
        Jout = ""

Next i
Jout = Jout & vbCrLf
```

```
s.WriteLine(Jout)
Jout = ""
```

```
Dim tmpBox As String
```

```
For i = 1 To Point(0, 0)
    If P_Client_SNS_Unit(1, i) = Client And
P_Client_SNS_Unit(2, i) = SNS And P_Client_SNS_Unit(3, i) = Unit Then
        Jout = Jout & "var point = new GLatLng(" &
cDStr(Point(2, i)) & "," & cDStr(Point(3, i)) & ");"
        Jout = Jout & vbCrLf

        tmpBox = ""
        If ShowBubble = True Then
            If Form1.CheckBox2.Checked = True Then tmpBox =
tmpBox & "Value .: " & cDStr(Point(1, i)) & " " &
P_Client_SNS_Unit(3, i) & "<br/>"
            If Form1.CheckBox3.Checked = True Then tmpBox =
tmpBox & "Date ...: " & P_Client_SNS_Unit(4, i) & "<br/>"
            If Form1.CheckBox4.Checked = True Then tmpBox =
tmpBox & "Time ...: " & P_Client_SNS_Unit(5, i) & "<br/>"
            If Form1.CheckBox5.Checked = True Then tmpBox =
tmpBox & "Segment ....: " & P_Client_SNS_Unit(2, i) & "<br/>"
            If Form1.CheckBox6.Checked = True Then tmpBox =
tmpBox & "Op. Client .: " & P_Client_SNS_Unit(1, i) & "<br/>"
            End If
            Jout = Jout & "Temp = '" & tmpBox & "';"
            Jout = Jout & vbCrLf
            Jout = Jout & "index = index + 1;"
            Jout = Jout & vbCrLf
            Jout = Jout & "Selected[index] = 0;"
            Jout = Jout & vbCrLf
            Jout = Jout & "SelectedSensorID[index] = '" &
P_Client_SNS_Unit(0, i) & "';"
            Jout = Jout & vbCrLf
            Jout = Jout & "markers[index] =
createInfoMarkerON(point, Temp, index);"
            Jout = Jout & vbCrLf
            Jout = Jout & "map.addOverlay(markers[index]);"

            Jout = Jout & vbCrLf & vbCrLf

            s.WriteLine(Jout)
            Jout = ""
        End If
    Next i
    Jout = Jout & vbCrLf

    If sat = True Then
        Jout = Jout & "map.setMapType(G_SATELLITE_MAP)"
    Else
        Jout = Jout & "map.setMapType(G_NORMAL_MAP)"
    End If

    Jout = Jout & vbCrLf
    Jout = Jout & "}"
    Jout = Jout & vbCrLf
    Jout = Jout & "}"
    Jout = Jout & vbCrLf
    Jout = Jout & "///]>"
```

```
Jout = Jout & vbCrLf
Jout = Jout & "</script>"
Jout = Jout & vbCrLf
Jout = Jout & "</head>"
Jout = Jout & vbCrLf
Jout = Jout & "<body onload=""load()"" onunload=""GUnload()"">"
Jout = Jout & vbCrLf
Jout = Jout & "<div id=""map"" style=""width: " &
Math.Round(Form1.WebBrowser1.Width).ToString & "; height: " &
(Math.Round((Form1.StatusStrip1.Top - Form1.MenuStrip1.Bottom - 20) *
25 / 26)).ToString & "px""></div>"
Jout = Jout & vbCrLf
Jout = Jout & "</body>"
Jout = Jout & vbCrLf
Jout = Jout & "</html>"
Jout = Jout & vbCrLf
```

```
s.WriteLine(Jout)
Jout = ""
s.Close()
```

```
Dim fs3 As New FileStream(GInfo_File, FileMode.Create,
FileAccess.Write)
Dim s3 As New StreamWriter(fs3)
s3.WriteLine("Selected Sensors (Index;SensorID;ColorUrl)")
For i = 1 To chart_N
    s3.WriteLine("0;0;0")
Next
s3.Close()
```

```
Form1.Timer2.Enabled = True
CreateJScript3 = S_path & "SJ_file.htm"
```

```
End Function
```

```
End Module
```

Εφαρμογή απεικόνισης μετρήσεων σε Google Maps

```
Imports Microsoft.Data.Odbc
```

```
Module DATA
```

```
    Dim oConn_2 As OdbcConnection
        Dim mystring As String = "Select * from Titles"
    Dim cmd As OdbcCommand = New OdbcCommand(mystring)
    Public oData_2 As Microsoft.Data.Odbc.OdbcDataAdapter

    Public oConn As OleDb.OleDbConnection
    Public oComm As OleDb.OleDbCommand
    Public oData As OleDb.OleDbDataAdapter
    Public DB_Connected As Boolean

    Public Sub DB_Connect(ByVal FileName As String, ByVal UserId As
String, ByVal Password As String)

        Dim Con As String
        Con = "Driver={Microsoft Access Driver (*.mdb)};DBQ=" &
FileName & ";UID=" & UserId & ";PWD=" & Password & ";"

        oConn_2 = New OdbcConnection(con)

        oConn_2.Open()

        DB_Connected = True
    End Sub

    Public Sub DB_Disconnect()
        oConn_2.Close()
        ' oConn.Close()
        DB_Connected = False
    End Sub

    Public Function Get_DB_Data(ByVal QQuery As String) As DataSet
        Dim resultSet As New DataSet

        oData_2 = New Microsoft.Data.Odbc.OdbcDataAdapter(QQuery,
oConn_2)
        'oData = New OleDb.OleDbDataAdapter(QQuery, oConn)
        oData_2.Fill(resultSet, "measurement")
        'oData.Fill(resultSet, "measurement")
        Return resultSet
    End Function

    Public Function Create_DB_Q2(ByVal Dmin As String, ByVal Dmax As
String, ByVal SensorID As String) As String
        Dim Out As String
        Dim i1, i2, i3, i4, i5 As Integer
        i1 = InStr(1, DB_Q2, "***DATE**")
        i2 = InStr(i1 + 1, DB_Q2, "***DATE**")
        i3 = InStr(i2 + 1, DB_Q2, "***SensorID**")
        i4 = InStr(1, DB_Q2, "***TIME**")
```

```
i5 = InStr(i4 + 1, DB_Q2, "***TIME**")
Out = Mid(DB_Q2, 1, i1 - 1)
Out = Out & Dmin
Out = Out & Mid(DB_Q2, i1 + 8, i2 - i1 - 8)
Out = Out & Dmax
Out = Out & Mid(DB_Q2, i2 + 8, i3 - i2 - 8)

Out = Out & " " & SensorID & " "

Out = Out & Mid(DB_Q2, i3 + 12, i4 - i3 - 12)
Out = Out & Form1.ComboBox6.Text
Out = Out & Mid(DB_Q2, i4 + 8, i5 - i4 - 8)
Out = Out & Form1.ComboBox7.Text
Out = Out & Mid(DB_Q2, i5 + 8, Len(DB_Q2) - i5 - 8 + 1)
Create_DB_Q2 = Out
End Function

Public Function FetchData(ByVal FileName As String) As DataSet

    'Define the connectors()

    Dim resultSet As New DataSet
    Dim oConnect, oQuery As String
    'Define connection string
    oConnect = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" &
FileName & ";User ID=Admin;Password="
    'Query String
    oQuery = "SELECT * FROM measurement"
    oQuery = "SELECT Measurement.Value, Sensor.WGS84_Lat,
Sensor.WGS84_Lon FROM Sensor INNER JOIN Measurement ON Sensor.SensorID
= Measurement.SensorID WHERE Measurement.SensorId=Sensor.SensorID"

    'Instantiate the connectors
    oConn = New OleDb.OleDbConnection(oConnect)
    oComm = New OleDb.OleDbCommand(oQuery, oConn)
    oData = New OleDb.OleDbDataAdapter(oQuery, oConn)

    Try 'Open connection
        oConn.Open()
        'Fill DataSet()
        oData.Fill(resultSet, "measurement")
        'Close connection()
        oConn.Close()
    Catch ex As OleDb.OleDbException
    Catch ex As Exception
        'Show error message and exit
        MsgBox(ex.Message & vbCrLf & ex.StackTrace)
        'Finally 'Dispose the connector objects
        If Not (oConn Is Nothing) Then oConn.Dispose()
        If Not (oComm Is Nothing) Then oComm.Dispose()
        If Not (oData Is Nothing) Then oData.Dispose()

