

Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial. Design and deploy a friendly geographic data application with the support of Microsoft Visual Studio and DotSpatial libraries.
Όνοματεπώνυμο Φοιτητή	Ζώτος Χρήστος
Πατρώνυμο	Νικόλαος
Αριθμός Μητρώου	ΜΠΠΛ 14017
Επιβλέπων	Ιωάννης Θεοδωρίδης, Καθηγητής

Ημερομηνία Παράδοσης **Μάιος 2018**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Ιωάννης Θεοδωρίδης
Καθηγητής

Άγγελος Πικράκης
Επίκουρος Καθηγητής

Νικόλαος Πελέκης
Επίκουρος Καθηγητής

Σχεδίαση και ανάπτυξη μίας
φιλικής εφαρμογής γεωγραφικών δεδομένων
με τη βοήθεια του Microsoft Visual Studio
και των βιβλιοθηκών DotSpatial.

Περίληψη

Η παρούσα μεταπτυχιακή διατριβή έχει ως σκοπό την ανάπτυξη μίας φιλικής desktop εφαρμογής με την ονομασία 'MyNauticalMap', της οποίας αντικείμενο είναι η υποτύπωση χρήσιμων πληροφοριών σε έναν ναυτιλιακό χάρτη. Πιο αναλυτικά, η συγκεκριμένη εφαρμογή δομημένη με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial, θα χρησιμοποιεί ελεύθερα χωρικά δεδομένα ναυτιλιακού περιεχομένου, ενώ παράλληλα θα παρέχεται στον χρήστη η δυνατότητα αποθήκευσης δεδομένων κειμένου (π.χ. θερμοκρασία ατμόσφαιρας, θερμοκρασία θαλάσσης, πυκνότητα, ναυτιλιακή κίνηση, ταχύτητα ήχου, τηλεφωνική κάλυψη, αλατότητα) στον χάρτη και της εισαγωγής δεδομένων τύπου εικόνας (π.χ. jpeg, png) σε μία βάση δεδομένων μέσω SQL Server, η οποία θα βρίσκεται εκτός Διαδικτύου. Λόγω των διαφορετικών αναγκών του κάθε τύπου πλοίου, θα πρέπει να υπάρχει η δυνατότητα τροποποίησης των στοιχείων που εισάγονται με ένα φιλικό ως προς τον χρήστη interface. Το αποτέλεσμα της διατριβής θα είναι λοιπόν μία desktop εφαρμογή, που θα παρέχει υδρογραφικές και όχι μόνο πληροφορίες για την ελληνική επικράτεια και θα έχει τη δυνατότητα να εξάγει τα δεδομένα που αποθηκεύονται στον πίνακα χαρακτηριστικών σε ένα αρχείο Excel. Στη συνέχεια το ανωτέρω αρχείο, θα μπορεί να φορτωθεί σε μία βάση δεδομένων στην οποία θα εκτελούνται λειτουργίες αναζήτησης για τη διευκόλυνση του τελικού χρήστη.

Abstract

The aim of this postgraduate dissertation is to develop a friendly desktop application called MyNauticalMap, the purpose of which is to provide useful information on a nautical map. More specifically, this application, built with the help of Microsoft Visual Studio and DotSpatial libraries, will freely use spatial data with maritime content, while allowing the user the ability to store string data (e.g. atmospheric temperature, sea temperature, density, availability of phone network) to a map and image type data (e.g. jpeg, png) to a SQL Server database, which will be offline. Thanks to the different needs of each type of ship, it should be possible to change the data imported with a user-friendly interface. The result of the dissertation will therefore be a desktop application that will provide hydrographic and not only information about the Greek territory and will be able to extract the stored attribute table data in an Excel file. The above file will then be loaded into a database in which search functions will be executed to facilitate the end user.

Περιεχόμενα

Περίληψη	3
Abstract	3
1. Εισαγωγή - Κίνητρο	9
1.1 Γενικά.....	9
1.2 Περίγραμμα εργασίας.....	9
2. Σχετικές εργασίες.....	10
2.1 Εισαγωγή.....	10
2.2 QGIS.....	10
2.3 ArcGIS.....	11
2.4 MapWindow.....	12
2.5 HydroDesktop.....	13
2.6 Η μεταπτυχιακή διατριβή «Δημιουργία ενός Ολοκληρωμένου Συστήματος παροχής χωρικής πληροφορίας, βασισμένο σε τεχνολογίες ανοιχτού κώδικα και ελεύθερα δεδομένα» του Δεληγιάννη Ιωάννη του Σπύρου στο Πανεπιστήμιο Πειραιώς με επιβλέποντα Καθηγητή τον κ. Ιωάννη Θεοδωρίδη	14
3. Αντικείμενο εργασίας.....	15
3.1 Τι είναι τα GIS (GEOGRAPHIC INFORMATION SYSTEMS)	16
3.2 Πως αναπαρίσταται η χωρική πληροφορία.....	17
3.2.1 Διανυσματικά δεδομένα.....	17
3.2.2 Ψηφιδωτά δεδομένα.....	19
3.2.3 Σύγκριση Ψηφιδωτών (raster) και διανυσματικών (vector) δεδομένων	20
3.3 Χωρικά συστήματα αναφοράς.....	21
3.3.1 Το παγκόσμιο σύστημα αναφοράς.....	21
3.3.2 Το ελληνικό σύστημα αναφοράς	22
3.3.3 EPSG:3857 - WGS84 Web Mercator (Auxiliary Sphere)	22
3.4 Ανάλυση απαιτήσεων.....	22
3.5 Σχεδιασμός.....	24
3.6 Εργαλεία Ανάπτυξης Εφαρμογής.....	28
3.6.1 Microsoft Visual Studio	28
3.6.2 Windows Forms.....	29
3.6.3 .NET Framework	29
3.6.4 Microsoft SQL Server	31
3.6.5 Microsoft Access Database Engine 2010 Redistributable	31
3.6.6 DotSpatial.....	31
3.6.7 Microsoft Excel	32
4. Δόμηση Εφαρμογής	33
4.1 Κατασκευή εφαρμογής	33
4.2 Παρουσίαση του συνολικού κώδικα της εφαρμογής	64
4.2.1 Κώδικας φόρμας form1	64
4.2.2 Κώδικας φόρμας form2	74

4.2.3 Κώδικας φόρμας form3	75
4.2.4 Κώδικας φόρμας form4	76
5. Περιγραφή αναπτυχθείσας εφαρμογής	78
5.1 Οθόνη Σύνδεσης χρήστη.....	78
5.2 Περιγραφή των βασικών λειτουργιών.....	80
6. Συμπεράσματα	93
7. Μελλοντικές επεκτάσεις.....	93
7.1 Σύνθετη αναζήτηση στη βάση δεδομένων του SQL Server	93
7.2 GPS	94
7.3 Δημιουργία μίας βάσης δεδομένων.....	94
7.4 Εξαγωγή πίνακα χαρακτηριστικών από polygon και linestring layer σε αρχείο Excel	94
8. Βιβλιογραφία.....	95

Κατάλογος εικόνων

Εικόνα 2.1 :QGIS (QUANTUM GEOGRAPHIC INFORMATION SYSTEM)	11
Εικόνα 2.2: ArcGIS	12
Εικόνα 2.3: MapWindow	13
Εικόνα 2.4: HydroDesktop	14
Εικόνα 3.1: Σημεία, γραμμές, πολύγωνα	17
Εικόνα 3.2: Vector και Raster δεδομένα.....	17
Εικόνα 3.3: Διανυσματικά δεδομένα	19
Εικόνα 3.4: Ψηφιδωτά δεδομένα	20
Εικόνα 3.5: Ελλειψοειδές προβολικό σύστημα.....	22
Εικόνα 3.6: Λογικό διάγραμμα ER ΒΔ στοιχεία χρήστη	25
Εικόνα 3.7: Λογικό διάγραμμα ER ΒΔ εικόνων χρήστη	25
Εικόνα 3.8: Λογικό διάγραμμα ER ΒΔ αρχείου .Dbf.....	26
Εικόνα 3.9: Διάγραμμα περιπτώσεων χρήσης ενός χρήστη	27
Εικόνα 3.10: Microsoft Visual Studio	28
Εικόνα 3.11: Windows Forms.....	29
Εικόνα 3.12: .Net Framework 4.0.....	30
Εικόνα 3.13: Microsoft SQL Server	31
Εικόνα 4.1: Δημιουργία ενός νέου έργου στο Visual Studio σε γλώσσα C # και Windows πλατφόρμα	34
Εικόνα 4.2: Δημιουργία ενός νέου έργου στο Visual Studio σε γλώσσα C # και Windows πλατφόρμα	34
Εικόνα 4.3: Προσθέτουμε μία καινούργια καρτέλα με το όνομα DotSpatial για να εισάγουμε το αρχείο DotSpatial.Controls.dll	35
Εικόνα 4.4: Η καρτέλα των εργαλείων DotSpatial.....	35
Εικόνα 4.5: Εισαγωγή του αρχείου DotSpatial.Controls.dll	36
Εικόνα 4.6: Επιλογή του αρχείου DotSpatial.Controls.dll από τα αρχεία που κατεβάσαμε από το Διαδίκτυο	36

Σχεδίαση και ανάπτυξη μίας
φιλικής εφαρμογής γεωγραφικών δεδομένων
με τη βοήθεια του Microsoft Visual Studio
και των βιβλιοθηκών DotSpatial.

Εικόνα 4.7: Επιλογή του αρχείου DotSpatial.Controls.dll από τα αρχεία που κατεβάσαμε από το Διαδίκτυο	37
Εικόνα 4.8: Εργαλεία DotSpatial στο toolbox του Visual Studio	37
Εικόνα 4.9: Διαδικασία εισαγωγής References στο νέο project	38
Εικόνα 4.10: Διαδικασία εισαγωγής References στο νέο project	38
Εικόνα 4.11: Επιλογή του συνόλου των dll εκτός του Dotspatial.Controls.dll το οποίο έχει ήδη προστεθεί προηγουμένως	39
Εικόνα 4.12: Στάδια δημιουργίας ενός νέου project	39
Εικόνα 4.13: Επιλογή του AppManager από τα εργαλεία	40
Εικόνα 4.14: Εμφάνιση του AppManager κάτω από τη φόρμα	40
Εικόνα 4.15 Διαδικασία εισαγωγής κώδικα	41
Εικόνα 4.16: Εμφάνιση του κώδικα	41
Εικόνα 4.17: Στάδια δημιουργίας ενός νέου project	42
Εικόνα 4.18: SpatialDockManager	42
Εικόνα 4.19: TabControl και Map	43
Εικόνα 4.20: Επιλογή στις ιδιότητες των dock TabControl και Map σε Fill	43
Εικόνα 4.21: Επιλογή στις ιδιότητες του dock του Legend1 σε Fill	44
Εικόνα 4.22: Δημιουργία ενός φακέλου Plugins στο bin/Debug του project μας	44
Εικόνα 4.23: Αντιγραφή του φακέλου tools στον φάκελο plugins	45
Εικόνα 4.24: Σύνδεση των εργαλείων μεταξύ τους	45
Εικόνα 4.25: Σύνδεση των εργαλείων μεταξύ τους	46
Εικόνα 4.26: Ο κώδικας για την εμφάνιση των εργαλείων	46
Εικόνα 4.27: Δημιουργία καρτέλας pictures data	47
Εικόνα 4.28: Εισαγωγή κώδικα για την δημιουργία νέας σύνδεσης στη βάση δεδομένων των εικόνων	48
Εικόνα 4.29: Εισαγωγή κώδικα για το Browse	49
Εικόνα 4.30: Εισαγωγή κώδικα για το Save	50
Εικόνα 4.31: Εισαγωγή κώδικα για το Save	51
Εικόνα 4.32: Εισαγωγή κώδικα για το View	52
Εικόνα 4.33: Εισαγωγή κώδικα για το Clear	53
Εικόνα 4.34: Δημιουργία μίας νέας φόρμας form2 για την σύνδεση χρήστη	54
Εικόνα 4.35: Δημιουργία μίας νέας καρτέλας με το όνομα export.....	56
Εικόνα 4.36: Επιλογή ιδιοτήτων του Project	63
Εικόνα 4.37: Επιλογή .Net Framework 4.5 ως target framework.....	63
Εικόνα 4.38: Επιλογή x64 ως Platform target.	64
Εικόνα 5.1: Το αρχικό παράθυρο της εφαρμογής	79
Εικόνα 5.2: Εμφάνιση του μηνύματος αν δεν είναι συμπληρωμένα και τα δύο πεδία	79
Εικόνα 5.3: Εμφάνιση μηνύματος λάθους εισαγωγής στοιχείων.....	80
Εικόνα 5.4: Αρχικό παράθυρο της εφαρμογής	80
Εικόνα 5.5: Άνοιγμα ενός υπάρχοντος project στην εφαρμογή	81
Εικόνα 5.6: Φόρτωση ενός χάρτη στην εφαρμογή	81
Εικόνα 5.7: Εμφάνιση ιδιοτήτων ενός επιπέδου (layer)	82

Εικόνα 5.8: Εργαλειοθήκη της εφαρμογής	82
Εικόνα 5.9: Η καρτέλα Pictures Data	83
Εικόνα 5.10: Η καρτέλα Pictures Data	83
Εικόνα 5.11: Προσθήκη ναυτιλιακής πληροφορίας στο χάρτη	84
Εικόνα 5.12: Εμφάνιση ιδιοτήτων ενός επιπέδου (layer)	84
Εικόνα 5.13: Αλλαγή ρυθμίσεων ενός επιπέδου (layer).....	85
Εικόνα 5.14: Εμφάνιση σημείων στον χάρτη με διαφορετικό εικονίδιο	85
Εικόνα 5.15: Ανάγνωση του πίνακα ιδιοτήτων (attribute table) για το σημείο.....	86
Εικόνα 5.16: Εισαγωγή Ανάγνωση ID στην καρτέλα Pictures Data για την εμφάνιση αποθηκευμένης εικόνας.....	86
Εικόνα 5.17: Εμφάνιση αποθηκευμένης εικόνας.....	87
Εικόνα 5.18: Εμφάνιση μηνύματος σφάλματος σε περίπτωση αποθήκευσης της ήδη αποθηκευμένης εικόνας.....	87
Εικόνα 5.19: Εμφάνιση του συνόλου των αποθηκευμένων ναυτιλιακών πληροφοριών	88
Εικόνα 5.20: Εξαγωγή δεδομένων ενός χάρτη.....	88
Εικόνα 5.21: Εξαγωγή επιλεγμένων δεδομένων ενός χάρτη	89
Εικόνα 5.22: Εμφάνιση μηνύματος αντικατάστασης υπάρχοντος αρχείου Excel	89
Εικόνα 5.23: Εξαγωγή δεδομένων σε αρχείο Excel με εμφάνιση αντίστοιχου μηνύματος επιτυχίας.....	90
Εικόνα 5.24: Το αρχείο Excel	90
Εικόνα 5.25: Εμφάνιση μηνύματος επιτυχίας της εισαγωγής του αρχείου στη ΒΔ	91
Εικόνα 5.26: Απεικόνιση των δεδομένων στη βάση δεδομένων.....	91
Εικόνα 5.27: Αναζήτηση με χρήση μόνο λέξης κλειδιού	92
Εικόνα 5.28: Αναζήτηση με τη χρήση διαστήματος μεταξύ δύο ημερομηνιών λήψης στοιχείων	92
Εικόνα 5.29: Αναζήτηση με τη χρήση διαστήματος μεταξύ δύο ημερομηνιών λήψης στοιχείων και λέξης κλειδιού	93

Περιεχόμενα πινάκων

Πίνακας 1: Συγκριτικός πίνακας εργασιών	15
Πίνακας 2: Συγκριτικός πίνακας χαρακτηριστικών raster και vector δεδομένων	21

Ευχαριστίες

Η παρούσα μεταπτυχιακή διατριβή εκπονήθηκε στο Πανεπιστήμιο Πειραιά, στα πλαίσια του προγράμματος Σπουδών του Π.Μ.Σ. «Πληροφορική» του τμήματος Πληροφορικής.

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα της παρούσας μελέτης Ιωάννη Θεοδωρίδη, Καθηγητή του Πανεπιστημίου Πειραιά και τον υποψήφιο διδάκτορα Γ. Κοντούλη, για την υποστήριξη και την καθοδήγηση που μου παρείχαν σε όλη τη διάρκεια της μεταπτυχιακής μου διατριβής. Θα ήθελα να ευχαριστήσω θερμά και τους συνεργάτες μου από τον χώρο εργασίας μου, για τις ιδέες που μου ενέπνευσαν και που τελικά οδήγησαν στον εμπλουτισμό της εφαρμογής με επιπλέον δυνατότητες.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου και τη γυναίκα μου, για τη στήριξη που μου προσέφεραν, ώστε να μπορώ αφιερώνω περισσότερο χρόνο και ενέργεια στους στόχους που θέτω.

Εισαγωγή - Κίνητρο

1.1 ΓΕΝΙΚΑ

Η παρούσα μεταπτυχιακή διατριβή έχει ως σκοπό την ανάπτυξη μίας φιλικής και εκτός δικτύου desktop εφαρμογής, με την ονομασία 'MyNauticalMap', της οποίας αντικείμενο είναι η υποτύπωση χρήσιμων πληροφοριών σε έναν ναυτιλιακό χάρτη. Εκπονείται στα πλαίσια του προγράμματος Σπουδών του Π.Μ.Σ. «Πληροφορική» του τμήματος Πληροφορικής στο Πανεπιστήμιο Πειραιά.

Για τη δόμηση της εφαρμογής χρησιμοποιήθηκε το ολοκληρωμένο περιβάλλον ανάπτυξης IDE (Integrated Development Environment) Visual Studio, η ανοιχτού λογισμικού βιβλιοθήκη DotSpatial για γεωγραφικά συστήματα πληροφοριών, το Microsoft Access Database Engine 2010 Redistributable, το Microsoft Excel και ο SQL Server για την αποθήκευση των εικόνων, του πίνακα χαρακτηριστικών και των δεδομένων username και password της εφαρμογής.

Η εφαρμογή θα χρησιμοποιεί ελεύθερα χωρικά δεδομένα ναυτιλιακού περιεχομένου, ενώ παράλληλα θα παρέχεται στον χρήστη η δυνατότητα αποθήκευσης δεδομένων (π.χ. θερμοκρασία ατμόσφαιρας, θερμοκρασία θαλάσσης, πυκνότητα, αλατότητα, ταχύτητα ήχου, ναυτιλιακή κίνηση, τηλεφωνική κάλυψη, θαλάσσια ρεύματα, κινδύνους κλπ.) στον χάρτη και της εισαγωγής δεδομένων τύπου εικόνας (π.χ. jpeg, png) σε μία βάση δεδομένων μέσω SQL Server, η οποία θα βρίσκεται εκτός Διαδικτύου. Λόγω των διαφορετικών αναγκών του κάθε τύπου πλοίου, θα υπάρχει η δυνατότητα τροποποίησης των στοιχείων που εισάγονται με ένα φιλικό ως προς τον χρήστη interface.

Το αποτέλεσμα της διατριβής θα είναι λοιπόν μία desktop εφαρμογή, που θα παρέχει υδρογραφικές και όχι μόνο πληροφορίες στον χρήστη και θα έχει και τη δυνατότητα να εξάγει τα δεδομένα που αποθηκεύονται στον πίνακα Attribute Table σε ένα αρχείο Excel. Στη συνέχεια το ανωτέρω αρχείο, θα μπορεί να φορτωθεί σε μία βάση δεδομένων εκτός Διαδικτύου, στην οποία θα εκτελούνται λειτουργίες αναζήτησης. Το αρχείο Excel θα δίνει τη δυνατότητα στον χρήστη να έχει πρόσβαση στα δεδομένα που έχει στην κατοχή του, χωρίς τη χρήση της εφαρμογής. Τυχόν φωτογραφικό υλικό, θα μπορεί να εισαχθεί σε μια βάση δεδομένων με χρήση SQL Server.

Για κάθε πλοίο θα υπάρχει ένας συγκεκριμένος χρήστης που θα χειρίζεται την εφαρμογή με τον ρόλο του χρήστη (user). Για τη σύνδεση (login) των χρηστών θα τηρείται πίνακας με το όνομα «Login» που θα υπάρχει στη βάση δεδομένων της εφαρμογής.

Οι βάσεις δεδομένων θα βρίσκονται εκτός Διαδικτύου. Κάθε πλοίο που φορτώνει πληροφορίες στις βάσεις δεδομένων του, για κάποιο λιμένα που επισκέφτηκε ή κάποιο αγκυροβόλι που έκανε, είναι αναγκαίο να μπορεί να μεταφορτώνει τα δεδομένα του σε μία κεντρική βάση δεδομένων. Οι πληροφορίες που συλλέγονται με αυτόν τον τρόπο, θα μπορούν στη συνέχεια να μεταφορτώνονται σε όλα τα πλοία με κάποιο φορητό μέσο αποθήκευσης.

Επιπρόσθετα η εφαρμογή αυτή, θα έχει τη δυνατότητα να εξελιχθεί ακόμα περισσότερο από οποιονδήποτε προγραμματιστή και να προσαρμοστεί στις εκάστοτε ανάγκες.

Στόχος της εφαρμογής είναι ο χειρισμός και η αξιοποίηση των χωρικών δεδομένων σε ένα ιδιαίτερα φιλικό περιβάλλον και η εύκολη μεταφορά τους σε όλους τους χρήστες.

1.2 ΠΕΡΙΓΡΑΦΜΑ ΕΡΓΑΣΙΑΣ

Στο πρώτο μέρος θα παρουσιαστούν εργασίες και προγράμματα που υπάρχουν διαθέσιμα στην αγορά τα οποία έχουν κοινά σημεία με την παρούσα μεταπτυχιακή διατριβή. Συγκεκριμένα, στην αρχή θα αναφερθεί μία γενική περιγραφή των δεδομένων και στη συνέχεια θα παρουσιαστούν τα προγράμματα QGIS, ARCGIS, MapWindow, HydroDesktop και η

Σχεδίαση και ανάπτυξη μίας
φιλικής εφαρμογής γεωγραφικών δεδομένων
με τη βοήθεια του Microsoft Visual Studio
και των βιβλιοθηκών DotSpatial.

μεταπτυχιακή διατριβή «Δημιουργία ενός Ολοκληρωμένου Συστήματος παροχής χωρικής πληροφορίας, βασισμένο σε τεχνολογίες ανοιχτού κώδικα και ελεύθερα δεδομένα» του Δεληγιάννη Ιωάννη του Σπύρου στο Πανεπιστήμιο Πειραιώς, με επιβλέποντα Καθηγητή τον κ. Ιωάννη Θεοδωρίδη.

Στο δεύτερο μέρος θα δοθεί έμφαση στο αντικείμενο της εργασίας και θα περιγραφούν τα συστήματα γεωγραφικών πληροφοριών GIS, το πως αναπαρίσταται η χωρική πληροφορία και τα χωρικά συστήματα αναφοράς. Στη συνέχεια, θα παρουσιαστεί η ανάλυση απαιτήσεων του προγράμματος που θα αναπτυχθεί, ο σχεδιασμός του καθώς και τα εργαλεία που χρησιμοποιήθηκαν για την υλοποίησή του.

Στο τρίτο μέρος θα αναπτυχθεί ο τρόπος κατασκευής της εφαρμογής, παρουσιάζοντας βήμα προς βήμα με εικόνες και κώδικα την υλοποίηση.

Στο τέταρτο μέρος θα περιγραφεί η αναπτυχθείσα εφαρμογή. Συγκεκριμένα, θα παρουσιαστεί το σύνολο των βασικών λειτουργιών του προγράμματος με εικόνες και επεξηγήσεις.

Στο τέλος της διατριβής, θα διατυπωθούν τα συμπεράσματα που προκύπτουν από τις λειτουργίες της εφαρμογής, μελλοντικές επεκτάσεις της εφαρμογής και θα παρουσιαστούν οι βιβλιογραφικές αναφορές.

Σχετικές εργασίες

2.1 ΕΙΣΑΓΩΓΗ

Τα χωρικά δεδομένα αναφέρονται σε όλους τους τύπους αντικειμένων ή στοιχείων δεδομένων που υπάρχουν σε γεωγραφικό χώρο ή ορίζοντα. Επιτρέπουν την παγκόσμια εύρεση και εντοπισμό ατόμων ή συσκευών οπουδήποτε στον κόσμο.

Τα χωρικά δεδομένα μπορούν να ταξινομηθούν ως ψηφιδωτά ή διανυσματικά δεδομένα και παρέχουν πληροφορίες σχετικά με γεωγραφικές ή χωρικές τοποθεσίες. Χρησιμοποιούνται σε γεωγραφικά συστήματα πληροφοριών (GIS) και σε άλλες υπηρεσίες γεωγραφικής τοποθέτησης ή εντοπισμού θέσης. Τα χωρικά δεδομένα αποτελούνται από σημεία, γραμμές, πολύγωνα και άλλα αρχέτυπα γεωγραφικών και γεωμετρικών δεδομένων, τα οποία μπορούν να χαρτογραφηθούν ανά τοποθεσία, να αποθηκευτούν ως μεταδεδομένα ή να χρησιμοποιηθούν από ένα σύστημα επικοινωνίας για τον εντοπισμό συσκευών τελικού χρήστη.

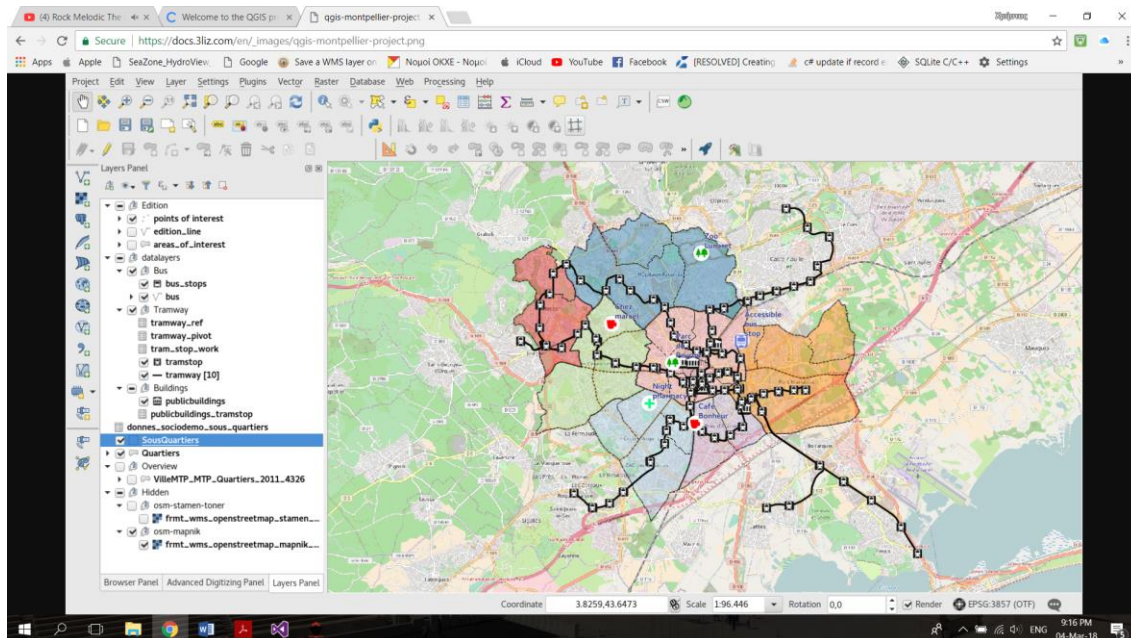
Εξαιτίας της συνεχούς αύξησης χρήσης του GPS (global positioning system), τα χωρικά δεδομένα καθίστανται πλέον ευρέως γνωστά και υπάρχει πληθώρα εφαρμογών για φορητές και όχι μόνο συσκευές που καλύπτουν διαφορετικού είδους ανάγκες χρηστών.

2.2 QGIS (QUANTUM GEOGRAPHIC INFORMATION SYSTEM)

Το QGIS (παλαιότερα γνωστό ως Quantum GIS) είναι μια ελεύθερη και ανοιχτού κώδικα εφαρμογή γεωγραφικών πληροφοριών (GIS) που υποστηρίζει την προβολή, την επεξεργασία και την ανάλυση γεωχωρικών δεδομένων.

Το QGIS λειτουργεί ως λογισμικό συστήματος γεωγραφικών πληροφοριών (GIS), επιτρέποντας στους χρήστες να αναλύουν και να επεξεργάζονται χωρικές πληροφορίες καθώς και να συνθέτουν και να εξάγουν χάρτες. Το QGIS υποστηρίζει τόσο τα ψηφιδωτά όσο και τα διανυσματικά δεδομένα. Τα διανυσματικά δεδομένα αποθηκεύονται με τη μορφή σημείων, γραμμών ή πολυγώνων.

Ως εφαρμογή ελεύθερου λογισμικού, το QGIS μπορεί να τροποποιηθεί ελεύθερα για να εκτελέσει διαφορετικές ή πιο εξειδικευμένες λειτουργίες. Οι εφαρμογές QGIS Browser και QGIS Server, αποτελούν δύο παραδείγματα της ανωτέρω δυνατότητας.



Εικόνα 2.1 – QGIS (QUANTUM GEOGRAPHIC INFORMATION SYSTEM).

2.3 ARCGIS

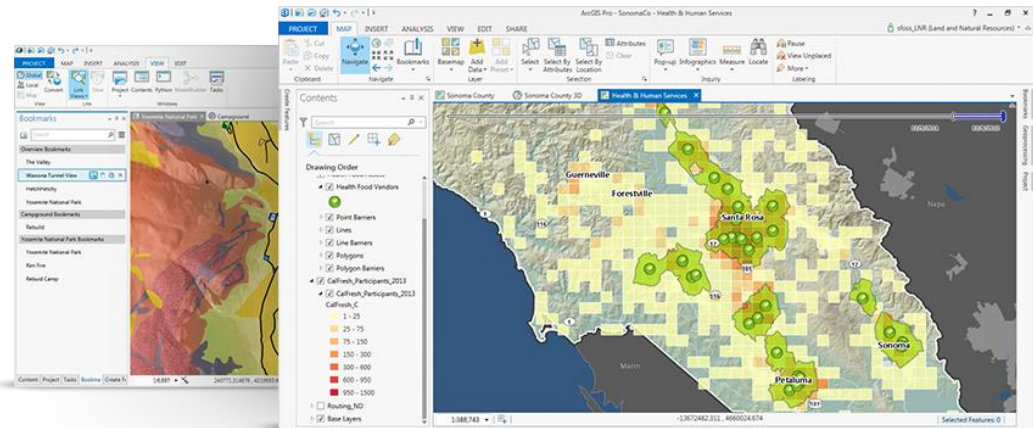
Το ArcGIS είναι ένα γεωγραφικό σύστημα πληροφοριών (GIS) για την επεξεργασία χαρτών και γεωγραφικών πληροφοριών. Χρησιμοποιείται για τη δημιουργία και τη χρήση χαρτών, τη σύνταξη γεωγραφικών δεδομένων, την ανάλυση χαρτογραφημένων πληροφοριών, την κοινή χρήση και την ανακάλυψη γεωγραφικών πληροφοριών, τη χρήση χαρτών και γεωγραφικών πληροφοριών σε ένα πλήθος εφαρμογών και τη διαχείριση γεωγραφικών πληροφοριών σε μια βάση δεδομένων. Το σύστημα παρέχει μια υποδομή για τη δημιουργία χαρτών και γεωγραφικών πληροφοριών σε έναν οργανισμό, σε μια κοινότητα και ελεύθερα στο Διαδίκτυο.

Το ArcGIS περιλαμβάνει το ακόλουθο λογισμικό επιφάνειας εργασίας των Windows:

- ArcReader, που επιτρέπει σε κάποιον να προβάλλει και να αναζητά τους χάρτες που έχουν δημιουργηθεί με τα άλλα προϊόντα του ArcGIS.
- Το ArcGIS for Desktop, το οποίο διαθέτει άδεια χρήσης σε τρία επίπεδα λειτουργικότητας:
 - ArcGIS for Desktop Basic (παλαιότερα γνωστό ως ArcView), το οποίο επιτρέπει σε κάποιον να προβάλλει χωρικά δεδομένα, να δημιουργεί χάρτες και να εκτελεί βασικές χωρικές αναλύσεις.
 - Το ArcGIS for Desktop Standard (παλαιότερα γνωστό ως ArcEditor), το οποίο εκτός από τη λειτουργικότητα του ArcView περιλαμβάνει πιο προηγμένα εργαλεία για το χειρισμό των shapefiles και των γεωγραφικών δεδομένων.
 - Το ArcGIS for Desktop Advanced (πρώην ArcInfo), το οποίο περιλαμβάνει δυνατότητες επεξεργασίας και ανάλυσης δεδομένων.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

Υπάρχουν επίσης προϊόντα ArcGIS που βασίζονται σε διακομιστές, καθώς και προϊόντα ArcGIS για PDA. Οι επεκτάσεις μπορούν να αγοραστούν ξεχωριστά για να αυξήσουν τη λειτουργικότητα του ArcGIS.



Εικόνα 2.2 – ARCGIS.

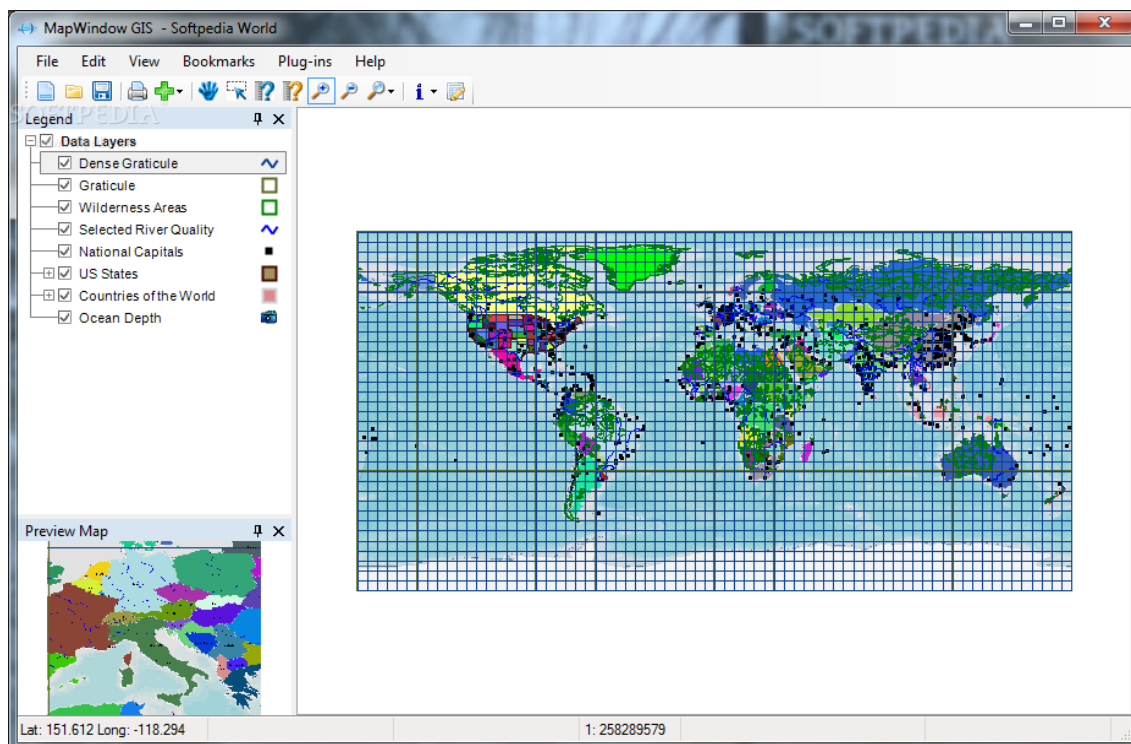
2.4 MapWindow

Το MapWindow GIS είναι μια εφαρμογή ανοιχτού κώδικα GIS και περιέχει ένα σύνολο προγραμματιζόμενων εργαλείων χαρτογράφησης. Έχει υιοθετηθεί από την Υπηρεσία Προστασίας του Περιβάλλοντος των Ηνωμένων Πολιτειών ως η κύρια πλατφόρμα GIS για τα λογισμικά ανάλυσης και μοντελοποίησης των λεκανών BASINS (Better Assessment Science Integration Point και Nonpoint Survey).

Το MapWindow GIS διανέμεται ως εφαρμογή ανοιχτού κώδικα υπό την άδεια διανομής του Mozilla Public License. Έχει τη δυνατότητα να επαναπρογραμματιστεί για να εκτελέσει διαφορετικές ή πιο εξειδικευμένες εργασίες. Υπάρχουν επίσης διαθέσιμες προσθήκες για την επέκταση της συμβατότητας και της λειτουργικότητας. Η εφαρμογή βασίζεται στην τεχνολογία Microsoft .NET. Οι ενημερώσεις για το MapWindow GIS κυκλοφορούν τακτικά από ομάδες φοιτητών και εθελοντών προγραμματιστών.

Το MapWindow εκτελεί περίπου το 90% των απαιτήσεων των χρηστών των GIS περιλαμβάνοντας την αναγνώριση χαρακτηριστικών, των εργαλείων επεξεργασίας χάρτη και εκτύπωσης. Περιλαμβάνει κάποια εργαλεία υψηλότερου επιπέδου, όπως το TauDEM για την αυτόματη οριοθέτηση της λεκάνης απορροής. Το MapWindow GIS, περιέχει ένα σύστημα γεωγραφικών πληροφοριών ελεύθερου και ανοιχτού κώδικα (GIS) με επεκτάσιμη αρχιτεκτονική plugin, ένα GIS ActiveX control και μια βιβλιοθήκη γεωγραφικών συστημάτων C# που ονομάζεται DotSpatial.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



Εικόνα 2.3 – MapWindow.

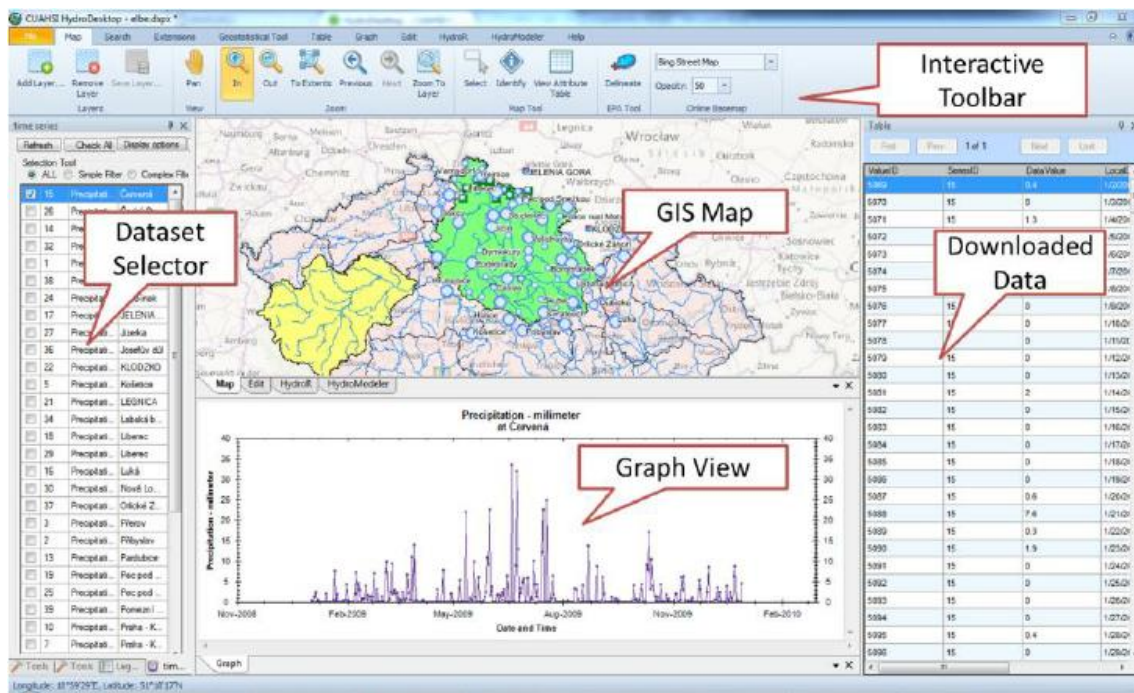
2.5 HydroDesktop

Το HydroDesktop είναι μια εφαρμογή ανοιχτού κώδικα υδρολογικής πληροφορίας (HIS), η οποία βοηθάει τους χρήστες να ανακαλύπτουν, να χρησιμοποιούν και να διαχειρίζονται δεδομένα υδρολογικών δεδομένων χρονοσειρών. Οι λειτουργίες του περιλαμβάνουν τη διερεύνηση, τη λήψη, την οπτικοποίηση, τη γραφική παράσταση, την ανάλυση και τη μοντελοποίηση των δεδομένων.

Το HydroDesktop χρησιμοποιεί μία βιβλιοθήκη GIS ανοιχτού κώδικα που ονομάζεται DotSpatial και ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων (relational database management system) ανοιχτού κώδικα που ονομάζεται SQLite, για να παρέχει μια βάση πληροφοριών με την οποία οι χρήστες μπορούν να συλλέγουν και να συνθέτουν αποτελεσματικά μεγάλο όγκο υδρολογικών δεδομένων.

Το HydroDesktop έχει σχεδιαστεί για να αξιοποιεί τη χρησιμότητα των υπηρεσιών Web WaterOneFlow που αναπτύχθηκαν από την Κοινπραξία Πανεπιστημίων για την Υδρολογική Επιστήμη, Inc. (CUAHSI) Hydrologic Information Systems project (CUAHSI-HIS). Το αποτέλεσμα είναι ένα σύστημα χωροθέτησης, που διευκολύνει τη συγκέντρωση χρονικών σειρών δεδομένων παρατήρησης που περιγράφουν το υδατικό περιβάλλον.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



Εικόνα 2.4 – HydroDesktop.

2.6 Η μεταπτυχιακή διατριβή «Δημιουργία ενός Ολοκληρωμένου Συστήματος παροχής χωρικής πληροφορίας, βασισμένο σε τεχνολογίες ανοιχτού κώδικα και ελεύθερα δεδομένα» του Δεληγιάννη Ιωάννη του Σπύρου στο Πανεπιστήμιο Πειραιώς με επιβλέποντα Καθηγητή τον κ. Ιωάννη Θεοδωρίδη.

Η Μεταπτυχιακή διατριβή «Δημιουργία ενός Ολοκληρωμένου Συστήματος παροχής χωρικής πληροφορίας, βασισμένο σε τεχνολογίες ανοιχτού κώδικα και ελεύθερα δεδομένα» του Δεληγιάννη Ιωάννη του Σπύρου στο Πανεπιστήμιο Πειραιώς με επιβλέποντα Καθηγητή τον κ. Ιωάννη Θεοδωρίδη, παρουσιάζει αρκετά κοινά σημεία με την παρούσα εργασία, και ως εκ τούτου κρίνεται σκόπιμη μια σύντομη παρουσίαση του περιεχομένου της, προκειμένου να γίνει μία σχετική σύνδεση με το αντικείμενο της παρούσας εργασίας.