    End Try 'Return results
    Return resultSet

End Function
```

```
Public Function FetchData(ByVal FileName As String, ByVal QQuery As
String) As DataSet

    'Define the connectors()

    Dim resultSet As New DataSet
    Dim oConnect, oQuery As String
    'Define connection string
    oConnect = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" &
FileName & ";User ID=Admin;Password="
    'Query String
    oQuery = "SELECT * FROM measurement"
    oQuery = QQuery

    'Instantiate the connectors
    oConn = New OleDb.OleDbConnection(oConnect)
    oComm = New OleDb.OleDbCommand(oQuery, oConn)
    oData = New OleDb.OleDbDataAdapter(oQuery, oConn)

    Try 'Open connection
        oConn.Open()
        'Fill DataSet()
        oData.Fill(resultSet, "measurement")
        'Close connection()
        oConn.Close()
    Catch ex As OleDb.OleDbException
    Catch ex As Exception
        'Show error message and exit
        MsgBox(ex.Message & vbCrLf & ex.StackTrace)
        'Finally 'Dispose the connector objects
        If Not (oConn Is Nothing) Then oConn.Dispose()
        If Not (oComm Is Nothing) Then oComm.Dispose()
        If Not (oData Is Nothing) Then oData.Dispose()

    End Try 'Return results
    Return resultSet

End Function

End Module
```


Εφαρμογή απεικόνισης μετρήσεων σε Google Maps

```
Imports system
Imports System.IO
Module Chart

    Public Structure FileHeader
        Dim FileType As String ' file type always 4D42h or "BM"
        Dim FileSize As Long ' size in bytes usually 0 for
uncompressed
        Dim Res1 As Integer ' always 0
        Dim Res2 As Integer ' always 0
        Dim BitmapOffset As Long ' starting position of image data in
bytes
    End Structure

    Public Sub Create_Chart_32(ByVal FileName As String, ByVal Data(,,)
As Byte, ByVal w As Long, ByVal h As Long)

        Dim fs As New FileStream(FileName, FileMode.OpenOrCreate)
        Dim Bw As New BinaryWriter(fs)
        fs.Position = 0

        'file header, total 14 bytes
        Bw.Write(Asc("B")) 'file type always 4D42h or "BM"
        fs.Position = 1
        Bw.Write(Asc("M"))
        fs.Position = 2
        Bw.Write(CLng(0)) 'size in bytes usually 0 for
uncompressed
        fs.Position = 6
        Bw.Write(CInt(0)) 'always 0
        fs.Position = 8
        Bw.Write(CInt(0)) 'always 0
        fs.Position = 10
        Bw.Write(CLng(14 + 40)) 'starting position of
image data in bytes' '.....'Long

        'image header, total 40 bytes
        fs.Position = 14
        Bw.Write(CLng(40)) 'Size of this header
        fs.Position = 18
        Bw.Write(CLng(w)) 'width of your image
        fs.Position = 22
        Bw.Write(CLng(h)) 'height of your image
        fs.Position = 26
        Bw.Write(CInt(1)) 'always 1
        fs.Position = 28
        Bw.Write(CInt(32)) 'number of bits per pixel 1, 4, 8,
or 24 or 32
        fs.Position = 30
        Bw.Write(CLng(0)) '0 data is not compressed
        fs.Position = 34
```

Εφαρμογή απεικόνισης μετρήσεων σε Google Maps

```
        Bw.Write(CLng(0))           'size of bitmap in bytes, typically
0 when uncompressed
        fs.Position = 38
        Bw.Write(CLng(0))           'preferred resolution in pixels per
meter
        fs.Position = 42
        Bw.Write(CLng(0))           'preferred resolution in pixels per
meter
        fs.Position = 46
        Bw.Write(CLng(0))           'number of colors that are actually
used (can be 0)
        fs.Position = 50
        Bw.Write(CLng(0))           'which color is most important (0 means all
of them)
```

```
Dim i, j As Long
For i = 1 To h
    For j = 1 To w
        Bw.Write(Data(i, j, 1))
        Bw.Write(Data(i, j, 2))
        Bw.Write(Data(i, j, 3))
        Bw.Write(Data(i, j, 3))
    Next
Next
```

```
fs.Close()
```

```
End Sub
```

```
Public Function CreateGraph(ByVal Unit As String, ByVal SensorId As
String, ByVal FileName As String, ByVal MyColor As String, ByVal index
As Integer) As Boolean
```

```
Dim plot1() As Date
Dim plot3() As Double
```

```
Dim n As Long
Dim k As Long
n = 0
```

```
ReDim Preserve Chart_par(4, Chart_N) 'xmin,xo,xbima, xmax
```

```
ReDim Preserve plot3(Measurements_N)
ReDim Preserve plot1(Measurements_N)
```

```
For k = 1 To Measurements_N
    If Measurements(1, k) = SensorId And Measurements(7, k) =
Unit Then
        n += 1
```

```
k)          plot1(n) = Measurements(8, k) & " " & Measurements(9,
          plot3(n) = Measurements(4, k)

          End If
Next
ReDim Preserve plot3(n)
ReDim Preserve plot1(n)

Dim xo As Long
Dim yo As Long
Dim xt As Long
Dim yt As Long
Dim xmin As Date
Dim xmax As Date
Dim ymin As Double
Dim ymax As Double

Dim bimaX As Double
Dim bimaY As Double
Dim w As Integer
Dim h As Integer

w = 1024
h = 768

xo = 50
yo = 50
xt = w - 80
yt = h - 80

If n = 0 Then GoTo NoData

xmin = plot1(1)
xmax = plot1(1)
ymin = plot3(1)
ymax = plot3(1)

For k = 1 To n
    If plot1(k) > xmax Then xmax = plot1(k)
    If plot1(k) < xmin Then xmin = plot1(k)
    If plot3(k) > ymax Then ymax = plot3(k)
    If plot3(k) < ymin Then ymin = plot3(k)
Next

bimaY = (ymax - ymin) / (yt - yo)
bimaX = (xmax - xmin).TotalMinutes / (xt - xo)

Dim plot1_2() As Double
ReDim plot1_2(n)
For k = 1 To n
    plot1_2(k) = (plot1(k) - xmin).TotalMinutes
Next
```

```
Dim tmp As Double
Dim tmpDate As Date
Dim sorted As Boolean
sorted = True
Do While sorted = True
    sorted = False
    For k = n To 2 Step -1
        If plot1_2(k) < plot1_2(k - 1) Then
            tmp = plot1_2(k)
            plot1_2(k) = plot1_2(k - 1)
            plot1_2(k - 1) = tmp
            tmp = plot3(k)
            plot3(k) = plot3(k - 1)
            plot3(k - 1) = tmp
            tmpDate = plot1(k)
            plot1(k) = plot1(k - 1)
            plot1(k - 1) = tmpDate
            sorted = True
        End If
    Next
Loop

Dim data(,) As Single
ReDim data(2, n)

For k = 1 To 1024
    Chart_Label(k, index) = ""
Next
For k = 1 To n
    data(1, k) = CSng(Math.Round((plot1_2(k) - plot1_2(1)) /
bimaX) + xo)
    data(2, k) = h - CSng(Math.Round((plot3(k) - ymin) / bimaY)
+ yo)
    Chart_Label(data(1, k), index) = "Value : " & plot3(k) & "
Date : " & plot1(k).Day.ToString & "/" & plot1(k).Month.ToString & "/"
& plot1(k).Year.ToString & "    " & Add0to1Dig(plot1(k).TimeOfDay.Hours)
& ":" & Add0to1Dig(plot1(k).TimeOfDay.Minutes.ToString)
Next

NoData:
Dim b As Bitmap
Dim g As Graphics
b = New Bitmap(w, h)
g = Graphics.FromImage(b)

Dim chartvalues As ArrayList = New ArrayList

Dim Cpen As Pen
If MyColor = "Green" Then Cpen = New Pen(Color.Green, 3)
If MyColor = "Blue" Then Cpen = New Pen(Color.Blue, 3)
If MyColor = "Red" Then Cpen = New Pen(Color.Red, 3)
If MyColor = "Brown" Then Cpen = New Pen(Color.Brown, 3)
If MyColor = "Black" Then Cpen = New Pen(Color.Black, 3)
If MyColor = "Yellow" Then Cpen = New Pen(Color.Yellow, 3)
```