Στην προαναφερθείσα μεταπτυχιακή διατριβή, χρησιμοποιώντας ελεύθερα χωρικά δεδομένα και εφαρμογές ανοιχτού κώδικα (εκτός από την ανάπτυξη της ιστοσελίδας που έγινε με εμπορικό λογισμικό), παρέχεται η δυνατότητα δρομολόγησης και εύρεσης διευθύνσεων στο Διαδίκτυο καθώς και η δυνατότητα αναβάθμισης δεδομένων, όταν νέα δεδομένα εμφανίζονται στις αντίστοιχες πηγές. Για την εισαγωγή των δεδομένων στη βάση δεδομένων, χρησιμοποιήθηκε η εφαρμογή osm2po, για την αποθήκευση των δεδομένων χρησιμοποιήθηκε το αντικείμενο-σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων (object-relational database management system, ORDBMS) PostgreSQL, με επεκτάσεις τις εφαρμογές PostGIS και PgRouting, και για την οπτικοποίηση των δεδομένων ο διακομιστής Geoserver. Το ολοκληρωμένο IDE που χρησιμοποιήθηκε είναι το Microsoft Visual Studio και οι συνδέσεις της βάσης δεδομένων και του διακομιστή χωρικών δεδομένων με το IDE, έγιναν με τις εφαρμογές Npgsql connection και SharpMap αντίστοιχα. Το αποτέλεσμα της διατριβής, είναι μία web-based εφαρμογή με φιλικό και εύκολο για το χρήστη περιβάλλον διεπαφής, που παρέχει πληροφορίες

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

για περιοχές, οδούς (διευθύνσεις και διαδρομές), σημεία ενδιαφέροντος, με τρόπο κατανοητό και προπάντων γρήγορο και αξιόπιστο.

Χαρακτηριστικά	Παρούσα εργασία	Μεταπτυχιακή διατριβή [1]
Τύπος Εφαρμογής	Desktop	Web based
Χωρικά Δεδομένα	Θαλάσσια	Γη
Βάση Δεδομένων	Ναι (Offline)	Ναι
Δυνατότητα μεταφόρτωσης νέων δεδομένων από το διαδίκτυο	Όχι	Ναι
Δυνατότητα Εισαγωγής Πληροφοριών από χρήστη	Ναι	Όχι
Ολοκληρωμένο IDE	Microsoft Visual Studio	Microsoft Visual Studio
Αποθήκευση Δεδομένων	SQL Server, Excel	SQL Server
Επεκτάσεις	-	PostGIS, PgRouting

Πίνακας 1. Συγκριτικός πίνακας εργασιών.

Αντικείμενο εργασίας

Αντικείμενο της εργασίας είναι η αξιοποίηση και οργάνωση πληροφοριών ναυτιλιακής φύσεως που συλλέγει ο χρήστης (όπως για παράδειγμα ναυτιλιακές πληροφορίες λιμένων και λοιπών περιοχών, τυχόν φωτογραφικό υλικό, μετρήσεις θερμοκρασιών θαλάσσης, μετρήσεις πυκνότητας θαλάσσης, αποθηκευμένα σίγματα, ναυαγία κλπ.). Για τη διευκόλυνση του χρήστη, θα εκτελείται εξαγωγή της βάσης δεδομένων του αρχείου shapefile σε ένα αρχείο Excel, ενώ παράλληλα θα δίνεται και η δυνατότητα εισαγωγής του εξαχθέντος Excel σε μια βάση δεδομένων (η οποία θα βρίσκεται και αυτή εκτός Διαδικτύου). Στην ανωτέρω βάση δεδομένων θα εκτελούνται εξειδικευμένου τύπου αναζητήσεις, με βάση τις ημερομηνίες καταγραφής των δεδομένων και μίας λέξης κλειδί. Η απεικόνιση των γεωγραφικών δεδομένων θα εκτελείται σε ένα φιλικό interface, το οποίο θα παρέχει αρκετές δυνατότητες στον χρήστη. Για την αποθήκευση των δεδομένων εικόνας, των δεδομένων του πίνακα χαρακτηριστικών και των δεδομένων username και password, θα χρησιμοποιηθεί το σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων (relational database management system - RDBMS) SQL Server.

Στο παρόν κεφάλαιο θα παρουσιαστεί μία εισαγωγή στα Γεωγραφικά Συστήματα Πληροφοριών (GIS) και των αντίστοιχων δεδομένων, λειτουργιών και εργαλείων που θα χρησιμοποιηθούν στην εφαρμογή της παρούσας μεταπτυχιακής διατριβής. Στη συνέχεια, θα ακολουθήσει η ανάλυση των απαιτήσεων και ο σχεδιασμός του συστήματος.

3.1 Τι είναι τα GIS (GEOGRAPHIC INFORMATION SYSTEMS)

Τα Συστήματα Γεωγραφικών Πληροφοριών (ΣΓΠ), γνωστά ευρέως και ως GIS (Geographic Information systems) είναι συστήματα διαχείρισης χωρικών δεδομένων (spatial data) και συσχετισμένων ιδιοτήτων. Ο όρος αυτός αναφέρεται σε συστήματα τα οποία έχουν τη δυνατότητα να χειρίζονται γεωγραφικά δεδομένα. Ουσιαστικά τα GIS είναι πληροφοριακά συστήματα (Information systems) που παρέχουν τη δυνατότητα συλλογής, διαχείρισης, αποθήκευσης, επεξεργασίας, ανάλυσης και οπτικοποίησης, σε ψηφιακό περιβάλλον των δεδομένων που σχετίζονται με τον χώρο. Το σύστημα συνολικά δεν περιλαμβάνει μόνο το αντίστοιχο λογισμικό και υλικό που απαιτείται σε όλα τα συστήματα, αλλά επιπροσθέτως περιλαμβάνει και ειδικές συσκευές για τη δημιουργία και την απεικόνιση χαρτών αλλά και συστήματα επικοινωνιών που απαιτούνται για να συνδέσουν τα διάφορα συστατικά από τα οποία αποτελούνται.

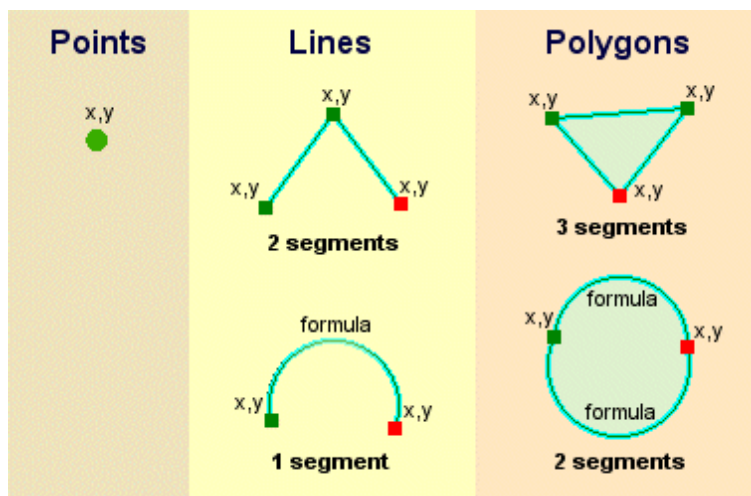
Σε σύγκριση με τους απλούς χάρτες, ένα σύστημα GIS έχει το πλεονέκτημα ότι η αποθήκευση των δεδομένων γίνεται χωριστά από την αναπαράστασή τους. Αυτό έχει σαν αποτέλεσμα τα ίδια δεδομένα να μπορούν να αναπαρασταθούν με διαφορετικούς τρόπους. Για παράδειγμα, μπορούμε να μεγεθύνουμε τον (ψηφιακό πλέον) χάρτη, να εμφανίσουμε συγκεκριμένες μόνο περιοχές, να κάνουμε υπολογισμούς αποστάσεων μεταξύ τοποθεσιών, να δημιουργήσουμε πίνακες που να δείχνουν τα διάφορα χαρακτηριστικά του χάρτη καθώς και να απεικονίσουμε επιπλέον πληροφορία πάνω στο χάρτη εάν χρειαστεί.

Όλα τα δεδομένα σε ένα σύστημα GIS είναι συνδεδεμένα με μια συγκεκριμένη γεωγραφική τοποθεσία της επιφάνειας της γης μέσω ενός συστήματος συντεταγμένων. Τα δεδομένα αυτά συνήθως λέγονται γεωγραφικά ή χαρτογραφικά ή χωρικά (spatial) και μπορεί να συσχετίζονται με μια σειρά από περιγραφικά δεδομένα τα οποία και τα χαρακτηρίζουν μοναδικά. Τα συστήματα GIS αποτυπώνουν χωρικά δεδομένα σε γεωγραφικό ή χαρτογραφικό ή καρτεσιανό σύστημα συντεταγμένων.

Βασικό χαρακτηριστικό των GIS είναι ότι τα χωρικά δεδομένα συνδέονται και με περιγραφικά δεδομένα, π.χ. μια ομάδα σημείων που αναπαριστούν θέσεις πόλεων συνδέεται με ένα πίνακα όπου κάθε εγγραφή εκτός από τη θέση περιέχει πληροφορίες όπως ονομασία, πληθυσμός κλπ.

Όλα αυτά μπορεί να φαίνονται πολύπλοκα. Παρ' όλα αυτά η αλήθεια είναι, πως τα ανωτέρω συστήματα δεν διαφέρουν στη λειτουργία τους από έναν απλό χάρτη. Για παράδειγμα στα διανυσματικά δεδομένα, τα σημεία (points) αναπαριστούν πόλεις, χωριά και γενικότερα οτότητες για τις οποίες η μορφή τους δεν έχει σημασία, οι γραμμές (lines, polylines) αναπαριστούν ποτάμια, δρόμους και τα πολύγωνα (polygons) περικλείουν μια γεωγραφική έκταση. Σε όλα αυτά τα χωρικά δεδομένα, έχουμε τη δυνατότητα να τα συνδέσουμε με περιγραφικά δεδομένα τα οποία ένα πρόγραμμα GIS μπορεί να διαχειριστεί με ευκολία και να προσφέρει δυνατότητες και λειτουργίες που θα χρειαζόταν ώρες κάποιος να τις εκτελέσει χωρίς αυτό.

Ένα από τα πιο συνηθισμένα συστήματα γεωγραφικών συντεταγμένων είναι αυτό του γεωγραφικού μήκους και γεωγραφικού πλάτους. Σε αυτό το σύστημα συντεταγμένων, κάθε τοποθεσία προσδιορίζεται σχετικά με τον ισημερινό και τη γραμμή μηδενικού γεωγραφικού μήκους που περνά από το αστεροσκοπείο Greenwich της Αγγλίας. Υπάρχουν πολλά άλλα γεωγραφικά συστήματα συντεταγμένων και κάθε GIS σύστημα θα πρέπει να υποστηρίζει τη μετατροπή των συντεταγμένων από το ένα σύστημα στο άλλο.

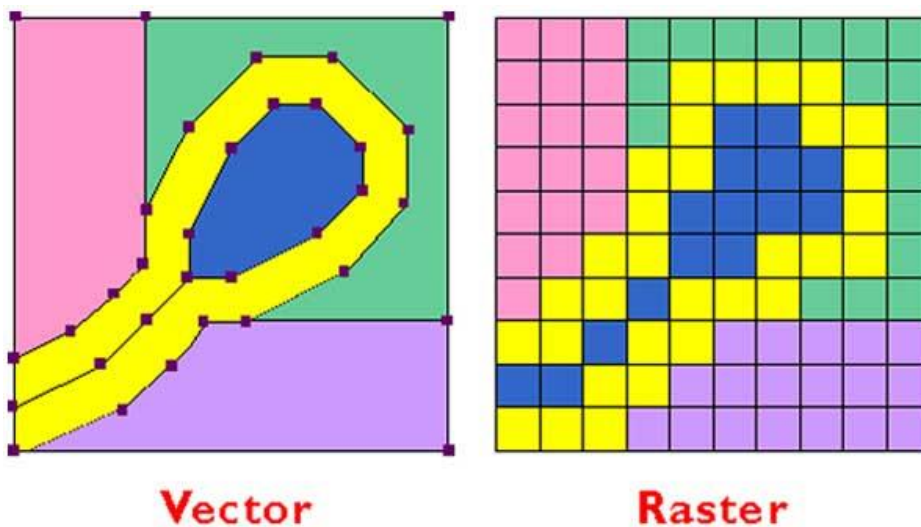


Εικόνα 3.1 – Σημεία, γραμμές, πολύγωνα.

3.2 Πως αναπαρίσταται η χωρική πληροφορία

Η χωρική πληροφορία αναπαρίσταται με δυο τρόπους:

- Ως διανυσματικά δεδομένα με τη μορφή σημείων, γραμμών και πολυγώνων.
- Ως ψηφιδωτά (raster) δεδομένα, οργανωμένα συστηματικά σε κελιά (όπως π.χ. μια ψηφιακή εικόνα).



Εικόνα 3.2 – Vector και Raster δεδομένα.

3.2.1 Διανυσματικά δεδομένα

Στα διανυσματικά δεδομένα τα χαρακτηρισικά καταγράφονται ένα προς ένα, με το σχήμα να ορίζεται από τις αριθμητικές τιμές των ζευγών των συντεταγμένων x και y .

- Ένα σημείο ορίζεται από ένα μόνο ζεύγος τιμών συντεταγμένων x και y .

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

- Μια γραμμή ορίζεται από μια ακολουθία ζευγών συντεταγμένων x και y .
- Μια περιοχή ορίζεται με παρόμοιο τρόπο, μόνο που το πρώτο και το τελευταίο σημείο ενώνονται για να κάνουν ένα πλήρες περίβλημα.

Τα διανυσματικού τύπου δεδομένα μπορούν να θεωρηθούν ως μια λίστα τιμών. Τα συστήματα GIS για να αναπαράγουν αυτά τα δεδομένα διαβάζουν αυτές τις τιμές και σχεδιάζουν γραμμές ενώνοντας τις τοποθεσίες των συντεταγμένων. Ακόμα μπορούν να αποθηκεύσουν επιπρόσθετες πληροφορίες για αυτά τα χαρακτηριστικά. Επιπλέον, μπορούν να μεγεθυνθούν άπειρα χωρίς να χαλάει η ποιότητά τους και να τους δοθεί χρώμα αναλόγως των προτιμήσεων του χρήστη.

Ο τύπος δεδομένων shapfile είναι από τις πιο διαδεδομένες μορφές αποθήκευσης διανυσματικών επιπέδων χάρτη στις εφαρμογές γεωγραφικών συστημάτων πληροφορίας (GIS). Αναπτύχθηκε και εξελίχθηκε από την Esri. Με τον συγκεκριμένο τύπο είναι δυνατή η αποθήκευση ή η απεικόνιση γεωγραφικών δεδομένων σημείων, γραμμών, πολύγωνων σε ένα πολύ μεγάλο εύρος προγραμμάτων όπως και του Dotspatial. Έτσι δίνεται η δυνατότητα να αποθηκευτούν σε ένα χάρτη με μεγάλη ευκολία δεδομένα όπως ποταμοί, λίμνες, υπόγεια ύδατα. Ο τύπος shapfile format χρησιμοποιήθηκε αρχικά στην εφαρμογή ArcView GIS v.2 στις αρχές του 1990s.

Αυτός ο τύπος είναι σχετικά απλός καθώς αποθηκεύει ταυτόχρονα με τα διανυσματικά γεωγραφικά δεδομένα, τον πίνακα χαρακτηριστικών. Το πραγματικό shapfile έχει κατάληξη .shp αλλά είναι ελλιπές για χρήση καθώς είναι υποχρεωτική η χρήση και των αρχείων με επεκτάσεις .shx και .dbf.

Το .shp είναι η ίδια η διανυσματική γεωμετρία χαρακτηριστικών, το .shx είναι ένας δείκτης θέσης της γεωμετρίας των χαρακτηριστικών που επιτρέπει τη γρήγορη αναζήτηση δεδομένων, το .dbf αποτελεί τη βάση δεδομένων των γεωγραφικών δεδομένων σε μορφή dBase IV. Μία ακόμα μορφή δεδομένων η .prj η οποία χρησιμοποιείται στο Dotspatial, περιέχει το σύστημα συντεταγμένων και τις πληροφορίες προβολής με ένα αρχείο απλού κειμένου. Η πρώτη εγγραφή του .shp ανταποκρίνεται στις πρώτες εγγραφές των αρχείων .shx και .dbf. Το αρχείο shapfile μπορεί λοιπόν να περιέχει και άλλους μορφότυπους επιπλέον των τριών υποχρεωτικών.

Μορφότυποι επιπλέον των υποχρεωτικών :

- .prj (projection format) : Το σύστημα συντεταγμένων και οι πληροφορίες προβολής. Είναι ένα αρχείο απλού κειμένου που περιγράφει την προβολή.
- .shn και .sbx: χωρικός δείκτης των χαρακτηριστικών.
- .fbi και .fbi: χωρικός δείκτης των χαρακτηριστικών, ο οποίος είναι μόνο για διάβασμα (read-only).
- .ain και .aih:— ένας δείκτης χαρακτηριστικών των ενεργών πεδίων σε έναν πίνακα.
- .ixs: ένας δείκτης γεωκωδικοποίησης για δεδομένα ανάγνωσης-εγγραφής.
- .mxs: ένας δείκτης γεωκωδικοποίησης για δεδομένα ανάγνωσης-εγγραφής (ODB format).
- .atx: ένας δείκτης χαρακτηριστικών για το αρχείο .dbf με τη μορφή shapfile.columnname.atx (ArcGIS 8 και νεότερη έκδοση).
- .shp.xml: γεωπεριβαλλοντικά μεταδεδομένα σε μορφή XML, όπως το ISO 19115.
- .cpg: χρησιμοποιείται για να καθορίσει τη σελίδα κώδικα (μόνο για .dbf) για τον προσδιορισμό της κωδικοποίησης χαρακτήρων που θα χρησιμοποιηθεί.
- .qix: ένας εναλλακτικός τετραπλός χωρικός δείκτης που χρησιμοποιείται από το λογισμικό MapServer και GDAL / OGR.

Σε κάθε ένα από τα αρχεία .shp, .shx και .dbf, η πρώτη εγγραφή στο αρχείο .shp αντιστοιχεί στην πρώτη εγγραφή στα αρχεία .shx και .dbf, και τα λοιπά.

Σχεδίαση και ανάπτυξη μίας

φιλικής εφαρμογής γεωγραφικών δεδομένων

με τη βοήθεια του Microsoft Visual Studio

και των βιβλιοθηκών DotSpatial.

Για την αποθήκευση των διανυσματικών δεδομένων ισχύουν τα ακόλουθα:

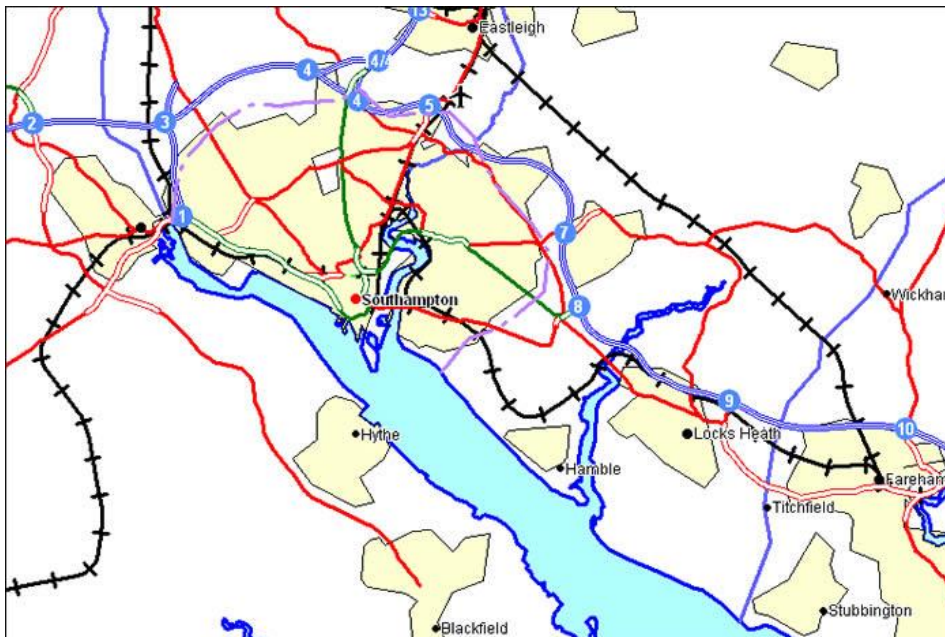
- Το μέγεθος των αρχείων .shp και .dbf δεν μπορεί να υπερβαίνει τα 2 GB - περίπου τα 70 εκατομμύρια σημεία στην καλύτερη περίπτωση. Η μορφή βάσης δεδομένων χαρακτηριστικών για το αρχείο .dbf βασίζεται σε παλαιότερο πρότυπο dBase.
- Μόνο οι πρώτοι 10 χαρακτήρες σε ένα όνομα πεδίου χαρακτηριστικών αναγνωρίζονται για τα αρχεία dbf.
- Κατά την εξαγωγή δεδομένων σε shapfiles από άλλη πηγή δεδομένων, ορισμένα από τα μοναδικά ονόματα πεδίων που είναι μεγαλύτερα από 10 χαρακτήρες μπορεί να γίνουν διπλότυπα αν οι πρώτοι 10 χαρακτήρες δεν είναι μοναδικοί.
- Ανάλογα με τη μορφή dBase, υπάρχει ένα ανώτατο όριο για τον αριθμό των πεδίων που μπορούν να αποθηκευτούν σε ένα αρχείο dbf.

DBF 2 (dBASE II): 32

DBF 3 (dBASE III): 128

DBF 4 (dBASE IV): 255

- Κακή υποστήριξη για ονόματα πεδίων Unicode ή αποθήκευση πεδίων.
- Οι υποστηριζόμενοι τύποι πεδίων και το μέγιστο μέγεθός τους είναι: τύπου κινητής υποδιαστολής (αποθήκευση 13 χαρακτήρων), τύπου ακεραίου (αποθηκευτικός χώρος 4 ή 9 χαρακτήρων), ημερομηνία (χωρίς αποθήκευση χρόνου, αποθήκευση 8 χαρακτήρων) και κείμενο (μέγιστη αποθήκευση 254 χαρακτήρων).
- Οι αριθμοί κινητής υποδιαστολής ενδέχεται να περιέχουν σφάλματα στρογγυλοποίησης, επειδή αποθηκεύονται ως κείμενο.



Εικόνα 3.3 – Διανυσματικά δεδομένα.

3.2.2 Ψηφιδωτά (raster) δεδομένα

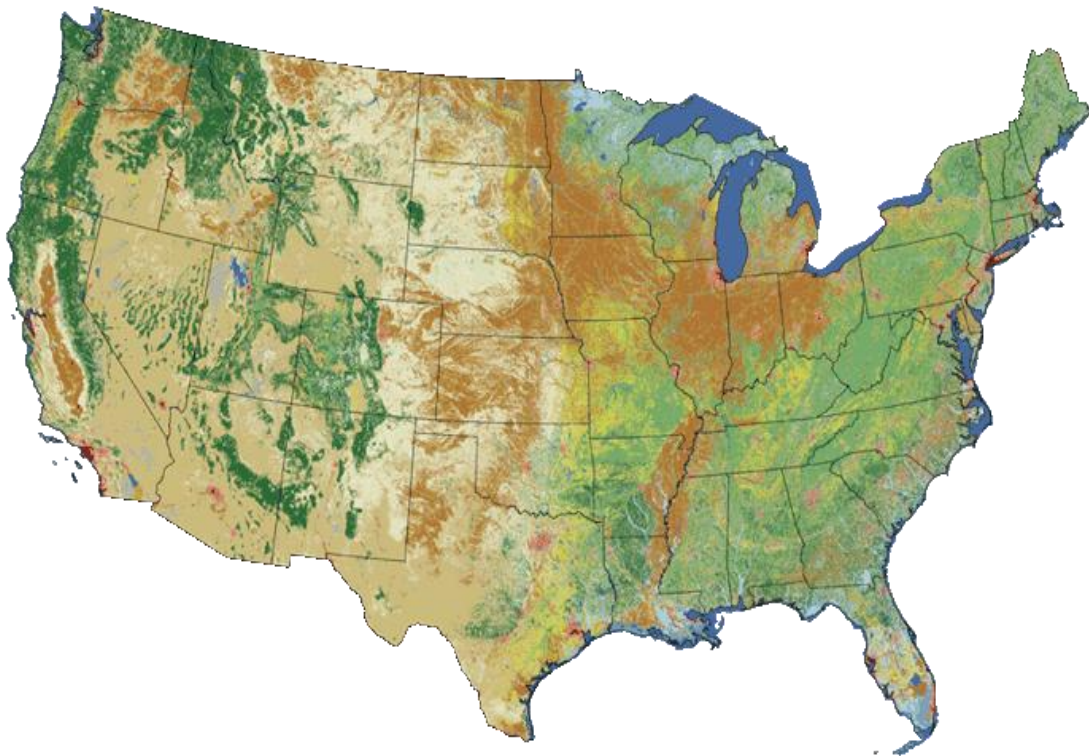
Στα ψηφιδωτά (raster) δεδομένα, ολόκληρη η περιοχή του χάρτη υποδιαιρείται σε ένα πλέγμα μικροσκοπικών κελιών. Μια τιμή αποθηκεύεται σε κάθε ένα από αυτά τα κελιά για να

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

αναπαριστά τη φύση στο οτιδήποτε υπάρχει στην αντίστοιχη θέση στο έδαφος. Τα ψηφιδωτά (raster) δεδομένα μπορούν να θεωρηθούν ως ένας πίνακας τιμών. Μία μεμονωμένη κουκκίδα είναι απλώς ένα χρώμα, αλλά ένα σύνολο έγχρωμων κουκκίδων συνθέτουν μια ζωντανή και λεπτομερή αναπαράσταση παρέχοντας πλούσιες λεπτομέρειες στον χρήστη.

Η κύρια χρήση των ψηφιδωτών (raster) δεδομένων είναι η αποθήκευση πληροφοριών ενός χάρτη ως ψηφιακές εικόνες, στις οποίες οι τιμές των κελιών σχετίζονται με τα χρώματα των εικονοστοιχείων της εικόνας. Για να αναπαράγει την εικόνα, ο υπολογιστής διαβάζει τις τιμές των κελιών μία προς μία και τις εφαρμόζει στα εικονοστοιχεία της οθόνης.

Υπάρχουν διάφοροι τύποι αρχείων raster όπως για παράδειγμα JPG, GIF και PNG και κάθε τύπος αρχείου έχει τις δικές του αποχρώσεις. Η ποιότητα συχνά υπαγορεύεται από τον αριθμό των εικονοστοιχείων που περιέχονται σε μια ίντσα, εκφρασμένα σε εικονοστοιχεία ανά ίντσα ή rpi, καθώς και από τις συνολικές διαστάσεις της εικόνας, εκφραζόμενες επίσης ως εικονοστοιχεία (για παράδειγμα, πλάτος 5.000 pixel με ύψος 2.500 pixel). Όσο μεγαλύτερες είναι οι τιμές των rpi και οι διαστάσεις μίας εικόνας, τόσο υψηλότερη είναι η ποιότητα. Αν θυσιαστούν τα ανωτέρω, δεν θα μπορεί να μεγεθυνθεί αρκετά, καθώς κάθε εικονοστοιχείο θα γίνεται ολοένα μεγαλύτερο με αποτέλεσμα η εικόνα να φαίνεται θολή.



Εικόνα 3.4 – Ψηφιδωτά δεδομένα.

3.2.3 Σύγκριση Ψηφιδωτών (raster) και διανυσματικών (vector) δεδομένων

Και οι δύο τύποι δεδομένων είναι πολύ χρήσιμοι, αλλά υπάρχουν σημαντικές διαφορές. Τα παρακάτω χαρακτηριστικά είναι ευρείες γενικεύσεις που δεν ισχύουν απαραίτητα σε όλες τις περιπτώσεις.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

Vector	Raster
Σχετικά χαμηλός όγκος δεδομένων	Σχετικά υψηλός όγκος δεδομένων
Ταχύτερη εμφάνιση	Αργή εμφάνιση
Μπορεί επίσης να αποθηκεύει ιδιότητες	Δεν έχει πληροφορίες χαρακτηριστικών
Απεικόνιση λιγότερο ευχάριστη στο μάτι	Πιο ευχάριστη απεικόνιση στο μάτι
Δεν υπαγορεύει τον τρόπο εμφάνισης των χαρακτηριστικών σε ένα GIS	Εγγενώς αποθηκεύει τον τρόπο εμφάνισης των χαρακτηριστικών σε ένα GIS

Πίνακας 2. Συγκριτικός πίνακας χαρακτηριστικών Raster και Vector δεδομένων

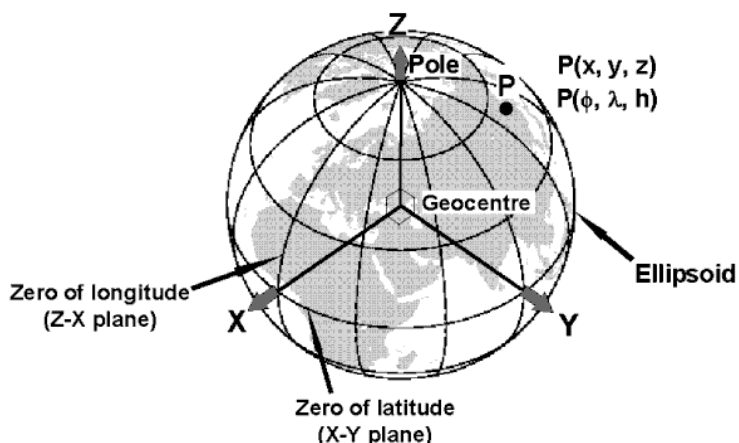
Ο πίνακας 2 παρουσιάζει μια απλή σύγκριση μεταξύ των δύο τύπων δεδομένων. Καθώς εκτελείται μεγέθυνση στα raster δεδομένα, η δομή των εικονοστοιχείων γίνεται εμφανής. Τελικά, η συνολική εικόνα ομοιάζει περισσότερο με απεικόνιση σύγχρονης τέχνης παρά με τη λεπτομερή αναπαράσταση των (χωρικών) δεδομένων που βρίσκουμε σε έναν κλασικό χάρτη. Τα διανυσματικά δεδομένα είναι περισσότερο σαν ένα γράφημα με μια γραμμή που σύρεται μεταξύ των σημείων, το πλάτος παραμένει το ίδιο, όσο μεγάλη μεγέθυνση και να πραγματοποιηθεί.

3.3 Χωρικά συστήματα αναφοράς

Το χωρικό σύστημα αναφοράς ή spatial reference system (SRS) μπορεί να είναι ένα τοπικό ή περιφερειακό ή παγκόσμιο σύστημα, το οποίο βασίζεται στις συντεταγμένες και χρησιμοποιείται για τον εντοπισμό χωρικών οντοτήτων. Αυτά τα συστήματα συντεταγμένων αποτελούν την καρδιά όλων των εφαρμογών GIS. Τα χωρικά συστήματα αναφοράς προσδιορίζονται από έναν ακέραιο SRID αριθμό, συμπεριλαμβανομένων και των EPSG κωδικών που ορίζονται από τη Διεθνή Ένωση παραγωγών πετρελαίου και αερίου. Υπάρχουν χιλιάδες συστήματα διαθέσιμα εκ των οποίων αξιολογημένα είναι αυτά με τους κωδικούς EPSG: 4326, 2100 και 3857.

3.3.1 Το παγκόσμιο σύστημα αναφοράς

Το παγκόσμιο σύστημα αναφοράς (WGS84) με κωδικό EPSG: 4326 καλύπτει τα όρια (-180.0000, -90.0000, 180.0000, 90.0000) και χρησιμοποιείται από το δορυφορικό σύστημα πλοήγησης GPS και για τη γεωδαιτική χωρομέτρηση του NATO.



Εικόνα 3.5 – Ελλειψοειδές προβολικό σύστημα.

3.3.2 Το ελληνικό σύστημα αναφοράς

Το ελληνικό σύστημα αναφοράς (GGRS87) με κωδικό EPSG: 2100 καλύπτει τα όρια (18.2700, 33.2300, 29.9700, 41.7700) και χρησιμοποιείται για τοπογραφική χαρτογράφηση μεγάλης και μεσαίας κλίμακας της περιοχής αναφοράς, καθώς και από τους τοπογράφους μηχανικούς.

3.3.3 EPSG:3857 - WGS84 Web Mercator (Auxiliary Sphere)

Προβολή που χρησιμοποιείται σε πολλές δημοφιλείς εφαρμογές χαρτογράφησης ιστού (Google / Bing / OpenStreetMap / etc). Μερικές φορές είναι γνωστή ως EPSG: 900913.

3.4 Ανάλυση απαιτήσεων

Η ανάπτυξη της εφαρμογής 'MyNauticalMap' θα εκτελεστεί μέσω της δημοφιλούς προγραμματιστικής πλατφόρμας Microsoft Visual Studio σε γλώσσα προγραμματισμού C# με ενσωμάτωση των βιβλιοθηκών του DotSpatial και τη χρήση του SQL Server για την αποθήκευση των δεδομένων που θα εισάγει ο χρήστης. Το DotSpatial είναι μια ανοιχτού λογισμικού βιβλιοθήκη για γεωγραφικά συστήματα πληροφοριών, η οποία λειτουργεί με το .Net Framework 4 του Visual Studio της Microsoft. Ενσωματώνοντας τη βιβλιοθήκη του DotSpatial σε μια Windows Form εφαρμογή, μπορούμε να εκτελέσουμε ενδεικτικά τα εξής:

- Απεικόνιση raster και vector αρχείων.
- Απεικόνιση βάσης δεδομένων sqlite και εκτέλεση εντολών sqlite.
- Επεξεργασία των ανωτέρω αρχείων.
- Δημιουργία νέου επιπέδου - διαγραφή επιπέδου (layer).
- Απεικόνιση ενός πίνακα ιδιοτήτων (attribute table), αναζήτηση στοιχείων με χρήση μίας λέξης κλειδί και απεικόνισή του στον χάρτη.
- Μεγέθυνση – σμίκρυνση του χάρτη.
- Αναγνώριση αντικειμένων πάνω στο χάρτη και εμφάνιση των στοιχείων τους.
- Μετακίνηση πάνω στον χάρτη.
- Μεγέθυνση σε συγκεκριμένες συντεταγμένες.
- Μεγέθυνση σε πλήρη περιεχόμενο χάρτη.
- Τοποθέτηση παραπάνω τους ενός επιπέδου (layer).

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

- Εισαγωγή - διαγραφή ναυτιλιακής πληροφορίας.
- Μέτρηση απόστασης.
- Εξαγωγή πίνακα χαρακτηριστικών ενός point shapefile σε ένα Excel.
- Αποθήκευση εικόνας για συγκεκριμένο ID σε μορφή Byte.
- Είσοδος στην εφαρμογή με χρήση συνθηματικού.
- Εισαγωγή του εξαχθέντος Excel σε μία βάση SQL.
- Εκτέλεση λειτουργιών αναζήτησης στην ανωτέρω βάση δεδομένων με βάση μία περίοδο δύο ημερομηνιών και μίας λέξης κλειδιού.
- Εκτύπωση απεικόνισης χάρτη.
- Αποσύνδεση χρήστη.
- Έξοδος από την εφαρμογή.

Η εφαρμογή 'MyNauticalMap' θα περιλαμβάνει κατά το ελάχιστο:

1. Ναυτιλιακές πληροφορίες λιμένων και λοιπών περιοχών.
2. Τυχόν Φωτογραφικό υλικό.
3. Μετρήσεις θερμοκρασιών θαλάσσης.
4. Μετρήσεις πυκνότητας θαλάσσης.
5. Μετρήσεις θερμοκρασιών ατμόσφαιρας.
6. Ημερομηνία λήψης των ανωτέρω στοιχείων.
7. Αλατότητα.
8. Ταχύτητα ήχου.
9. Ναυτιλιακή Κίνηση.
10. Τηλεφωνική κάλυψη.
11. Θέση της λήψης των στοιχείων.
12. Θαλάσσια ρεύματα.

Οι ναυτιλιακές πληροφορίες λιμένων και λοιπών περιοχών π.χ. οι διάφορες μετρήσεις θερμοκρασιών και πυκνοτήτων θαλάσσης θα εισάγονται από τον χρήστη από τις μετρήσεις που έκανε ο ίδιος και θα ενσωματώνονται στον χάρτη της εφαρμογής, ενώ το τυχόν φωτογραφικό υλικό θα μπορεί να εισαχθεί σε μια βάση δεδομένων με χρήση SQL Server. Ο πίνακας Attribute Table που θα περιλαμβάνει όλα αυτά τα στοιχεία, θα έχει τη δυνατότητα να εξαχθεί σε μορφή Excel για τυχόν χρήση εκτός εφαρμογής, ενώ παράλληλα θα έχει τη δυνατότητα να εισαχθεί σε μία βάση δεδομένων στην οποία θα μπορούν να εκτελούνται λειτουργίες αναζήτησης για τη διευκόλυνση του χρήστη.

Ως προς τις λειτουργικές απαιτήσεις, θα υπάρχει συγκεκριμένος χρήστης ανά πλοίο που θα χειρίζεται την εφαρμογή με τον ρόλο του χρήστη (user). Για τη σύνδεση (login) των χρηστών θα τηρείται πίνακας με το όνομα «Login» στη βάση δεδομένων της εφαρμογής.

Για τον οποιοδήποτε χρήστη θα πρέπει να ισχύουν τα ακόλουθα:

- Να έχει τη δυνατότητα να συνδεθεί με ένα username και password.
- Να μπορεί να φορτώνει διανυσματικούς ή ψηφιδωτούς χάρτες (Raster -Vector).
- Να μπορεί να εισάγει, να τροποποιεί και να διαγράφει ναυτιλιακές πληροφορίες
- Να μπορεί να εισάγει και να τροποποιεί οποιαδήποτε δεδομένα της βάσης δεδομένων των εικόνων.
- Να μπορεί να εκτυπώσει την απεικόνιση του χάρτη
- Να εξαγάγει τα δεδομένα του attribute table σε ένα Excel
- Να μπορεί να εισάγει το εξαχθέν Excel σε μία βάση δεδομένων.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

- Να εκτελεί λειτουργίες αναζήτησης στην ανωτέρω βάση δεδομένων.
- Να μπορεί να αποσυνδεθεί.
- Να μπορεί να εξέλθει από την εφαρμογή.

Ως προς τις μη λειτουργικές απαιτήσεις, η εφαρμογή θα πρέπει να πληροί τα κάτωθι:

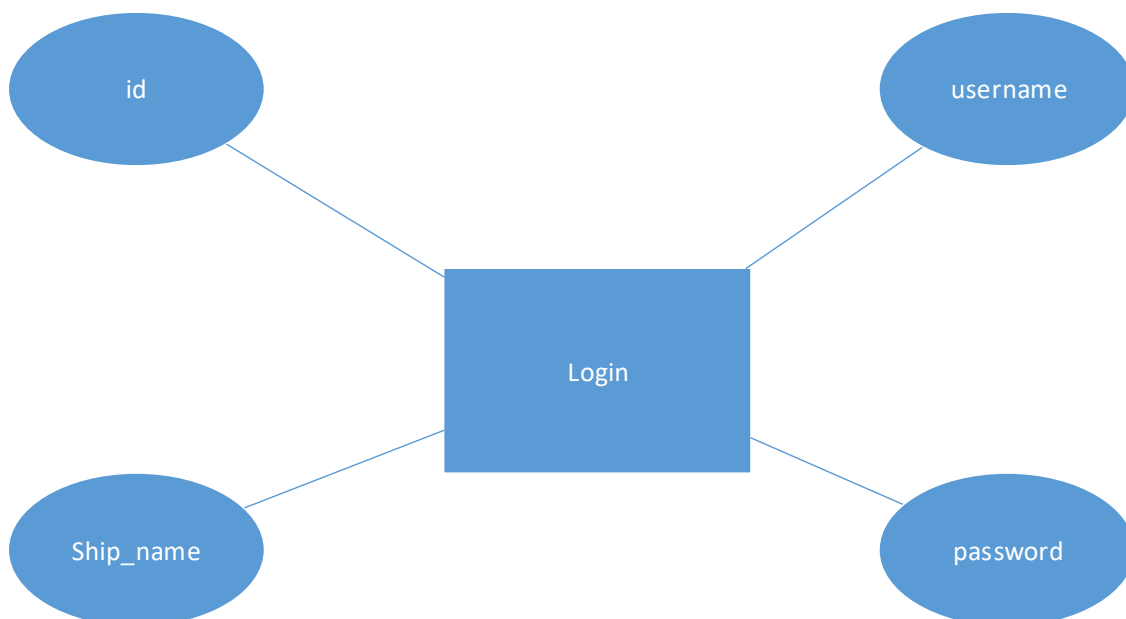
- Να λειτουργεί αδιάλειπτα.
- Να μπορεί να εκτελείται σε λειτουργικό Windows XP και άνω, ενώ υπολογιστές χωρίς μεγάλη υπολογιστική ισχύ, να μπορούν να εκτελούν την εφαρμογή με ικανοποιητικό χρόνο απόκρισης.
- Οι βάσεις δεδομένων πρέπει να βρίσκονται εκτός Διαδικτύου. Κάθε πλοίο που αποθηκεύει πληροφορίες στις βάσεις δεδομένων του, για κάποιο λιμένα που επισκέφτηκε ή κάποιο αγκυροβόλι που έκανε, είναι αναγκαίο να μπορεί να τα μεταφορτώνει σε μία κεντρική βάση δεδομένων, ώστε οι πληροφορίες αυτές να μπορούν στη συνέχεια να μεταφορτώνονται σε όλα τα πλοία με κάποιο φορητό μέσο αποθήκευσης.
- Να περιέχει ενδεικτικά δεδομένα – πληροφορίες, τα οποία έχουν αντληθεί από διαδικτυακές σελίδες με ελεύθερα χωρικά datasets.

3.5 Σχεδιασμός

Για τον σχεδιασμό της εφαρμογής 'MyNauticalMap' θα απαιτηθεί να περιγραφούν οι τρεις βάσεις δεδομένων των στοιχείων των χρηστών (user_id, password), των εικόνων και της εισαγωγής του πίνακα χαρακτηριστικών από το Excel, όπως επίσης και οι λειτουργίες ενός χρήστη στην εφαρμογή μέσω διαγράμματος περιπτώσεων χρήσης (use case diagram - UML).