```
If MyColor = "Galazio" Then Cpen = New Pen(Color.Cyan, 3)
If MyColor = "Purple" Then Cpen = New Pen(Color.Purple, 3)
Dim Cpen2 As Pen = New Pen(Color.Black, 3)

g.DrawLine(Cpen2, xo, h - yo, xo, h - yt)
g.DrawLine(Cpen2, xo, h - yo, xt, h - yo)
Dim axesFontN As Font = New Font("arial", 22)
If n > 0 Then
    For k = 2 To n
        g.DrawLine(Cpen, data(1, k - 1), data(2, k - 1),
data(1, k), data(2, k))
    Next
    g.DrawString(plot1(1).Day.ToString & "/" &
plot1(1).Month.ToString & "/" & plot1(1).Year.ToString & " " &
Add0tolDig(plot1(1).Hour.ToString) & ":" &
Add0tolDig(plot1(1).Minute.ToString), axesFontN, Brushes.Black, data(1,
1) - 50, h - yo + 10)
    g.DrawString(plot1(n).Day.ToString & "/" &
plot1(n).Month.ToString & "/" & plot1(n).Year.ToString & " " &
Add0tolDig(plot1(n).Hour.ToString) & ":" &
Add0tolDig(plot1(n).Minute.ToString), axesFontN, Brushes.Black, data(1,
n) - 180, h - yo + 10)
    g.DrawString(ymin.ToString, axesFontN, Brushes.Black, 1, h
- 80)
    g.DrawString(ymax.ToString, axesFontN, Brushes.Black, 1, h
- (ymax - ymin) / bimaY - 90)
    g.DrawString("SensorID : " & SensorId, axesFontN,
Brushes.Black, w / 2 - 100, 10)
End If

g.Dispose()
b.Dispose()

Chart_par(1, index) = xmin.ToOADate
Chart_par(2, index) = xo
Chart_par(3, index) = bimaX
Chart_par(4, index) = xmax.ToOADate

CreateGraph = True

End Function

Public Function CreateGraphMax(ByVal Unit As String, ByVal SensorId
As String, ByVal FileName As String, ByVal MyColor As String, ByVal
index As Integer) As Boolean

    Dim plot1() As Date
    Dim plot3() As Double
```

```
Dim n As Long
Dim k As Long
n = 0

ReDim Preserve plot3(Measurements_N)
ReDim Preserve plot1(Measurements_N)
For k = 1 To Measurements_N
    If Measurements(1, k) = SensorId And Measurements(7, k) =
Unit Then
        n += 1

        plot1(n) = Measurements(8, k) & " " & Measurements(9,
k)
        plot3(n) = Measurements(4, k)

    End If
Next
ReDim Preserve plot3(n)
ReDim Preserve plot1(n)

Dim SumY As Double
Dim Ncount As Long
Dim S0 As Double
SumY = 0
Ncount = 0
S0 = 0

Dim xo As Long
Dim yo As Long
Dim xt As Long
Dim yt As Long
Dim xmin As Date
Dim xmax As Date
Dim ymin As Double
Dim ymax As Double

Dim bimaX As Double
Dim bimaY As Double
Dim w As Integer
Dim h As Integer

w = 1024 * 4
h = 768 * 4

xo = 50 * 4
yo = 50 * 4
xt = w - 80 * 4
yt = h - 80 * 4

If n = 0 Then GoTo NoData

xmin = plot1(1)
xmax = plot1(1)
ymin = plot3(1)
```

```
ymax = plot3(1)

''Brisko max kai ypologizo
statistika'' .....

For k = 1 To n
  If plot1(k) > xmax Then xmax = plot1(k)
  If plot1(k) < xmin Then xmin = plot1(k)
  If plot3(k) > ymax Then
    ymax = plot3(k)
    Chart_Stat(1, index) = plot1(k).ToOADate
  End If
  If plot3(k) < ymin Then
    ymin = plot3(k)
    Chart_Stat(3, index) = plot1(k).ToOADate
  End If
  SumY = SumY + plot3(k)
Next
Ncount = n
Chart_Stat(2, index) = ymax
Chart_Stat(4, index) = ymin
Chart_Stat(5, index) = SumY / Ncount
For k = 1 To n
  S0 = S0 + (plot3(k) - Chart_Stat(5, index)) ^ 2
Next
S0 = Math.Sqrt(S0 / (Ncount - 1))
Chart_Stat(6, index) = S0
Chart_Stat(7, index) = Ncount

.....
.....

bimaY = (ymax - ymin) / (yt - yo)
bimaX = (xmax - xmin).TotalMinutes / (xt - xo)

Dim plot1_2() As Double
ReDim plot1_2(n)
For k = 1 To n
  plot1_2(k) = (plot1(k) - xmin).TotalMinutes
Next

Dim tmp As Double
Dim tmpDate As Date
Dim sorted As Boolean
sorted = True
Do While sorted = True
  sorted = False
  For k = n To 2 Step -1
    If plot1_2(k) < plot1_2(k - 1) Then
      tmp = plot1_2(k)
      plot1_2(k) = plot1_2(k - 1)
      plot1_2(k - 1) = tmp
      tmp = plot3(k)
      plot3(k) = plot3(k - 1)
      plot3(k - 1) = tmp
      tmpDate = plot1(k)
```

```
        plot1(k) = plot1(k - 1)
        plot1(k - 1) = tmpDate
        sorted = True
    End If
Next
Loop

Dim data(,) As Single
ReDim data(2, n)

For k = 1 To n
    data(1, k) = CSng(Math.Round((plot1_2(k) - plot1_2(1)) /
bimaX) + xo)
    data(2, k) = h - CSng(Math.Round((plot3(k) - ymin) / bimaY)
+ yo)
Next

NoData:
'brisko index tou sensor sto point(0,0) kai sto
P_Client_SNS_Unit
Dim CurSensor As Integer
For k = 1 To Point(0, 0)
    If P_Client_SNS_Unit(0, k) = SensorId Then CurSensor = k
Next

Dim b As Bitmap
Dim g As Graphics
b = New Bitmap(w, h)
g = Graphics.FromImage(b)

g.FillRectangle(Brushes.White, 1, 1, b.Size.Width - 1,
b.Size.Height - 1)
Dim chartvalues As ArrayList = New ArrayList

Dim Cpen As Pen
If MyColor = "Green" Then Cpen = New Pen(Color.Green, 3)
If MyColor = "Blue" Then Cpen = New Pen(Color.Blue, 3)
If MyColor = "Red" Then Cpen = New Pen(Color.Red, 3)
If MyColor = "Brown" Then Cpen = New Pen(Color.Brown, 3)
If MyColor = "Black" Then Cpen = New Pen(Color.Black, 3)
If MyColor = "Yellow" Then Cpen = New Pen(Color.Yellow, 3)
If MyColor = "Galazio" Then Cpen = New Pen(Color.Cyan, 3)
If MyColor = "Purple" Then Cpen = New Pen(Color.Purple, 3)
Dim Cpen2 As Pen = New Pen(Color.Black, 3)

g.DrawLine(Cpen2, xo, h - yo, xo, h - yt)
g.DrawLine(Cpen2, xo, h - yo, xt, h - yo)
Dim axesFontN As Font = New Font("arial", 30)
If n > 0 Then
    For k = 2 To n
        g.DrawLine(Cpen, data(1, k - 1), data(2, k - 1),
data(1, k), data(2, k))
```



```
Next
    g.DrawString(plot1(1).Day.ToString & "/" &
plot1(1).Month.ToString & "/" & plot1(1).Year.ToString & " " &
Add0to1Dig(plot1(1).Hour.ToString) & ":" &
Add0to1Dig(plot1(1).Minute.ToString), axesFontN, Brushes.Black, data(1,
1) - 50, h - yo + 10 * 4)
    g.DrawString(plot1(n).Day.ToString & "/" &
plot1(n).Month.ToString & "/" & plot1(n).Year.ToString & " " &
Add0to1Dig(plot1(n).Hour.ToString) & ":" &
Add0to1Dig(plot1(n).Minute.ToString), axesFontN, Brushes.Black, data(1,
n) - 180, h - yo + 10 * 4)
    g.DrawString(ymin.ToString, axesFontN, Brushes.Black, 20, h
- 80 * 4)
    g.DrawString(ymax.ToString, axesFontN, Brushes.Black, 20, h
- (ymax - ymin) / bimaY - 90 * 4)
    g.DrawString("SensorID", axesFontN, Brushes.Black, w / 4,
10 * 4)
    g.DrawString(": " & SensorId, axesFontN, Brushes.Black, w
/ 4 + 200, 10 * 4)
    g.DrawString("Latitude", axesFontN, Brushes.Black, w / 4, 2
* 10 * 4)
    g.DrawString(": " & Point(2, CurSensor).ToString,
axesFontN, Brushes.Black, w / 4 + 200, 2 * 10 * 4)
    g.DrawString("Longitude", axesFontN, Brushes.Black, w / 4,
3 * 10 * 4)
    g.DrawString(": " & Point(3, CurSensor).ToString,
axesFontN, Brushes.Black, w / 4 + 200, 3 * 10 * 4)
    g.DrawString("Unit", axesFontN, Brushes.Black, w / 4 + 600,
1 * 10 * 4)
    g.DrawString(": " & P_Client_SNS_Unit(3, CurSensor),
axesFontN, Brushes.Black, w / 4 + 600 + 200, 1 * 10 * 4)
    g.DrawString("Client Op.", axesFontN, Brushes.Black, w / 4
+ 600, 2 * 10 * 4)
    g.DrawString(": " & P_Client_SNS_Unit(1, CurSensor),
axesFontN, Brushes.Black, w / 4 + 600 + 200, 2 * 10 * 4)
    g.DrawString("Segment", axesFontN, Brushes.Black, w / 4 +
600, 3 * 10 * 4)
    g.DrawString(": " & P_Client_SNS_Unit(2, CurSensor),
axesFontN, Brushes.Black, w / 4 + 600 + 200, 3 * 10 * 4)
End If

'System.IO.File.Delete(FileName)
b.Save(FileName)

g.Dispose()
b.Dispose()

CreateGraphMax = True

End Function

Public Function SID_Found_In_tmpSID(ByVal SID As String) As Boolean
```

```
Dim out As Boolean
out = False

Dim i As Long
For i = 1 To Chart_N
    If tmpSID(i) = SID Then
        out = True
    End If
Next

SID_Found_In_tmpSID = out
End Function
Public Function CreateUniqueGraph(ByVal Unit As String, ByVal
FileName As String) As Boolean

Dim w As Integer
Dim h As Integer

w = 1024
h = 768

Dim b As Bitmap
Dim g As Graphics
b = New Bitmap(w, h)
g = Graphics.FromImage(b)
g.FillRectangle(Brushes.White, 1, 1, b.Size.Width - 1,
b.Size.Height - 1)

Dim data(,) As Single
Dim xo As Long
Dim yo As Long
Dim xt As Long
Dim yt As Long
Dim xmin As Date
Dim xmax As Date
Dim ymin As Double
Dim ymax As Double
xo = 50
yo = 50
xt = w - 80
yt = h - 80
Dim xtemp As Date
Dim ytemp As Double

Dim bimaX As Double
Dim bimaY As Double
```

```
Dim n As Long
Dim k As Long
n = 0

For k = 1 To Measurements_N
    If SID_Found_In_tmpSID(Measurements(1, k)) = True And
Measurements(7, k) = Unit Then
        n += 1
        If n = 1 Then
            xmax = Measurements(8, k) & " " & Measurements(9, k)
            xmin = xmax
            ymax = Measurements(4, k)
            ymin = ymax
        End If
        xtemp = Measurements(8, k) & " " & Measurements(9, k)
        ytemp = Measurements(4, k)
        If xtemp > xmax Then xmax = xtemp
        If xtemp < xmin Then xmin = xtemp
        If ytemp > ymax Then ymax = ytemp
        If ytemp < ymin Then ymin = ytemp
    End If
Next

bimaY = (ymax - ymin) / (yt - yo)
bimaX = (xmax - xmin).TotalMinutes / (xt - xo)

Dim plot1() As Date
Dim plot1_2() As Double
Dim plot3() As Double

ReDim Preserve Chart_par(3, 1) 'xmin,xo,xbima

Dim i As Long
n = 0
For i = 1 To Chart_N
    ReDim Preserve plot3(Measurements_N)
    ReDim Preserve plot1(Measurements_N)
    For k = 1 To Measurements_N
        If Measurements(1, k) = tmpSID(i) And Measurements(7,
k) = Unit Then
            n += 1

            plot1(n) = Measurements(8, k) & " " &
Measurements(9, k)
            plot3(n) = Measurements(4, k)

        End If
    Next
    ReDim Preserve plot3(n)
```

```
ReDim Preserve plot1(n)
ReDim plot1_2(n)

For k = 1 To n
    plot1_2(k) = (plot1(k) - xmin).TotalMinutes
Next

'      sort

Dim tmp As Double
Dim tmpDate As Date
Dim sorted As Boolean
sorted = True
Do While sorted = True
    sorted = False
    For k = n To 2 Step -1
        If plot1_2(k) < plot1_2(k - 1) Then
            tmp = plot1_2(k)
            plot1_2(k) = plot1_2(k - 1)
            plot1_2(k - 1) = tmp
            tmp = plot3(k)
            plot3(k) = plot3(k - 1)
            plot3(k - 1) = tmp
            tmpDate = plot1(k)
            plot1(k) = plot1(k - 1)
            plot1(k - 1) = tmpDate
            sorted = True
        End If
    Next
Loop

ReDim data(2, n)

For k = 1 To n
    data(1, k) = CSng(Math.Round((plot1_2(k) - plot1_2(1))
/ bimaX) + xo)
    data(2, k) = h - CSng(Math.Round((plot3(k) - ymin) /
bimaY) + yo)
Next

For k = 2 To n
    g.DrawLine(Pens.Aqua, data(1, k - 1), data(2, k - 1),
data(1, k), data(2, k))
Next

Next
```

```
Dim chartvalues As ArrayList = New ArrayList

Dim Cpen As Pen
Dim axesFontN As Font = New Font("arial", 22)
Dim Cpen2 As Pen = New Pen(Color.Black, 3)

g.DrawLine(Cpen2, xo, h - yo, xo, h - yt)
g.DrawLine(Cpen2, xo, h - yo, xt, h - yo)
g.DrawString(plot1(1).Day.ToString & "/" &
plot1(1).Month.ToString & "/" & plot1(1).Year.ToString & " " &
Add0tolDig(plot1(1).Hour.ToString) & ":" &
Add0tolDig(plot1(1).Minute.ToString), axesFontN, Brushes.Black, DATA(1,
1) - 50, h - yo + 10)
g.DrawString(plot1(n).Day.ToString & "/" &
plot1(n).Month.ToString & "/" & plot1(n).Year.ToString & " " &
Add0tolDig(plot1(n).Hour.ToString) & ":" &
Add0tolDig(plot1(n).Minute.ToString), axesFontN, Brushes.Black, DATA(1,
n) - 150, h - yo + 10)
g.DrawString(ymin.ToString, axesFontN, Brushes.Black, 1, h -
80)
g.DrawString(ymax.ToString, axesFontN, Brushes.Black, 1, h -
(ymax - ymin) / bimaY - 50)

b.Save(FileName)

g.Dispose()
b.Dispose()

End Function

End Module
```

```
Imports System.Xml
Imports System.Drawing
Imports System.Drawing.Graphics
Imports System.Drawing.Drawing2D
Imports System.IO

Public Class Form1

    Public fs2 As New FileStream(S_path & "Proc.txt", FileMode.Create,
FileAccess.Write)
    Public s2 As New StreamWriter(fs2)
    Public savedIndex As Integer      'krataei posa saved metavlites
exo kanei

Public Sub Charts_Reset()
    ReDim Measurements(9, 1)
    Dim ff As Integer
    For ff = 1 To Chart_N
        Try
            ChartBox(ff).Image = Image.FromFile(S_path & "null_" &
ff & ".bmp")
            ChartBoxFile(ff) = S_path & "null_" & ff & ".bmp"