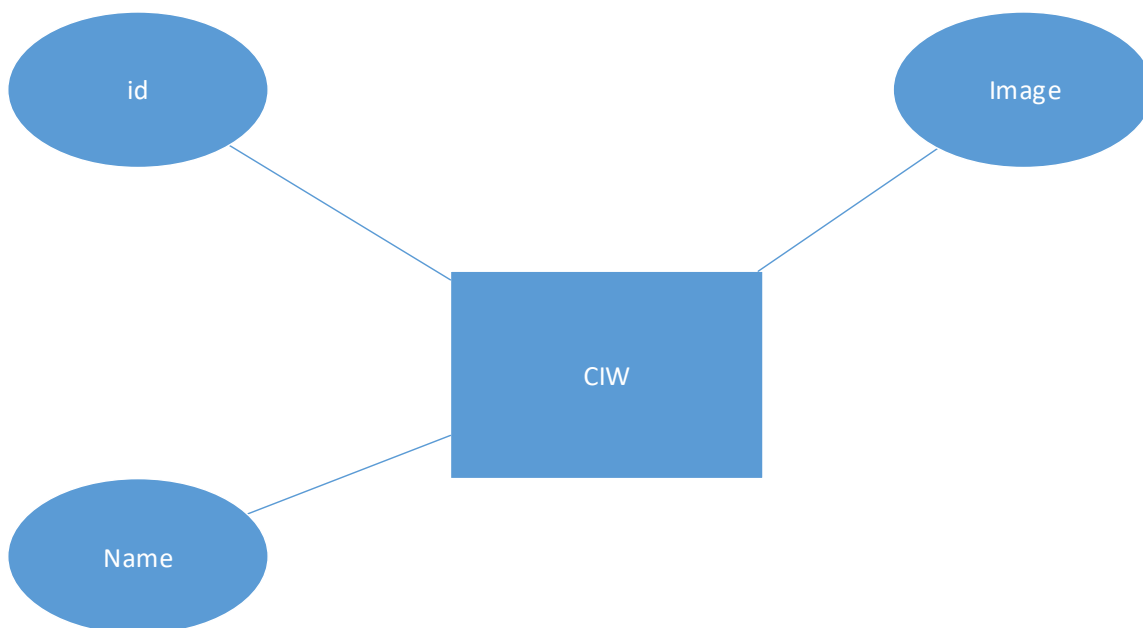
Κάθε υπεύθυνος ναυτιλίας ενός πλοίου θα διαθέτει ένα username και ένα password. Με αυτά τα στοιχεία θα μπορεί να συνδεθεί στο σύνολο των λειτουργιών της εφαρμογής.

Όσον αφορά τη βάση δεδομένων των στοιχείων των χρηστών της εφαρμογής, αυτή θα περιέχει τον πίνακα 'Login'. Ο πίνακας 'Login' θα περιέχει τα πεδία id, ship_name, username και password. Το διάγραμμα ER απεικονίζεται παρακάτω:



Εικόνα 3.6 – Λογικό διάγραμμα ER ΒΔ Στοιχεία χρήστη.

Η βάση δεδομένων για την εισαγωγή εικόνων περιέχει τον πίνακα CIW με τα πεδία id, Name και Image. Το διάγραμμα ER απεικονίζεται παρακάτω:

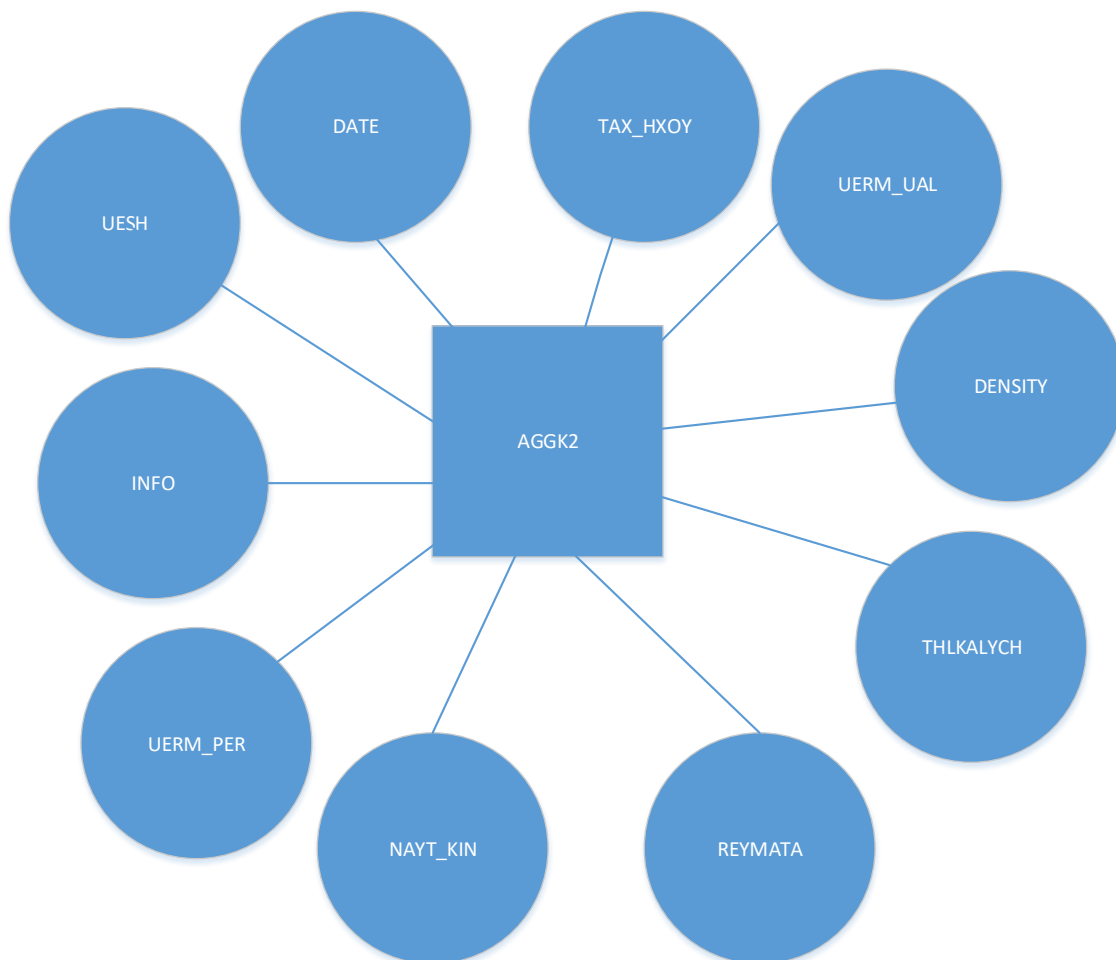


Εικόνα 3.7 – Λογικό διάγραμμα ER ΒΔ Εικόνων χρήστη.

Η βάση δεδομένων για την εισαγωγή του αρχείου Excel περιέχει τον πίνακα AGGK2 με τα πεδία INFO, UESH (ως Primary key, καθώς τα στοιχεία θα δίνονται με συντεταγμένες και όνομα τοποθεσίας), DENSITY, UERM_UAL, UERM_PER, TAX_HXOY, DATE, NAYT_KIN, REYMATATA, THLKALYCH.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

Το διάγραμμα ER απεικονίζεται παρακάτω:



Εικόνα 3.8 – Λογικό διάγραμμα ER ΒΔ αρχείου .dbf.

Οι λειτουργίες ενός χρήστη πάνω στην εφαρμογή 'MyNauticalMap' μπορούν να απεικονιστούν με τη βοήθεια του διαγράμματος περιπτώσεων χρήσης (use case diagram) της (πρότυπης) γλώσσας μοντελοποίησης UML. Ο οποιοσδήποτε χρήστης θεωρείται ως actor στο σύστημα – εφαρμογή, ενώ οι διάφορες λειτουργίες που μπορεί να εκτελέσει σε αυτή, ονομάζονται περιπτώσεις χρήσης (use case). Μεταξύ των πολλών περιπτώσεων χρήσης είναι δυνατό να υπάρχει μια σχέση. Το διάγραμμα περιπτώσεων χρήσης ενός χρήστη απεικονίζεται παρακάτω:



Εικόνα 3.9 – Διάγραμμα Περιπτώσεων Χρήσης ενός χρήστη (Use case diagram).

Όπως απεικονίζεται παραπάνω, ο χρήστης αφού συνδεθεί στην εφαρμογή με τη χρήση μοναδικού username και password που του αντιστοιχεί θα μπορεί:

- Να προσθέσει ένα ή περισσότερα επίπεδα (layer)
- Να προβάλει εικόνες από τη βάση δεδομένων
- Εκτέλεση λειτουργιών αναζήτησης σε βάση δεδομένων του SQL Server
- Να αποσυνδεθεί από την εφαρμογή

Στη συνέχεια, αφού προσθέσει ένα ή περισσότερα επίπεδα (layers) θα υπάρχει η δυνατότητα για:

- Προβολή ναυτιλιακών πληροφοριών από οποιοδήποτε επίπεδο
- Εκτύπωση απεικόνισης χάρτη
- Αφαίρεση επιπέδου (layer)
- Εισαγωγή ναυτιλιακής πληροφορίας σε υπάρχον επίπεδο (layer)
- Εισαγωγή ναυτιλιακής πληροφορίας σε νέο επίπεδο (layer)
- Εξαγωγή του δεδομένων του αρχείου .dbf ενός point shapefile σε αρχείο Excel

Μετά την εισαγωγή ναυτιλιακής πληροφορίας σε υπάρχον ή σε νέο επίπεδο θα υπάρχει η δυνατότητα για:

- Αφαίρεση ναυτιλιακής πληροφορίας
- Αποθήκευση εικόνων σε βάση δεδομένων
- Τροποποίηση πίνακα ιδιοτήτων (attribute table)
- Εξαγωγή επιπέδου σε αρχείο .shp

Ο χρήστης μπορεί ακόμη να αποθηκεύσει και να εξαγάγει επιλεγμένα δεδομένα ώστε να είναι διαθέσιμα στον ίδιο αλλά και σε άλλους χρήστες της εφαρμογής μέσα από τον Attribute table editor. Τέλος, μπορεί να εξαγάγει το σύνολο των δεδομένων του αρχείου .dbf ενός point shapefile

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

σε αρχείο Excel, στη συνέχεια να το μεταφορτώσει σε μία βάση δεδομένων και στη συνέχεια να εκτελέσει λειτουργίες αναζήτησης.

3.6 Εργαλεία Ανάπτυξης Εφαρμογής

Σε αυτό το κεφάλαιο θα περιγραφούν συνοπτικά τα σημαντικότερα εργαλεία προγραμματισμού που χρησιμοποιήθηκαν για την κατασκευή της εφαρμογής.

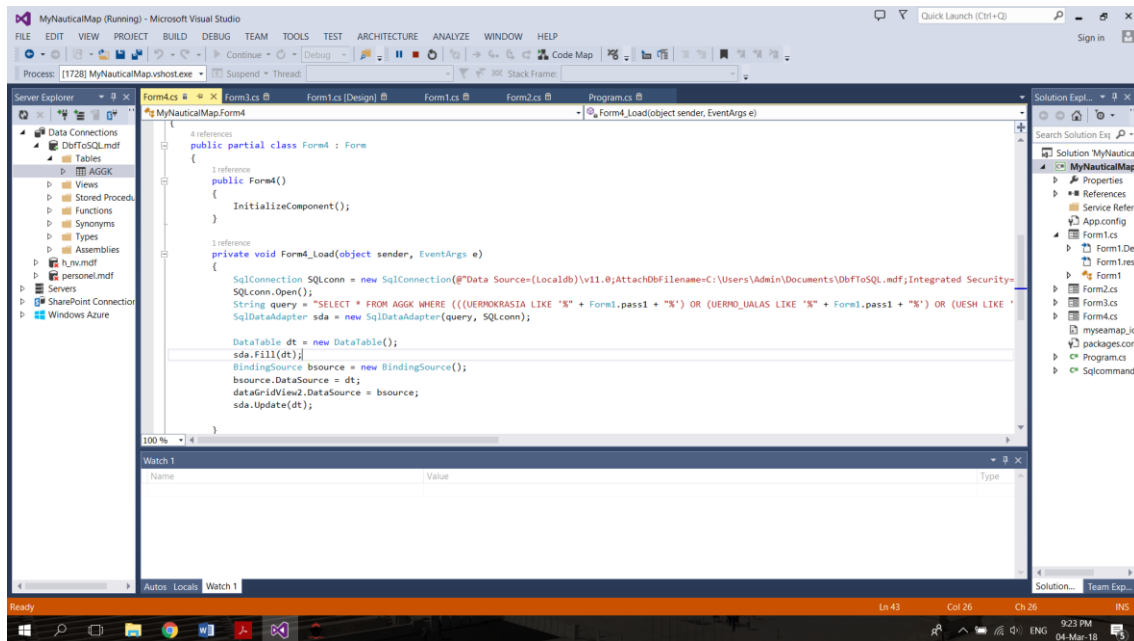
3.6.1 Microsoft Visual Studio

Το Microsoft Visual Studio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) από τη Microsoft. Χρησιμοποιείται για την ανάπτυξη προγραμμάτων ηλεκτρονικών υπολογιστών για τα Microsoft Windows, καθώς και για την κατασκευή ιστοσελίδων, διαδικτυακών εφαρμογών και υπηρεσιών web. Το Visual Studio χρησιμοποιεί πλατφόρμες ανάπτυξης λογισμικού της Microsoft, όπως Windows API, Windows Forms, Windows Presentation Foundation, Windows Store και το Microsoft Silverlight. Μπορεί να παράγει εγγενή και διαχειριζόμενο κώδικα.

Περιέχει ενσωματωμένα εργαλεία σχεδίασης μορφών για την κατασκευή εφαρμογών GUI, ιστοσελίδων, βάσης δεδομένων και μεταξύ άλλων αποδέχεται plugins που βελτιώνουν τη λειτουργικότητα σχεδόν σε όλα τα επίπεδα και την προσθήκη νέων εργαλείων.

Το Visual Studio υποστηρίζει 36 διαφορετικές γλώσσες προγραμματισμού και επιτρέπει στον επεξεργαστή κειμένου να υποστηρίζει εντοπισμό σφαλμάτων (σε διαφορετικούς βαθμούς) σχεδόν σε οποιαδήποτε γλώσσα προγραμματισμού. Οι ενσωματωμένες γλώσσες περιλαμβάνουν την C, C ++, C ++ / CLI, Visual Basic .NET, C #, F #, JavaScript, TypeScript, XML, XSLT, HTML και CSS. Παρέχει υποστήριξη και για άλλες γλώσσες όπως Python, Ruby, Node.js μέσω plug-ins.

Η πιο βασική έκδοση του Visual Studio διατίθεται δωρεάν.



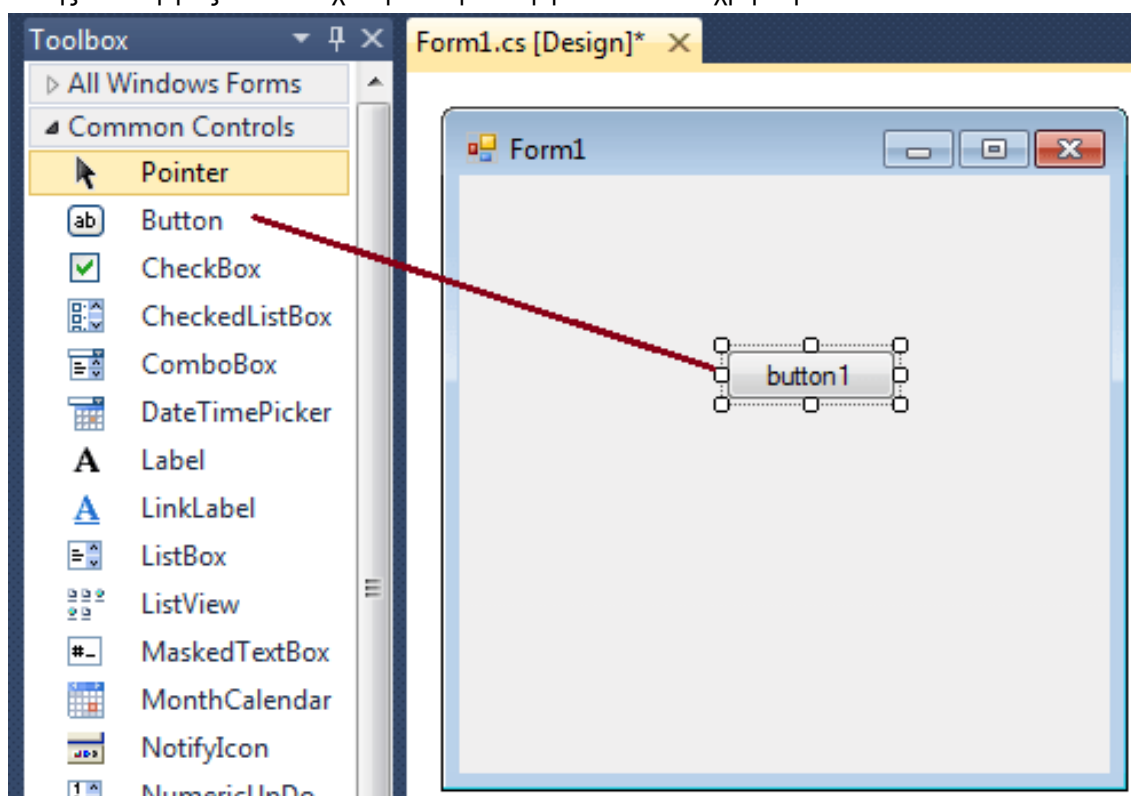
Εικόνα 3.10 – Microsoft Visual Studio.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

3.6.2 Windows Forms

Τα Windows Forms (WinForms) είναι μια βιβλιοθήκη γραφικών (GUI) αποτελεί μέρος του Microsoft .NET Framework, παρέχοντας μια πλατφόρμα δημιουργίας εφαρμογών για επιτραπέζιους, φορητούς υπολογιστές και tablet PCs. Αυτή η πλατφόρμα ξοδεύει το μεγαλύτερο μέρος του χρόνου της απλώς περιμένοντας τον χρήστη να κάνει κάτι, όπως το να συμπληρώσει ένα πλαίσιο κειμένου ή να κάνει κλικ σε ένα κουμπί.

Για τα στοιχεία ελέγχου των Windows όπως το button, το textbox, το checkbox και η listview υφίστανται οι δυνατότητες επιλογής της τοποθεσίας τους, του μεγέθους τους, του χρώματος που θα έχουν, της γραμματοσειράς που θα χρησιμοποιείται, του κειμένου που θα περιέχουν και άλλων ιδιοτήτων του από τον χρήστη. Ακόμα με την εκτέλεση κάποιου συμβάντος όπως του πατήματος ενός κλικ, μπορεί να εκτελούνται κώδικες που θα αποσκοπούν την εκτέλεση μίας άλλης λειτουργίας που θα έχει οριστεί με σαφήνεια από τον χρήστη.



Εικόνα 3.11 – Windows Forms.

3.6.3 .NET Framework

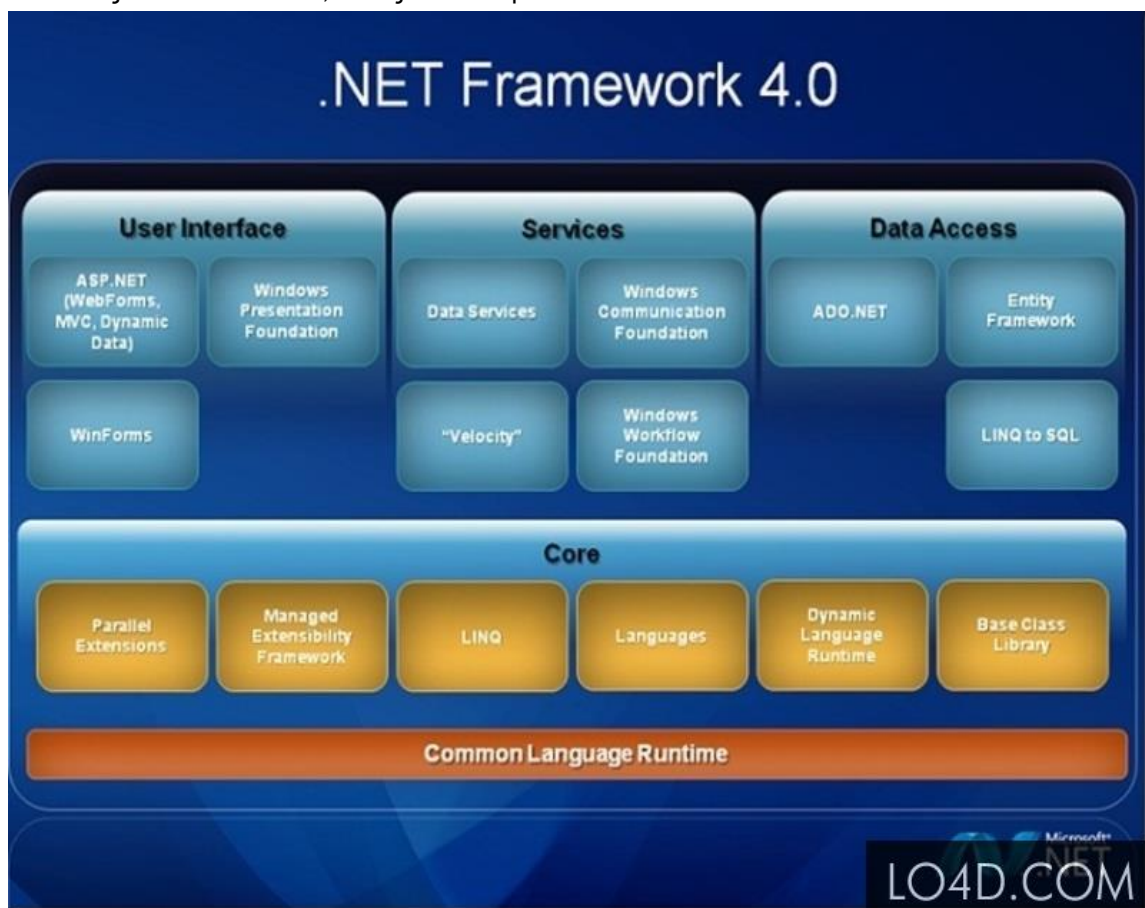
Το .Net Framework είναι μία πλατφόρμα ανάπτυξης λογισμικού για τα Microsoft Windows, το Διαδίκτυο, τα κινητά και τις υπηρεσίες cloud όπως το Microsoft Azure. Δημιουργήθηκε από την εταιρεία Microsoft και κυκλοφόρησε στην αγορά πρώτη φορά το 2002. Περιέχει μια μεγάλη βιβλιοθήκη με το όνομα Framework Class Library (FCL) και κάθε γλώσσα προγραμματισμού μπορεί να χρησιμοποιεί κώδικα ο οποίος είναι γραμμένος σε άλλη γλώσσα προγραμματισμού. Προγράμματα τα οποία είναι γραμμένα για το .NET Framework εκτελούνται σε ένα προγραμματιστικό περιβάλλον το οποίο καλείται Common Language Runtime (CLR), μια εικονική μηχανή η οποία παρέχει δυνατότητες όπως διαχείριση μνήμης, ασφάλεια, διαχείριση

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

λαθών, διαχείριση νημάτων για κάθε γλώσσα προγραμματισμού που συνεργάζεται με το .NET Framework. Ο εκάστοτε μεταφραστής/μεταγλωττιστής (compiler) της γλώσσας προγραμματισμού που υποστηρίζεται από το .Net Framework μεταφράζει τον πηγαίο κώδικα και εξάγει ένα bytecode αρχείο ή αλλιώς το CIL κώδικα (Common Intermediate Language). Στη συνέχεια, ο CLR μεταφράζει το CIL κώδικα και εξάγει το εκτελέσιμο αρχείο. Το FCL και το CLR μαζί αποτελούν το .NET Framework.

Το Framework Class Library είναι μια μεγάλη βιβλιοθήκη που περιέχει κώδικα για προγραμματιστικά θέματα όπως νήματα, εγγραφή/ανάγνωση αρχείων, υποστήριξη βάσεων δεδομένων, μετατροπή σε XML, δομές δεδομένων όπως στοιβές και ουρές κλπ. Η βιβλιοθήκη αυτή είναι διαθέσιμη σε κάθε γλώσσα που συνεργάζεται με το .NET Framework. Οι προγραμματιστές παράγουν προγράμματα συνδυάζοντας τον πηγαίο κώδικά τους με το .NET Framework και με άλλες βιβλιοθήκες. Χρησιμοποιείται στις περισσότερες από τις καινούργιες εφαρμογές οι οποίες δημιουργήθηκαν για τα Windows. Επιπρόσθετα, η Microsoft παράγει ένα εξειδικευμένο περιβάλλον δημιουργίας εφαρμογών το οποίο καλείται όπως αναφέρθηκε και στην ενότητα "3.6.1 Microsoft Visual Studio".

Το .NET Framework οδήγησε στη δημιουργία μίας οικογένειας από .NET πλατφόρμες, οι οποίες στοχεύουν στην κινητή τηλεφωνία, ενσωματωμένα συστήματα, σε εναλλακτικά λειτουργικά συστήματα, φυλλομετρητές και plug-ins. Μία περιορισμένη έκδοση του, το .NET Compact Framework, είναι διαθέσιμο στα Windows CE, συμπεριλαμβάνοντας τις φορητές συσκευές Windows Mobile, όπως τα smartphones.



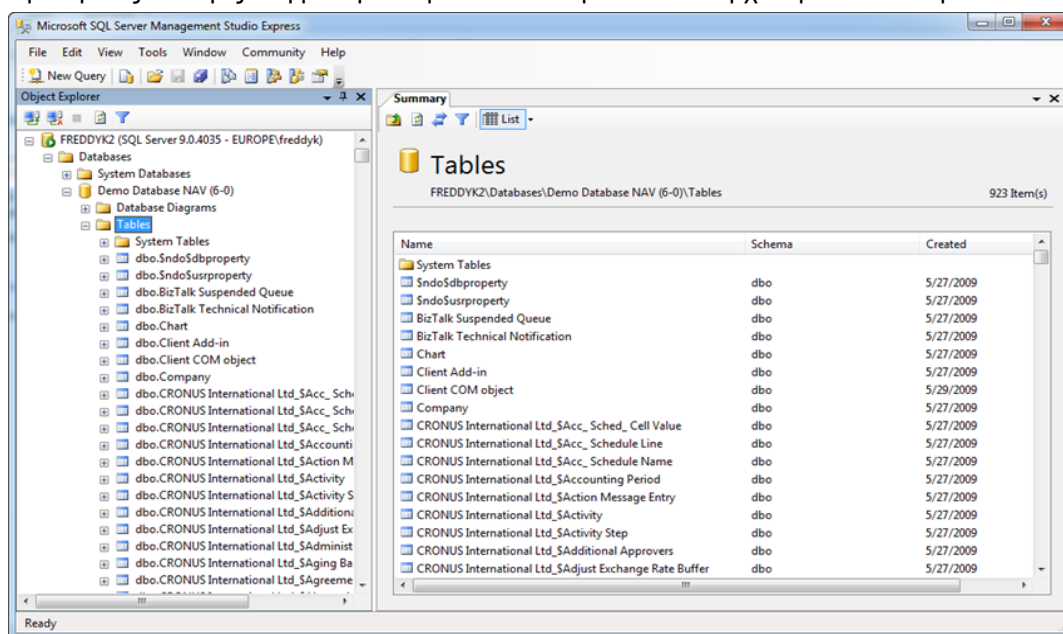
Εικόνα 3.12 – .NET Framework 4.0.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

3.6.4 Microsoft SQL Server

Ο SQL Server είναι ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων (relational database management system), που αναπτύσσεται από τη Microsoft. Οι κύριες γλώσσες που χρησιμοποιούνται είναι η T-SQL και η ANSI SQL. Ο SQL Server βγήκε για πρώτη φορά στην αγορά το 1989 σε συνεργασία με την Sybase. Η κύρια μονάδα αποθήκευσης στοιχείων είναι μια βάση δεδομένων, η οποία αποτελείται από μια συλλογή πινάκων και κώδικα.

Η κεντρική βάση δεδομένων του SQL υποστηρίζει διαφορετικούς τύπους, συμπεριλαμβανομένων των ακεραίων αριθμών, αριθμών κινητής υποδιαστολής, δεκαδικών, αλφαριθμητικών, Varchar (σειρές χαρακτήρων μεταβλητού μήκους), δυαδικών αριθμών (για τα μη δομημένα δεδομένα), κειμένων (για κείμενα). Τα στοιχεία στη βάση δεδομένων αποθηκεύονται σε ένα (ή περισσότερα) αρχεία με επέκταση .mdf. Τα δευτεροβάθμια στοιχεία αποθηκεύονται στο αρχείο με επέκταση .ndf. Το αρχείο καταγραφής το οποίο περιέχει όλες τις πρόσφατες αλλαγές στη βάση δεδομένων αποθηκεύεται σε αρχείο με επέκταση .ldf.



Εικόνα 3.13 – Microsoft SQL Server.

3.6.5 Microsoft Access Database Engine 2010 Redistributable

Το Microsoft Access Database Engine 2010 Redistributable είναι ένα σύνολο στοιχείων που διευκολύνουν τη μεταφορά δεδομένων μεταξύ υφιστάμενων αρχείων του Microsoft Office, όπως αρχεία Microsoft Office Access 2010 (* .mdb και * .accdb) και του Microsoft Office Excel 2010 (* .xls, * .xlsx και * .xlsb) προς άλλες πηγές δεδομένων, όπως ο Microsoft SQL Server. Επίσης, υποστηρίζεται συνδεσιμότητα σε υπάρχοντα αρχεία κειμένου. Τα προγράμματα οδήγησης ODBC και OLEDB χρησιμοποιούνται από προγραμματιστές εφαρμογών, για την ανάπτυξη λογισμικού που στοχεύει στη συνδεσιμότητα των αρχείων του Office.

3.6.6 DotSpatial

Το DotSpatial είναι μια βιβλιοθήκη συστήματος γεωγραφικών πληροφοριών γραμμένη για το .NET 4. Επιτρέπει στους προγραμματιστές την ενσωμάτωση των λειτουργιών των χωρικών

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

δεδομένων, την ανάλυση και τη χαρτογράφηση στις εφαρμογές τους ή τη συνεισφορά στη δημιουργία επεκτάσεων GIS στην ανοιχτή κοινότητα. Ο Δρ. Dan Ames είναι ο διαχειριστής του έργου της ανάπτυξης του DotSpatial. Το DotSpatial παρέχει έλεγχο χάρτη για το .NET και πολλές δυνατότητες GIS, όπως:

- Εμφάνιση χάρτη σε .NET Windows Forms ή σε εφαρμογή Web.
- Συμβατότητα με shapefiles, grids, raster αρχεία και εικόνες.
- Ετικέτες.
- Χειρισμός και εμφάνιση δεδομένων χαρακτηριστικών.
- Επιστημονική ανάλυση.
- Χειρισμό δεδομένων GPS.

Χρησιμοποιείται από πολλές γνωστές εφαρμογές, οι κυριότερες των οποίων είναι οι κάτωθι:

- MapWindow 6.
- HydroDesktop.
- MAD.
- LineSiter.
- Marine Life.
- PVMapper Site Designer.

3.6.7 Microsoft Excel

Το Microsoft Excel είναι πρόγραμμα λογιστικών φύλλων που αναπτύχθηκε από τη Microsoft για τα Microsoft Windows, macOS, Android και iOS. Διαθέτει υπολογισμούς, εργαλεία γραφημάτων, συγκεντρωτικούς πίνακες και μια γλώσσα προγραμματισμού macro με όνομα Visual Basic for Applications. Είναι μια πολύ ευρέως διαδεδομένη εφαρμογή υπολογιστικών φύλλων, ειδικά από την έκδοση 5 το 1993, και έχει αντικαταστήσει το Lotus 1-2-3 ως το βιομηχανικό πρότυπο για υπολογιστικά φύλλα. Το Microsoft Excel αποτελεί μέρος του Microsoft Office.

Το Microsoft Excel έχει τα βασικά χαρακτηριστικά όλων των λογιστικών φύλλων, χρησιμοποιώντας ένα πλέγμα κελιών διατεταγμένων σε αριθμημένες σειρές και στήλες με σκοπό την οργάνωση χειρισμών δεδομένων, όπως των αριθμητικών πράξεων. Έχει πολλές παρεχόμενες λειτουργίες, οι οποίες εξυπηρετούν στατιστικές, μηχανικές και οικονομικές ανάγκες. Επιπλέον, μπορεί να εμφανίζει δεδομένα ως γραμμικά γραφικές παραστάσεις, ιστογράμματα και γραφήματα ακόμα και με πολύ περιορισμένη τρισδιάστατη γραφική απεικόνιση. Επιτρέπει τη διατομή των δεδομένων για την προβολή των εξαρτήσεων τους από διάφορους παράγοντες για διαφορετικές οπτικές γωνίες (χρησιμοποιώντας συγκεντρωτικούς πίνακες περιστροφής και τον διαχειριστή σεναρίων). Περιέχει, ακόμα, μια πτυχή προγραμματισμού, το Visual Basic for Applications, που επιτρέπει στο χρήστη να χρησιμοποιεί μια μεγάλη ποικιλία αριθμητικών μεθόδων, όπως για παράδειγμα, την επίλυση των διαφορικών εξισώσεων της μαθηματικής φυσικής, και στη συνέχεια να αναφέρει τα αποτελέσματα πίσω στο υπολογιστικό φύλλο.

Έχει επίσης μια ποικιλία διαδραστικών λειτουργιών που επιτρέπουν στις διεπαφές χρήστη να αποκρύψουν πλήρως το υπολογιστικό φύλλο, έτσι ώστε το λογιστικό φύλλο να παρουσιάζεται ως μια "εφαρμογή" ή "σύστημα υποστήριξης αποφάσεων" (DSS), μέσω μιας προσαρμοσμένης διεπαφής χρήστη. Για παράδειγμα, μπορεί να χρησιμοποιηθεί ως ένας αναλυτής αποθέματος ή ως ένα εργαλείο σχεδιασμού, που ζητά από το χρήστη ερωτήσεις και παρέχει απαντήσεις και αναφορές. Σε μια πιο περίπλοκη υλοποίηση, μια εφαρμογή του Excel μπορεί αυτόματα να κάνει δημοσκόπηση σε εξωτερικές βάσεις δεδομένων και όργανα μέτρησης χρησιμοποιώντας ένα πρόγραμμα ενημέρωσης, να αναλύσει τα αποτελέσματα, να κάνει μια έκθεση Word ή μια παρουσίαση PowerPoint και να τα στείλει ηλεκτρονικά σε μια λίστα συμμετεχόντων. Το Excel δεν σχεδιάστηκε για να χρησιμοποιηθεί ως βάση δεδομένων. Τέλος, η

Microsoft επιτρέπει έναν αριθμό προαιρετικών διακοπών της γραμμής εντολών για τον έλεγχο του τρόπου εκκίνησης του Excel.

Ο αυτοματισμός στο Excel, επιτρέπει να πραγματοποιηθούν ενέργειες όπως η δημιουργία ενός νέου βιβλίου εργασίας, η προσθήκη δεδομένων σε ένα βιβλίο εργασίας ή η δημιουργία γραφημάτων. Με το Excel και άλλες εφαρμογές του Microsoft Office, σχεδόν όλες οι ενέργειες που μπορούν να πραγματοποιηθούν με μη αυτόματο τρόπο μέσω του περιβάλλοντος εργασίας χρήστη, μπορούν επίσης να πραγματοποιηθούν μέσω προγραμματισμού, χρησιμοποιώντας τον "Αυτοματισμό" (Automation).

Το Excel εκθέτει αυτή τη λειτουργία μέσω προγραμματισμού, χρησιμοποιώντας ένα μοντέλο αντικειμένου. Το μοντέλο αντικειμένου είναι μια συλλογή κλάσεων και μεθόδων, οι οποίες λειτουργούν ως ισοδύναμα των λογικών στοιχείων του Excel. Για παράδειγμα, υπάρχει ένα αντικείμενο Εφαρμογή (Application), ένα αντικείμενο Βιβλίο εργασίας (Workbook) και ένα αντικείμενο Φύλλο εργασίας (Worksheet), καθένα από τα οποία περιέχει τη λειτουργικότητα αυτών των τμημάτων του Excel. Για να αποκτηθεί πρόσβαση στο μοντέλο αντικειμένου από το Visual C# .NET, μπορεί να οριστεί μία αναφορά έργου για τη βιβλιοθήκη τύπων.

Δόμηση Εφαρμογής

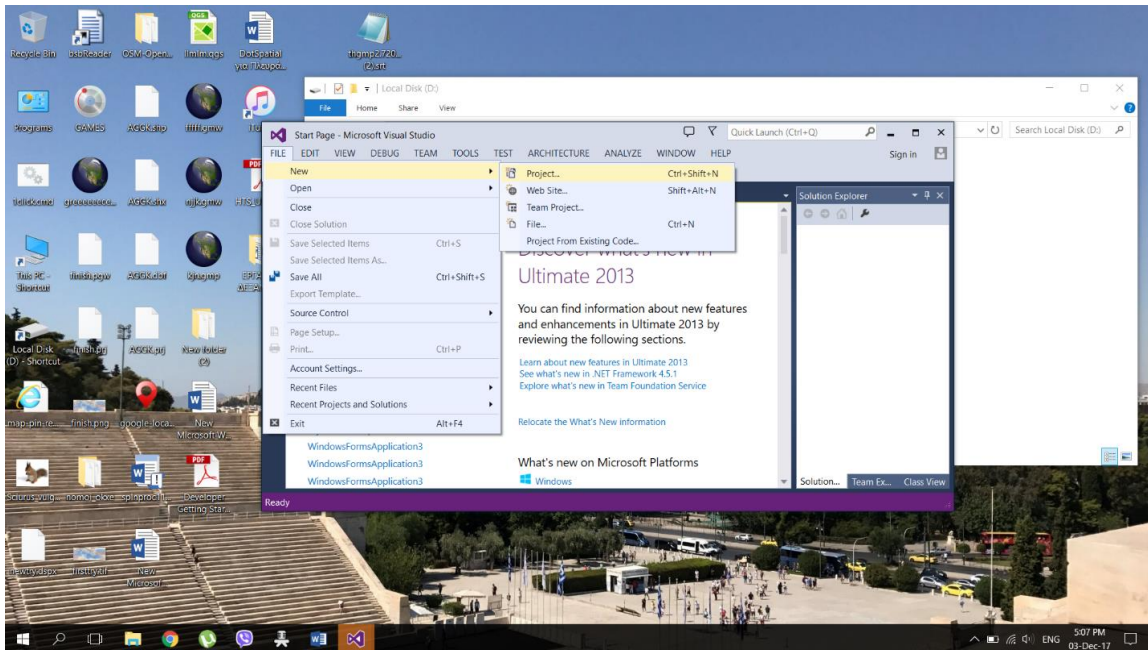
Η ανάπτυξη ενός προγράμματος με γεωγραφικές δυνατότητες έχει γίνει πλέον μία εύκολη διαδικασία. Ακόμα και ένας αρχάριος προγραμματιστής μπορεί με έτοιμους κώδικες- εργαλεία, να δημιουργήσει διαφορετικά περιβάλλοντα διεπαφής χρήστη ανάλογα με τις απαιτήσεις που υφίστανται.

Το σημαντικότερο βήμα είναι η εισαγωγή των εργαλείων του DotSpatial στα εργαλεία του Visual Studio. Με αυτόν τον τρόπο θα επιτευχθεί η αξιοποίησή τους στην εφαρμογή χάρτη που θα δημιουργηθεί.

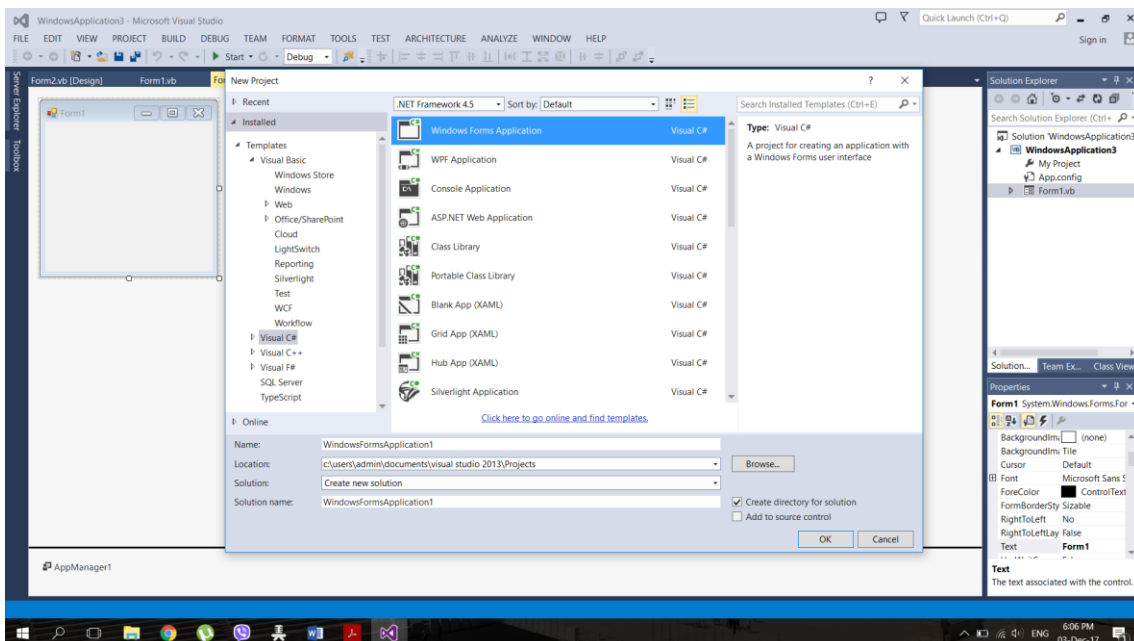
Σε αυτό το κεφάλαιο, αρχικά θα παρουσιαστούν τα βασικότερα βήματα για την κατασκευή της εφαρμογής, ενώ στη συνέχεια θα παρουσιαστεί το σύνολο του κώδικα που χρησιμοποιήθηκε ανά φόρμα.

4.1 Κατασκευή εφαρμογής

Πρώτα δημιουργούμε ένα νέο έργο στο Visual Studio σε γλώσσα C# και Windows πλατφόρμα.



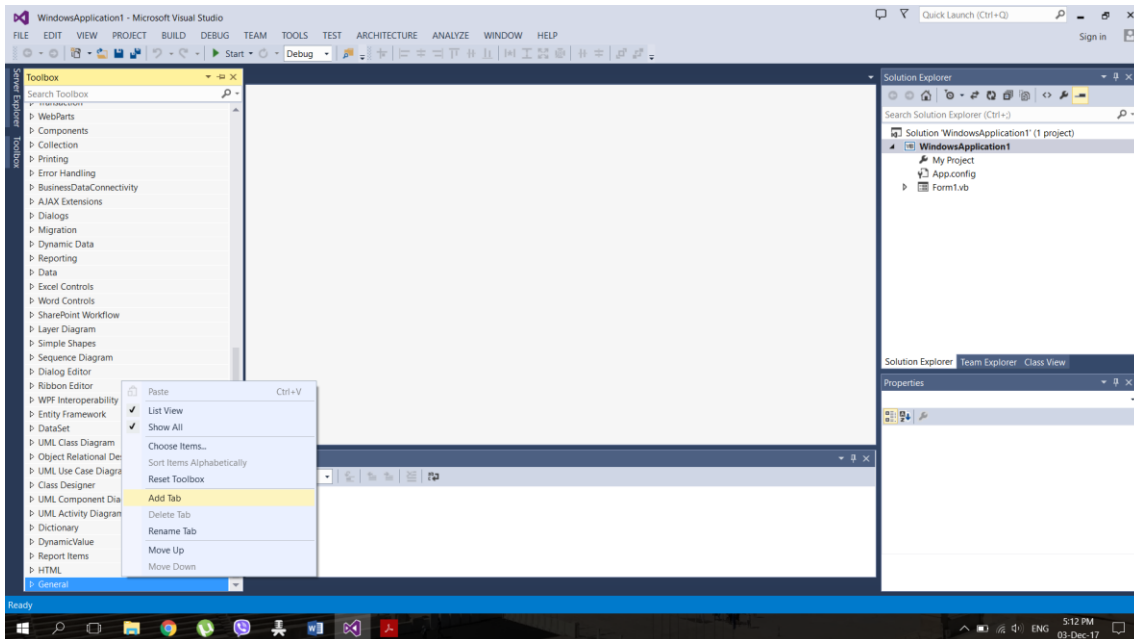
Εικόνα 4.1 – Δημιουργία ενός νέου έργου στο Visual Studio σε γλώσσα C# και Windows πλατφόρμα.