            ChartBox(ff).BackColor = Color.Empty
        Catch ex As Exception

        End Try
    Next

    CurSensorChart = -1

    ReDim CSensorID(Chart_N)
End Sub

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Startup()

    savedIndex = 0
    ProcIndex = 0
    LastProcTime = Date.Now

    ReDim Measurements(9, 1)
    ReDim Chart_Label(1024, Chart_N)

    Dim i As Integer
    Dim w1 As Integer
    'ReDim ChartBox(Chart_N)
```

```
ReDim ChartBox(7)

ChartBox(1) = P_B_1
ChartBox(2) = P_B_2
ChartBox(3) = P_B_3
ChartBox(4) = P_B_4
ChartBox(5) = P_B_5
ChartBox(6) = P_B_6
ChartBox(7) = P_B_7
For i = 1 To 7
    ChartBox(i).Visible = False
Next

For i = 1 To Chart_N
    ChartBox(i).Visible = True
    'ChartBox(i) = New Windows.Forms.PictureBox
    ChartBox(i).BorderStyle = BorderStyle.Fixed3D
    'ChartBox(i).Parent = Me

    ChartBox(i).Top = MenuStrip1.Bottom + 10

    ChartBox(i).Height = 300 - 10 - MenuStrip1.Bottom - 10
    w1 = CLng(ChartBox(i).Height * 4 / 3)
    ChartBox(i).Left = WebBrowser1.Left + (i - 1) * (w1 + 10)
    ChartBox(i).Width = w1
    ChartBox(i).Visible = True
    ChartBox(i).SizeMode = PictureBoxSizeMode.StretchImage
    'ChartBox(i).Cursor = Cursors.Cross
    ChartBox(i).Refresh()

    File.Copy(S_path & "null.bmp", S_path & "null_" & i &
".bmp", True)

Next

ReDim CSensorID(Chart_N)

Charts_Reset()

Timer2.Enabled = False

End Sub
```

```
Private Sub ExitToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ExitToolStripMenuItem.Click
    s2.Close()
    Me.Close()

End Sub

Private Sub Button2_Click_1(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click

    If DB_Connected = False Then
        DB_Connect(DB_File, DB_UserID, DB_Password)
        DataS = Get_DB_Data(DB_Q1)

        Dim Client_Found As Boolean
        Dim SNS_Found As Boolean
        Dim Unit_Found As Boolean

        Dim k As Integer
        Dim i As Integer
        Dim j As Integer
        Dim tmp As String

        i = 0
        tmp = ""
        For j = 0 To DataS.Tables(i).Rows.Count - 1
            ReDim Preserve Point(4, 0 To j + 1)
            ReDim Preserve P_Client_SNS_Unit(7, j + 1)
            Point(0, 0) = Point(0, 0) + 1
            Point(1, j + 1) = CDb1(DataS.Tables(i).Rows(j)(0))
            Point(2, j + 1) = CDb1(DataS.Tables(i).Rows(j)(1))
            Point(3, j + 1) = CDb1(DataS.Tables(i).Rows(j)(2))
            P_Client_SNS_Unit(0, j + 1) =
CStr(DataS.Tables(i).Rows(j)(6))
            P_Client_SNS_Unit(1, j + 1) =
CStr(DataS.Tables(i).Rows(j)(3))
            P_Client_SNS_Unit(2, j + 1) =
CStr(DataS.Tables(i).Rows(j)(4))
            P_Client_SNS_Unit(3, j + 1) =
CStr(DataS.Tables(i).Rows(j)(5))
            P_Client_SNS_Unit(4, j + 1) =
CStr(DataS.Tables(i).Rows(j)(7))
            P_Client_SNS_Unit(5, j + 1) =
CStr(DataS.Tables(i).Rows(j)(8))
            P_Client_SNS_Unit(6, j + 1) =
CStr(DataS.Tables(i).Rows(j)(9)) 'Icon ON
            P_Client_SNS_Unit(7, j + 1) =
CStr(DataS.Tables(i).Rows(j)(10)) 'Icon OFF
```



```
    If j = 0 Then
        Client_N = 1
        SNS_N = 1
        Unit_N = 1

        ReDim Client(1)
        ReDim SNS(1)
        ReDim Unit(3, 1)

        Client(1) = P_Client_SNS_Unit(1, 1)
        SNS(1) = P_Client_SNS_Unit(2, 1)
        Unit(1, 1) = P_Client_SNS_Unit(3, 1)
        Unit(2, 1) = P_Client_SNS_Unit(6, 1)
        Unit(3, 1) = P_Client_SNS_Unit(7, 1)
        'ComboBox1.Items.Add(P_Client_SNS_Unit(2, 1))
    End If

    Client_Found = False
    SNS_Found = False
    Unit_Found = False

    For k = 1 To Client_N
        If Client(k) = P_Client_SNS_Unit(1, j + 1) Then
Client_Found = True
        Next
    For k = 1 To SNS_N
        If SNS(k) = P_Client_SNS_Unit(2, j + 1) Then
SNS_Found = True
        Next
    For k = 1 To Unit_N
        If Unit(1, k) = P_Client_SNS_Unit(3, j + 1) Then
Unit_Found = True
        Next

    If Client_Found = False Then
        Client_N = Client_N + 1
        ReDim Preserve Client(Client_N)
        Client(Client_N) = P_Client_SNS_Unit(1, j + 1)
        'ComboBox1.Items.Add(P_Client_SNS_Unit(1, j + 1))
    End If
    If SNS_Found = False Then
        SNS_N = SNS_N + 1
        ReDim Preserve SNS(SNS_N)
        SNS(SNS_N) = P_Client_SNS_Unit(2, j + 1)
        'ComboBox1.Items.Add(P_Client_SNS_Unit(2, j + 1))
    End If
    If Unit_Found = False Then
        Unit_N = Unit_N + 1
        ReDim Preserve Unit(3, Unit_N)
        Unit(1, Unit_N) = P_Client_SNS_Unit(3, j + 1)
        Unit(2, Unit_N) = P_Client_SNS_Unit(6, j + 1)
```

```
        Unit(3, Unit_N) = P_Client_SNS_Unit(7, j + 1)
        'ComboBox4.Items.Add(P_Client_SNS_Unit(3, j + 1))
    End If

    tmp = tmp & "Value:" & DataS.Tables(i).Rows(j)(0)
    tmp = tmp & vbTab & vbTab & "Lat :" &
DataS.Tables(i).Rows(j)(1)
    tmp = tmp & vbTab & "Long:" &
DataS.Tables(i).Rows(j)(2)
    tmp = tmp & vbTab & "SNS:" & DataS.Tables(i).Rows(j)(3)

    If Not j = DataS.Tables(i).Rows.Count - 1 Then tmp =
tmp & vbCrLf
    Next

    For i = 1 To Unit_N
        ComboBox4.Items.Add(Unit(1, i))
    Next i

    StatusStrip1.Items.Add(DB_File & "    Connected")
    StatusStrip1.Items(0).RightToLeft =
Windows.Forms.RightToLeft.Yes

    Me.ComboBox2.Items.Clear()
    Me.ComboBox5.Items.Clear()
    Dim DataStmp As DataSet
    'DataStmp.Clear()
    DataStmp = Get_DB_Data("SELECT MIN(Date) FROM
SNM_Measurement;")
    MinDate = DataStmp.Tables(0).Rows(0)(0)
    DataStmp.Clear()
    DataStmp = Get_DB_Data("SELECT MAX(Date) FROM
SNM_Measurement;")
    MaxDate = DataStmp.Tables(0).Rows(0)(0)
    DataStmp.Clear()
    DataStmp = Get_DB_Data("SELECT DISTINCT Date FROM
SNM_Measurement ORDER BY Date;")
    For i = 0 To DataStmp.Tables(0).Rows.Count - 1
        Me.ComboBox2.Items.Add(DataStmp.Tables(0).Rows(i)(0))
        Me.ComboBox5.Items.Add(DataStmp.Tables(0).Rows(i)(0))
    Next
```

```
Me.ComboBox2.Text = (MinDate)
Me.ComboBox5.Text = (MaxDate)