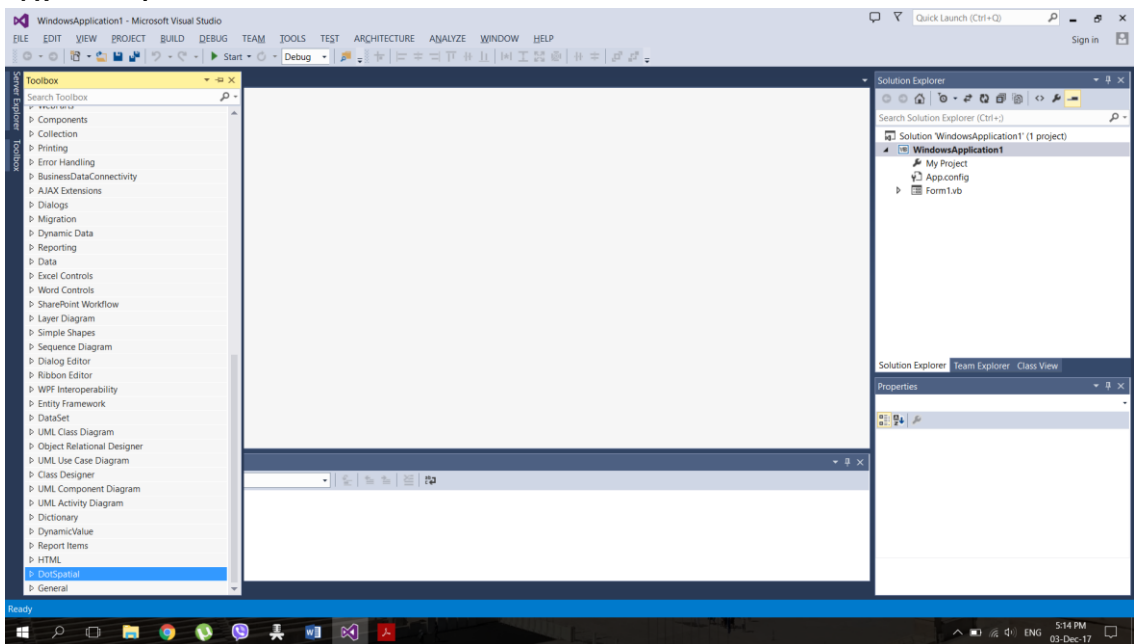
Εικόνα 4.2 – Δημιουργία ενός νέου έργου στο Visual Studio σε γλώσσα C# και Windows πλατφόρμα.

Εγκαθιστούμε το αρχείο Microsoft Access Database Engine 2010 Redistributable από το site της Microsoft. Στη συνέχεια προσθέτουμε μία καινούργια καρτέλα με το όνομα DotSpatial με σκοπό την αποθήκευση των εργαλείων του DotSpatial. Αυτό μπορεί να γίνει πατώντας δεξί κλικ και Add Tab στην καρτέλα General στο Toolbox του Visual Studio.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



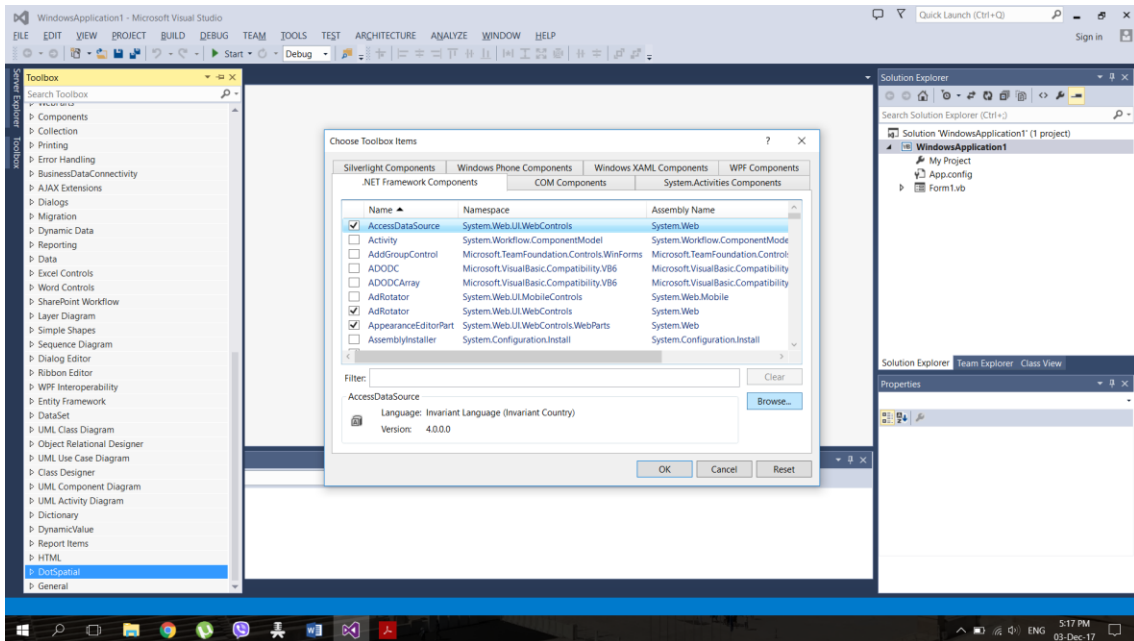
Εικόνα 4.3 – Προσθέτουμε μία καινούργια καρτέλα με το όνομα DotSpatial για να εισάγουμε το αρχείο DotSpatial.Controls.dll.



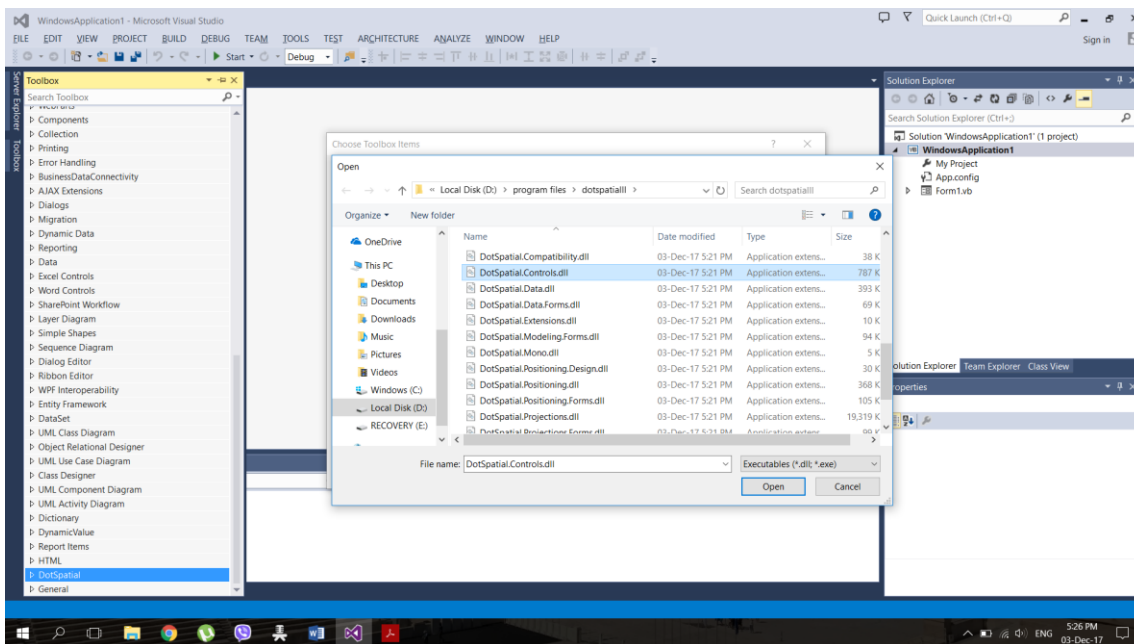
Εικόνα 4.4 – Η καρτέλα των εργαλείων DotSpatial.

Στη συγκεκριμένη καρτέλα πατάμε Choose Items από το μενού των περιεχομένων και πατάμε Browse για να εισάγουμε το αρχείο DotSpatial.Controls.dll που βρίσκεται στα αρχεία που κατεβάσαμε από το Διαδίκτυο.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

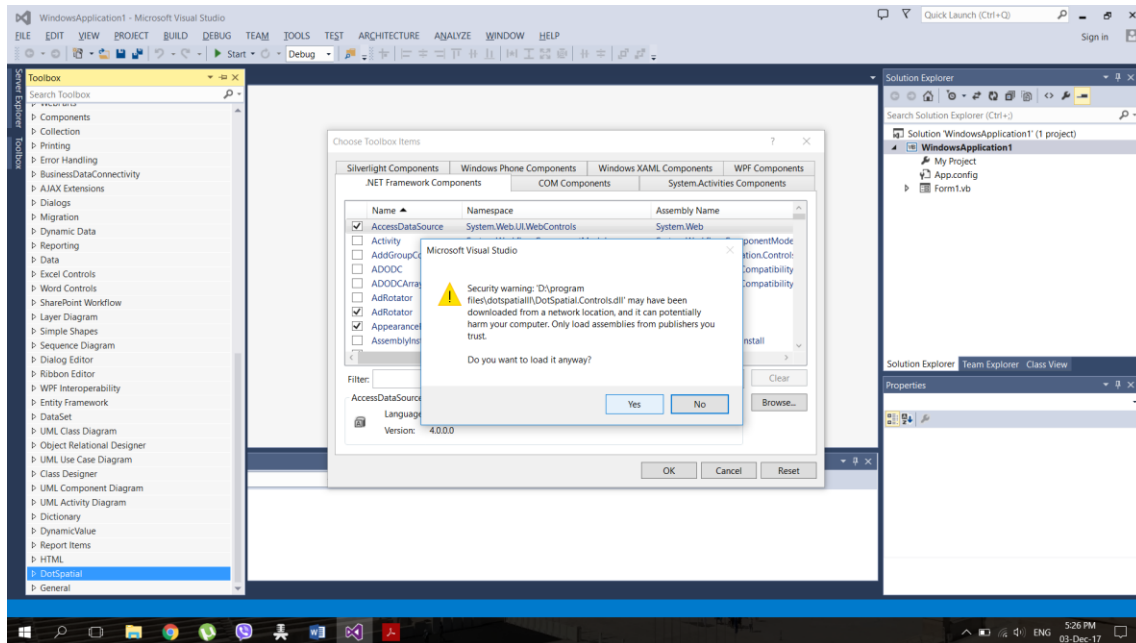


Εικόνα 4.5 – Εισαγωγή αρχείου Dotspatial.Controls.dll.



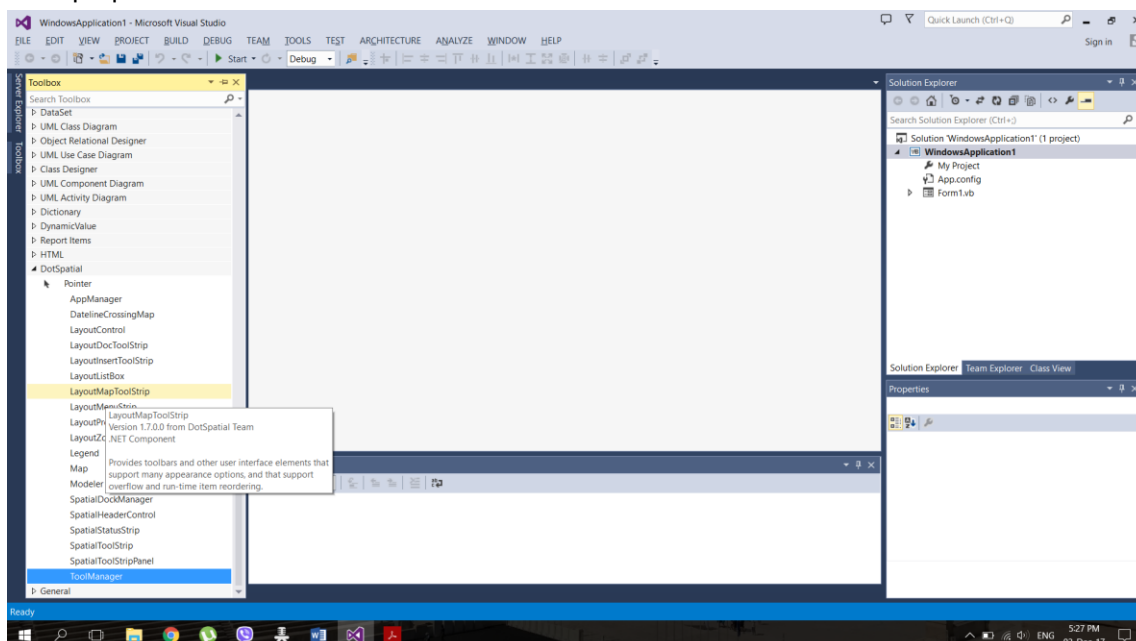
Εικόνα 4.6 – Επιλογή του αρχείου Dotspatial.Controls.dll από τα αρχεία που κατεβάσαμε από το Διαδίκτυο.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



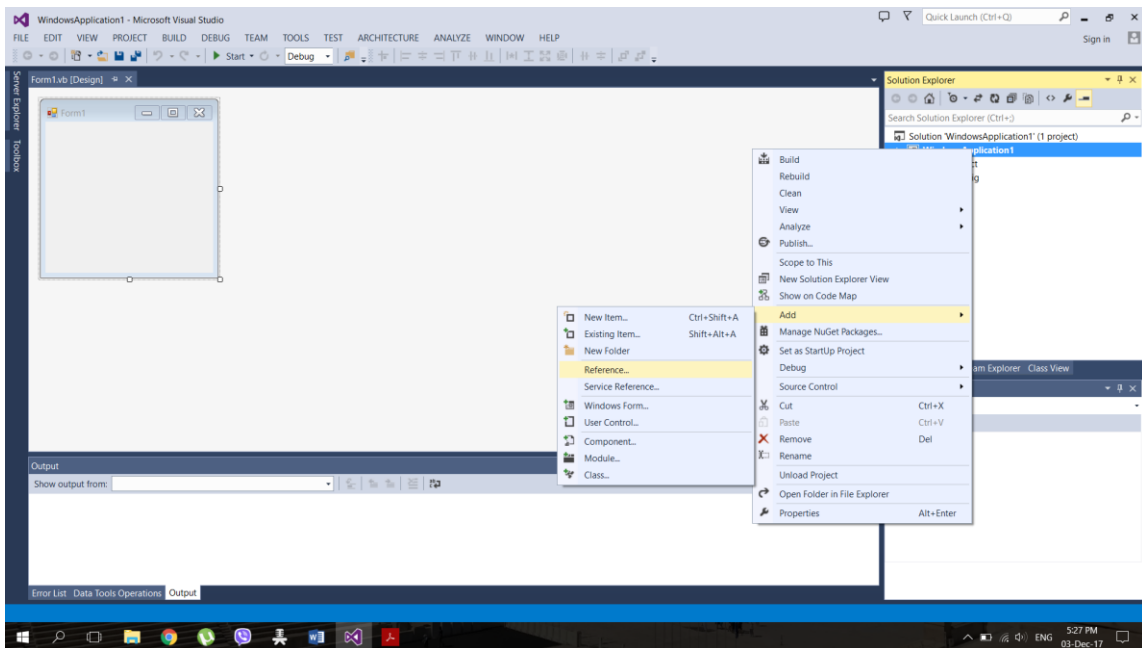
Εικόνα 4.7 – Επιλογή του αρχείου Dotspatial.Controls.dll από τα αρχεία που κατεβάσαμε από το διαδίκτυο.

Μόλις επιλεγθεί το ανωτέρω αρχείο θα εμφανιστεί ένα μεγάλο πλήθος εργαλείων στο toolbox του Visual Studio. Στο πεδίο References στα δεξιά μας πατάμε δεξί κλικ και στη συνέχεια επιλέγουμε Add References.



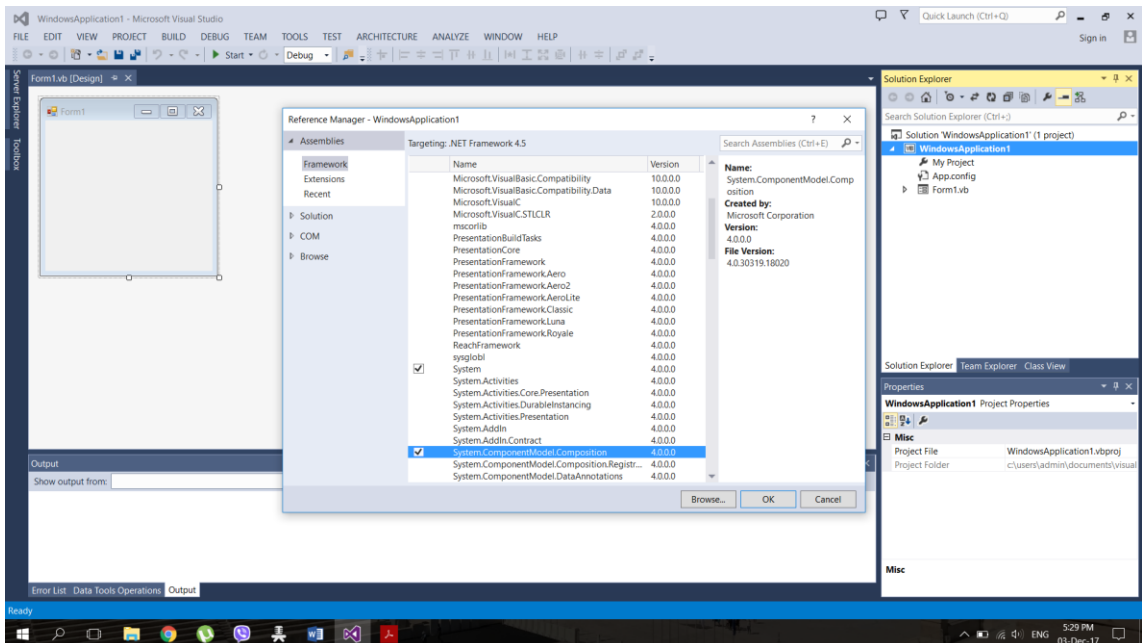
Εικόνα 4.8 – Εργαλεία DotSpatial στο toolbox του Visual Studio.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



Εικόνα 4.9 – Διαδικασία εισαγωγής References στο νέο project.

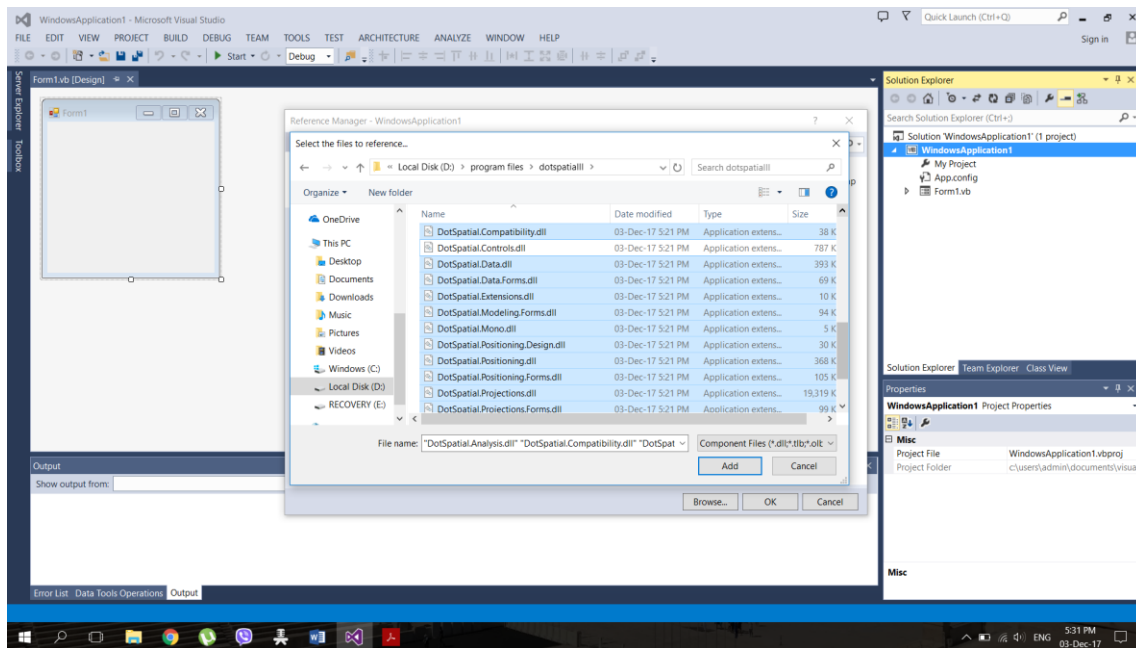
Στον φάκελο που εμφανίζεται επιλέγουμε το System.ComponentModel.Composition, μετά πατάμε Browse και επιλέγουμε σαν διαδρομή, τη διαδρομή των αρχείων του DotSpatial που κατεβάσαμε στην οποία βρίσκονται μέσα όλα τα dll.



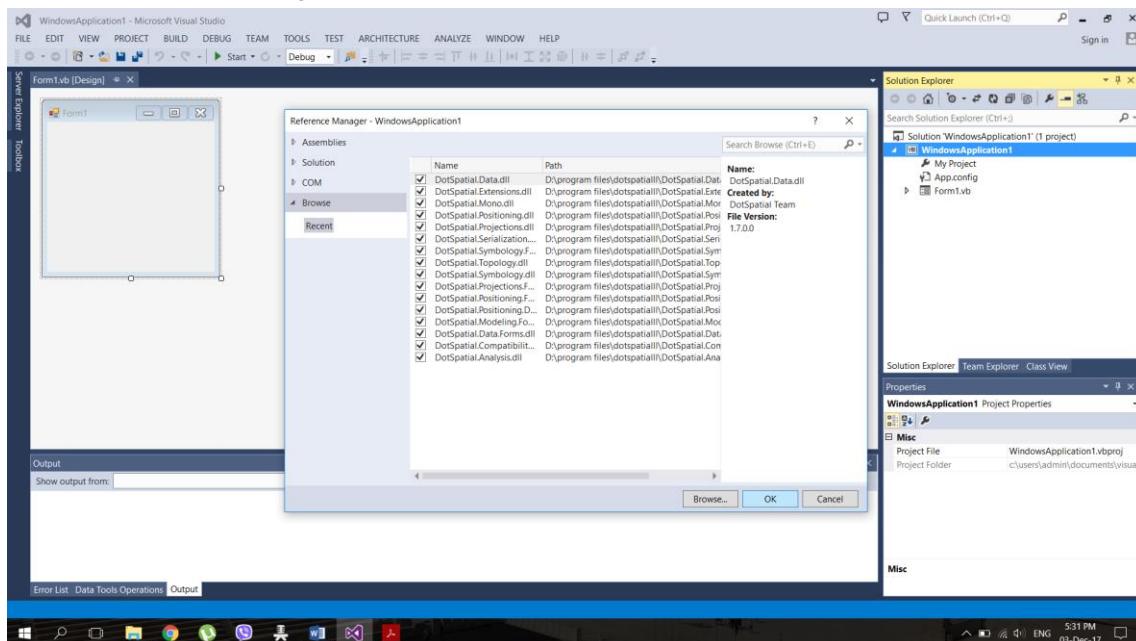
Εικόνα 4.10 – Διαδικασία εισαγωγής References στο νέο project.

Επιλέγουμε το σύνολο των dll εκτός του DotSpatial.Controls.dll, το οποίο έχει ήδη προστεθεί προηγουμένως.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



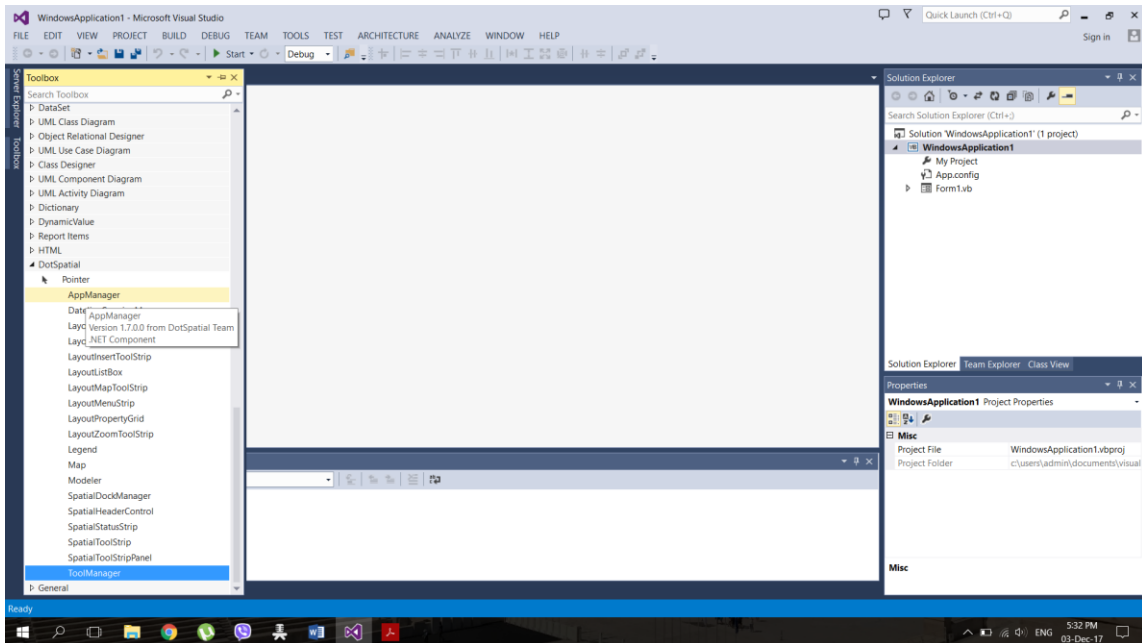
Εικόνα 4.11 – Επιλογή του συνόλου των dll εκτός του Dotspatial.Controls.dll, το οποίο έχει ήδη προστεθεί προηγουμένως.



Εικόνα 4.12 – Στάδια δημιουργίας ενός νέου project.

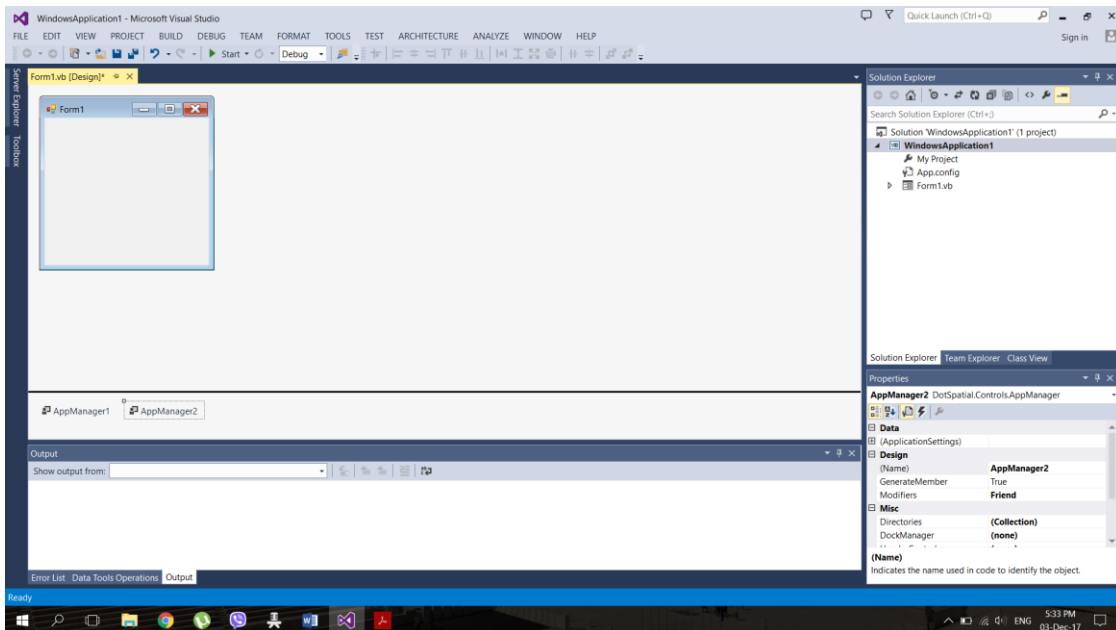
Βλέπουμε την ανανέωση της λίστας του references. Πηγαίνουμε στα εργαλεία του Dotspatial και σύρουμε το AppManager στο παράθυρο της φόρμας.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



Εικόνα 4.13 – Επιλογή του AppManager από τα εργαλεία.

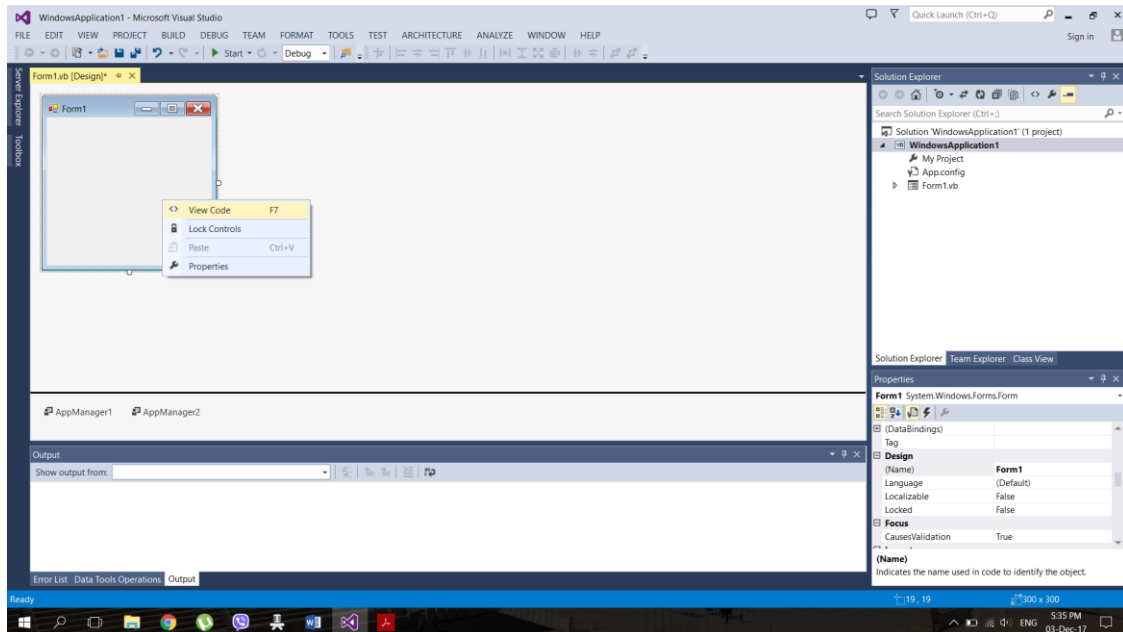
Με αυτόν τον τρόπο εμφανίζεται κάτω από τη φόρμα.



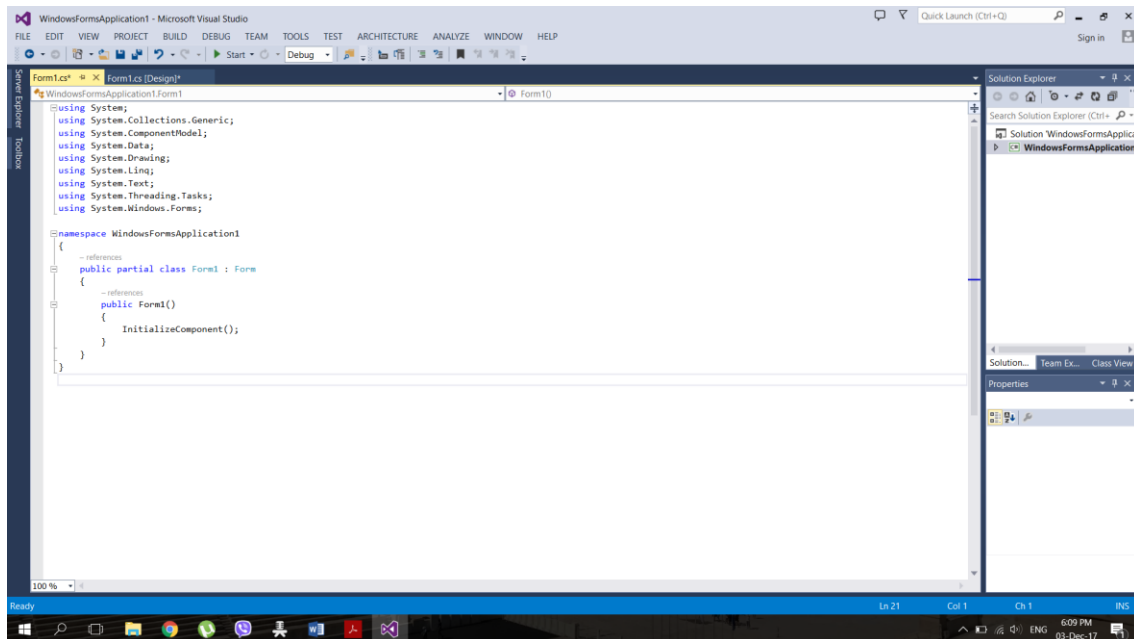
Εικόνα 4.14 – Εμφάνιση του AppManager κάτω από τη φόρμα.

Πατάμε δεξί κλικ στη φόρμα και πατάμε να δούμε τον κώδικα.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



Εικόνα 4.15 – Διαδικασία εισαγωγής κώδικα.



Εικόνα 4.16 – Εμφάνιση του κώδικα.

Προσθέτουμε κάτω από το τελευταίο using που υπάρχει, τον ακόλουθο κώδικα:

```
using System.ComponentModel.Composition;
```

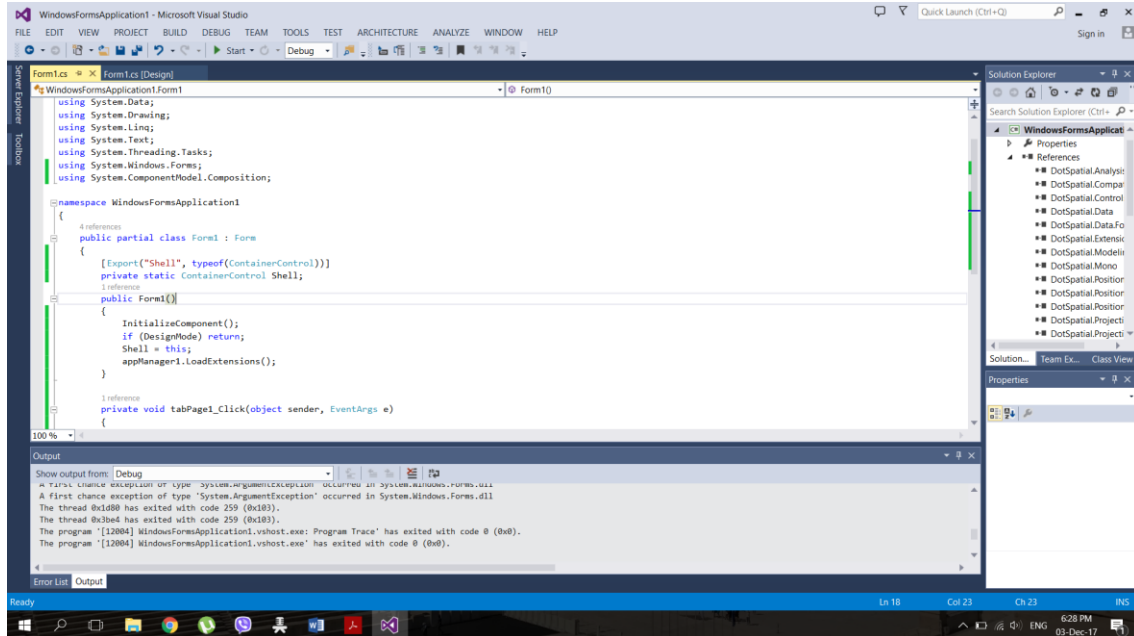
Μετά από αυτό το βήμα, μετά την αγκύλη { του public partial class Form1 : Form προσθέτουμε τον παρακάτω κώδικα:

```
[Export("Shell", typeof(ContainerControl))]
private static ContainerControl Shell;
```

Μετά το InitializeComponent(); προσθέτουμε:

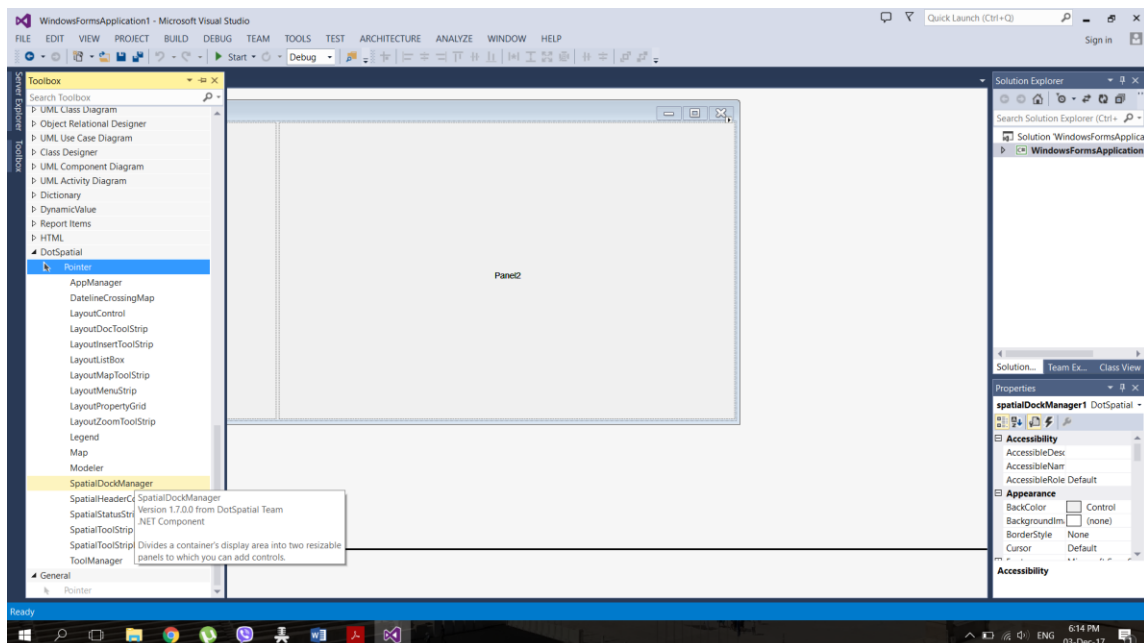
Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

```
If (DesignMode) return;
Shell = this;
appManager1.LoadExtensions();
```



Εικόνα 4.17 – Στάδια δημιουργίας ενός νέου project.

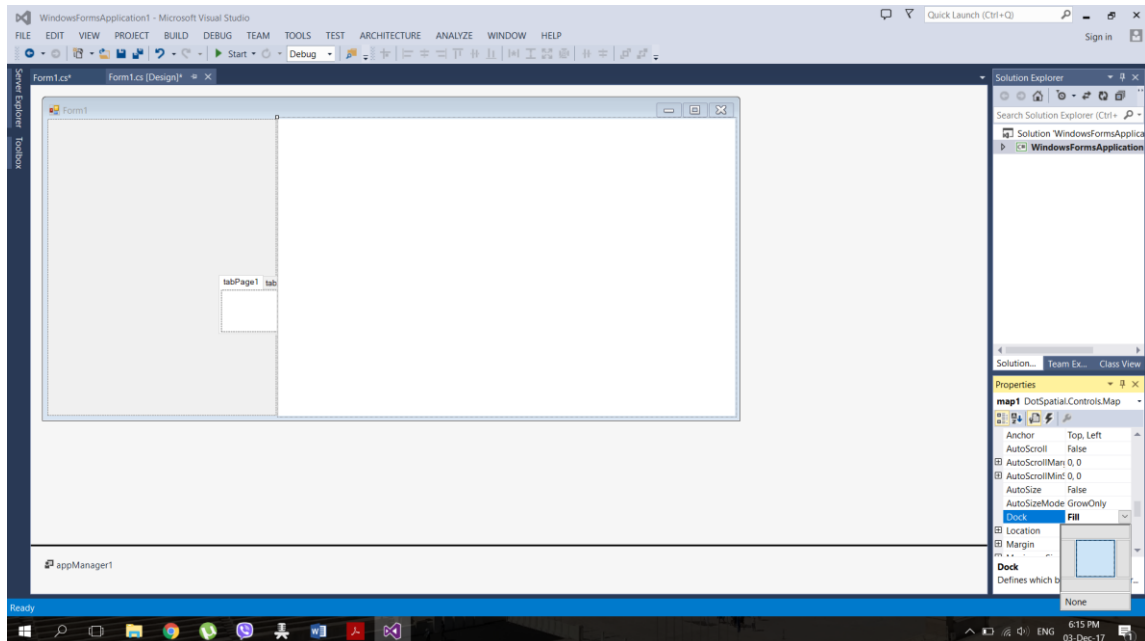
Η μπάρα του μενού δεν θα εμφανιστεί μέχρι να γίνει compile το project. Μπορούμε όμως ακόμα να προσθέσουμε δυνατότητες στον χάρτη, καθώς τα υπόλοιπα θα εκτελεστούν αυτόματα. Μοιράζουμε τη φόρμα μας στα δύο χρησιμοποιώντας το SpatialDockManager, απλά σέρνοντας το και αφήνοντας το πάνω στη φόρμα.



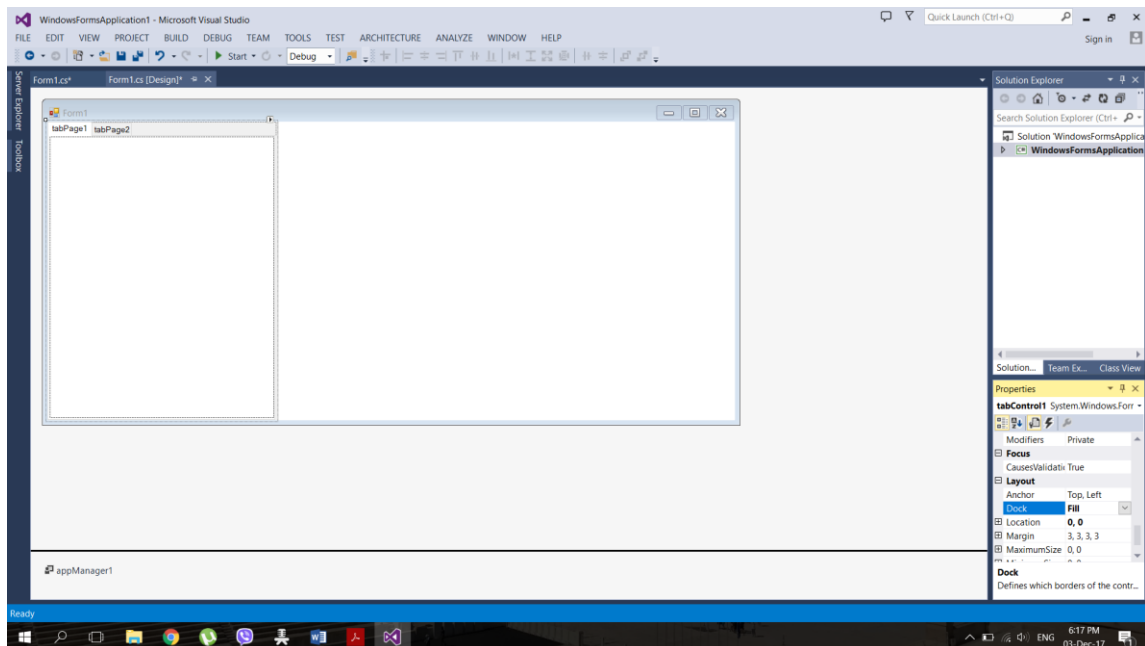
Εικόνα 4.18 – SpatialDockManager.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

Σέρνουμε και αφήνουμε αριστερά το TabControl και δεξιά το Map και επιλέγουμε στις ιδιότητες και των δύο Dock σε Fill.



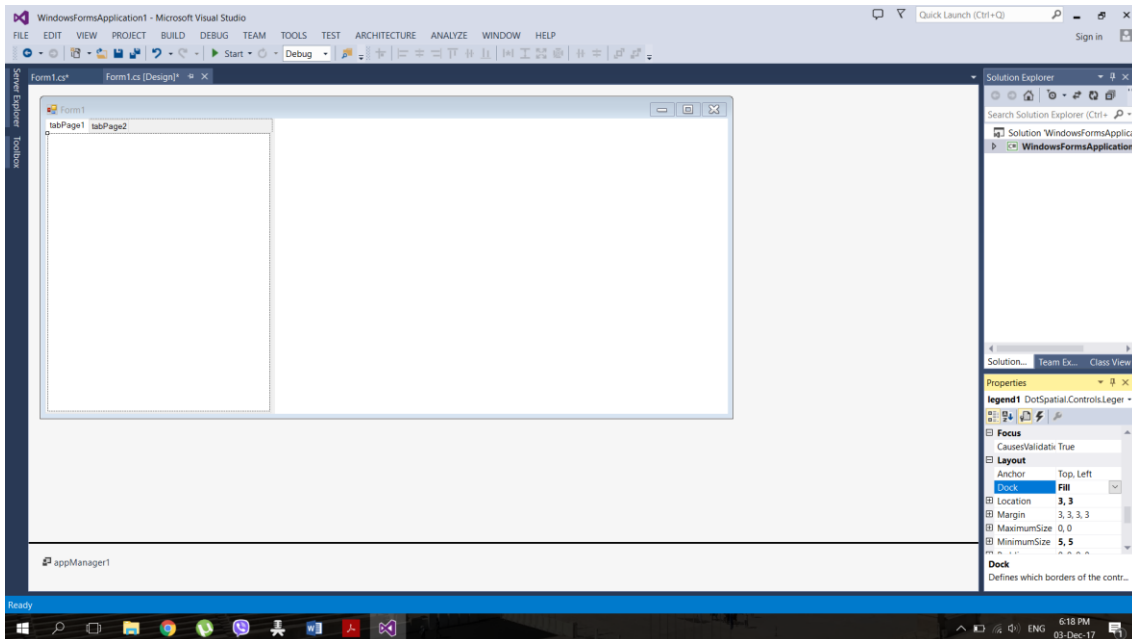
Εικόνα 4.19 - TabControl και Map.



Εικόνα 4.20 – Επιλογή στις ιδιότητες των dock TabControl και Map σε Fill.

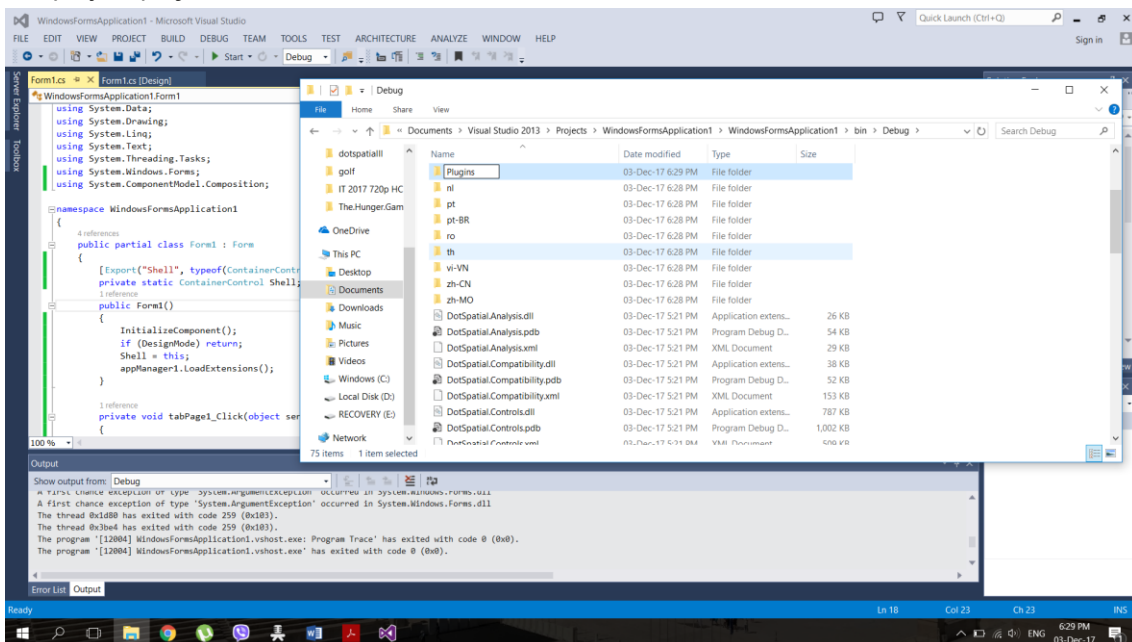
Στο TabControl θα προσθέσουμε το legend και τα DotSpatial Tools. Ονομάζουμε την πρώτη καρτέλα Legend και σέρνουμε και αφήνουμε το Legend.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



Εικόνα 4.21 – Επιλογή στις ιδιότητες του dock legend1 σε Fill.

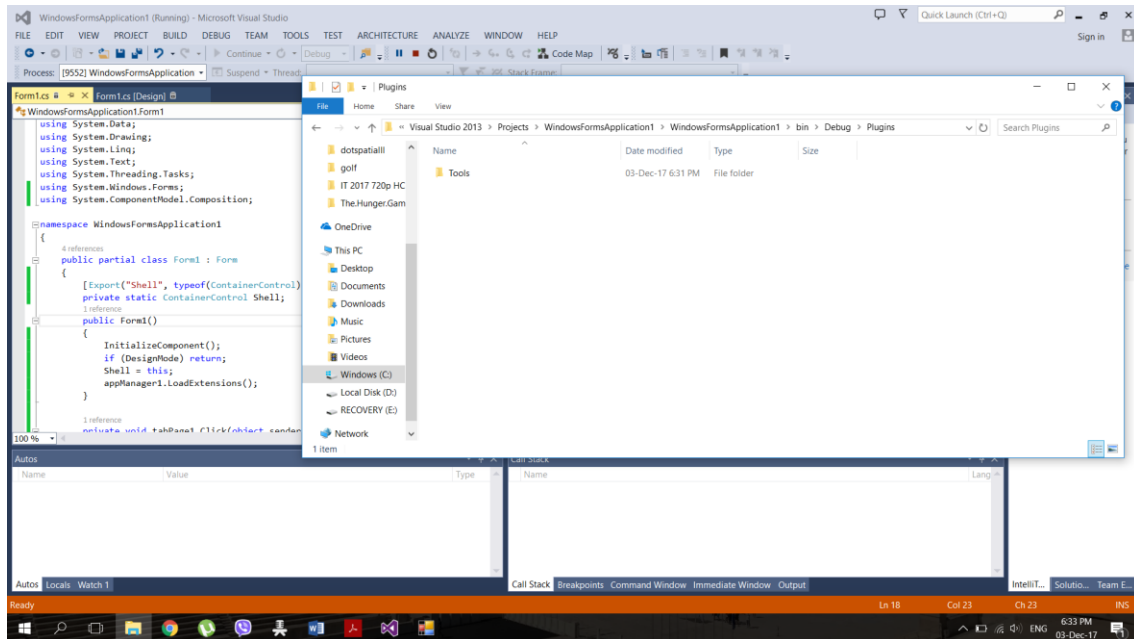
Τα toolbox θα χρησιμοποιηθούν σαν plugin. Δημιουργούμε έναν φάκελο Plugins στο bin/Debug του project μας.



Εικόνα 4.22 – Δημιουργία ενός φακέλου Plugins στο bin/Debug του project μας.

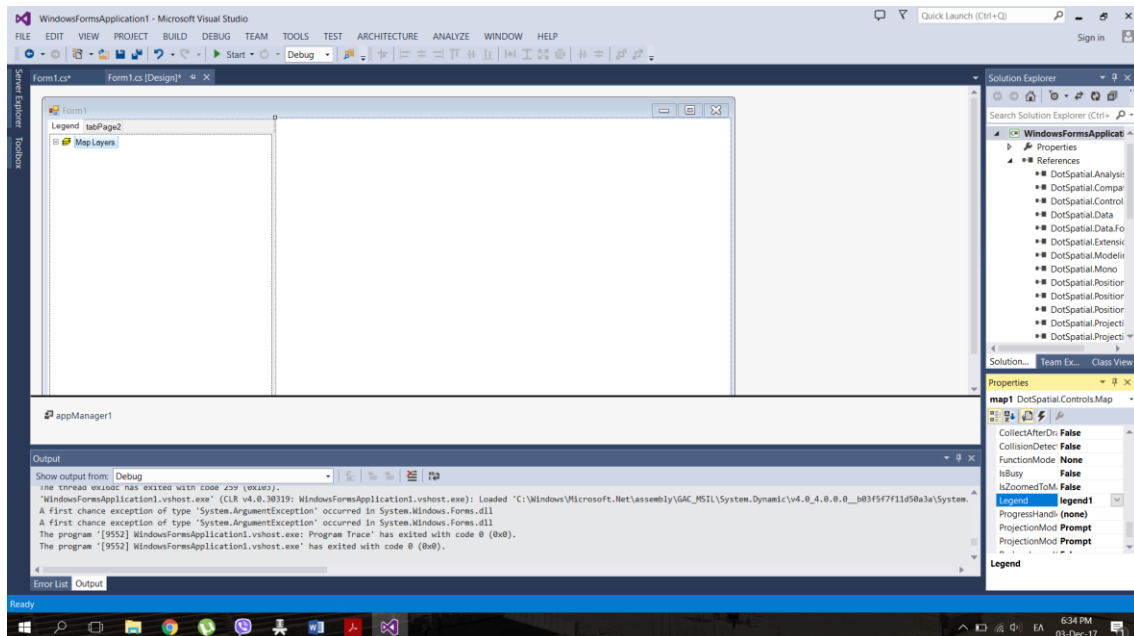
Από τα αρχεία που κατεβάσαμε αντιγράφουμε τον φάκελο tools και κάνουμε επικόλληση στον φάκελο Plugins που δημιουργήσαμε προηγουμένως.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



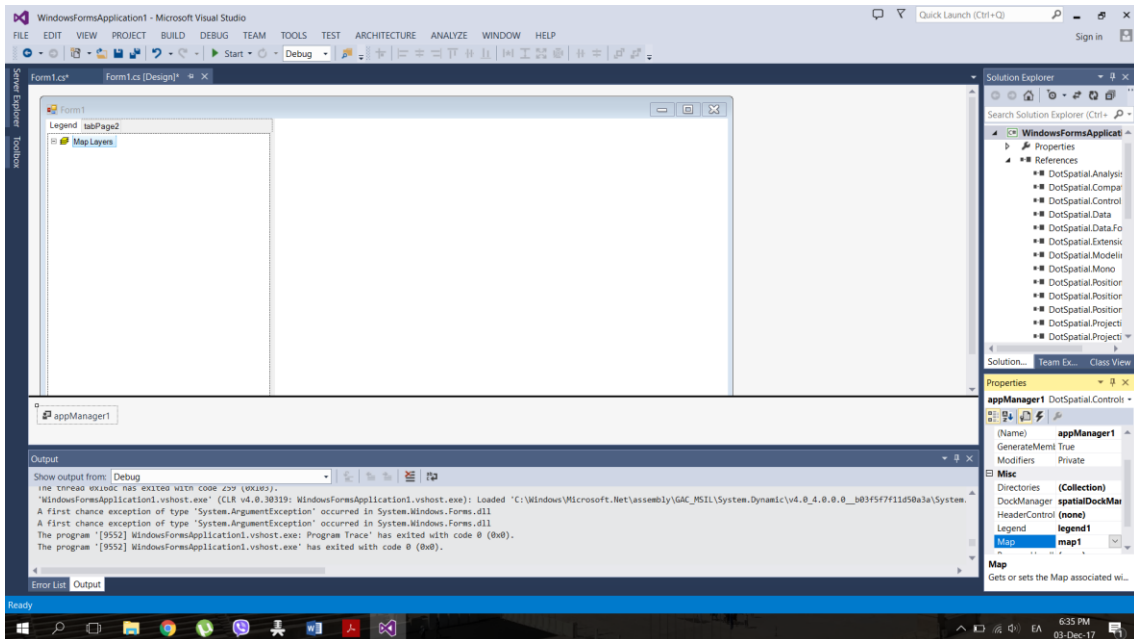
Εικόνα 4.23 – Αντιγραφή του φακέλου tools στον φάκελο Plugins

Τώρα θα εμφανίζεται μία καρτέλα δίπλα από το Legend με το όνομα Tools και έχουμε λειτουργικότητα όλων των εργαλείων. Μπορούμε να σβήσουμε την καρτέλα που περισσεύει. Το μόνο που μένει είναι να συνδέσουμε τα πάντα μεταξύ τους. Αυτό θα το καταφέρουμε απλά βάζοντας στις ιδιότητες των AppManager, DockManager, Legend, Map settings τα ονόματα που δώσαμε στο καθένα. Όταν εκτελεστεί με επιτυχία το συγκεκριμένο βήμα, θα εμφανιστεί στο Legend της φόρμας, το Map Layers.



Εικόνα 4.24 – Σύνδεση των εργαλείων μεταξύ τους.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



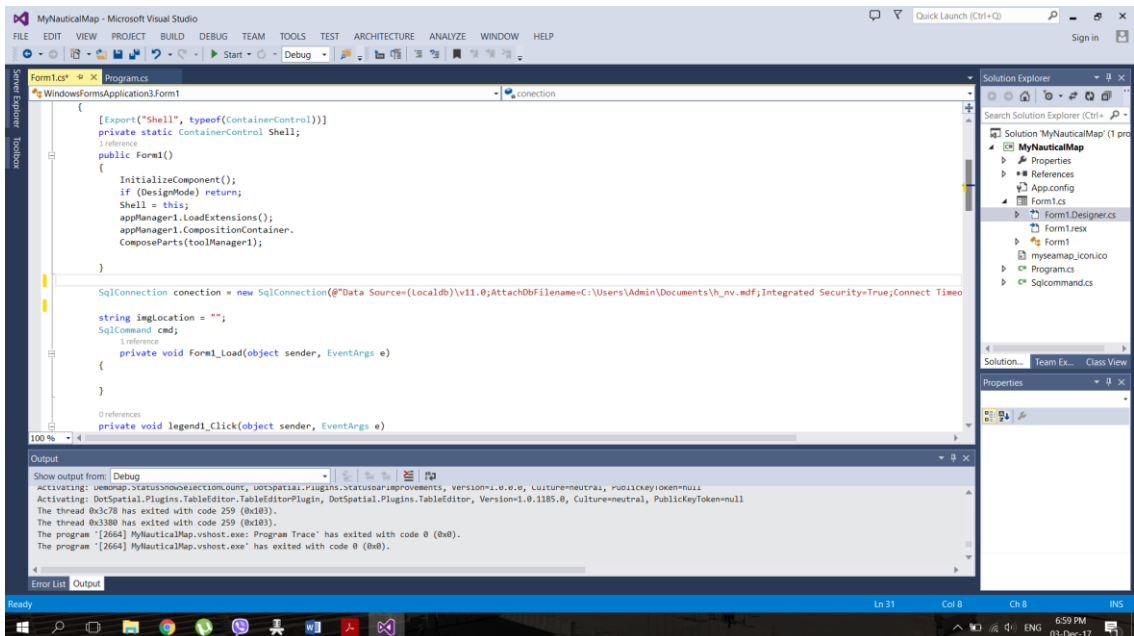
Εικόνα 4.25 – Σύνδεση των εργαλείων μεταξύ τους.

Τέλος προσθέτουμε τον κώδικα για την εμφάνιση των εργαλείων όπως φαίνεται στην εικόνα παρακάτω :

appManager1.CompositionContainer.

ComposeParts(toolManager1);

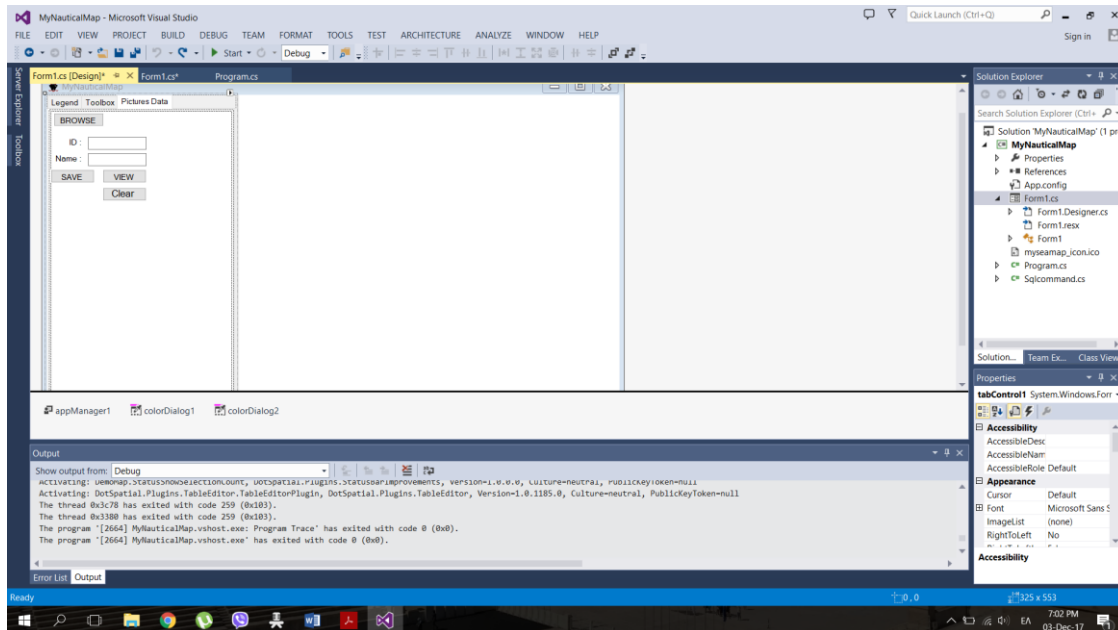
Κάτω από το InitializeComponent();



Εικόνα 4.26 – Ο κώδικας για την εμφάνιση των εργαλείων.

Δημιουργούμε μία νέα καρτέλα στο TabControl με το όνομα Pictures Data και τη μορφοποιούμε όπως την επόμενη εικόνα.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



Εικόνα 4.27 – Δημιουργία καρτέλας Pictures Data.

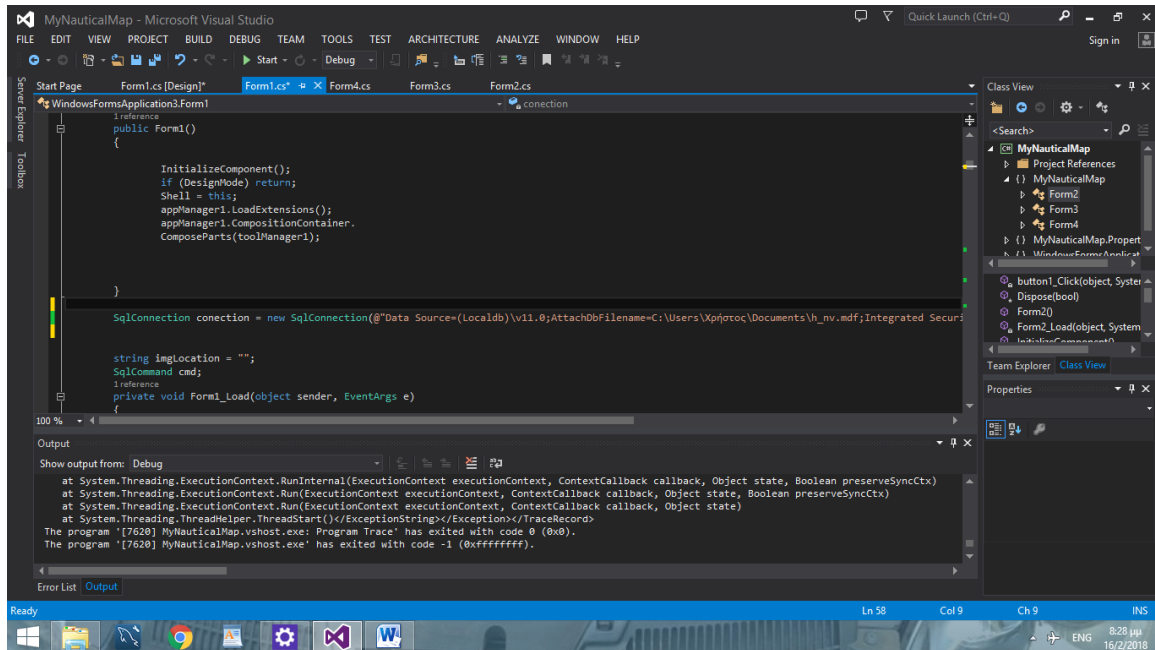
Ανοίγουμε μία νέα σύνδεση σε μία βάση δεδομένων με τον κώδικα :

```
SqlConnection connection = new SqlConnection(@"Data
Source=(Localdb)\v11.0;AttachDbFilename=C:\Users\Χρήστος\Documents\h_nv.mdf;Integrated
Security=True;Connect Timeout=30");
```

Και ακόμα δημιουργούμε μία μεταβλητή string και μία εντολή Sql:

```
string imgLocation = "";
SqlCommand cmd;
```

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



Εικόνα 4.28 – Εισαγωγή κώδικα για τη δημιουργία νέας σύνδεσης στη βάση δεδομένων των εικόνων.

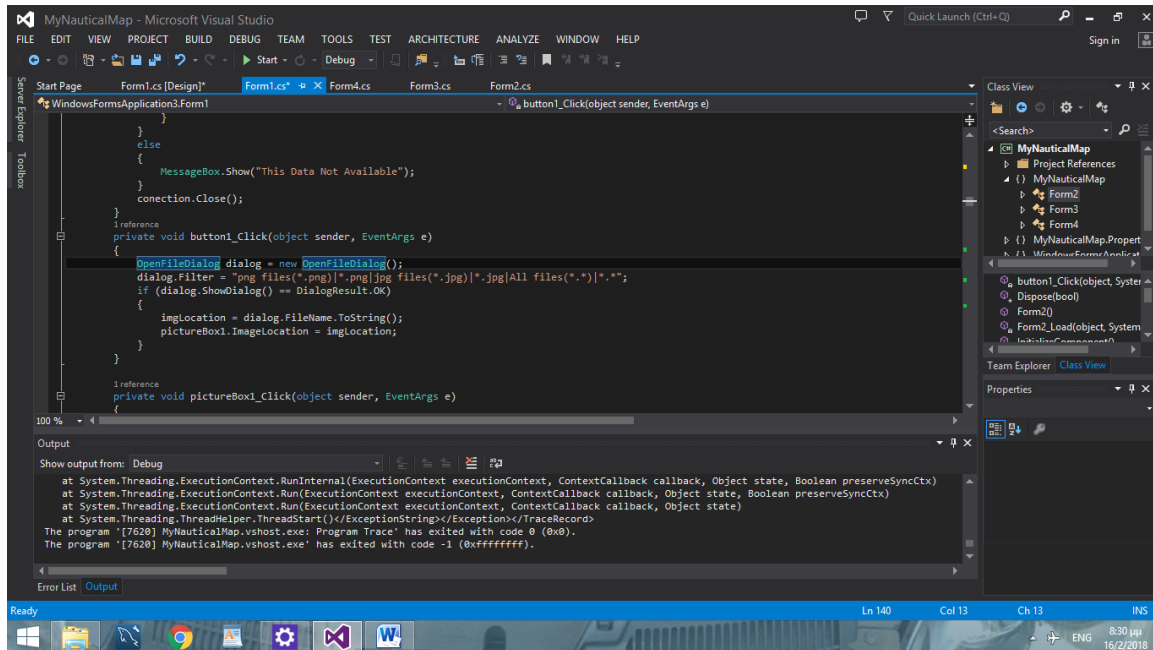
Πατάμε δύο φορές το button Browse, εμφανίζεται ο κώδικας και εισάγουμε τα παρακάτω:

```

OpenFileDialog dialog = new OpenFileDialog();
dialog.Filter = "png files (*.png)|*.png|jpg files (*.jpg)|*.jpg|All files (*.*)|*.*";
if (dialog.ShowDialog() == DialogResult.OK)
{
    imgLocation = dialog.FileName.ToString();
    pictureBox1.ImageLocation = imgLocation;
}

```

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



Εικόνα 4.29 – Εισαγωγή κώδικα για το Browse.

Πατάμε δύο φορές το Save, εμφανίζεται ο κώδικας και εισάγουμε τα παρακάτω:

```

conection.Open();
byte[] images = null;
if (textID.Text == "" || textName.Text == "")
{
    MessageBox.Show("Κάποιο ή κάποια από τα πεδία είναι κενά!");
    conection.Close();
}
else if(imgLocation == "")
{
    MessageBox.Show("Εισάγετε νέα φωτογραφία και όχι αυτή που υπάρχει ήδη στην
εικόνα!");
    conection.Close();
}
else
{
    FileStream Stream = new FileStream(imgLocation, FileMode.Open,
FileAccess.Read);
    BinaryReader brs = new BinaryReader(Stream);
    images = brs.ReadBytes((int)Stream.Length);
    SqlCommand cmdCount = new SqlCommand("SELECT count(*) from CIW WHERE
Id = " + textID.Text + "", conection);
    int count = (int)cmdCount.ExecuteScalar();
    if (count > 0)

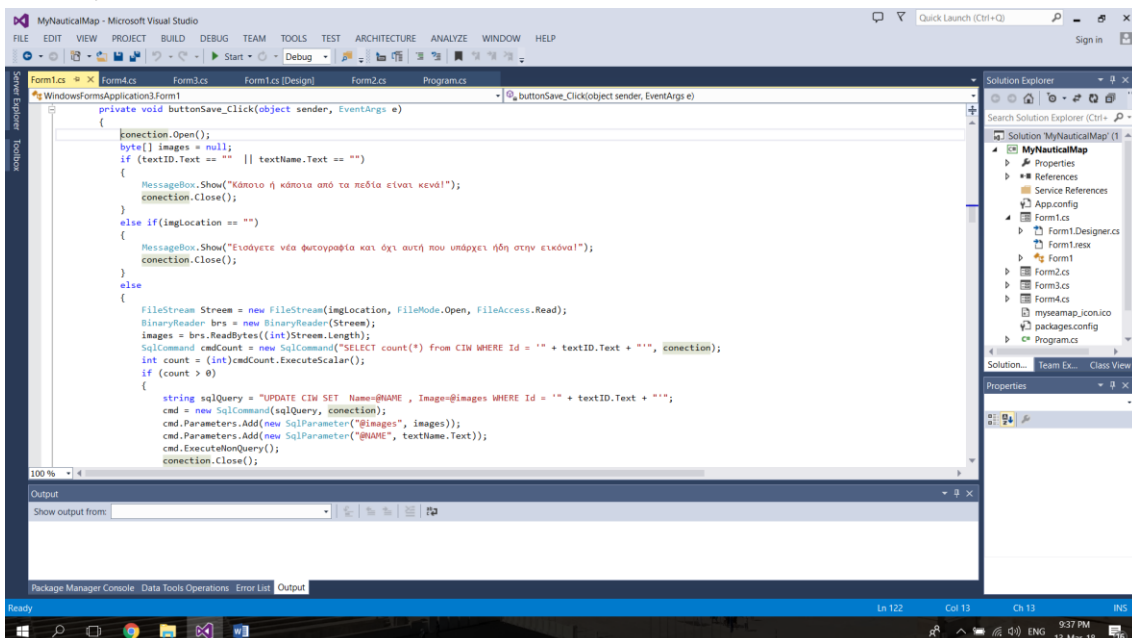
```

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

```

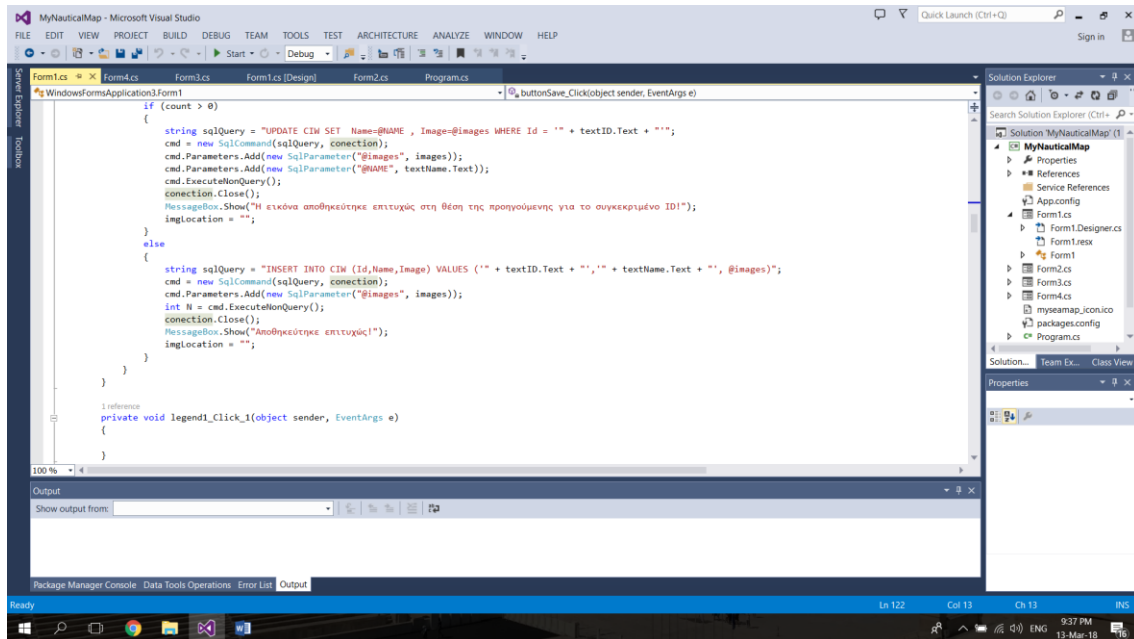
{
    string sqlQuery = "UPDATE CIW SET Name=@NAME , Image=@images WHERE
Id = '" + textID.Text + "'";
    cmd = new SqlCommand(sqlQuery, conection);
    cmd.Parameters.Add(new SqlParameter("@images", images));
    cmd.Parameters.Add(new SqlParameter("@NAME", textName.Text));
    cmd.ExecuteNonQuery();
    conection.Close();
    MessageBox.Show("Η εικόνα αποθηκεύτηκε επιτυχώς στη θέση της προηγούμενης
για το συγκεκριμένο ID!");
    imgLocation = "";
}
else
{
    string sqlQuery = "INSERT INTO CIW (Id,Name,Image) VALUES ('" + textID.Text +
",'" + textName.Text + "', @images)";
    cmd = new SqlCommand(sqlQuery, conection);
    cmd.Parameters.Add(new SqlParameter("@images", images));
    int N = cmd.ExecuteNonQuery();
    conection.Close();
    MessageBox.Show("Αποθηκεύτηκε επιτυχώς!");
    imgLocation = "";
}
}
}

```



Εικόνα 4.30 – Εισαγωγή κώδικα για το Save.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



Εικόνα 4.31 – Εισαγωγή κώδικα για το Save.

Πατάμε δύο φορές το View, εμφανίζεται ο κώδικας και εισάγουμε τα παρακάτω:

```

conection.Open();
conection.Open();

string sqlQuery = "select Name,Image from CIW where id='" + textID.Text + "'";
cmd = new SqlCommand(sqlQuery, conection);
SqlDataReader Dataread = cmd.ExecuteReader();
Dataread.Read();
if (Dataread.HasRows)
{
    textName.Text = Dataread[0].ToString();
    byte[] images = (byte[])Dataread[1];
    if (images == null)
    {
        pictureBox1.Image = null;
    }
    else
    {
        MemoryStream mstream = new MemoryStream(images);
        pictureBox1.Image = Image.FromStream(mstream);
    }
}
else
{
    MessageBox.Show("Δεν υπάρχει εικόνα για αυτό το ID");
}

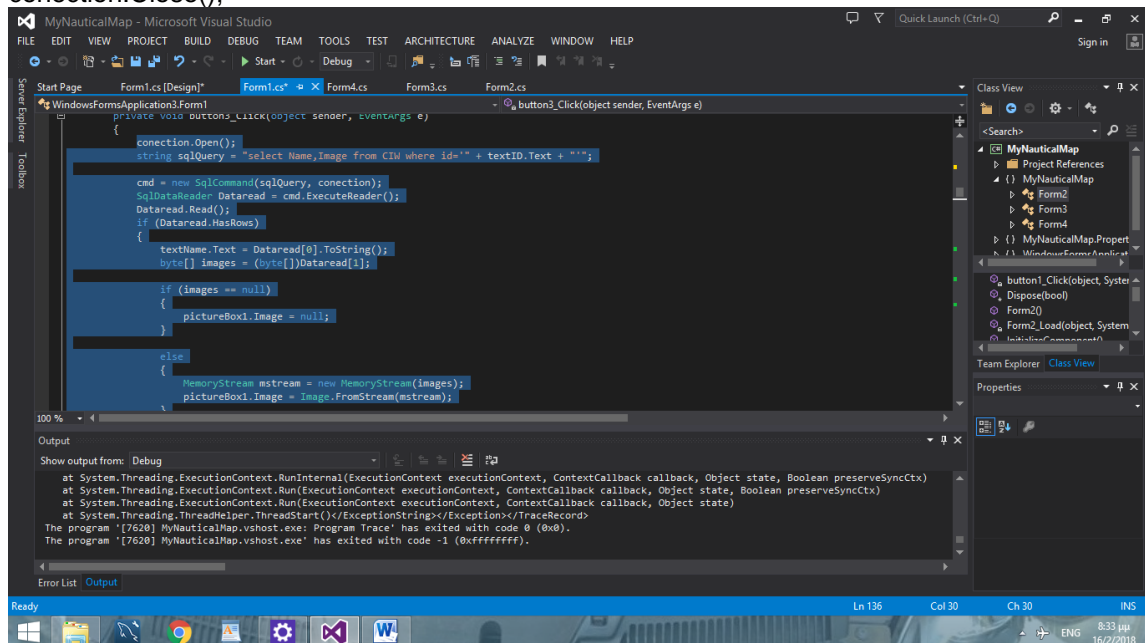
```

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

```

    }
    connection.Close();

```



Εικόνα 4.32 – Εισαγωγή κώδικα για το View.

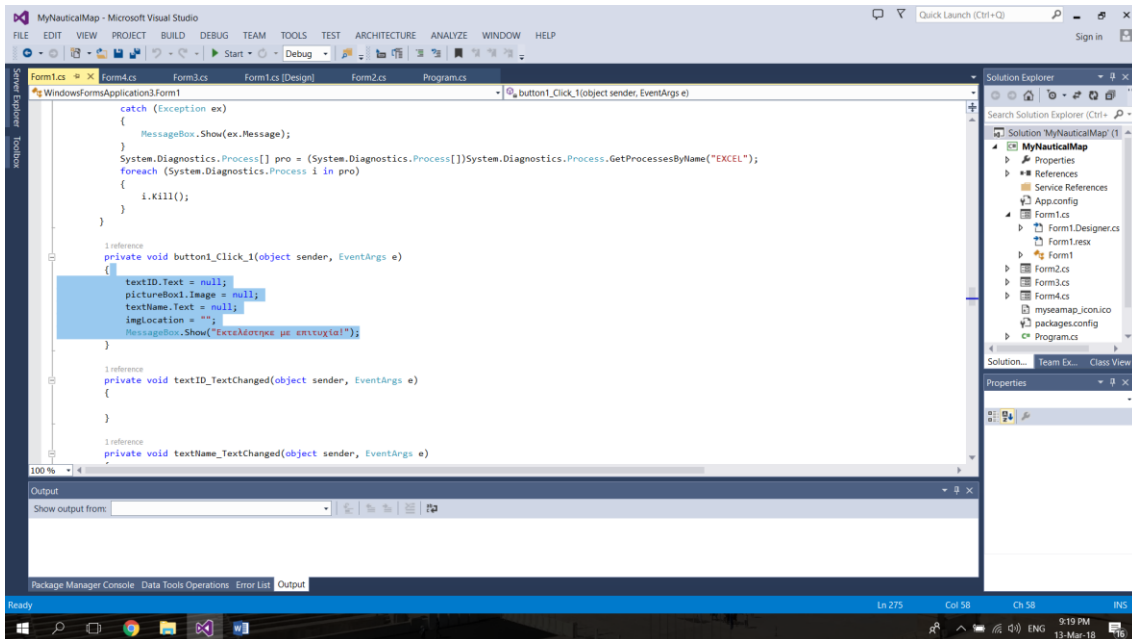
Πατάμε δύο φορές το clear, εμφανίζεται ο κώδικας και εισάγουμε τα παρακάτω:

```

textID.Text = null;
pictureBox1.Image = null;
textName.Text = null;
imgLocation = "";
MessageBox.Show("Εκτελέστηκε με επιτυχία!");

```

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



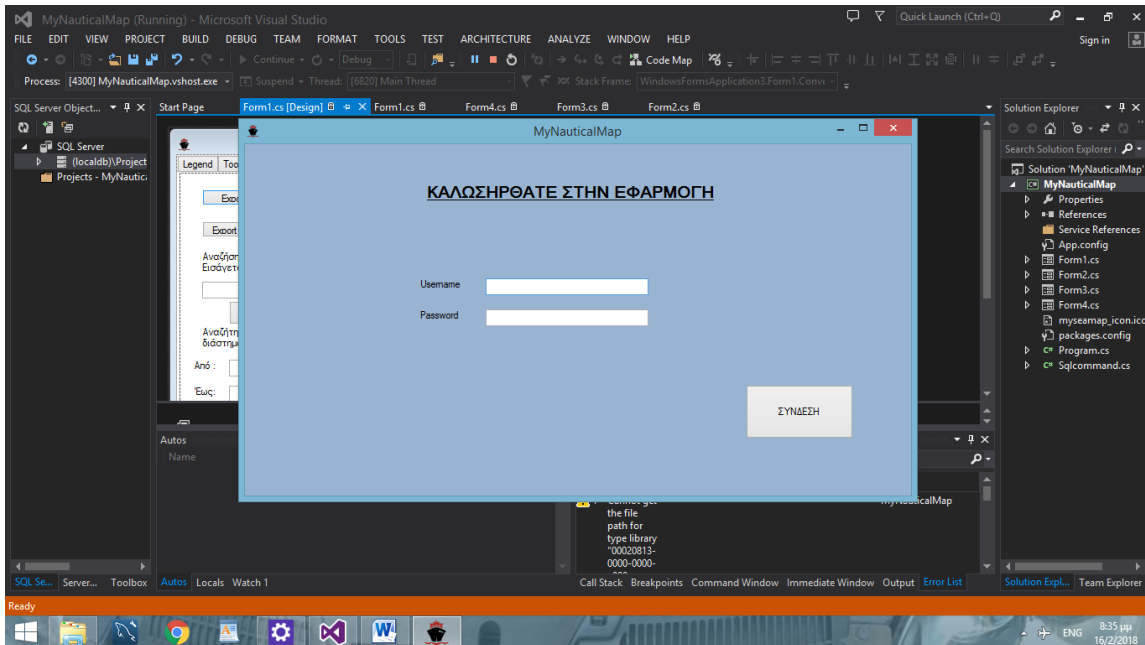
Εικόνα 4.33 – Εισαγωγή κώδικα για το clear.

Στο form1 κάτω από το public partial class Form1 : Form{, εισάγουμε τις στατικές δημόσιες μεταβλητές που θα χρησιμεύσουν στις λειτουργίες αναζήτησης της βάσης δεδομένων του πίνακα χαρακτηριστικών μας:

```
public static string pass;
public static string pass1;
public static DateTime pass2;
public static DateTime pass3;
```

Δημιουργούμε μία νέα φόρμα form2 με τον κάτωθι κώδικα και μορφοποίηση:

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



Εικόνα 4.34 – Δημιουργία μίας νέας φόρμα form2 για τη σύνδεση χρήστη.