Button2.Text = "Disconnect - Exit"
DB_Connected = True
Else
DB_Disconnect()
Timer2.Enabled = False
Button2.Text = "Connect"
Data_Reset()
Me.Dispose()

End If

End Sub

Private Sub ComboBox1_SelectedIndexChanged_1(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ComboBox1.SelectedIndexChanged
If CheckBox1.Checked = False Then
WebBrowser1.Navigate(CreateJScript3(0, Point, P_Client_SNS_Unit,
ComboBox3.Text, ComboBox1.Text, ComboBox4.Text, Coors, Chart_N,
CurIconON, CurIconOFF))
If CheckBox1.Checked = True Then
WebBrowser1.Navigate(CreateJScript3(1, Point, P_Client_SNS_Unit,
ComboBox3.Text, ComboBox1.Text, ComboBox4.Text, Coors, Chart_N,
CurIconON, CurIconOFF))

End Sub

Private Sub ComboBox3_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ComboBox3.SelectedIndexChanged
Dim i, j As Integer
ComboBox1.Items.Clear()
ComboBox1.Text = ""

Dim Found As Boolean
Found = False
i = 1

Do While Found = False
If P_Client_SNS_Unit(1, i) = ComboBox3.Text Then
ComboBox1.Items.Add(P_Client_SNS_Unit(2, i))
Found = True
End If
i = i + 1
Loop
```

```
For i = 1 To Point(0, 0)
    Found = False

    If P_Client_SNS_Unit(1, i) = ComboBox3.Text Then
        For j = 1 To ComboBox1.Items.Count
            If P_Client_SNS_Unit(2, i) = ComboBox1.Items(j -
1).ToString Then Found = True
        Next
        If Found = False Then
            ComboBox1.Items.Add(P_Client_SNS_Unit(2, i))
        End If
    End If
Next i

If ComboBox1.Items.Count = 1 Then
    ComboBox1.SelectedIndex = 0
End If

End Sub

Private Sub ComboBox4_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ComboBox4.SelectedIndexChanged
    Dim i, j As Integer
    ComboBox3.Items.Clear()
    ComboBox1.Items.Clear()
    ComboBox3.Text = ""
    ComboBox1.Text = ""

    Dim Found As Boolean
    Found = False
    i = 1

    Do While Found = False
        If P_Client_SNS_Unit(3, i) = ComboBox4.Text Then
            ComboBox3.Items.Add(P_Client_SNS_Unit(1, i))
            Found = True
        End If
        i = i + 1
    Loop

    For i = 1 To Point(0, 0)
        Found = False

        If P_Client_SNS_Unit(3, i) = ComboBox4.Text Then
            For j = 1 To ComboBox3.Items.Count
                If P_Client_SNS_Unit(1, i) = ComboBox3.Items(j -
1).ToString Then Found = True
            Next
            If Found = False Then
                ComboBox3.Items.Add(P_Client_SNS_Unit(1, i))
            End If
        End If
    End If
```

```
Next i

For i = 1 To Unit_N
    If ComboBox4.Text = Unit(1, i) Then
        CurIconON = Unit(2, i)
        CurIconOFF = Unit(3, i)
    End If
Next

If ComboBox3.Items.Count = 1 Then
    ComboBox3.SelectedIndex = 0
End If

End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick

    If Graph_Flag = True Then
        'Timer2.Enabled = False
        WebBrowser1.Top = WebBrowser1.Top + 4
        WebBrowser1.Height = WebBrowser1.Height - 4
        If WebBrowser1.Top > 300 Then
            Timer1.Enabled = False
            If CheckBox1.Checked = False Then
                WebBrowser1.Navigate(CreateJScript3(0, Point, P_Client_SNS_Unit,
                ComboBox3.Text, ComboBox1.Text, ComboBox4.Text, Coors, Chart_N,
                CurIconON, CurIconOFF))
                If CheckBox1.Checked = True Then
                    WebBrowser1.Navigate(CreateJScript3(1, Point, P_Client_SNS_Unit,
                    ComboBox3.Text, ComboBox1.Text, ComboBox4.Text, Coors, Chart_N,
                    CurIconON, CurIconOFF))
                End If
                'Timer2.Enabled = True
            End If
        End If

    If Graph_Flag = False Then
        'Timer2.Enabled = False
        WebBrowser1.Top = WebBrowser1.Top - 4
        WebBrowser1.Height = WebBrowser1.Height + 4
        If WebBrowser1.Top < MenuStrip1.Bottom + 9 Then
            Timer1.Enabled = False
            If CheckBox1.Checked = False Then
                WebBrowser1.Navigate(CreateJScript3(0, Point, P_Client_SNS_Unit,
                ComboBox3.Text, ComboBox1.Text, ComboBox4.Text, Coors, Chart_N,
                CurIconON, CurIconOFF))
                If CheckBox1.Checked = True Then
                    WebBrowser1.Navigate(CreateJScript3(1, Point, P_Client_SNS_Unit,
                    ComboBox3.Text, ComboBox1.Text, ComboBox4.Text, Coors, Chart_N,
                    CurIconON, CurIconOFF))
                End If
                'Timer2.Enabled = True
            End If
        End If
    End If
End Sub
```

```
Public Sub GraphToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
GraphToolStripMenuItem.Click
    If sender.checked = False Then
        'Timer2.Enabled = False
        Graph_Flag = True
        'Timer1.Enabled = True
        Dim i As Integer

        For i = 1 To Chart_N
            ChartBox(i).Visible = False
        Next

        sender.checked = True
        WebBrowser1.Top = 300
        WebBrowser1.Height = StatusStrip1.Top - 300 - 10
        '

        For i = 1 To Chart_N
            ChartBox(i).Image = Image.FromFile(ChartBoxFile(i))
            ChartBox(i).SizeMode = PictureBoxSizeMode.StretchImage
            ChartBox(i).Visible = True
        Next

    Else

        Graph_Flag = False
        sender.checked = False
        WebBrowser1.Top = MenuStrip1.Bottom + 10
        WebBrowser1.Height = StatusStrip1.Top - MenuStrip1.Bottom -
20

    End If
End Sub

Public Sub ChartHide()
    'Timer2.Enabled = False
    Graph_Flag = False
    'Timer1.Enabled = True

    WebBrowser1.Top = MenuStrip1.Bottom + 10
    WebBrowser1.Height = StatusStrip1.Top - MenuStrip1.Bottom - 20

End Sub
```

```
Public Sub ChartShow()  
  
    Dim i As Integer  
  
    For i = 1 To Chart_N  
        ChartBox(i).Visible = False  
    Next  
  
    WebBrowser1.Top = 300  
    WebBrowser1.Height = StatusStrip1.Top - 300 - 10  
    '  
  
    For i = 1 To Chart_N  
        ChartBox(i).Image = Image.FromFile(ChartBoxFile(i))  
        ChartBox(i).SizeMode = PictureBoxSizeMode.StretchImage  
        ChartBox(i).Visible = True  
    Next  
  
End Sub  
  
Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Timer2.Tick  
    Dim tmp(Chart_N) As String  
    ReDim Preserve tmpSID(Chart_N)  
    Dim tmpIndex(Chart_N) As String  
    Dim tmpIn As Integer  
    If (Date.Now - LastProcTime).TotalSeconds > 0.5 Then  
        Try  
  
            Dim i As Integer  
  
  
            FileOpen(1, GInfo_File, OpenMode.Input)  
            tmp(1) = (LineInput(1))  
  
            For i = 1 To Chart_N  
                tmp(i) = (LineInput(1))  
                tmpIn = InStr(tmp(i), ";")  
                tmpIndex(i) = Mid(tmp(i), 1, tmpIn - 1)  
                tmpSID(i) = Mid(tmp(i), tmpIn + 1, InStr(tmpIn + 1,  
tmp(i), ";") - tmpIn - 1)  
  
            Next
```

```
FileClose(1)
Dim Changed As Boolean
Changed = False
For i = 1 To Chart_N
    If tmpSID(i) <> CSensorID(i) Then Changed = True
Next