Κώδικας:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using WindowsFormsApplication3;
using System.Data.SqlClient;
```

```
namespace MyNauticalMap
```

```
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
```

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

```

    if (textBox1.Text == "" || textBox2.Text == "")
    {
        MessageBox.Show("Παρακαλώ εισάγετε τα στοιχεία σας και στα δύο πεδία");
        textBox1.Text = "";
        textBox2.Text = "";
    }
    else
    {
        SqlConnection con = new SqlConnection(@"Data
Source=(LocalDB)\v11.0;AttachDbFilename=C:\Users\Χρήστος\Documents\personel.mdf;Integr
ated Security=True;Connect Timeout=30;");
        SqlDataAdapter sda = new SqlDataAdapter("Select Count(*) From login where
username=" + textBox1.Text + " and password=" + textBox2.Text + """, con);
        DataTable dt = new DataTable();
        sda.Fill(dt);
        if (dt.Rows[0][0].ToString() == "1")
        {
            this.Hide();
            Form1 = new Form1();
            form1.Show();
        }
        else
        {
            MessageBox.Show("Λάθος Στοιχεία!Παρακαλώ ξαναεισάγετε τα στοιχεία σας");
            textBox1.Text = "";
            textBox2.Text = "";
        }
    }
}

private void label1_Click(object sender, EventArgs e)
{
}

private void label3_Click(object sender, EventArgs e)
{
}

private void Form2_Load(object sender, EventArgs e)
{

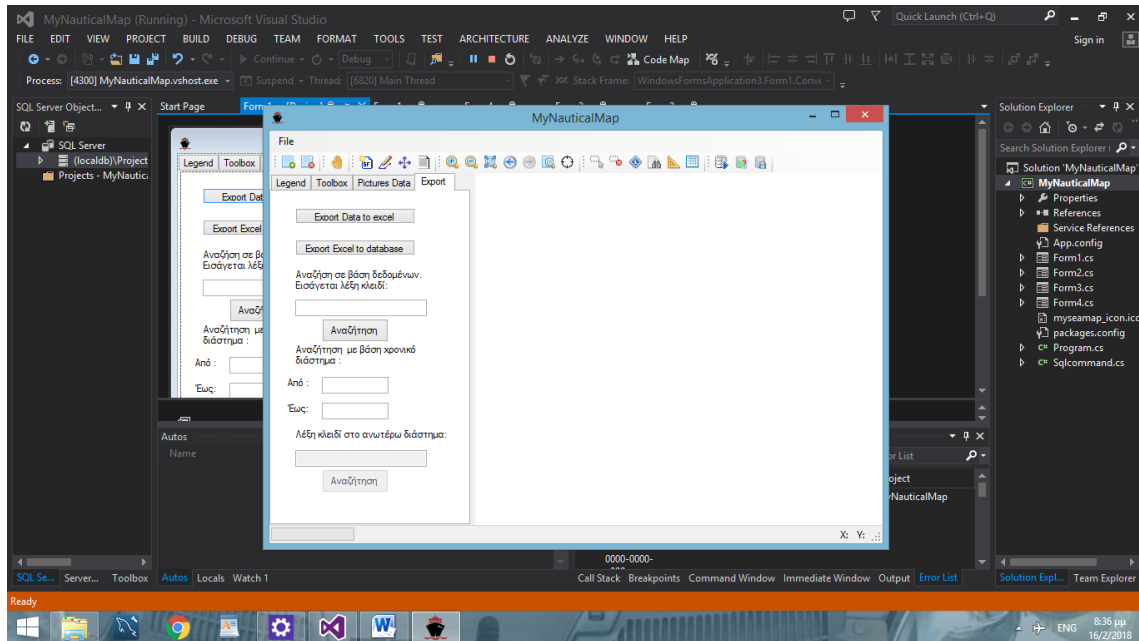
```

```

}
}
}

```

Δημιουργούμε μία νέα καρτέλα με το όνομα export όπως φαίνεται στην εικόνα:



Εικόνα 4.35 – Δημιουργία μίας νέας καρτέλα με το όνομα export.

Πατάμε δύο φορές το export data to Excel και εισάγουμε τον παρακάτω κώδικα:

```

System.Data.DataTable dt = null;
if (map1.Layers.Count > 0)
{
    MapPointLayer stateLayer = default(MapPointLayer);
    stateLayer = map1.Layers[0] as MapPointLayer;
    if (stateLayer == null)
    {
        MessageBox.Show(" Αυτό το layer δεν είναι τύπου σημείων ή εργάζεστε με
πολλαπλά layers!");
    }
    else
    {
        //Απουκεία του shapefile's attribute table στον πίνακα dt
        dt = stateLayer.DataSet.DataTable;
        ExportToExcel(dt);
    }
}
else

```

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.


```

    {
        MessageBox.Show("Παρακαλώ προσθέστε layer στον χάρτη.");
    }

```

Πατάμε δύο φορές το export Excel to database και εισάγουμε τον παρακάτω κώδικα:

```

private void button3_Click_1(object sender, EventArgs e)
{
    OpenExcelFile();
}

public static FeatureSet OpenExcelFile()
{
    OpenFileDialog openFileDialog = new OpenFileDialog();
    openFileDialog.Filter = "Excel Files|*.xls";
    if (openFileDialog.ShowDialog() == DialogResult.OK)
    {
        DataTable excelTable = ConvertExcelFileToDataTable(openFileDialog.FileName);
    }
    return null;
}

private static DataTable ConvertExcelFileToDataTable(string excelFileName)
{
    string connectionString =
        String.Format("Provider=Microsoft.ACE.OLEDB.12.0;Data Source={0};Extended
        Properties=\"Excel 12.0 Xml;HDR=YES;IMEX=1\"", excelFileName);

    using (OleDbConnection connection = new OleDbConnection(connectionString))
    {
        string query = "SELECT * FROM [Sheet1$]";
        connection.Open();
        OleDbCommand objCmdSelect = new OleDbCommand(query, connection);
        OleDbDataReader objDR = null;
        OleDbDataAdapter adapter = new OleDbDataAdapter(objCmdSelect);
        DataTable excelTable = new DataTable();
        adapter.Fill(excelTable);
        SqlConnection SQLconn = new SqlConnection(@"Data
        Source=(Localdb)\v11.0;AttachDbFilename=C:\Users\Admin\Documents\DbfToSQL.mdf;Integra
        ted Security=True;Connect Timeout=30");
        SQLconn.Open();
        string sqlTrunc = "TRUNCATE TABLE AGGK";
        SqlCommand cmdlast = new SqlCommand(sqlTrunc, SQLconn);
        cmdlast.ExecuteNonQuery();
    }
}

```

Σχεδίαση και ανάπτυξη μίας
φιλικής εφαρμογής γεωγραφικών δεδομένων
με τη βοήθεια του Microsoft Visual Studio
και των βιβλιοθηκών DotSpatial.

```

using (SqlBulkCopy bulkCopy = new SqlBulkCopy(SQLconn))
{
    bulkCopy.DestinationTableName = "AGGK";
    // Set up the column mappings by name.
    SqlBulkCopyColumnMapping mapUESH =
        new SqlBulkCopyColumnMapping("UESH", "UESH");
    bulkCopy.ColumnMappings.Add(mapUESH);
    SqlBulkCopyColumnMapping mapUERMOKRASIA =
        new SqlBulkCopyColumnMapping("UERM_PER", "UERM_PER");
    bulkCopy.ColumnMappings.Add(mapUERMOKRASIA);
    SqlBulkCopyColumnMapping mapUERMOKRASIA_U =
        new SqlBulkCopyColumnMapping("UERM_UAL", "UERM_UAL");
    bulkCopy.ColumnMappings.Add(mapUERMOKRASIA_U);
    SqlBulkCopyColumnMapping mapPYKNOTHTA_U =
        new SqlBulkCopyColumnMapping("DENSITY", "DENSITY");
    bulkCopy.ColumnMappings.Add(mapPYKNOTHTA_U);
    SqlBulkCopyColumnMapping mapPLHROFORIES =
        new SqlBulkCopyColumnMapping("INFO", "INFO");
    bulkCopy.ColumnMappings.Add(mapPLHROFORIES);
    SqlBulkCopyColumnMapping mapTAXYTHTA_HX =
        new SqlBulkCopyColumnMapping("TAX_HXOY", "TAX_HXOY");
    bulkCopy.ColumnMappings.Add(mapTAXYTHTA_HX);
    SqlBulkCopyColumnMapping mapHMEROMHНИЯ =
        new SqlBulkCopyColumnMapping("DATE", "DATE");
    bulkCopy.ColumnMappings.Add(mapHMEROMHНИЯ);
    SqlBulkCopyColumnMapping mapTHL_KALYCH =
        new SqlBulkCopyColumnMapping("THLKALYCH", "THLKALYCH");
    bulkCopy.ColumnMappings.Add(mapTHL_KALYCH);
    SqlBulkCopyColumnMapping mapREYMATA =
        new SqlBulkCopyColumnMapping("REYMATA", "REYMATA");
    bulkCopy.ColumnMappings.Add(mapREYMATA);
    SqlBulkCopyColumnMapping mapNAYT_KINSHS =
        new SqlBulkCopyColumnMapping("NAYT_KIN", "NAYT_KIN");
    bulkCopy.ColumnMappings.Add(mapNAYT_KINSHS);
    try
    {
        objDR = objCmdSelect.ExecuteReader();
        bulkCopy.WriteToServer(objDR);
        connection.Close();
        SQLconn.Close();
        MessageBox.Show("Εκτελέστηκε με επιτυχία!");
    }
}

```

```

    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
    }
    return excelTable;
}
}
}

```

Στην πρώτη αναζήτηση πατώντας δύο φορές το κλικ εισάγουμε τον παρακάτω κώδικα:

```

    pass = textBox1.Text;
    Form3 = new Form3();
    form3.Show();

```

Σε αυτό το σημείο ανοίγει το form3 που έχει σαν κώδικα τα εξής:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.ComponentModel.Composition;
using System.Data.SqlClient;
using System.Windows;
using System.IO;
using MyNauticalMap;
using System.Data.OleDb;
using DotSpatial.Controls;
using DotSpatial.Data;
using DotSpatial.Symbology;
using DotSpatial.Topology;
//using System.Data.OleDb;
using DotSpatial.Projections;
//using DotSpatial.Topology;
using SQL = System.Data;
using WindowsFormsApplication3;

```

```

namespace MyNauticalMap
{

```

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

```

public partial class Form3 : Form
{
    public Form3()
    {
        InitializeComponent();
    }

    private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
    {
    }

    private void Form3_Load(object sender, EventArgs e)
    {
        SqlConnection SQLconn = new SqlConnection(@"Data
Source=(Localdb)\v11.0;AttachDbFilename=C:\Users\Admin\Documents\DbfToSQL.mdf;Integra
ted Security=True;Connect Timeout=30");
        SQLconn.Open();
        String query = "SELECT * FROM AGGK WHERE ((UERM_PER LIKE '%" + Form1.pass
+ "%') OR (THLKALYCH LIKE '%" + Form1.pass + "%')OR (NAYT_KIN LIKE '%" + Form1.pass
+ "%') OR (UERM_UAL LIKE '%" + Form1.pass + "%') OR (UESH LIKE '%" + Form1.pass +
"%') OR (INFO LIKE '%" + Form1.pass + "%') OR (DENSITY LIKE '%" + Form1.pass + "%')OR
(REYMATA LIKE '%" + Form1.pass + "%') OR (TAX_HXOY LIKE '%" + Form1.pass + "%')) ";
        SqlDataAdapter sda = new SqlDataAdapter(query, SQLconn);
        DataTable dt = new DataTable();
        sda.Fill(dt);
        BindingSource bsource = new BindingSource();
        bsource.DataSource = dt;
        dataGridView1.DataSource = bsource;
        sda.Update(dt);
    }

    private void button1_Click(object sender, EventArgs e)
    {
        this.Close();
    }
}

```

Στη δεύτερη αναζήτηση πατώντας δύο φορές το κλικ εισάγουμε τον παρακάτω κώδικα:

```

try
{
    pass1 = textBox2.Text;

```

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

```

        DateTime dt = Convert.ToDateTime(textBox3.Text);
        string s2 = dt.ToString("dd-MM-yyyy");
        pass2 = Convert.ToDateTime(s2);
        DateTime dt1 = Convert.ToDateTime(textBox4.Text);
        string s3 = dt1.ToString("dd-MM-yyyy");
        pass3 = Convert.ToDateTime(s3);
        Form4 = new Form4();
        form4.Show();
    }
    catch (Exception ex)
    {
        if (!(
            ex is FormatException ||
            ex is OverflowException))
        {
            throw;
        }
        Console.WriteLine("Fill both textboxes with appropriate dates");
    }
}

```

Σε αυτό το σημείο ανοίγει το form4 που έχει σαν κώδικα τα εξής:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.ComponentModel.Composition;
using System.Data.SqlClient;
using System.Windows;
using System.IO;
using MyNauticalMap;
using System.Data.OleDb;
using DotSpatial.Controls;
using DotSpatial.Data;
using DotSpatial.Symbology;
using DotSpatial.Topology;
//using System.Data.OleDb;
using DotSpatial.Projections;

```

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

```

//using DotSpatial.Topology;
using SQL = System.Data;
using WindowsFormsApplication3;

namespace MyNauticalMap
{
    public partial class Form4 : Form
    {
        public Form4()
        {
            InitializeComponent();

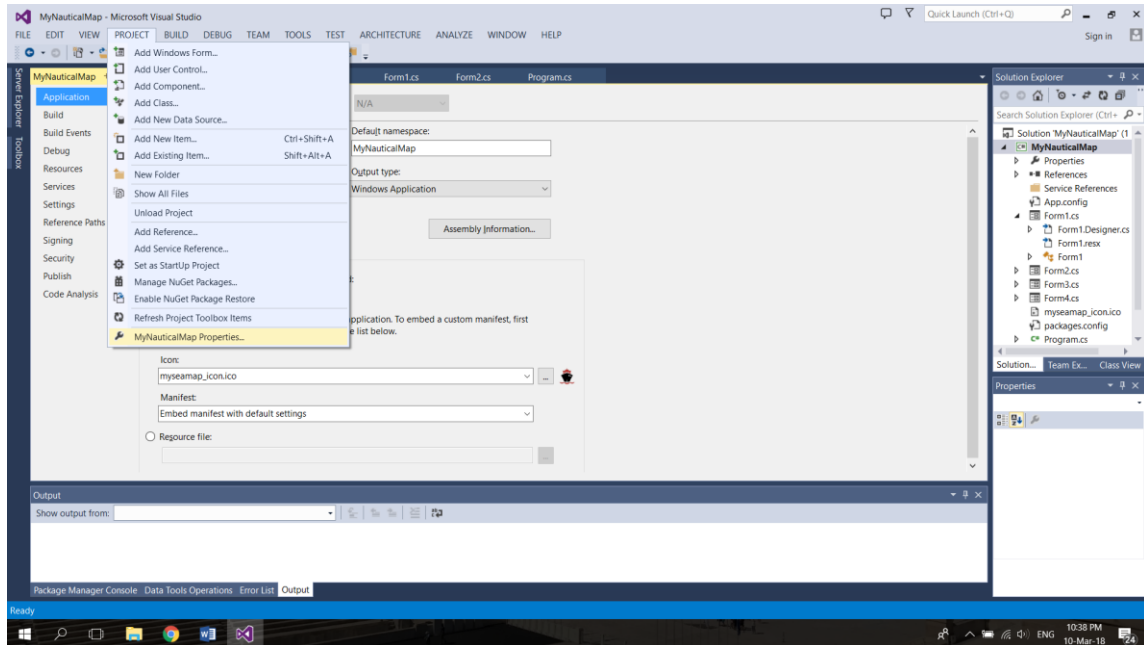
            private void Form4_Load(object sender, EventArgs e)
            {
                SqlConnection SQLconn = new SqlConnection(@"Data
                Source=(Localdb)\v11.0;AttachDbFilename=C:\Users\Admin\Documents\DbfToSQL.mdf;Integra
                ted Security=True;Connect Timeout=30");
                SQLconn.Open();
                String query = "SELECT * FROM AGGK WHERE (((UERM_PER LIKE '%' +
                Form1.pass1 + "') OR (UERM_UAL LIKE '%' + Form1.pass1 + "')OR (THLKALYCH LIKE
                '%" + Form1.pass1 + "')OR (NAYT_KIN LIKE '%" + Form1.pass1 + "') OR (UESH LIKE '%"
                + Form1.pass1 + "') OR (INFO LIKE '%" + Form1.pass1 + "')OR (INFO LIKE '%" +
                Form1.pass1 + "') OR (DENSITY LIKE '%" + Form1.pass1 + "') OR (TAX_HXOY LIKE '%" +
                Form1.pass1 + "')AND(DATE >= '" + Form1.pass2 + "' AND DATE <= '" + Form1.pass3 + "' )
                ";
                SqlDataAdapter sda = new SqlDataAdapter(query, SQLconn);
                DataTable dt = new DataTable();
                sda.Fill(dt);
                BindingSource bsource = new BindingSource();
                bsource.DataSource = dt;
                dataGridView2.DataSource = bsource;
                sda.Update(dt);
            }

            private void button1_Click(object sender, EventArgs e)
            {
                this.Close();
            }
        }
    }
}

```

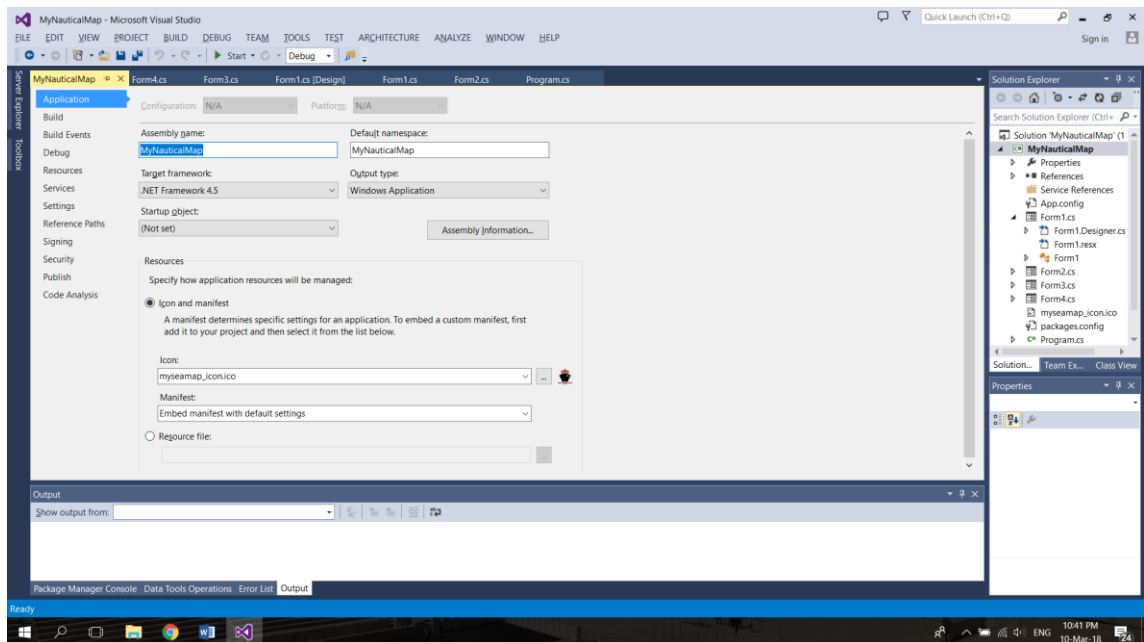
Τέλος, δεν παραλείπουμε το βήμα της εκτέλεσης της εφαρμογής σε 64 Bit και της χρήσης του .Net Framework 4.5. Αυτό μπορεί να εκτελεστεί ως εξής:

Πατάμε αριστερό κλικ στη καρτέλα PROJECT και στη συνέχεια πατάμε το MyNauticalMap Properties....



Εικόνα 4.36 – Επιλογή ιδιοτήτων του Project

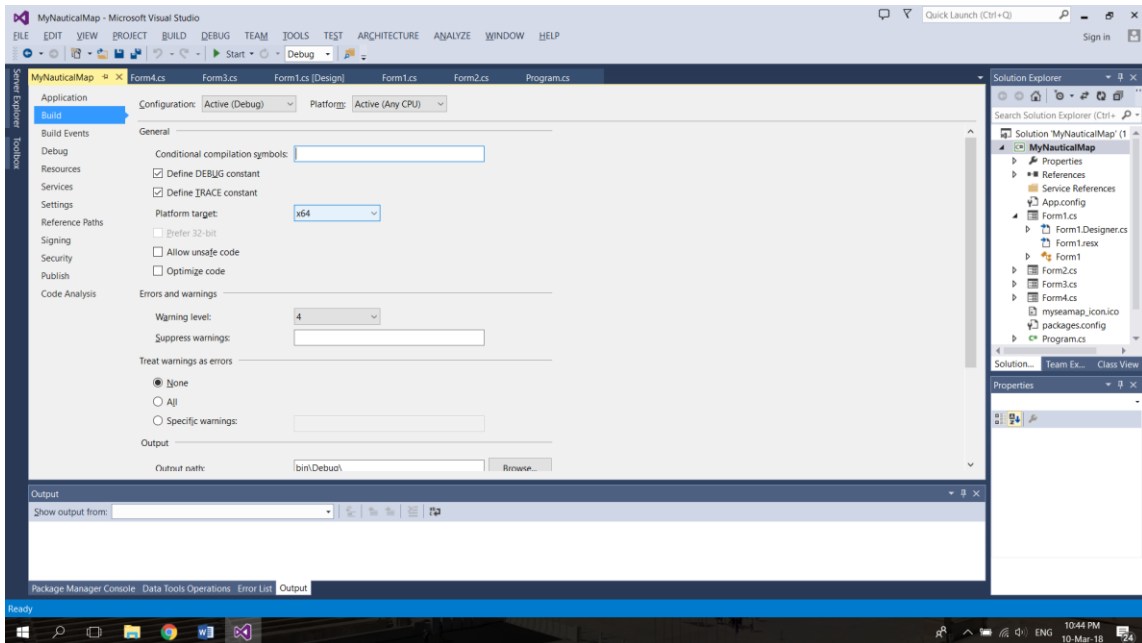
Στη συνέχεια τροποποιούμε από το application το target framework σε .NET Framework 4.5.



Εικόνα 4.37 – Επιλογή .Net Framework 4.5 ως target framework.

Από το Build επιλέγουμε στο Platform target: X64 όπως φαίνεται στην εικόνα παρακάτω.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



Εικόνα 4.38 – Επιλογή x64 ως Platform target.

4.2 Παρουσίαση του συνολικού κώδικα της εφαρμογής

Σε αυτό το κεφάλαιο θα πραγματοποιηθεί παρουσίαση του συνολικού κώδικα που χρησιμοποιήθηκε για την κατασκευή της εφαρμογής. Με αυτόν τον τρόπο θα μπορεί ο καθένας να κατασκευάσει την συγκεκριμένη εφαρμογή, να την τροποποιήσει ή και να της προσθέσει επιπλέον δυνατότητες.

4.2.1 Κώδικας φόρμας form1

Παρακάτω αναγράφεται ο κώδικας του form1 :

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.ComponentModel.Composition;
using System.Data.SqlClient;
using System.Windows;
using System.IO;
using MyNauticalMap;
using System.Data.OleDb;
using DotSpatial.Controls;
using DotSpatial.Data;
using DotSpatial.Symbology;
```

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.


```

using DotSpatial.Topology;
using DotSpatial.Projections;
using SQL = System.Data;

namespace WindowsFormsApplication3
{
    public partial class Form1 : Form
    {
        public static string pass;
        public static string pass1;
        public static DateTime pass2;
        public static DateTime pass3;
        /// <param name="objDT">attribute table as a datatable input</param>
        [Export("Shell", typeof(ContainerControl))]
        private static ContainerControl Shell;
        public Form1()
        {
            InitializeComponent();
            if (DesignMode) return;
            Shell = this;
            appManager1.LoadExtensions();
            appManager1.CompositionContainer.
                ComposeParts(toolManager1);
        }
        SqlConnection conection = new SqlConnection(@"Data
Source=(Localdb)\v11.0;AttachDbFilename=C:\Users\Admin\Documents\h_nv.mdf;Integra
ted Security=True;Connect Timeout=30");
        string imgLocation = "";
        SqlCommand cmd;
        private void Form1_Load(object sender, EventArgs e)
        {
            Tooltip TP = new Tooltip();
            TP.ShowAlways = true;
            TP.ToolTipTitle = "Τρόπος εισαγωγής...";
            TP.UseFading = true;
            TP.UseAnimation = true;
            TP.IsBalloon = true;
            TP.ShowAlways = true;
            TP.AutoPopDelay = 0;
            TP.InitialDelay = 0;
            TP.ReshowDelay = 0;
            TP.SetToolTip(textBox3, "Παρακαλώ εισάγετε ημερομηνία (π.χ 01-Jan-
18).");
            TP.SetToolTip(textBox4, "Παρακαλώ εισάγετε ημερομηνία (π.χ 01-Jan-
18).");
        }

        private void legend1_Click(object sender, EventArgs e)
        {

        }

        private void label1_Click(object sender, EventArgs e)
        {

```

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

```

    }

    private void label2_Click(object sender, EventArgs e)
    {
    }

    private void button3_Click(object sender, EventArgs e)
    {
        conection.Open();
        string sqlQuery = "select Name,Image from CIW where id='" +
textID.Text + "'";
        cmd = new SqlCommand(sqlQuery, conection);
        SqlDataReader Dataread = cmd.ExecuteReader();
        Dataread.Read();
        if (Dataread.HasRows)
        {
            textName.Text = Dataread[0].ToString();
            byte[] images = (byte[])Dataread[1];
            if (images == null)
            {
                pictureBox1.Image = null;
            }
            else
            {
                MemoryStream mstream = new MemoryStream(images);
                pictureBox1.Image = Image.FromStream(mstream);
            }
        }
        else
        {
            MessageBox.Show("Δεν υπάρχει εικόνα για αυτό το ID");
        }
        conection.Close();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        OpenFileDialog dialog = new OpenFileDialog();
        dialog.Filter = "png files(*.png)|*.png|jpg files(*.jpg)|*.jpg|All
files(*.*)|*.*";
        if (dialog.ShowDialog() == DialogResult.OK)
        {
            imgLocation = dialog.FileName.ToString();
            pictureBox1.ImageLocation = imgLocation;
        }
    }

    private void pictureBox1_Click(object sender, EventArgs e)
    {
    }

    private void buttonSave_Click(object sender, EventArgs e)

```

Σχεδίαση και ανάπτυξη μίας
φιλικής εφαρμογής γεωγραφικών δεδομένων
με τη βοήθεια του Microsoft Visual Studio
και των βιβλιοθηκών DotSpatial.

```

    {
        conection.Open();
        byte[] images = null;
        if (textID.Text == "" || textName.Text == "")
        {
            MessageBox.Show("Κάποιο ή κάποια από τα πεδία είναι κενά!");
            conection.Close();
        }
        else if(imgLocation == "")
        {
            MessageBox.Show("Εισάγετε νέα φωτογραφία και όχι αυτή που υπάρχει  

            ήδη στην εικόνα!");
            conection.Close();
        }
        else
        {
            FileStream Stream = new FileStream(imgLocation, FileMode.Open,
            FileAccess.Read);
            BinaryReader brs = new BinaryReader(Stream);
            images = brs.ReadBytes((int)Stream.Length);
            SqlCommand cmdCount = new SqlCommand("SELECT count(*) from CIW
            WHERE Id = '" + textID.Text + "'", conection);
            int count = (int)cmdCount.ExecuteScalar();
            if (count > 0)
            {
                string sqlQuery = "UPDATE CIW SET Name=@NAME , Image=@images
            WHERE Id = '" + textID.Text + "'";
                cmd = new SqlCommand(sqlQuery, conection);
                cmd.Parameters.Add(new SqlParameter("@images", images));
                cmd.Parameters.Add(new SqlParameter("@NAME", textName.Text));
                cmd.ExecuteNonQuery();
                conection.Close();
                MessageBox.Show("Η εικόνα αποθηκεύτηκε επιτυχώς στη θέση της  

                προηγούμενης για το συγκεκριμένο ID!");
                imgLocation = "";
            }
            else
            {
                string sqlQuery = "INSERT INTO CIW (Id,Name,Image) VALUES ('"
            + textID.Text + "', '" + textName.Text + "', @images)";
                cmd = new SqlCommand(sqlQuery, conection);
                cmd.Parameters.Add(new SqlParameter("@images", images));
                int N = cmd.ExecuteNonQuery();
                conection.Close();
                MessageBox.Show("Αποθηκεύτηκε επιτυχώς!");
                imgLocation = "";
            }
        }
    }

    private void legend1_Click_1(object sender, EventArgs e)
    {
    }
}

```

Σχεδίαση και ανάπτυξη μίας
 φιλικής εφαρμογής γεωγραφικών δεδομένων
 με τη βοήθεια του Microsoft Visual Studio
 και των βιβλιοθηκών DotSpatial.

```

private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    Application.Exit();
}

private void label3_Click(object sender, EventArgs e)
{
}

private void label4_Click(object sender, EventArgs e)
{
}

private void button2_Click(object sender, EventArgs e)
{
    System.Data.DataTable dt = null;
    if (map1.Layers.Count > 0)
    {
        MapPointLayer stateLayer = default(MapPointLayer);
        stateLayer = map1.Layers[0] as MapPointLayer;
        if (stateLayer == null)
        {
            MessageBox.Show(" Αυτό το layer δεν είναι τύπου σημείων ή
εργάζεστε με πολλαπλά layers!");
        }
        else
        {
            //Απουκεία του shapefile's attribute table στον πίνακα dt
            dt = stateLayer.DataSet.DataTable;
            ExportToExcel(dt);
        }
    }
    else
    {
        MessageBox.Show("Παρακαλώ προσθέστε layer στον χάρτη.");
    }
}

private void ExportToExcel(System.Data.DataTable objDT)
{
    Microsoft.Office.Interop.Excel.Application xlApp = new
Microsoft.Office.Interop.Excel.Application();
    string strFilename = null;
    int intCol = 0;
    int intRow = 0;
    string strPath = "C:\\Users\\Admin\\Desktop\\";
    if (xlApp == null)
    {
        MessageBox.Show("It appears that Excel is not installed on this
machine. This operation requires MS Excel to be installed on this machine.", "",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}

```

Σχεδίαση και ανάπτυξη μίας
φιλικής εφαρμογής γεωγραφικών δεδομένων
με τη βοήθεια του Microsoft Visual Studio
και των βιβλιοθηκών DotSpatial.

```

        return;
    }
    try
    {
        xlApp.SheetsInNewWorkbook = 1;
        xlApp.Workbooks.Add();
        xlApp.Worksheets[1].Select();
        xlApp.Cells[1, 1].EntireRow.Font.Bold = true;
        int intI = 1;
        for (intCol = 0; intCol <= objDT.Columns.Count - 1; intCol++)
        {
            xlApp.Cells[1, intI].value =
objDT.Columns[intCol].ColumnName;
            xlApp.Cells[1, intI].EntireRow.Font.Bold = true;
            intI += 1;
        }
        intI = 3;
        int intK = 1;
        for (intCol = 0; intCol <= objDT.Columns.Count - 1; intCol++)
        {
            intI = 2;
            for (intRow = 0; intRow <= objDT.Rows.Count - 1; intRow++)
            {
                xlApp.Cells[intI, intK].Value =
objDT.Rows[intRow].ItemArray[intCol];
                intI += 1;
            }
            intK += 1;
        }

        if (strPath.Substring(strPath.Length - 1, 1) != "\\")
        {
            strPath = strPath + "\\";
        }
        strFilename = strPath + "AttributeTable.xls";
        xlApp.ActiveCell.Worksheet.SaveAs(strFilename);
        System.Runtime.InteropServices.Marshal.ReleaseComObject(xlApp);
        xlApp = null;
        MessageBox.Show("Εκτελέστηκε εξαγωγή δεδομένων στο αρχείο Excel:
'" + strFilename + "'", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    System.Diagnostics.Process[] pro =
(System.Diagnostics.Process[])System.Diagnostics.Process.GetProcessesByName("EXCEL");
    foreach (System.Diagnostics.Process i in pro)
    {
        i.Kill();
    }
}

```

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

```

private void button1_Click_1(object sender, EventArgs e)
{
    textID.Text = null;
    pictureBox1.Image = null;
    textName.Text = null;
    imgLocation = "";
    MessageBox.Show("Εκτελέστηκε με επιτυχία!");
}

private void textID_TextChanged(object sender, EventArgs e)
{
}

private void textName_TextChanged(object sender, EventArgs e)
{
}

private void button3_Click_1(object sender, EventArgs e)
{
    OpenExcelFile();
}

public static FeatureSet OpenExcelFile()
{
    OpenFileDialog openFileDialog = new OpenFileDialog();
    openFileDialog.Filter = "Excel Files|*.xls";
    if (openFileDialog.ShowDialog() == DialogResult.OK)
    {
        DataTable excelTable =
ConvertExcelFileToDataTable(openFileDialog.FileName);
    }
    return null;
}

private static DataTable ConvertExcelFileToDataTable(string
excelFileName)
{
    string connectionString =
        String.Format("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source={0};Extended Properties=\\\"Excel 12.0 Xml;HDR=YES;IMEX=1\\\"",
excelFileName);

    using (OleDbConnection connection = new
OleDbConnection(connectionString))
    {
        string query = "SELECT * FROM [Sheet1$]";
        connection.Open();
        OleDbCommand objCmdSelect = new OleDbCommand(query, connection);
        OleDbDataReader objDR = null;
        OleDbDataAdapter adapter = new OleDbDataAdapter(objCmdSelect);
        DataTable excelTable = new DataTable();

```

Σχεδίαση και ανάπτυξη μίας
φιλικής εφαρμογής γεωγραφικών δεδομένων
με τη βοήθεια του Microsoft Visual Studio
και των βιβλιοθηκών DotSpatial.

```

adapter.Fill(excelTable);
SqlConnection SQLconn = new SqlConnection(@"Data
Source=(Localdb)\v11.0;AttachDbFilename=C:\Users\Admin\Documents\DbfToSQL.mdf;Int
egrated Security=True;Connect Timeout=30");
SQLconn.Open();
string sqlTrunc = "TRUNCATE TABLE AGGK";
SqlCommand cmdlast = new SqlCommand(sqlTrunc, SQLconn);
cmdlast.ExecuteNonQuery();
using (SqlBulkCopy bulkCopy = new SqlBulkCopy(SQLconn))
{
    bulkCopy.DestinationTableName = "AGGK";
    // Set up the column mappings by name.
    SqlBulkCopyColumnMapping mapUESH =
        new SqlBulkCopyColumnMapping("UESH", "UESH");
    bulkCopy.ColumnMappings.Add(mapUESH);

    SqlBulkCopyColumnMapping mapUERMOKRASIA =
        new SqlBulkCopyColumnMapping("UERM_PER", "UERM_PER");
    bulkCopy.ColumnMappings.Add(mapUERMOKRASIA);

    SqlBulkCopyColumnMapping mapUERMOKRASIA_U =
        new SqlBulkCopyColumnMapping("UERM_UAL", "UERM_UAL");
    bulkCopy.ColumnMappings.Add(mapUERMOKRASIA_U);

    SqlBulkCopyColumnMapping mapPYKNOTHTA_U =
        new SqlBulkCopyColumnMapping("DENSITY", "DENSITY");
    bulkCopy.ColumnMappings.Add(mapPYKNOTHTA_U);

    SqlBulkCopyColumnMapping mapPLHROFORIES =
        new SqlBulkCopyColumnMapping("INFO", "INFO");
    bulkCopy.ColumnMappings.Add(mapPLHROFORIES);

    SqlBulkCopyColumnMapping mapTAXYTHTA_HX =
        new SqlBulkCopyColumnMapping("TAX_HXOY", "TAX_HXOY");
    bulkCopy.ColumnMappings.Add(mapTAXYTHTA_HX);

    SqlBulkCopyColumnMapping mapHMEROMHNIA =
        new SqlBulkCopyColumnMapping("DATE", "DATE");
    bulkCopy.ColumnMappings.Add(mapHMEROMHNIA);

    SqlBulkCopyColumnMapping mapTHL_KALYCH =
        new SqlBulkCopyColumnMapping("THLKALYCH", "THLKALYCH");
    bulkCopy.ColumnMappings.Add(mapTHL_KALYCH);

    SqlBulkCopyColumnMapping mapREYMATA =
        new SqlBulkCopyColumnMapping("REYMATA", "REYMATA");
    bulkCopy.ColumnMappings.Add(mapREYMATA);

    SqlBulkCopyColumnMapping mapNAYT_KINSH =
        new SqlBulkCopyColumnMapping("NAYT_KIN", "NAYT_KIN");
    bulkCopy.ColumnMappings.Add(mapNAYT_KINSH);
    try
    {
        objDR = objCmdSelect.ExecuteReader();
    }
}

```

Σχεδίαση και ανάπτυξη μίας
φιλικής εφαρμογής γεωγραφικών δεδομένων
με τη βοήθεια του Microsoft Visual Studio
και των βιβλιοθηκών DotSpatial.

```

        bulkCopy.WriteToServer(objDR);
        connection.Close();
        SQLconn.Close();
        MessageBox.Show("Εκτελέστηκε με επιτυχία!");
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
    }
    return excelTable;
}
}

private void map1_Load(object sender, EventArgs e)
{
}

private void label3_Click_1(object sender, EventArgs e)
{
}

private void textBox1_TextChanged(object sender, EventArgs e)
{
}

private void label5_Click(object sender, EventArgs e)
{
}

//anazhthsh
private void button4_Click(object sender, EventArgs e)
{
    pass = textBox1.Text;
    Form3 = new Form3();
    form3.Show();
}

private void textBox2_TextChanged(object sender, EventArgs e)
{
    if ((textBox3.Text != "") && (textBox4.Text != ""))
    {
        textBox2.ReadOnly = false;
    }
}

private void button6_Click(object sender, EventArgs e)
{
    try

```



```

    {
        pass1 = textBox2.Text;
        DateTime dt = Convert.ToDateTime(textBox3.Text);
        string s2 = dt.ToString("dd-MM-yyyy");
        pass2 = Convert.ToDateTime(s2);
        DateTime dt1 = Convert.ToDateTime(textBox4.Text);
        string s3 = dt1.ToString("dd-MM-yyyy");
        pass3 = Convert.ToDateTime(s3);
        Form4 = new Form4();
        form4.Show();
    }
    catch (Exception ex)
    {
        if (!(
            ex is FormatException ||
            ex is OverflowException))
        {
            throw;
        }
        Console.WriteLine("Fill both textboxes with appropriate dates");
    }
}

private void textBox3_TextChanged(object sender, EventArgs e)
{
    if ((textBox3.Text != "") && (textBox4.Text != ""))
    {
        textBox2.ReadOnly = false;
        button6.Enabled = true;
    }
    if ((textBox3.Text == "") || (textBox4.Text == ""))
    {
        textBox2.ReadOnly = true;
        textBox2.Text = "";
        button6.Enabled = false;
    }
}

private void textBox4_TextChanged(object sender, EventArgs e)
{
    if ((textBox3.Text != "") && (textBox4.Text != ""))
    {
        textBox2.ReadOnly = false;
        button6.Enabled = true;
    }
    if ((textBox3.Text == "") || (textBox4.Text == ""))
    {
        textBox2.ReadOnly = true;
        textBox2.Text = "";
        button6.Enabled = false;
    }
}
}
}

```

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

```
}

```

4.2.2 Κώδικας φόρμας form2

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using WindowsFormsApplication3;
using System.Data.SqlClient;

namespace MyNauticalMap
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (textBox1.Text == "" || textBox2.Text == "")
            {
                MessageBox.Show("Παρακαλώ εισάγετε τα στοιχεία σας και στα δύο πεδία");
                textBox1.Text = "";
                textBox2.Text = "";
            }
            else
            {
                SqlConnection con = new SqlConnection(@"Data
Source=(LocalDB)\v11.0;AttachDbFilename=C:\Users\Admin\Documents\person1.mdf;Int
egrated Security=True;Connect Timeout=30;");
                SqlDataAdapter sda = new SqlDataAdapter("Select Count(*) From
login where username='" + textBox1.Text + "' and password='" + textBox2.Text +
"", con);
                DataTable dt = new DataTable();
                sda.Fill(dt);
                if (dt.Rows[0][0].ToString() == "1")
                {
                    this.Hide();
                    Form1 = new Form1();
                    form1.Show();
                }
                else
                {

```

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

```

        MessageBox.Show("Λάθος Στοιχεία! Παρακαλώ ξαναεισάγετε τα
στοιχεία σας");
        textBox1.Text = "";
        textBox2.Text = "";
    }
}

private void label1_Click(object sender, EventArgs e)
{
}

private void label3_Click(object sender, EventArgs e)
{
}

private void Form2_Load(object sender, EventArgs e)
{
}
}
}

```

4.2.3 Κώδικας φόρμας form3

Ο κώδικας του form3

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.ComponentModel.Composition;
using System.Data.SqlClient;
using System.Windows;
using System.IO;
using MyNauticalMap;
using System.Data.OleDb;
using DotSpatial.Controls;
using DotSpatial.Data;
using DotSpatial.Symbology;
using DotSpatial.Topology;
//using System.Data.OleDb;
using DotSpatial.Projections;
//using DotSpatial.Topology;
using SQL = System.Data;
using WindowsFormsApplication3;

```

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

```

namespace MyNauticalMap
{
    public partial class Form3 : Form
    {
        public Form3()
        {
            InitializeComponent();
        }

        private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {
        }

        private void Form3_Load(object sender, EventArgs e)
        {
            SqlConnection SQLconn = new SqlConnection(@"Data
Source=(Localdb)\v11.0;AttachDbFilename=C:\Users\Admin\Documents\DbfToSQL.mdf;Int
egrated Security=True;Connect Timeout=30");
            SQLconn.Open();
            String query = "SELECT * FROM AGGK WHERE ((UERM_PER LIKE '%" +
Form1.pass + "%') OR (THLKALYCH LIKE '%" + Form1.pass + "%')OR (NAYT_KIN LIKE '%"
+ Form1.pass + "%') OR (UERM_UAL LIKE '%" + Form1.pass + "%') OR (UESH LIKE '%" +
Form1.pass + "%') OR (INFO LIKE '%" + Form1.pass + "%') OR (DENSITY LIKE '%" +
Form1.pass + "%')OR (REYMATA LIKE '%" + Form1.pass + "%') OR (TAX_HXOY LIKE '%" +
Form1.pass + "%')) ";
            SqlDataAdapter sda = new SqlDataAdapter(query, SQLconn);
            DataTable dt = new DataTable();
            sda.Fill(dt);
            BindingSource bsource = new BindingSource();
            bsource.DataSource = dt;
            dataGridView1.DataSource = bsource;
            sda.Update(dt);
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}

```

4.2.4 Κώδικας φόρμας form4

Ο κώδικας του form4

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;

```

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

```

using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.ComponentModel.Composition;
using System.Data.SqlClient;
using System.Windows;
using System.IO;
using MyNauticalMap;
using System.Data.OleDb;
using DotSpatial.Controls;
using DotSpatial.Data;
using DotSpatial.Symbology;
using DotSpatial.Topology;
//using System.Data.OleDb;
using DotSpatial.Projections;
//using DotSpatial.Topology;
using SQL = System.Data;
using WindowsFormsApplication3;

namespace MyNauticalMap
{
    public partial class Form4 : Form
    {
        public Form4()
        {
            InitializeComponent();
        }

        private void Form4_Load(object sender, EventArgs e)
        {
            SqlConnection SQLconn = new SqlConnection(@"Data
Source=(Localdb)\v11.0;AttachDbFilename=C:\Users\Admin\Documents\DbfToSQL.mdf;Int
egrated Security=True;Connect Timeout=30");
            SQLconn.Open();
            String query = "SELECT * FROM AGGK WHERE (((UERM_PER LIKE '%" +
Form1.pass1 + "%') OR (UERM_UAL LIKE '%" + Form1.pass1 + "%')OR (THLKALYCH LIKE
'" + Form1.pass1 + "%')OR (NAYT_KIN LIKE '%" + Form1.pass1 + "%') OR (UESH LIKE
'" + Form1.pass1 + "%') OR (INFO LIKE '%" + Form1.pass1 + "%')OR (INFO LIKE '%"
+ Form1.pass1 + "%') OR (DENSITY LIKE '%" + Form1.pass1 + "%') OR (TAX_HXOY LIKE
'" + Form1.pass1 + "%'))AND(DATE >= '" + Form1.pass2 + "' AND DATE <= '" +
Form1.pass3 + "') ) ";
            SqlDataAdapter sda = new SqlDataAdapter(query, SQLconn);

            DataTable dt = new DataTable();
            sda.Fill(dt);
            BindingSource bsource = new BindingSource();
            bsource.DataSource = dt;
            dataGridView2.DataSource = bsource;
            sda.Update(dt);
        }

        private void button1_Click(object sender, EventArgs e)

```

Σχεδίαση και ανάπτυξη μίας
φιλικής εφαρμογής γεωγραφικών δεδομένων
με τη βοήθεια του Microsoft Visual Studio
και των βιβλιοθηκών DotSpatial.

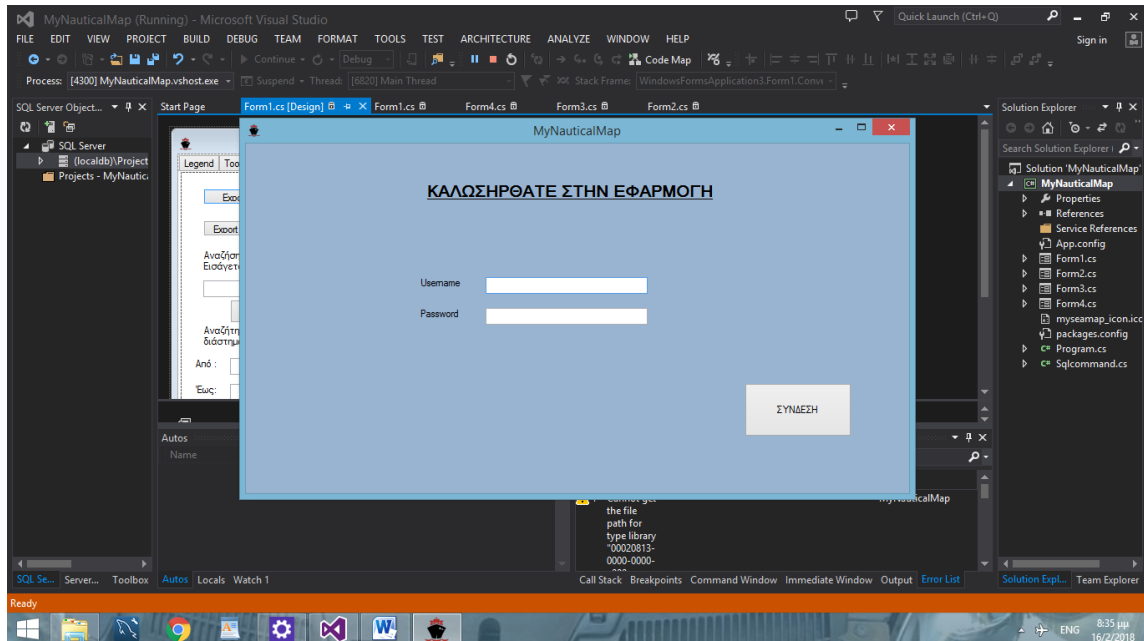
```
    {  
        this.Close();  
    }  
}
```

Περιγραφή αναπτυχθείσας εφαρμογής

Ο χρήστης μετά τη σύνδεση του στην εφαρμογή με τη χρήση μοναδικού username και password θα αποκτά πρόσβαση σε ένα παράθυρο που στα αριστερά θα έχει μία καρτέλα για τα map layers, μία για τα εργαλεία, μία για πρόσβαση στη βάση δεδομένων εικόνων και μία τελευταία καρτέλα για την εξαγωγή των δεδομένων του πίνακα χαρακτηριστικών του shapefile σε Excel και για την τυχόν εισαγωγή τους σε μία βάση δεδομένων. Δεξιά θα εκτελείται η εμφάνιση του χάρτη. Στο πάνω μέρος της εφαρμογής θα υπάρχει μία λωρίδα μενού (menu strip) με αρκετές λειτουργίες. Από αυτή λοιπόν τη λωρίδα μενού, θα εκτελείται η δημιουργία νέου επιπέδου (layer) που έχει τη δυνατότητα να είναι τύπου bmp, emf, exf, gif, ico, jpg, mbr, png, tif, wmf, bgd, shp, η διαγραφή επιπέδου (layer), ο χειρισμός και απεικόνιση ενός πίνακα ιδιοτήτων (attribute table), η μεγέθυνση – σμίκρυνση του χάρτη, μεγέθυνση σε πλήρες περιεχόμενο χάρτη, η αναγνώριση αντικειμένων πάνω στο χάρτη και εμφάνιση των στοιχείων τους, η μετακίνηση πάνω στον χάρτη, η μεγέθυνση σε συγκεκριμένες συντεταγμένες, η τοποθέτηση παραπάνω τους ενός επιπέδου (layer), εισαγωγή - διαγραφή ναυτιλιακής πληροφορίας που μπορεί να είναι τύπου point, line, polygon, multipoint και μέτρηση απόστασης. Επιπλέον των ανωτέρω δυνατοτήτων που θα προσφέρει η λωρίδα μενού, θα μπορεί να εκτελέσει πλήθος διαφορετικών δυνατοτήτων, όπως το να αποθηκεύει σε μορφή .shp επιλεγμένα ή και όλα δεδομένα που έχουν εισαχθεί από τους χρήστες, να τροποποιεί την απεικόνιση των σημείων στο χάρτη ανάλογα με τις προτιμήσεις του, να εκτυπώνει την απεικόνιση του χάρτη, να εξάγει τα δεδομένα του Attribute Table σε ένα αρχείο Excel καθώς και να μπορεί να εισάγει το συγκεκριμένο αρχείο σε μία βάση δεδομένων η οποία θα χρησιμοποιείται για την εκτέλεση λειτουργιών αναζήτησης.

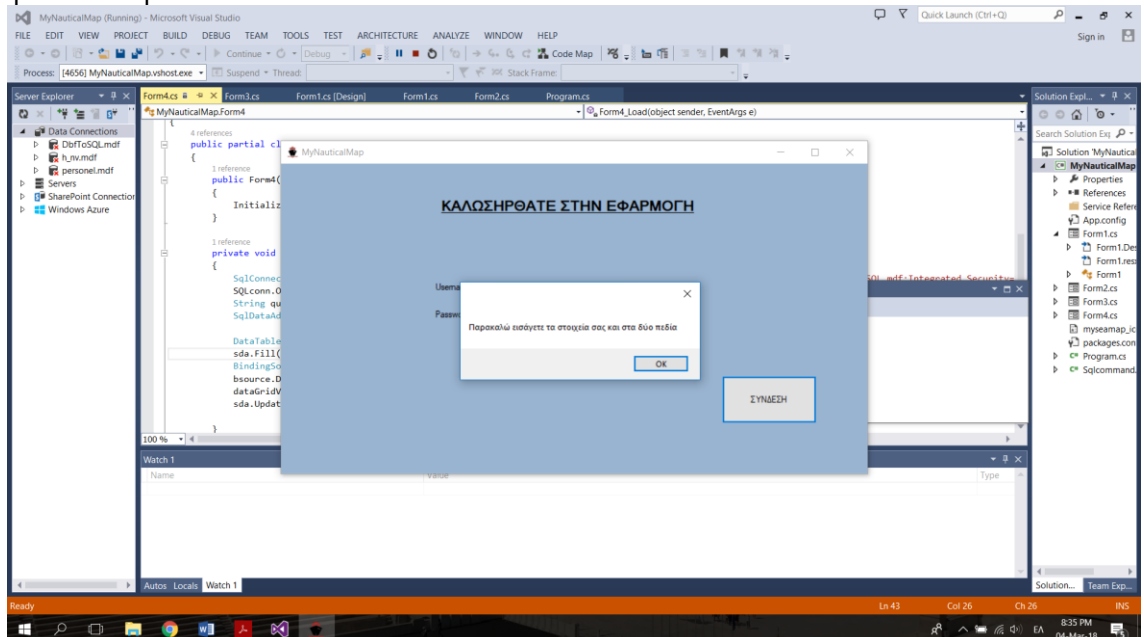
5.1 Οθόνη Σύνδεσης χρήστη

Η οθόνη σύνδεσης του χρήστη (log in screen) που φαίνεται παρακάτω, εμφανίζεται ανοίγοντας την εφαρμογή. Ο χρήστης χρησιμοποιεί το προσωπικό όνομα χρήσης (User name) και το συνθηματικό (Password) που του έχουν δοθεί και επιλέγει τη σύνδεση για επιβεβαίωση των στοιχείων του.



Εικόνα 5.1 – Το αρχικό παράθυρο της εφαρμογής.

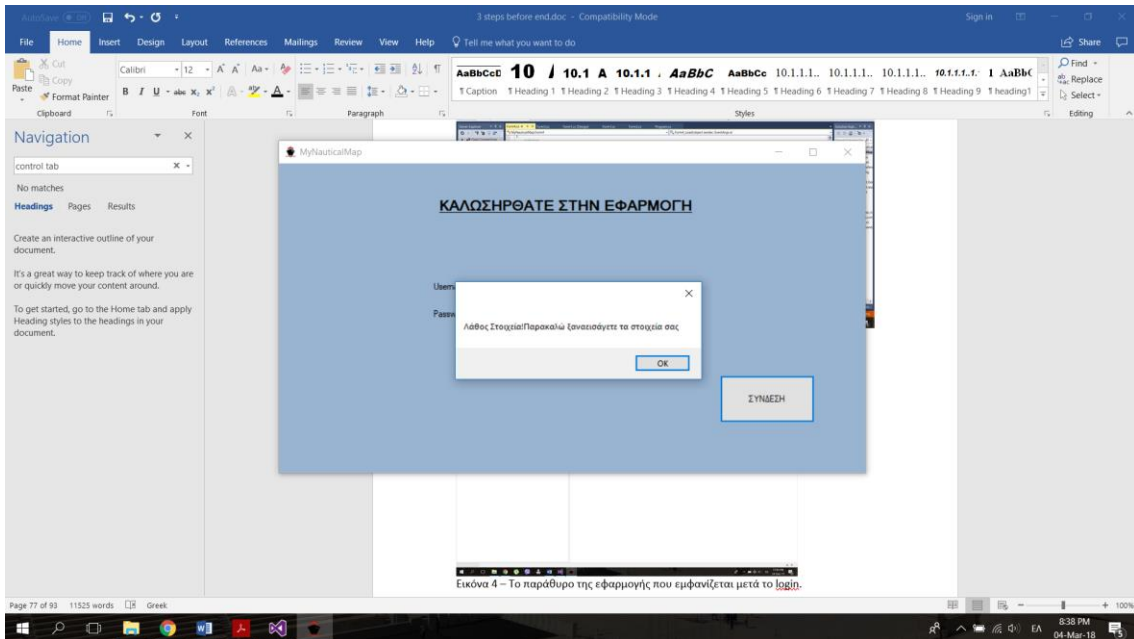
Αν δεν εισαχθούν και τα δύο πεδία στην οθόνη εμφανίζεται ένα αντίστοιχο μήνυμα όπως φαίνεται παρακάτω:



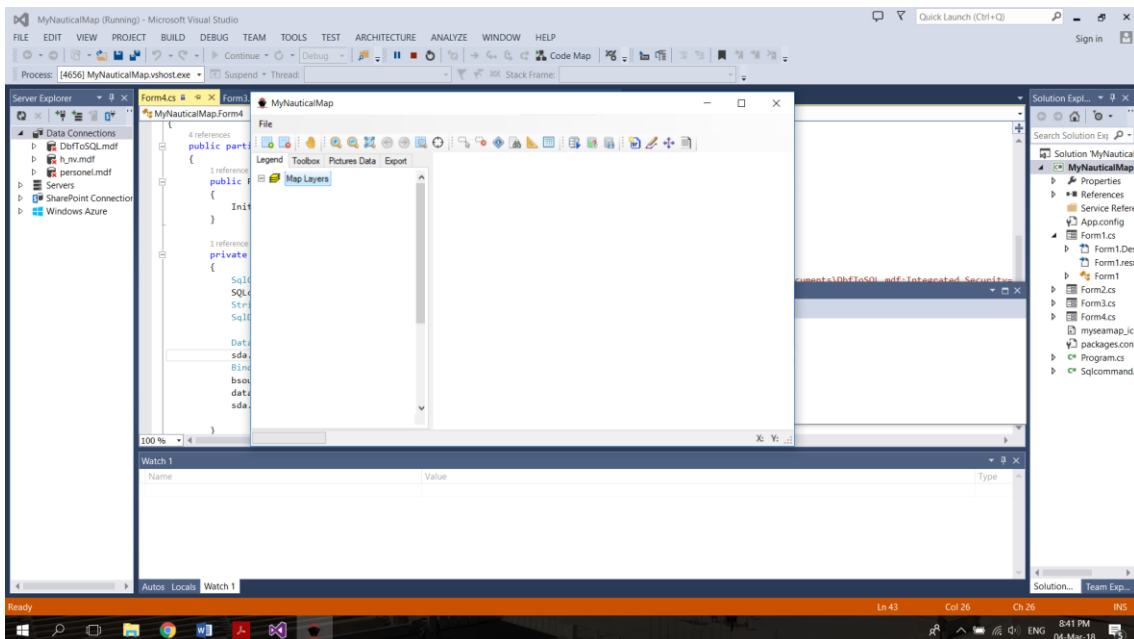
Εικόνα 5.2 – Εμφάνιση μηνύματος αν δεν είναι συμπληρωμένα και τα δύο πεδία .

Σε περίπτωση λάθους στην εισαγωγή των στοιχείων εμφανίζεται αντίστοιχο μήνυμα:

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



Εικόνα 5.3 – Εμφάνιση μηνύματος λάθους εισαγωγής στοιχείων.
Μόλις συνδεθεί ο χρήστης, εμφανίζεται το αρχικό παράθυρο της εφαρμογής;

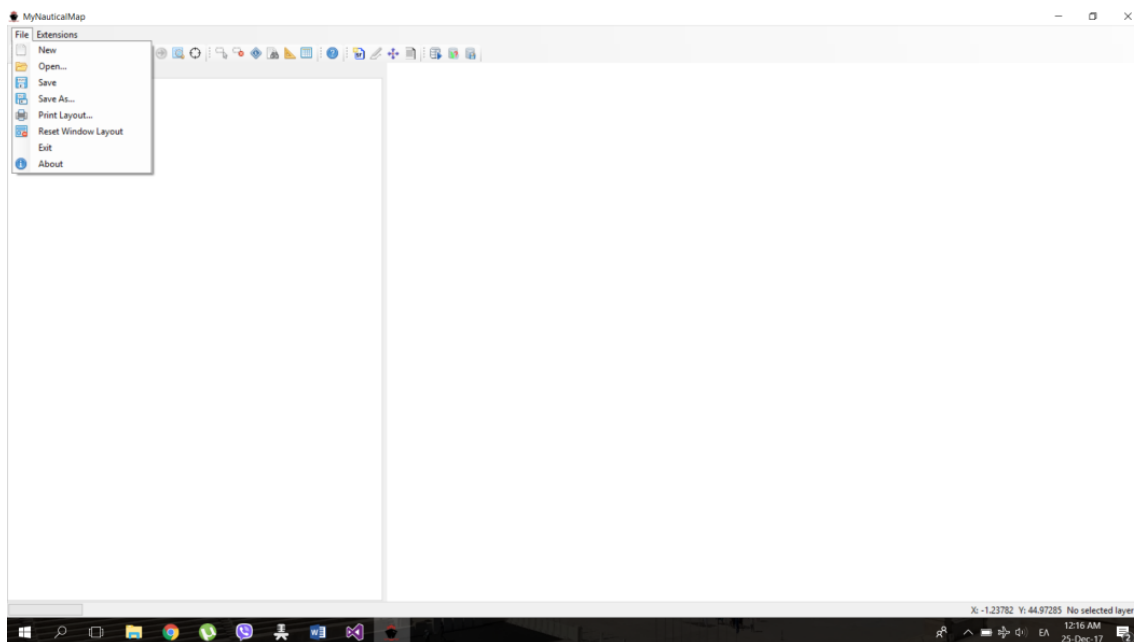


Εικόνα 5.4 – Αρχικό παράθυρο της εφαρμογής.

5.2 Περιγραφή των βασικών λειτουργιών

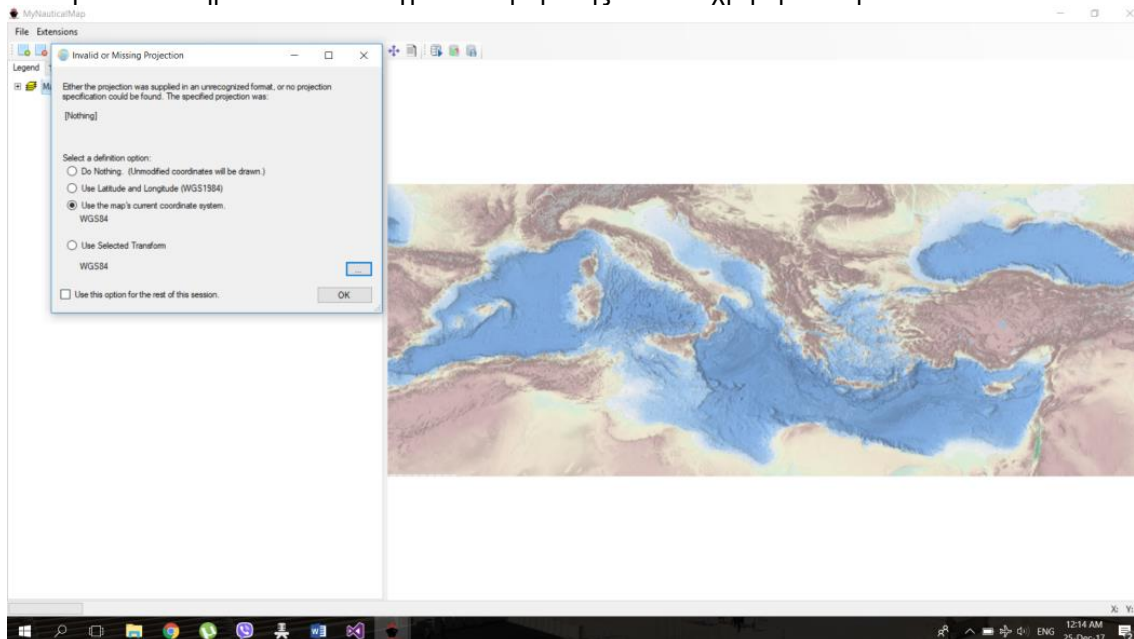
Μπαίνοντας στην επιλογή file ο χρήστης θα μπορεί να σώσει το περιεχόμενό του, να φορτώσει ένα έτοιμο project και να εκτυπώσει την απεικόνιση του χάρτη.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



Εικόνα 5.5 – Άνοιγμα ενός υπάρχοντος project στην εφαρμογή.

Όταν αποφασίσει ο χρήστης ποιον χάρτη θα φορτώσει, θα εμφανιστεί ένα παράθυρο που θα επιλέγει το σύστημα των συντεταγμένων προβολής που θα χρησιμοποιηθεί.



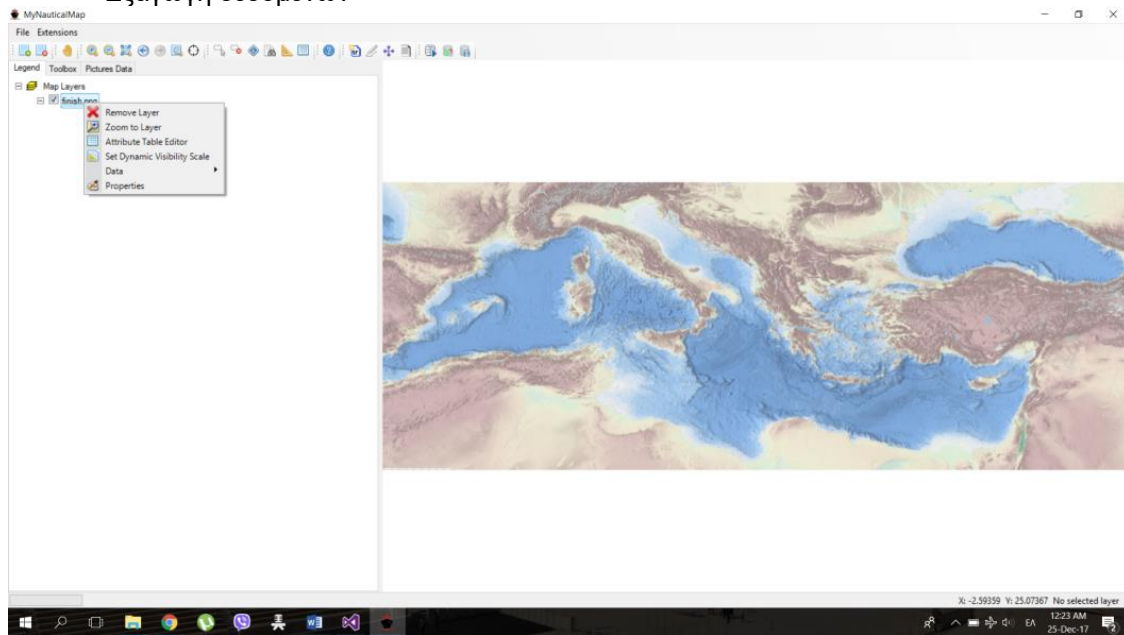
Εικόνα 5.6 – Φόρτωση ενός χάρτη στην εφαρμογή.

Με δεξί κλικ στο επίπεδο θα εμφανίζονται οι δυνατότητες για :

- Αφαίρεση επιπέδου
- Μεγέθυνση στο επίπεδο
- Εμφάνιση πίνακα
- Ιδιότητες

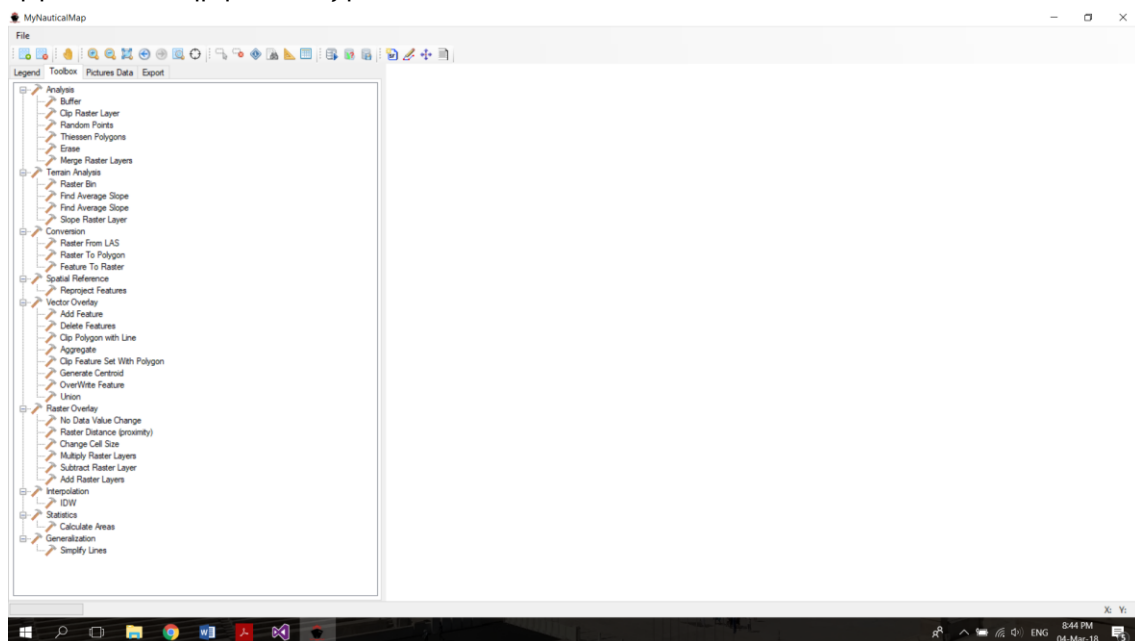
Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

- Δυναμική ορατότητα
- Εξαγωγή δεδομένων



Εικόνα 5.7 – Εμφάνιση ιδιοτήτων ενός επιπέδου (layer).

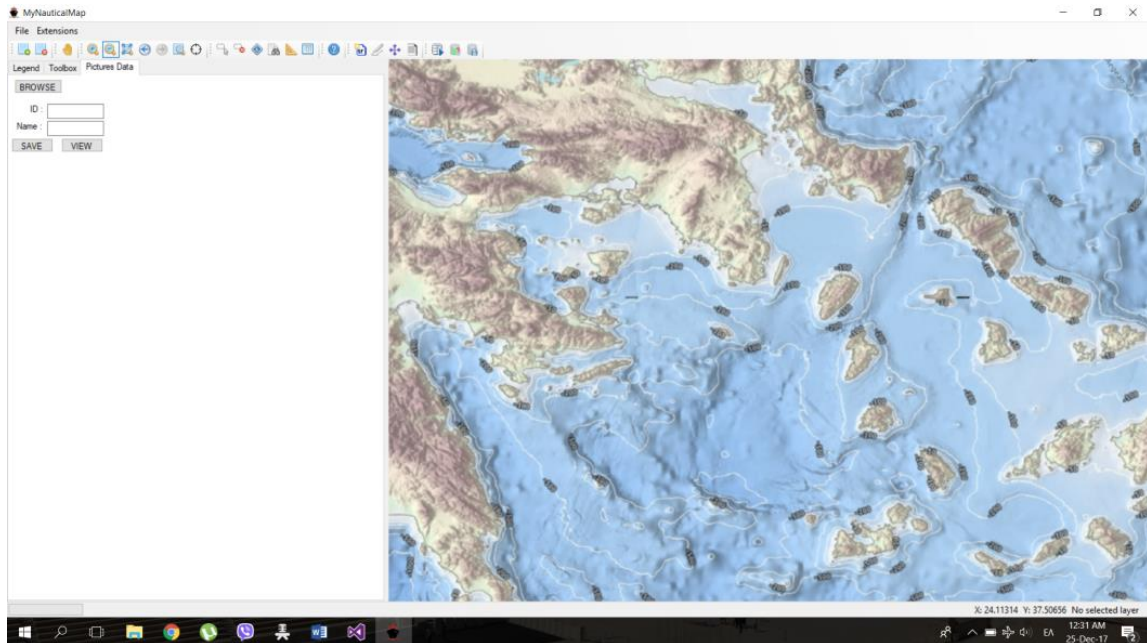
Κάτω δεξιά του χάρτη φαίνονται και οι συντεταγμένες από όπου περνάει ο δείκτης. Επιλέγοντας την καρτέλα toolbox θα εμφανιστούν όλα τα εργαλεία. Στην κάτωθι εικόνα φαίνονται όλα τα εργαλεία σε πλήρη ανάπτυξη.



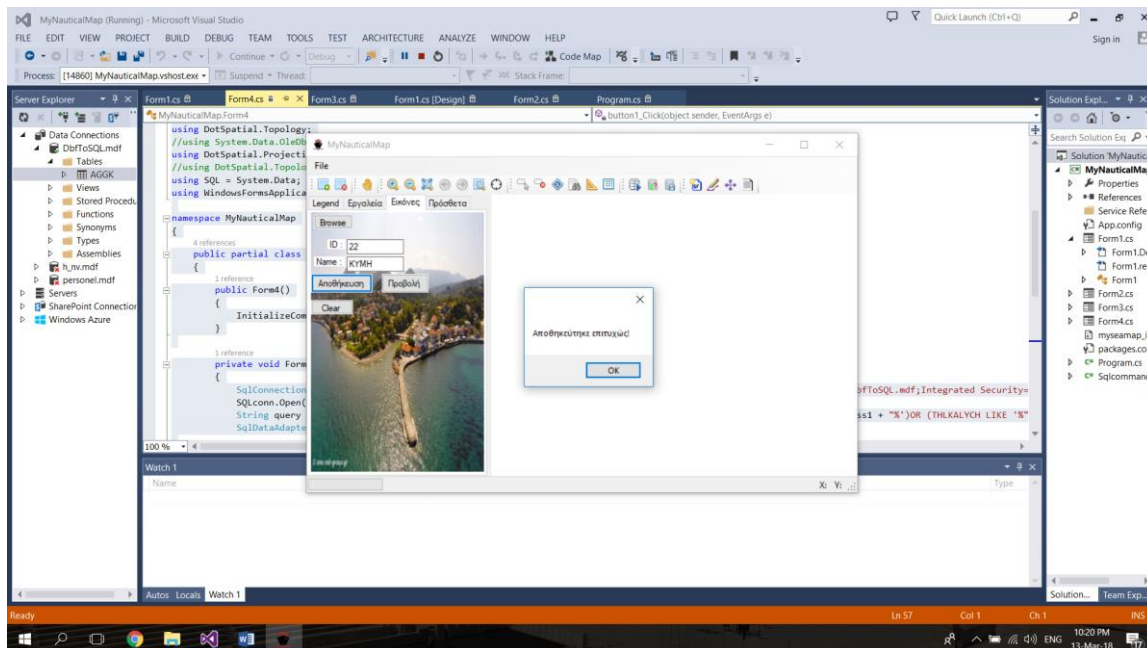
Εικόνα 5.8 – Εργαλειοθήκη της εφαρμογής.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

Στην καρτέλα Pictures Data, ο χρήστης αποκτά πρόσβαση στη βάση δεδομένων των εικόνων για να προβάλει μία αποθηκευμένη εικόνα, για να αποθηκεύει μία νέα εικόνα ή για να αντικαταστήσει μία ήδη υπάρχουσα εικόνα.



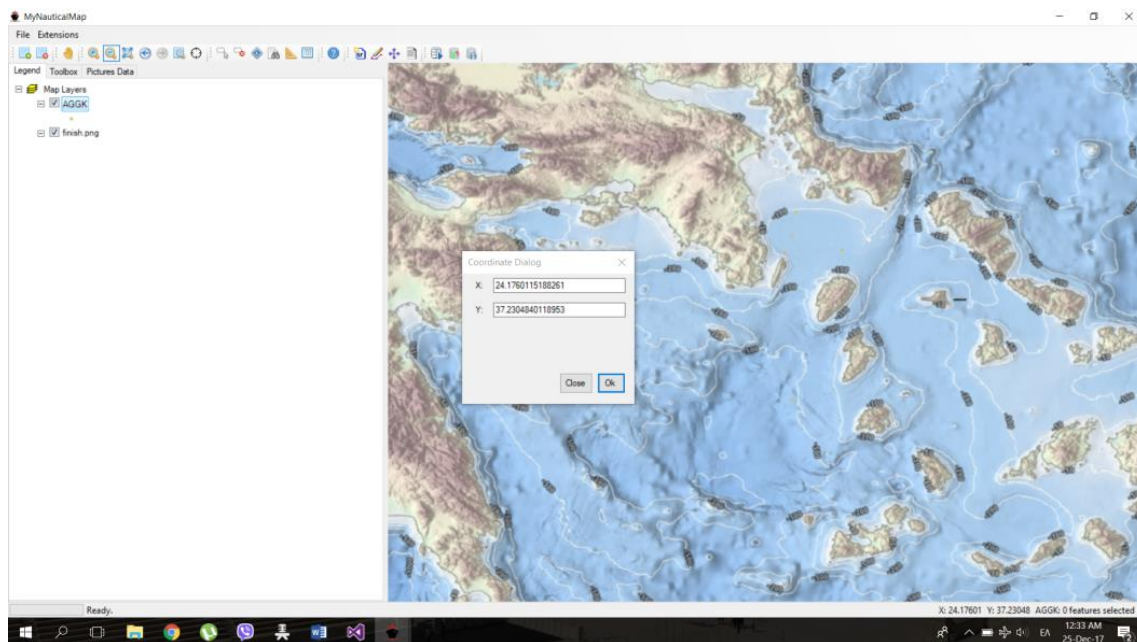
Εικόνα 5.9 – Η καρτέλα Pictures Data.



Εικόνα 5.10 – Η καρτέλα Pictures Data.

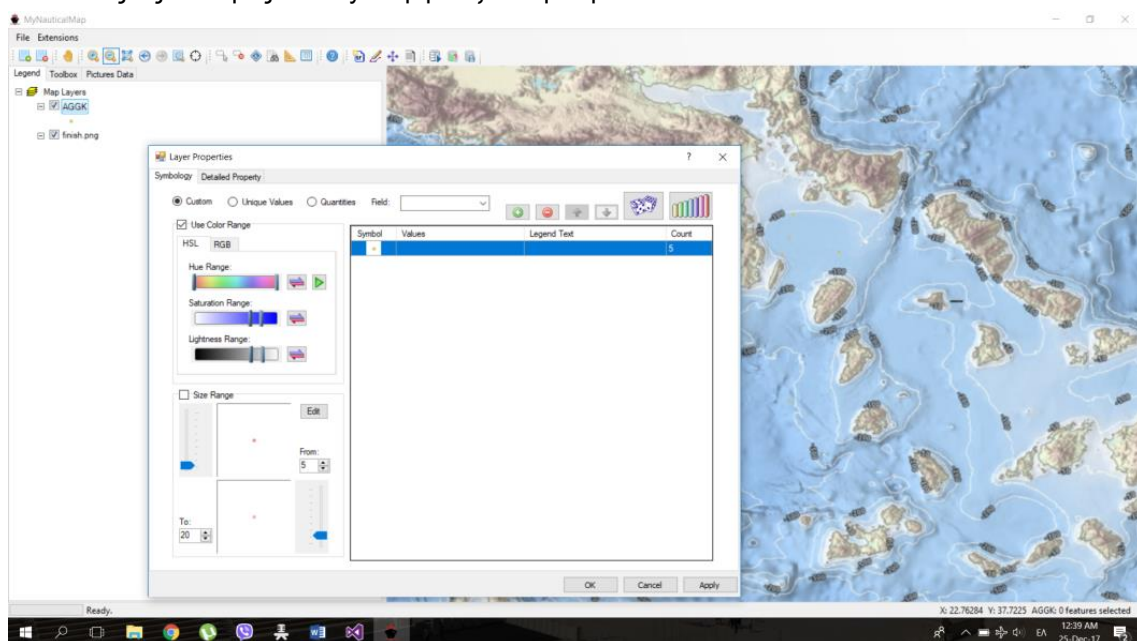
Στη συνέχεια, αφού προσθέσει ένα νέο επίπεδο μπορεί να εισάγει ναυτιλιακές πληροφορίες πατώντας το κουμπί Add share από το menu strip. Μπορεί ακόμα να προσθέσει πληροφορία και στο ίδιο αρχικό επίπεδο.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



Εικόνα 5.11 – Προσθήκη ναυτιλιακής πληροφορίας στο χάρτη.

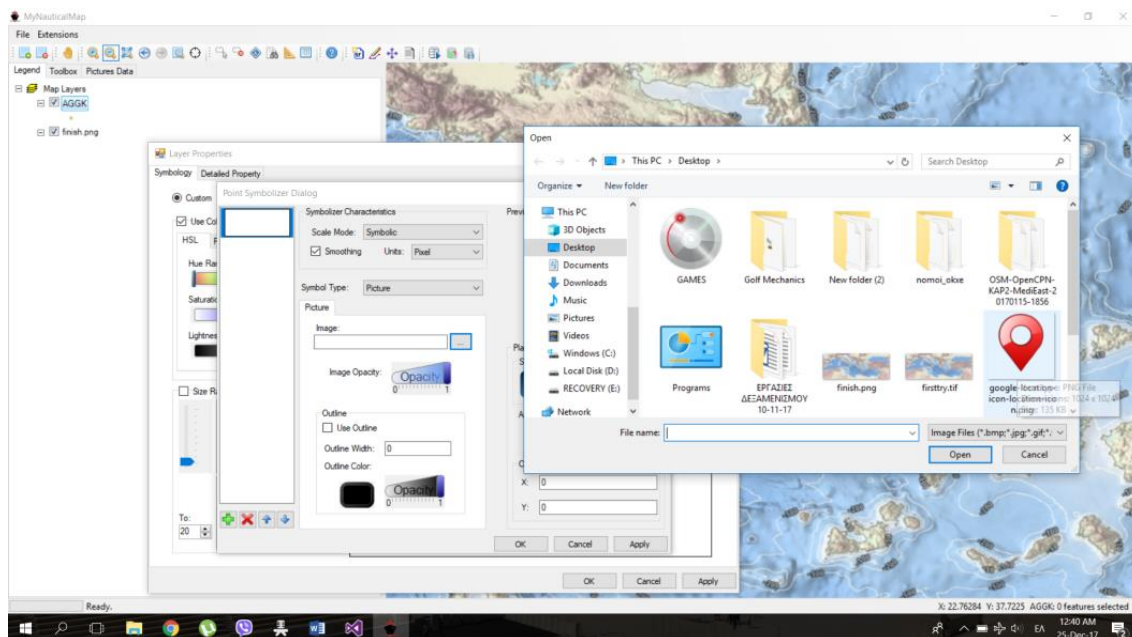
Πατώντας τις ιδιότητες του layer εμφανίζεται η παρακάτω εικόνα:



Εικόνα 5.12 – Εμφάνιση ιδιοτήτων ενός επιπέδου (layer).

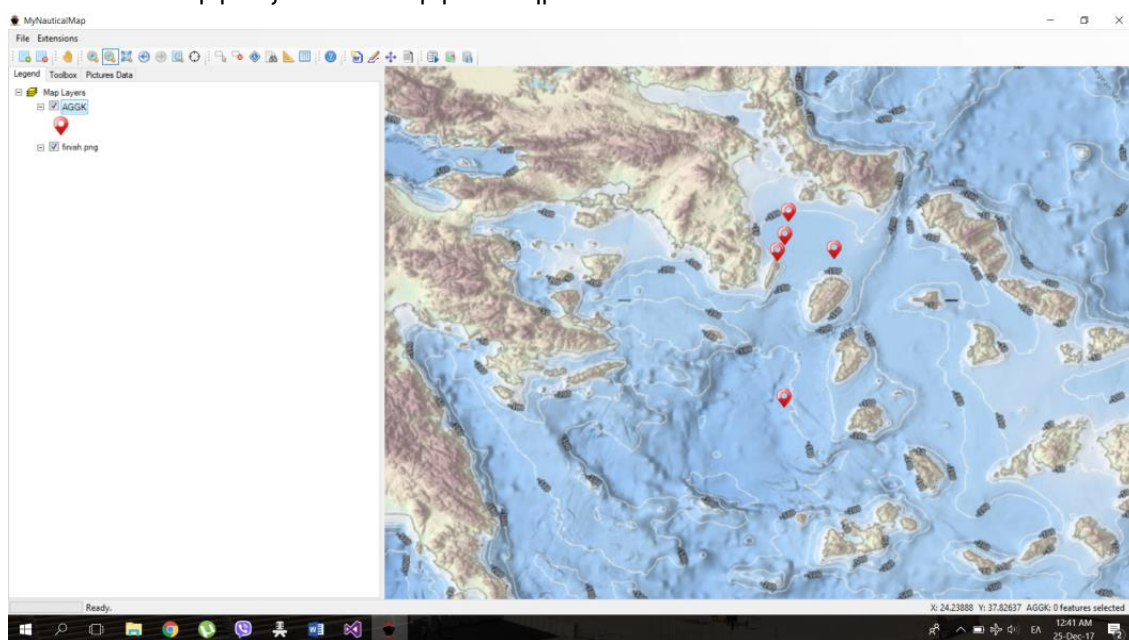
Εκεί, μεταξύ και των άλλων δυνατοτήτων που παρέχονται στο συγκεκριμένο παράθυρο, μπορεί να αλλάξει ο τρόπος παρουσίασης του σημείου που εμφανίζεται στον χάρτη.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



Εικόνα 5.13 – Αλλαγή ρυθμίσεων ενός επιπέδου (layer).

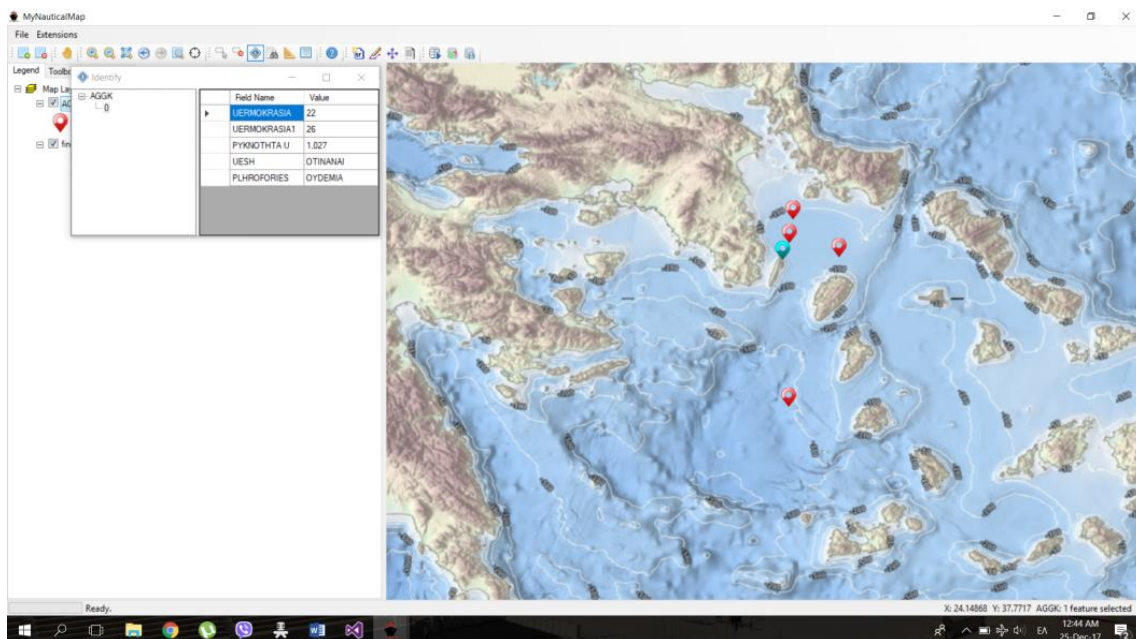
Έτσι πλέον θα εμφανίζονται πιο κομψά τα σημεία.



Εικόνα 5.14 – Εμφάνιση σημείων στο χάρτη με διαφορετικό εικονίδιο.

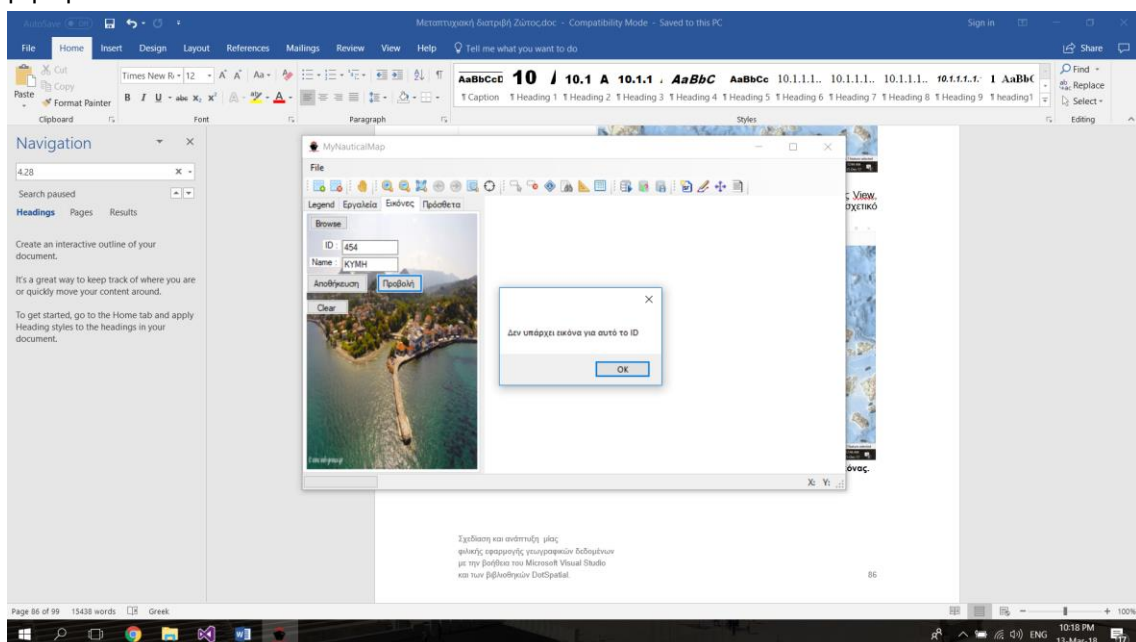
Πατώντας το κουμπί Identify πάνω σε ένα σημείο, βλέπει όλες τις διαθέσιμες πληροφορίες για αυτό.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



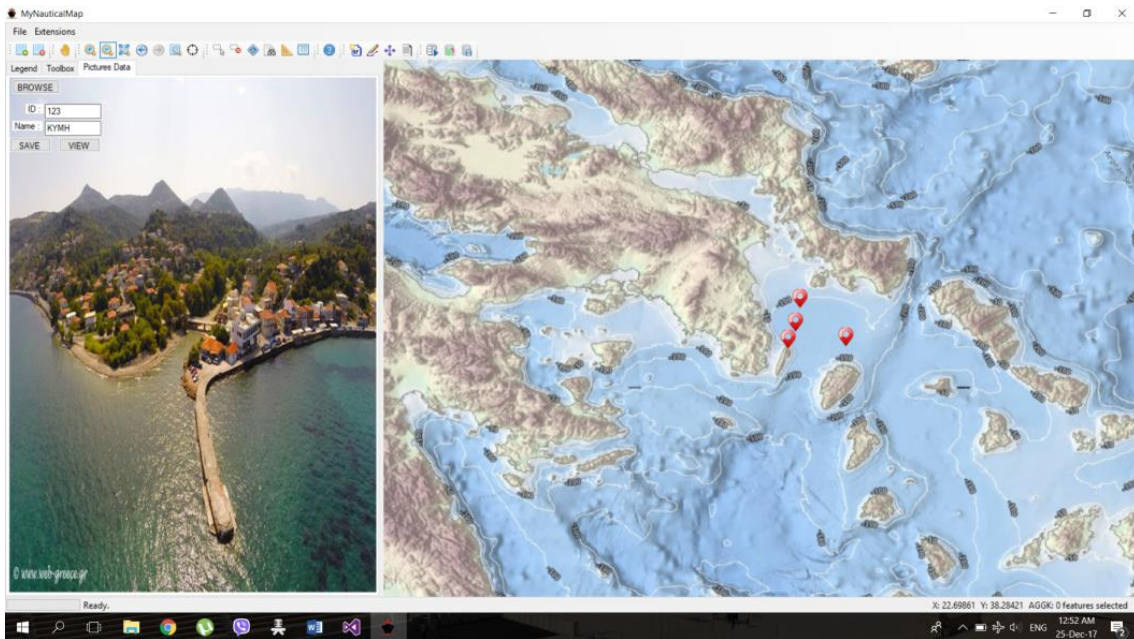
Εικόνα 5.15 – Ανάγνωση του πίνακα ιδιοτήτων (attribute table) για το σημείο.

Πηγαίνοντας στο Pictures Data, βάζοντας το Id του σημείου που θέλει και πατώντας View, μπορεί να δει την εικόνα που του αντιστοιχεί. Αν δεν υπάρχει εικόνα εμφανίζεται σχετικό μήνυμα.



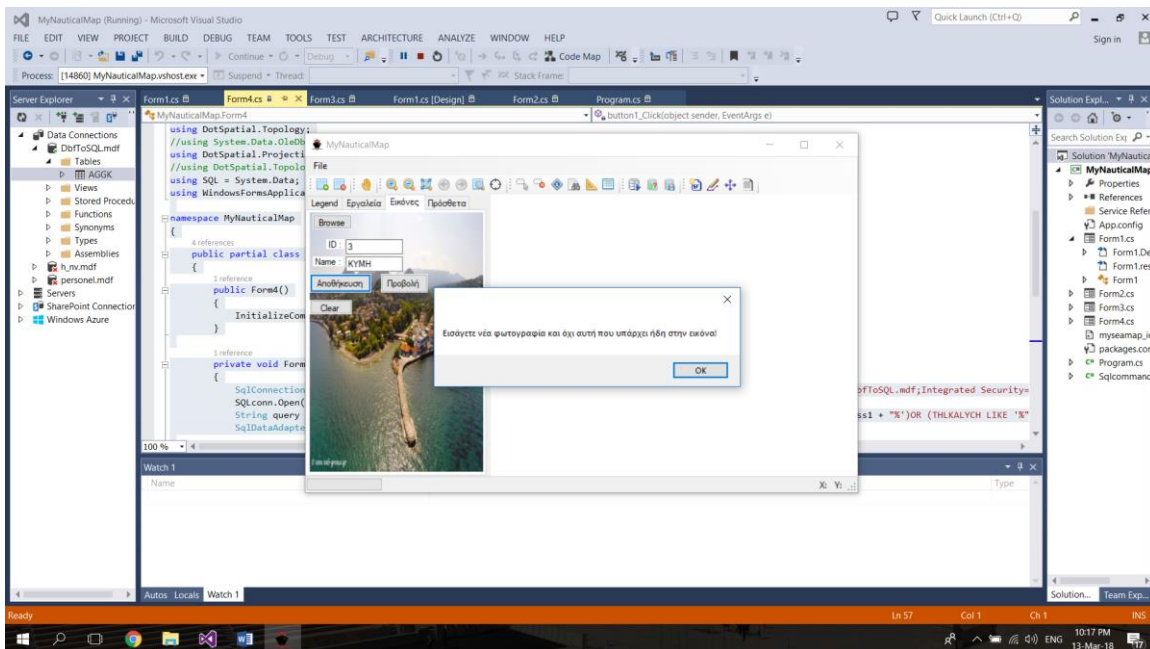
Εικόνα 5.16 – Εισαγωγή Id στην καρτέλα Pictures Data για την εμφάνιση αποθηκευμένης εικόνας.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



Εικόνα 5.17 – Εμφάνιση αποθηκευμένης εικόνας.

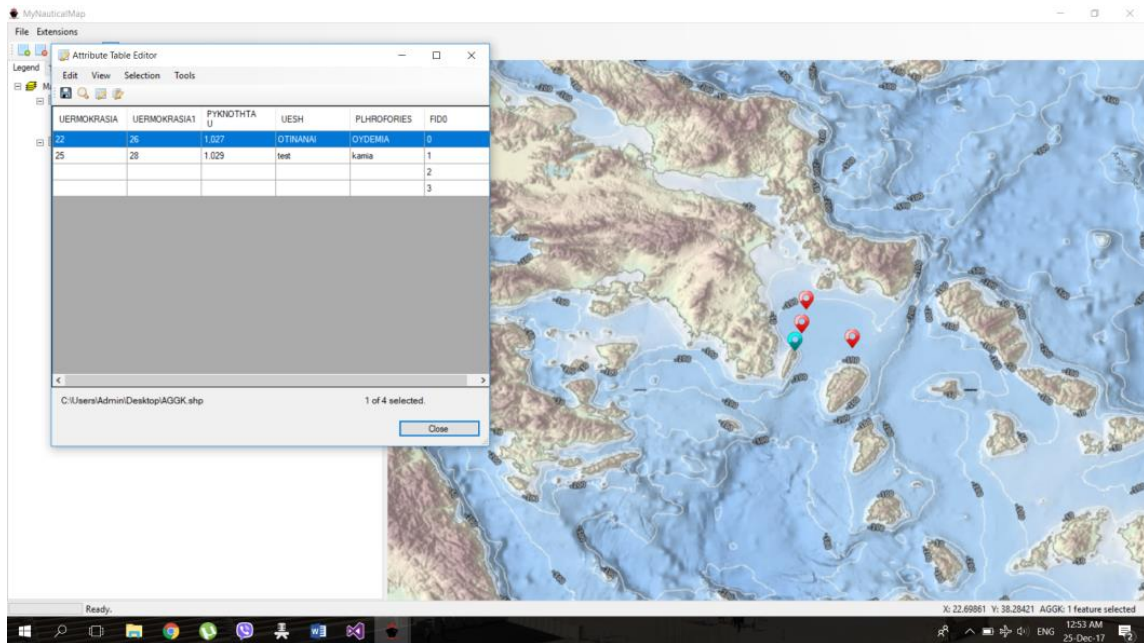
Αν ο χρήστης πατήσει λανθασμένα save σε μία ήδη αποθηκευμένη εικόνα θα εμφανιστεί ένα αντίστοιχο μήνυμα.



Εικόνα 5.18 – Εμφάνιση μηνύματος σφάλματος σε περίπτωση αποθήκευσης της ήδη αποθηκευμένης εικόνας.

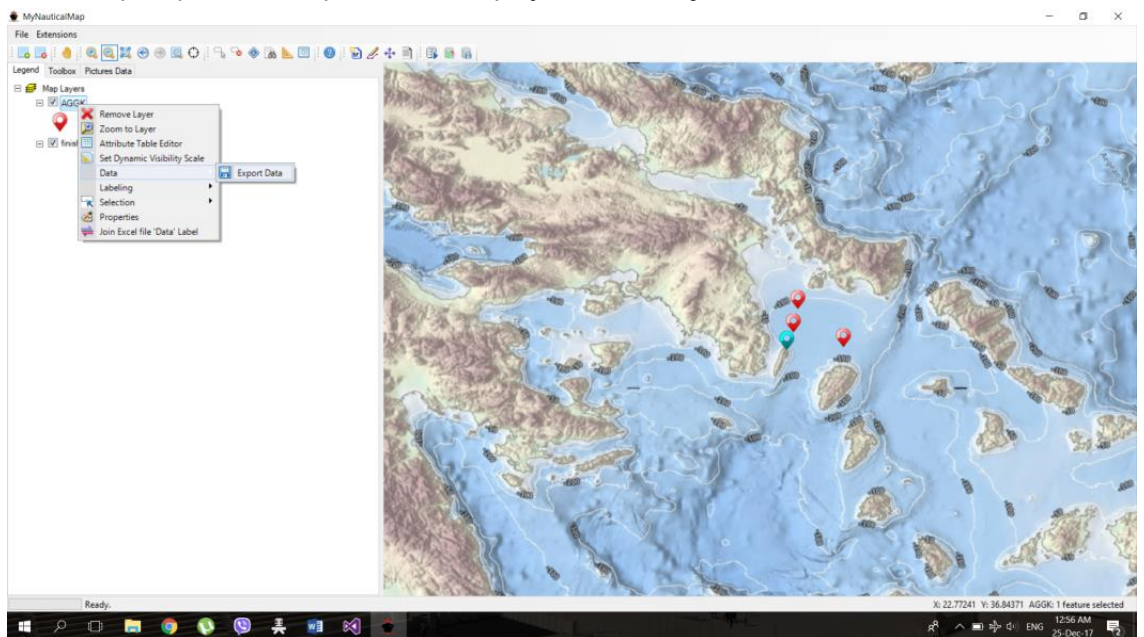
Ο χρήστης, εκτός από το να δει ξεχωριστά τα στοιχεία κάθε ναυτιλιακής πληροφορίας, μπορεί να δει το σύνολο αν πατήσει δεξί κλικ στο επίπεδο και Attribute table editor.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



Εικόνα 5.19 – Εμφάνιση του συνόλου των αποθηκευμένων ναυτιλιακών πληροφοριών.

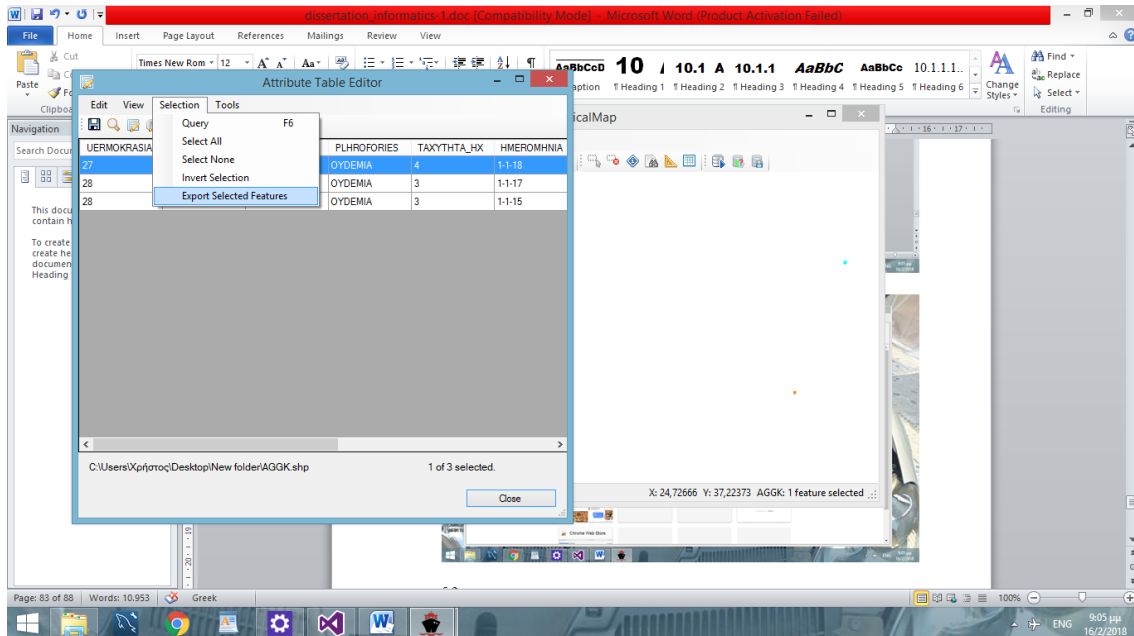
Ο χρήστης μπορεί να αποθηκεύσει και να εξαγάγει τα δεδομένα του, ώστε να είναι διαθέσιμα στον ίδιο αλλά και σε άλλους χρήστες της εφαρμογής πατώντας Data export με δεξί κλικ στο ίδιο επίπεδο ή ακόμα να αποθηκεύσει όλο το project πατώντας το file και save .



Εικόνα 5.20 – Εξαγωγή δεδομένων ενός χάρτη.

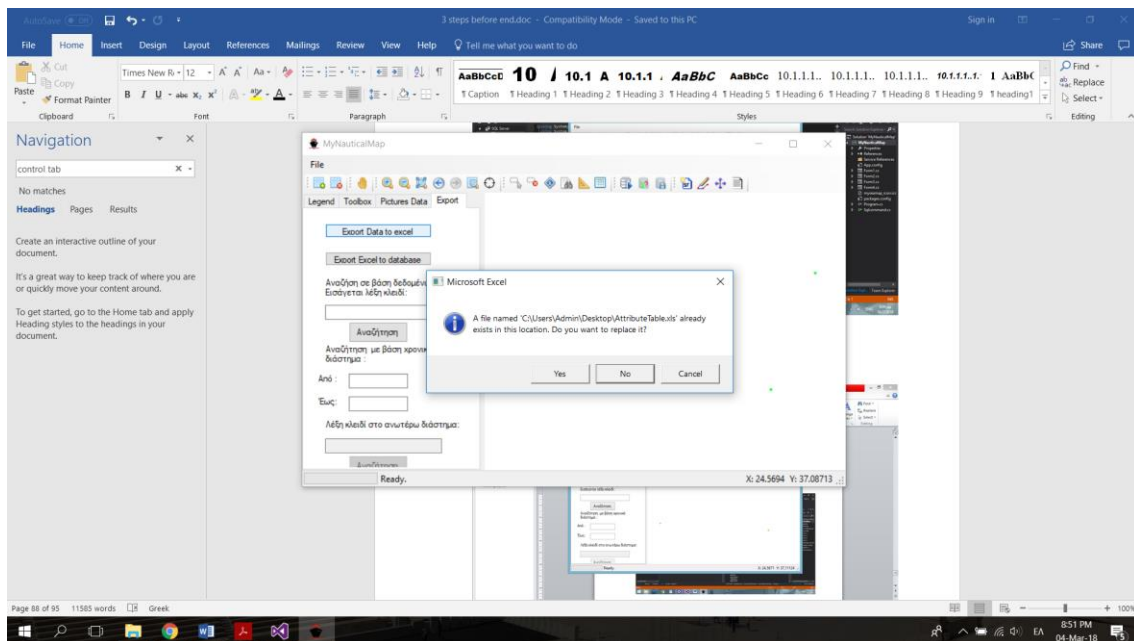