If Changed = True Then

    ProcIndex += 1
    s2.WriteLine("Proc Index : " & ProcIndex & "
MapIndex : " & savedIndex & "    CurSensorChart : " & CurSensorChart)
    For i = 1 To Chart_N
        s2.WriteLine("tmpSID(" & i & ") : " &
tmpSID(i))
    Next
    For i = 1 To Chart_N
        s2.WriteLine("CSensorID(" & i & ") : " &
CSensorID(i))
    Next
    For i = 1 To Chart_N
        s2.WriteLine("tmpIndex(" & i & ") : " &
tmpIndex(i))
    Next
    For i = 1 To Chart_N
        s2.WriteLine("tmp(" & i & ") : " & tmp(i))
    Next
    s2.WriteLine()

    If CurSensorChart < Chart_N Then
        CurSensorChart += 1
    Else
        CurSensorChart = 1

    Dim j As Long
    Dim Past_Measurements(,) As String
    Dim Past_Measurements_N As Long
    Past_Measurements = Measurements.Clone
    Past_Measurements_N = Measurements_N

    Measurements_N = 0
    For i = 1 To Past_Measurements_N
        For j = 2 To Chart_N
            If tmpSID(j) = Past_Measurements(1, i) Then
Measurements_N = Measurements_N + 1
            Next
        Next

    Dim New_Meas_N As Long
    Dim DataS_A As DataSet
```



```
        If CurSensorChart > 0 Then
            'Me.Enabled = False
            Form6.Show()
            Form6.Pro1.Value = 0
            Form6.Label11.Text = "Quering Database"
            Form6.Refresh()
        End If
        DataS_A = Get_DB_Data(Create_DB_Q2(MinDate,
MaxDate, tmpSID(1)))
        If CurSensorChart > 0 Then
            Form6.Pro1.Value = 40
            Form6.Label11.Text = "Processing Data"
            Form6.Refresh()
        End If
        New_Meas_N = DataS_A.Tables(0).Rows.Count
        Measurements_N = Measurements_N + New_Meas_N
        ReDim Measurements(9, Measurements_N)

        Dim k As Long
        Dim Put_N As Long
        Put_N = 0
        For i = 1 To Past_Measurements_N
            For j = 2 To Chart_N
                If tmpSID(j) = Past_Measurements(1, i) Then
                    Put_N += 1
                    For k = 1 To 9
                        Measurements(k, Put_N) =
Past_Measurements(k, i)
                    Next
                End If
            Next
        Next

        For i = 0 To New_Meas_N - 1
            Put_N += 1
            Measurements(1, Put_N) =
CStr(DataS_A.Tables(0).Rows(i)(0))
            Measurements(2, Put_N) =
CStr(DataS_A.Tables(0).Rows(i)(1))
            Measurements(3, Put_N) =
CStr(DataS_A.Tables(0).Rows(i)(2))
            Measurements(4, Put_N) =
CStr(DataS_A.Tables(0).Rows(i)(3))
            Measurements(5, Put_N) =
CStr(DataS_A.Tables(0).Rows(i)(4))
            Measurements(6, Put_N) =
CStr(DataS_A.Tables(0).Rows(i)(5))
            Measurements(7, Put_N) =
CStr(DataS_A.Tables(0).Rows(i)(6))
            Measurements(8, Put_N) =
CStr(DataS_A.Tables(0).Rows(i)(7))
            Measurements(9, Put_N) =
CStr(DataS_A.Tables(0).Rows(i)(8))
        Next
```

```
    If CurSensorChart > 0 Then
        Form6.Pro1.Value = 70
        Form6.Label1.Text = "Creating Charts"
        Form6.Refresh()
    End If

    For i = 1 To Chart_N
        Try
            ChartBox(i).Image.Dispose()
        Catch ex As Exception

        End Try
    Next
    If IsNumeric(tmpIndex(1)) = True Then
        If CLng(tmpIndex(1)) > 0 Then
            tmpIn = InStr(tmp(1), "icons\")
            CreateGraph(ComboBox4.Text, tmpSID(1),
S_path & "Graph_" & CStr(CurSensorChart) & ".bmp", Mid(tmp(1), tmpIn +
6, Len(tmp(1)) - tmpIn - 1 - 4 - 4), CurSensorChart)
            CreateGraphMax(ComboBox4.Text, tmpSID(1),
S_path & "GraphMax_" & CStr(CurSensorChart) & ".bmp", Mid(tmp(1), tmpIn
+ 6, Len(tmp(1)) - tmpIn - 1 - 4 - 4), CurSensorChart)

        End If
    End If

    For i = 1 To Chart_N
        CSensorID(i) = tmpSID(i)
        If IsNumeric(tmpIndex(i)) = True Then
            If CLng(tmpIndex(i)) > 0 Then

                If i = CurSensorChart Then
                    ChartBox(i).BackColor =
Color.Lavender

                Else
                    ChartBox(i).BackColor = Color.White
                End If

                tmpIn = InStr(tmp(i), "icons/")
                ChartBox(i).Image =
Image.FromFile(S_path & "Graph_" & CStr(i) & ".bmp")
                ChartBoxFile(i) = S_path & "Graph_" &
CStr(i) & ".bmp"

                ChartBox(i).SizeMode = PictureBoxSizeMode.StretchImage
                ChartBox(i).BorderStyle = BorderStyle.Fixed3D

                AddHandler ChartBox(i).MouseMove,
AddressOf Me.PictureBox1_MouseMove
                AddHandler ChartBox(i).DoubleClick,
AddressOf Me.PictureBox1_DoubleClick

            End If
        End If
    Next i
    If CurSensorChart > 0 Then
```

```
        Form6.Pro1.Value = 100
        Form6.Label1.Text = "Done"
        Form6.Refresh()
        Form6.Dispose()
    End If

    LastProcTime = Date.Now

    Dim curIndex As Long
    For i = 1 To Point(0, 0)
        If P_Client_SNS_Unit(0, i) = tmpSID(1) Then
            curIndex = i
        End If
    Next

    PutStat(CurSensorChart, curIndex)
End If

Catch ex As Exception

End Try

End If

End Sub

Private Sub DataShownToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
DataShownToolStripMenuItem.Click
    If GroupBox3.Visible = True Then
        'sender.checked = False
        GroupBox4.Left = GroupBox1.Left
        GroupBox3.Visible = False
    Else
        'sender.checked = True
        GroupBox3.Visible = True
        GroupBox4.Left = -250
    End If
End Sub

Private Sub CheckBox7_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CheckBox7.CheckedChanged

    If CheckBox7.Checked = True Then
        CheckBox2.Enabled = True
        CheckBox3.Enabled = True
        CheckBox4.Enabled = True
        CheckBox5.Enabled = True
        CheckBox6.Enabled = True
    End If
End Sub
```

```
        ShowBubble = True

    Else
        CheckBox2.Enabled = False
        CheckBox3.Enabled = False
        CheckBox4.Enabled = False
        CheckBox5.Enabled = False
        CheckBox6.Enabled = False

        ShowBubble = False
    End If
End Sub

Private Sub FilterToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
FilterToolStripMenuItem.Click
    ' If GroupBox4.Visible = True Then
    If GroupBox4.Left <> -250 Then

        GroupBox4.Left = -250

    Else

        GroupBox4.Left = GroupBox1.Left

    End If
End Sub

Private Sub KeyToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
KeyToolStripMenuItem.Click
    Form2.Show()
    Form2.TextBox1.Text = G_Key
    Me.Enabled = False
End Sub

Private Sub SqlQueryToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
SqlQueryToolStripMenuItem.Click
    Form3.Show()
    Form3.TextBox1.Text = DB_Q1
    Form3.TextBox2.Text = DB_Q2
    Me.Enabled = False
End Sub

Private Sub OpenToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
OpenToolStripMenuItem.Click
    Form4.Show()
    Form4.TextBox1.Text = DB_File
    Form4.TextBox2.Text = DB_UserID
    Form4.TextBox3.Text = DB_Password
    'Me.Enabled = False
End Sub
```

Εφαρμογή απεικόνισης μετρήσεων σε Google Maps

```
Private Sub WebBrowser1_PreviewKeyDown(ByVal sender As Object,
ByVal e As System.Windows.Forms.PreviewKeyDownEventArgs) Handles
WebBrowser1.PreviewKeyDown
    ' MsgBox(e.KeyValue)
End Sub

Private Sub ComboBox2_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ComboBox2.SelectedIndexChanged
    MinDate = ComboBox2.Text

End Sub

Private Sub ComboBox5_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ComboBox5.SelectedIndexChanged
    MaxDate = ComboBox5.Text

End Sub

Private Sub PictureBox1_DoubleClick(ByVal sender As Object, ByVal e
As System.EventArgs) Handles PictureBox1.DoubleClick

    Dim i As Integer

    For i = 1 To Chart_N
        ChartBox(i).Visible = False
    Next

    'sender.checked = True
    WebBrowser1.Top = MenuStrip1.Bottom + 10
    WebBrowser1.Height = StatusStrip1.Top - MenuStrip1.Bottom - 20
    '

    Dim char_index As Integer
    For i = 1 To Chart_N
        If sender.Left = ChartBox(i).Left Then
            char_index = i
        End If
    Next

    MaxImageFile = S_path & "GraphMax_" & CStr(char_index) & ".bmp"

    If Chart_Maximized = False Then

    Else

    End If

    Form5.Show()

End Sub
```

Εφαρμογή απεικόνισης μετρήσεων σε Google Maps

```
Private Sub PictureBox1_MouseMove(ByVal sender As Object, ByVal e
As System.Windows.Forms.MouseEventArgs) Handles PictureBox1.MouseMove

    If CurSensorChart > 0 Then

        Dim i As Integer
        Dim char_index As Integer
        For i = 1 To Chart_N
            If sender.Left = ChartBox(i).Left Then
                char_index = i
            End If
        Next

        Dim diff As Integer          'krataei to max abs x-
xzitoumeno
        Dim maxDate As Date = Date.FromOADate(Chart_par(4,
char_index))

        Dim tt As String
        tt = ""

        Dim x As Double
        x = e.Location.X

        x = Math.Round(x * 1024 / sender.Width)
        If Chart_Label(x, char_index) <> "" Then
            Label6.Text = Chart_Label(x, char_index)
        Else
            For i = -2 To 2
                Try
                    If Chart_Label(x + i, char_index) <> "" Then tt
= Chart_Label(x + i, char_index)
                Catch ex As Exception

                End Try
            Next
        End If

        If tt <> "" Then Label6.Text = tt

EXIT_FOR:
        If tt <> "" Then
            Label6.Left = -Label6.Width +
ChartBox(char_index).Width + ChartBox(char_index).Left '+ x / 1024 *
sender.Width '+ e.X
            Label6.Top = ChartBox(char_index).Top - 25 '-
        End If
    End If

End Sub
```

Εφαρμογή απεικόνισης μετρήσεων σε Google Maps

```
Private Sub Button5_Click_1(ByVal sender As System.Object, ByVal e  
As System.EventArgs) Handles Button5.Click
```

```
Try
```

```
Dim tmp(Chart_N) As String  
ReDim Preserve tmpSID(Chart_N)  
Dim tmpIndex(Chart_N) As String  
Dim tmpIn As Integer
```

```
FileOpen(1, GInfo_File, OpenMode.Input)  
tmp(1) = (LineInput(1))
```

```
Dim i As Integer  
For i = 1 To Chart_N  
    tmp(i) = (LineInput(1))  
    tmpIn = InStr(tmp(i), ";")  
    tmpIndex(i) = Mid(tmp(i), 1, tmpIn - 1)  
    tmpSID(i) = Mid(tmp(i), tmpIn + 1, InStr(tmpIn + 1,  
tmp(i), ";") - tmpIn - 1)  
Next  
FileClose(1)
```

```
Measurements_N = 0  
Dim Meas_N As Long  
Meas_N = 0  
Dim DataS_A(Chart_N) As DataSet  
Dim j As Integer  
Form6.Show()  
Form6.Pro1.Value = 0  
For i = 1 To Chart_N  
    'DataS2.Clear()
```

```
Chart_N          Form6.Label1.Text = "Querying Database " & i & "/" &  
  
                Form6.Refresh()  
                DataS_A(i) = Get_DB_Data(Create_DB_Q2(MinDate, MaxDate,  
tmpSID(i)))  
  
                Form6.Pro1.Value += 100 / Chart_N / 3  
                Form6.Label1.Text = "Processing Data " & i & "/" &  
Chart_N  
  
                Form6.Refresh()
```

```
Meas_N = Meas_N + DataS_A(i).Tables(0).Rows.Count  
ReDim Preserve Measurements(9, 0 To Meas_N + 1)
```

```
For j = 0 To DataS_A(i).Tables(0).Rows.Count - 1
    Measurements_N += 1

    Measurements(1, Measurements_N) =
CStr(DataS_A(i).Tables(0).Rows(j)(0))
    Measurements(2, Measurements_N) =
CStr(DataS_A(i).Tables(0).Rows(j)(1))
    Measurements(3, Measurements_N) =
CStr(DataS_A(i).Tables(0).Rows(j)(2))
    Measurements(4, Measurements_N) =
CStr(DataS_A(i).Tables(0).Rows(j)(3))
    Measurements(5, Measurements_N) =
CStr(DataS_A(i).Tables(0).Rows(j)(4))
    Measurements(6, Measurements_N) =
CStr(DataS_A(i).Tables(0).Rows(j)(5))
    Measurements(7, Measurements_N) =
CStr(DataS_A(i).Tables(0).Rows(j)(6))
    Measurements(8, Measurements_N) =
CStr(DataS_A(i).Tables(0).Rows(j)(7))
    Measurements(9, Measurements_N) =
CStr(DataS_A(i).Tables(0).Rows(j)(8))
Next

Form6.Pro1.Value += 100 / Chart_N / 3
Next

For i = 1 To Chart_N
    Try
        ChartBox(i).Image.Dispose()
    Catch ex As Exception

    End Try
Next

Dim kk As Integer
i = CurSensorChart
For kk = 1 To Chart_N
    Form6.Label1.Text = "Creating Charts " & kk & "/" &
Chart_N

    Form6.Refresh()
    If IsNumeric(tmpIndex(i)) = True Then
        If CLng(tmpIndex(i)) > 0 Then

            tmpIn = InStr(tmp(i), "icons\")
            CreateGraph(ComboBox4.Text, tmpSID(i), S_path &
"Graph_" & CStr(kk) & ".bmp", Mid(tmp(i), tmpIn + 6, Len(tmp(i)) -
tmpIn - 1 - 4 - 4), kk)
```


Εφαρμογή απεικόνισης μετρήσεων σε Google Maps

```
        CreateGraphMax(ComboBox4.Text, tmpSID(i),
S_path & "GraphMax_" & CStr(kk) & ".bmp", Mid(tmp(i), tmpIn + 6,
Len(tmp(i)) - tmpIn - 1 - 4 - 4), kk)

        If i = CurSensorChart Then
            ChartBox(i).BackColor = Color.Lavender
        Else
            ChartBox(i).BackColor = Color.White
        End If

        ChartBox(kk).Image = Image.FromFile(S_path &
"Graph_" & CStr(kk) & ".bmp")

        ChartBox(kk).SizeMode =
PictureBoxSizeMode.StretchImage
        ChartBox(kk).BorderStyle = BorderStyle.Fixed3D

        End If
    End If
    If i > 1 Then
        i -= 1
    Else
        i = Chart_N
    End If
    Form6.Pro1.Value += 100 / Chart_N / 3
Next kk

Form6.Pro1.Value = 100
Form6.Label1.Text = "Done"
Form6.Refresh()
Form6.Dispose()

Dim curIndex As Long
For i = 1 To Point(0, 0)
    If P_Client_SNS_Unit(0, i) = tmpSID(1) Then
        curIndex = i
    End If
Next

PutStat(CurSensorChart, curIndex)

Catch ex As Exception

End Try

End Sub
```

Εφαρμογή απεικόνισης μετρήσεων σε Google Maps

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    If CheckBox1.Checked = False Then
WebBrowser1.Navigate(CreateJScript3(0, Point, P_Client_SNS_Unit,
ComboBox3.Text, ComboBox1.Text, ComboBox4.Text, Coors, Chart_N,
CurIconON, CurIconOFF))
    If CheckBox1.Checked = True Then
WebBrowser1.Navigate(CreateJScript3(1, Point, P_Client_SNS_Unit,
ComboBox3.Text, ComboBox1.Text, ComboBox4.Text, Coors, Chart_N,
CurIconON, CurIconOFF))

    End Sub

End Class
```