Ο χρήστης μπορεί να αποθηκεύσει και να εξαγάγει επιλεγμένα δεδομένα από τον attribute table editor :

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



Εικόνα 5.21 – Εξαγωγή επιλεγμένων δεδομένων ενός χάρτη.

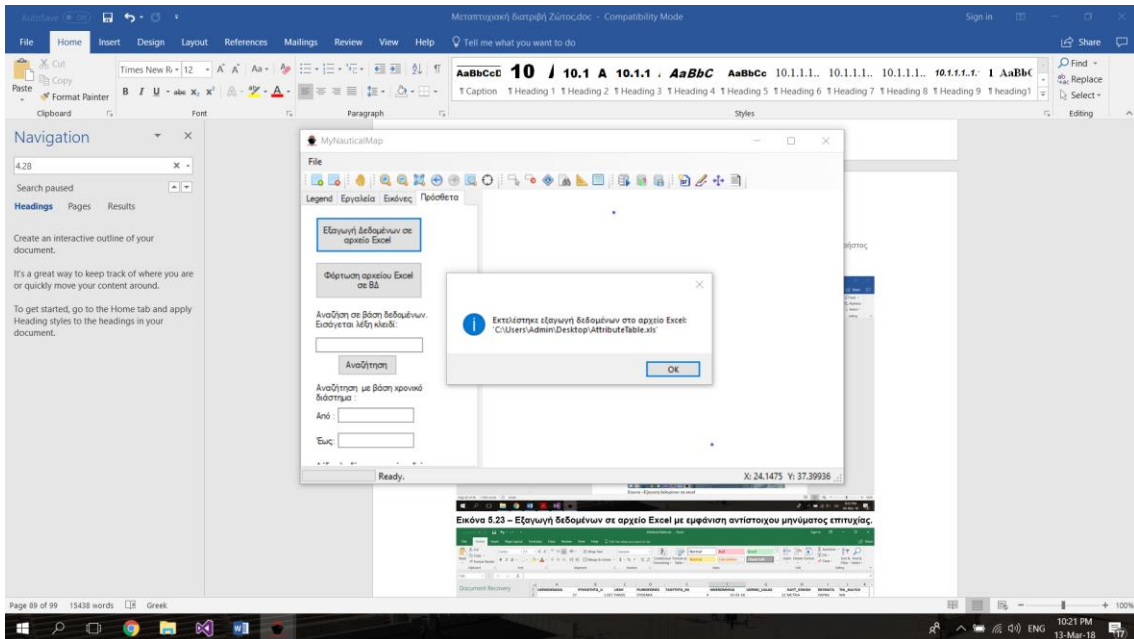
Μπορεί αν θέλει να εξαγάγει το σύνολο των δεδομένων του σε ένα Excel για χρήση τους εκτός της εφαρμογής και να τα εισάγει σε μία βάση δεδομένων για να εκτελεί λειτουργίες αναζήτησης σε αυτή. Αν υπάρχει ήδη Excel με ίδιο όνομα στην επιφάνεια εργασίας τότε εμφανίζεται μήνυμα αντικατάστασης.



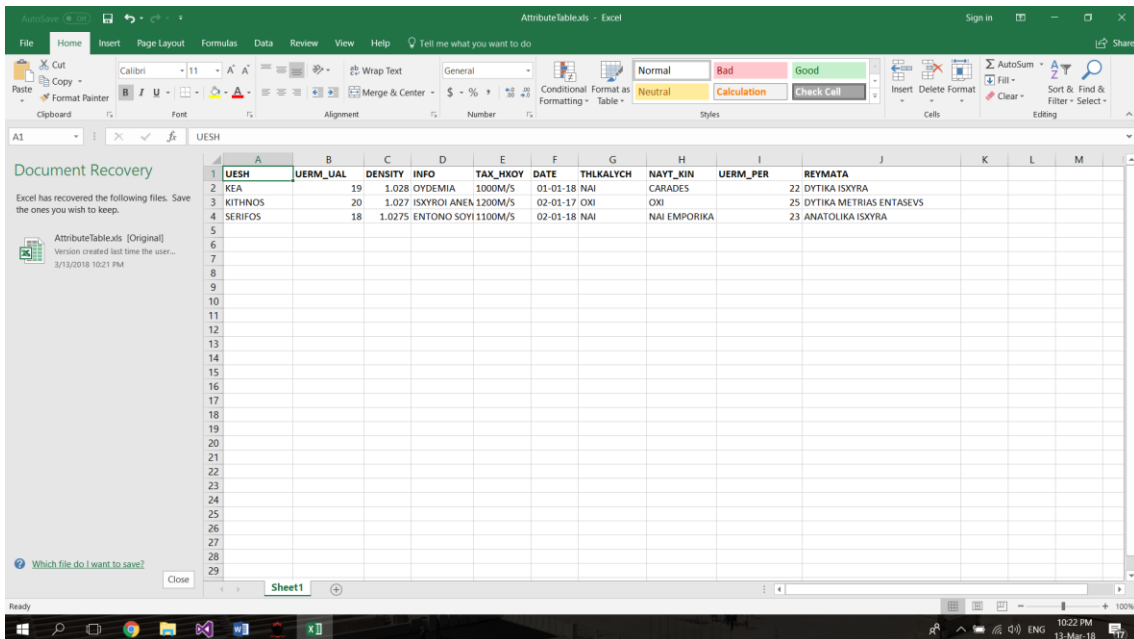
Εικόνα 5.22 – Εμφάνιση μηνύματος αντικατάστασης υπάρχοντος αρχείου Excel.

Αν δεν υπάρχει αρχείο Excel με το ίδιο όνομα τότε τα δεδομένα εξαγονται στο αρχείο με αντίστοιχο μήνυμα.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



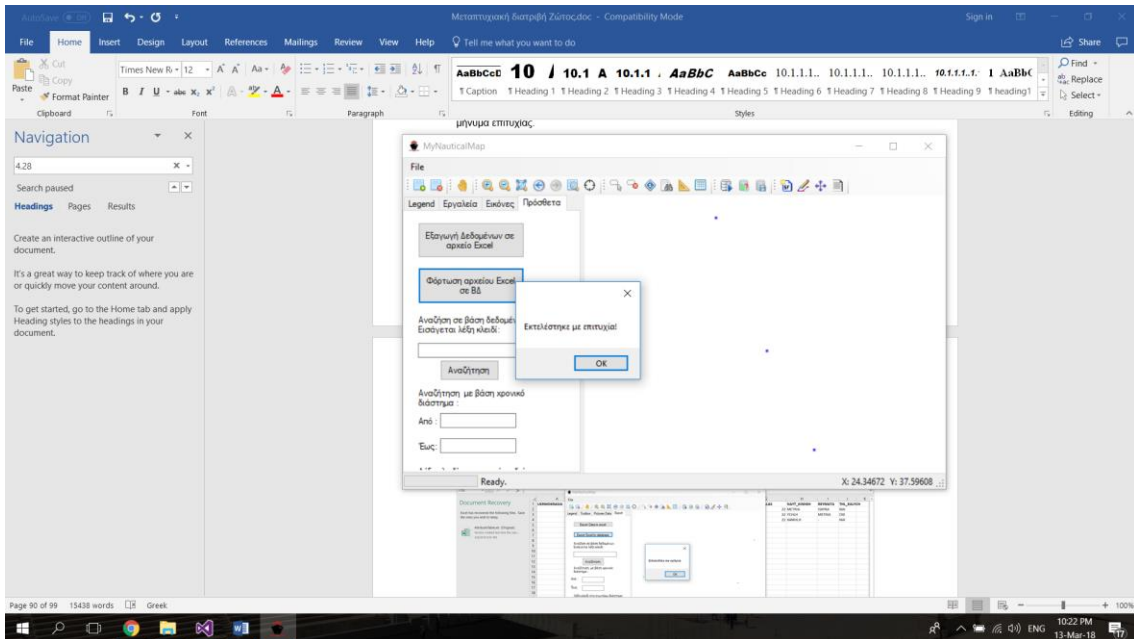
Εικόνα 5.23 – Εξαγωγή δεδομένων σε αρχείο Excel με εμφάνιση αντίστοιχου μηνύματος επιτυχίας.



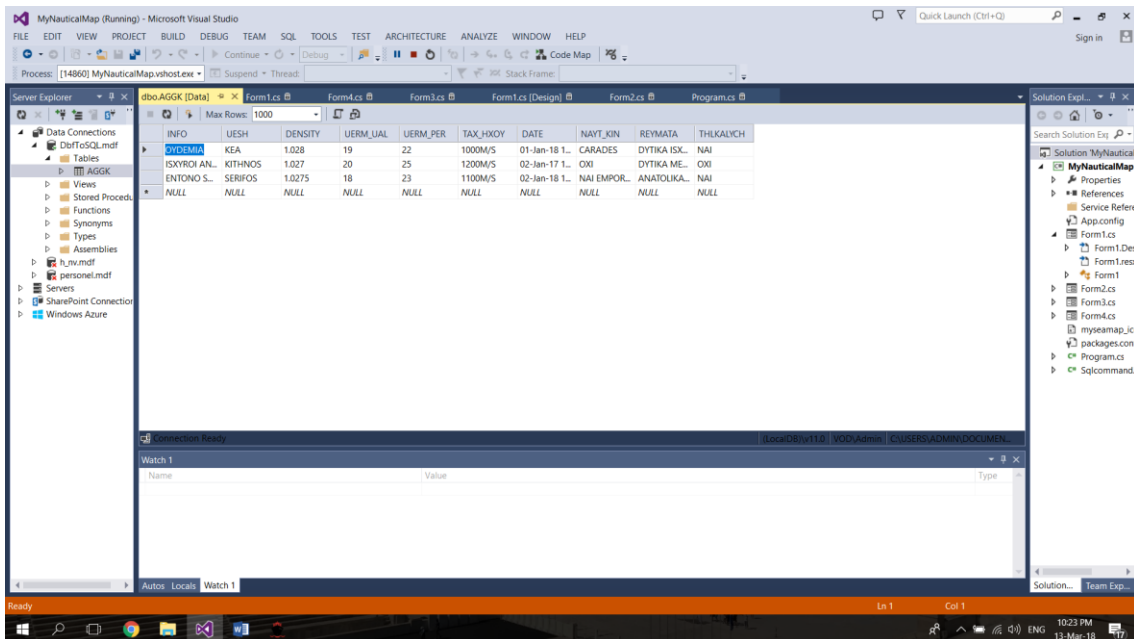
Εικόνα 5.24 – Το αρχείο Excel.

Επιλέγοντας την εισαγωγή του αρχείου Excel στη βάση δεδομένων εμφανίζεται αντίστοιχο μήνυμα επιτυχίας.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



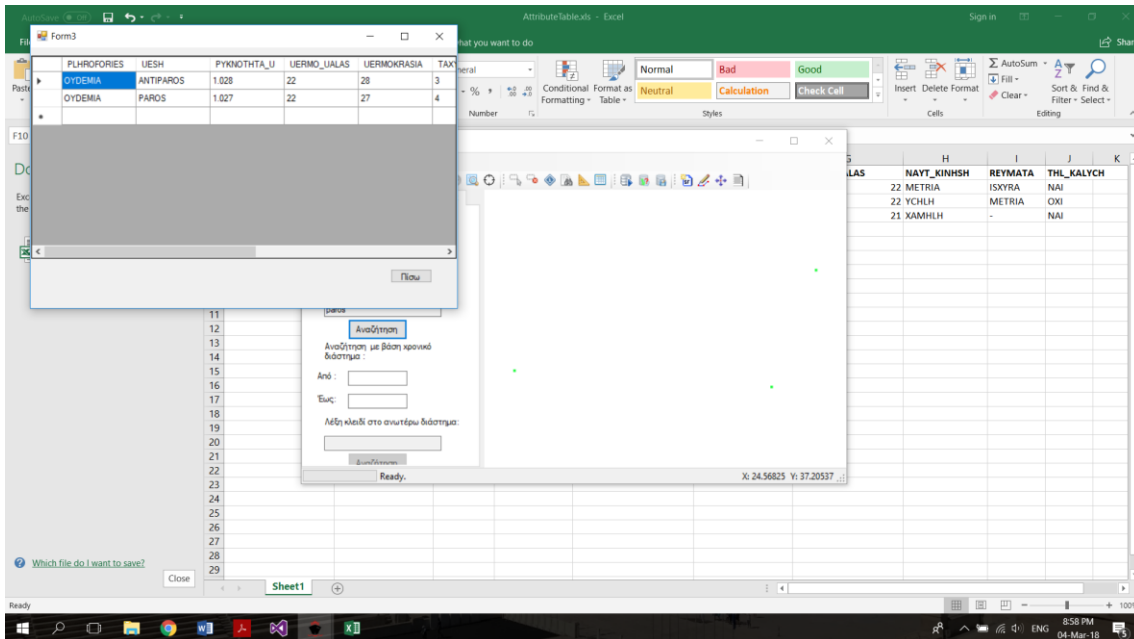
Εικόνα 5.25 – Εμφάνιση μηνύματος επιτυχίας της εισαγωγής του αρχείου Excel στη ΒΔ.



Εικόνα 5.26 – Απεικόνιση των δεδομένων στη βάση δεδομένων.

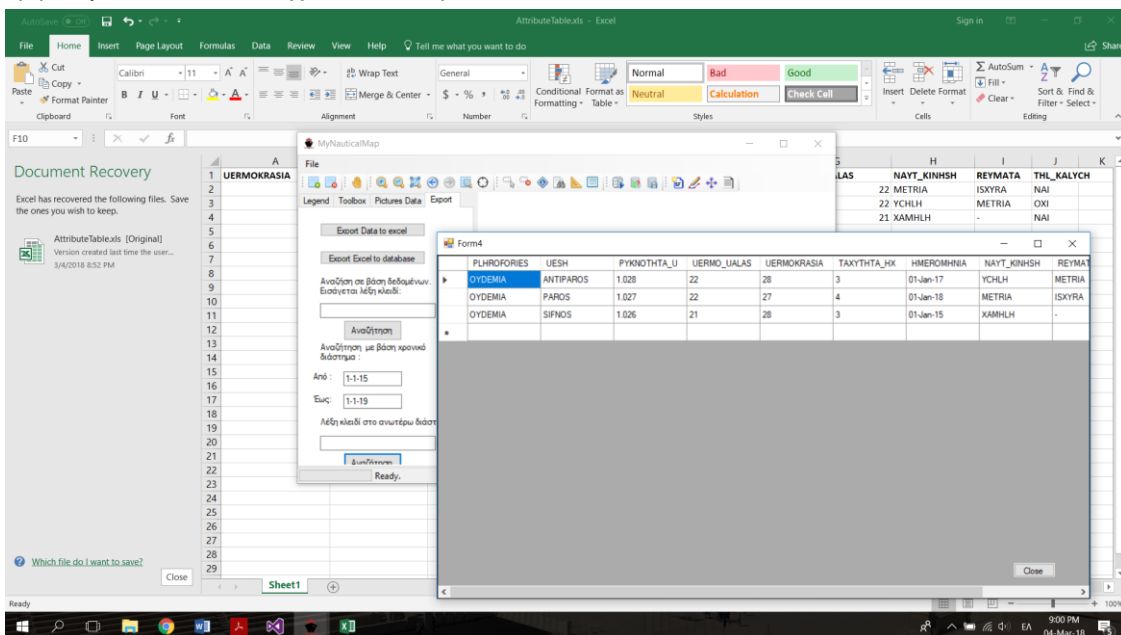
Πατώντας αναζήτηση με χρήση μόνο λέξης κλειδιού, εμφανίζονται τα αντίστοιχα αποτελέσματα.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



Εικόνα 5.27 – Αναζήτηση με χρήση μόνο λέξης κλειδιού.

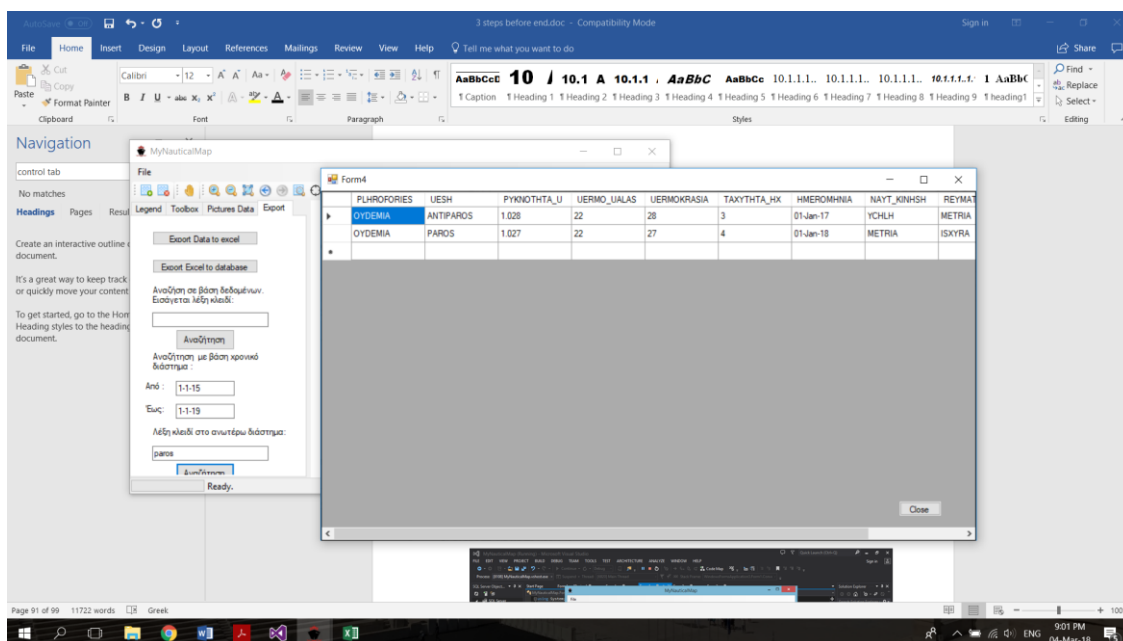
Πατώντας αναζήτηση με τη χρήση διαστήματος μεταξύ δύο ημερομηνιών λήψης στοιχείων, εμφανίζονται τα αντίστοιχα αποτελέσματα.



Εικόνα 5.28 – Αναζήτηση με τη χρήση διαστήματος μεταξύ δύο ημερομηνιών λήψης στοιχείων.

Πατώντας αναζήτηση με τη χρήση διαστήματος μεταξύ δύο ημερομηνιών λήψης των στοιχείων και λέξης κλειδιού, εμφανίζονται τα αντίστοιχα αποτελέσματα.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.



Εικόνα 5.29 – Αναζήτηση με τη χρήση διαστήματος μεταξύ δύο ημερομηνιών λήψης στοιχείων και λέξης κλειδιού.

Συμπεράσματα

Σε αυτή τη μεταπτυχιακή διατριβή αναπτύχθηκε η εφαρμογή 'MyNauticalMap' με τη βοήθεια του Visual Studio της Microsoft, της βιβλιοθήκης του DotSpatial της CodePlex και του SQL Server, με σκοπό την υποτύπωση στην εφαρμογή, ναυτιλιακού περιεχομένου πληροφοριών σε ένα χάρτη από το χρήστη. Με τη συγκεκριμένη εφαρμογή, ο οποιοσδήποτε χρήστης σε ένα πλοίο, αφού συνδεθεί σε αυτή με τους κωδικούς που θα του παρέχονται, θα μπορεί αρχικά να εισάγει έναν χάρτη και έπειτα μια πληροφορία ναυτιλιακού περιεχομένου (π.χ. αγκυροβόλι ενός πλοίου, ναυτιλιακός κίνδυνος). Στη συνέχεια θα εισάγει δεδομένα στον πίνακα χαρακτηριστικών των ναυτιλιακών πληροφοριών και θα έχει τη δυνατότητα να εισάγει εικόνες. Το σύνολο των δεδομένων που εισάγει θα μπορεί να εξαχθεί σε ένα Excel κυρίως για χρήση των στοιχείων εκτός της εφαρμογής, αλλά και για την υποβοήθηση του χρήστη σε λειτουργίες αναζήτησης, αφού εισαχθεί σε μία βάση δεδομένων.

Στόχος της εφαρμογής είναι η υποβοήθηση του πληρώματος της γέφυρας ενός πλοίου, με την παροχή πληροφοριών εύκολα μεταβιβάσιμων σε ένα ιδιαίτερα φιλικό περιβάλλον.

Μελλοντικές επεκτάσεις

Στην ενότητα αυτή θα αναφερθούμε σε επεκτάσεις δυνατοτήτων που μπορεί να υλοποιηθούν στη συγκεκριμένη εφαρμογή.

7.1 Σύνθετη αναζήτηση στη βάση δεδομένων του SQL Server

Η εφαρμογή θα μπορούσε να έχει ένα πιο εξελιγμένο σύστημα αναζήτησης στοιχείων στη βάση δεδομένων της. Σε αυτό θα μπορούσαν να χρησιμοποιούνται επιπλέον εργαλεία όπως drop – down lists, checkboxes, checkboxlists, radiobuttons που παρέχονται από τα Windows forms,

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

δημιουργώντας έναν ιδανικό τρόπο για εύκολη και γρήγορη αναζήτηση των δεδομένων της εφαρμογής με τον κατάλληλο κώδικα.

7.2 GPS

Μία άλλη επιπλέον δυνατότητα που θα μπορούσε να έχει η εφαρμογή, είναι το GPS. Συγκεκριμένα, θα μπορούσε να συνδεθεί απευθείας με το σύστημα GPS του πλοίου και να έχει πρόσβαση στα δεδομένα του τηλεγραφήματος. Αυτά τα δεδομένα που θα λάμβανε, θα μπορούσαν να χρησιμοποιηθούν με πολλούς τρόπους. Για παράδειγμα, τα δεδομένα στίγματος x και y θα μπορούσαν να αξιοποιηθούν χαράζοντας την τροχιά του πλοίου μέσω `linestring`. Η τροχιά του πλοίου θα μπορούσε να εξαχθεί σαν αρχείο `.shp` ή και σαν αρχείο `.csv`.

7.3 Δημιουργία μίας βάσης δεδομένων

Στην παρούσα εφαρμογή υπάρχουν τρεις βάσεις δεδομένων και σε κάθε βάση δεδομένων υπάρχει ένας πίνακας. Σε αυτή τη μορφή οδήγησε η σταδιακή εξέλιξη της εφαρμογής. Θα μπορούσε λοιπόν, να υπάρχει μία βάση δεδομένων με τρεις πίνακες, οι οποίοι να συνδέονται μεταξύ τους.

7.4 Εξαγωγή πίνακα χαρακτηριστικών από `polygon` και `linestring layer` σε αρχείο `Excel`.

Η εφαρμογή μπορεί να εξαγει τον πίνακα χαρακτηριστικών μόνο από `layer` τύπου `Point`, λόγω των απαιτήσεων που υπάρχουν. Με πρόσθεση επιπλέον κώδικα θα ήταν όμως δυνατή και η εξαγωγή πίνακα χαρακτηριστικών και από `layer` τύπου `linestring` και `polygon`.

Βιβλιογραφικές Αναφορές

1. Microsoft Visual Studio, Ολοκληρωμένο IDE για ανάπτυξη εφαρμογών, <https://www.visualstudio.com/>.
2. DotSpatial, Βιβλιοθήκη για γεωγραφικό σύστημα πληροφοριών, <http://dotspatial.codeplex.com/>.
3. Microsoft SQL Server, Πλατφόρμα για δημιουργία βάσεων δεδομένων, <https://www.microsoft.com/en-us/sql-server/sql-server-2016>.
4. Geofabrik, Εξαγωγή χωρικών δεδομένων από Openstreetmap, <http://download.geofabrik.de/>
5. Openstreetmap, Εξαγωγή χωρικών δεδομένων, <https://www.openstreetmap.org/>.
6. UML, Εκμάθηση της UML, <https://www.tutorialspoint.com/uml/>.
7. NOAA, Εθνικό κέντρο περιβαλλοντολογικής πληροφορίας, <https://www.ngdc.noaa.gov/mgg/ibcm/bathy/>.
8. EMODnet, European Marine Observation and Data Network, <http://www.emodnet-bathymetry.eu/data-products/web-services-and-standards>.
9. BODC, British Oceanographic Data Centre, https://www.bodc.ac.uk/data/hosted_data_systems/gebco_gridded_bathymetry_data/.
10. A survey of official online sources of high-quality free-of-charge geospatial data for maritime geographic information systems applications, Christos Kalyvas, Athanasios Kokkos, Theodoros Tzouramanis.
11. Γιάννης Θεοδωρίδης, Νίκος Πελέκης, vol. 2017.03 Γεωπληροφορική 4ο εξ.
12. Γιάννης Θεοδωρίδης, Νίκος Πελέκης, vol. 2016.03, Γεωγραφικά Πληροφοριακά Συστήματα (GIS) & Βάσεις Γεωγραφικών Δεδομένων.
13. Υδροσκόπιο, Εθνική τράπεζα υδρολογικής και μετεωρολογικής πληροφορίας, <http://www.hydroscope.gr/>.
14. Σημειώσεις μαθήματος Πανεπιστήμιο Πειραιώς, Τμήμα πληροφορικής, Γεωγραφικά πληροφοριακά συστήματα (GIS), Γιάννης Θεοδωρίδης, Νίκος Πελέκης, Ομάδα Διαχείρισης Δεδομένων, Εργαστήριο Πληροφοριακών συστημάτων.
15. Παρουσίαση μαθήματος GIS, May 2011, InfoLab mobility data collection.
16. M. Vodas, N. Pelekis, Y. Theodoridis, C. Ray, V. Karkaletsis, S. Petridis, A. Miliou, 2012, Big AIS Data Processing for Environmentally Safe Shipping'.
17. C. Claramunt, T. Devogele, S. Fourier, V. Noyon, M. Petit, C. Ray, 2007, Maritime GIS: From Monitoring to Simulation Systems.
18. Alexandru Boicea, 2004, Spatial Database for Geographical Information System (GIS) Applications.
19. N. Pelekis, E. Frentzos, N. Giatrakos, Y. Theodoridis, 2008, HERMES: Aggregative LBS via a Trajectory DB Engine.
20. M. Perez, R. Chang, R. Billings, 2009, Automatic Identification systems (AIS) Data Use in Marine Vessel Emission Estimation.
21. V. Popovich, C. Claramunt, V. Osipov, C. Ray, T. Wang, D. Berbenev, 2009, Integration of Vessel Traffic Control System and Geographical Information System.
22. AA. Alesheikh, H. Helali, HA. Behroz, 2002, Web GIS: Technologies and Its Applications.
23. Ιστοσελίδα τεκμηρίωσης των AIS (Automatic Identification Systems) συστημάτων, http://en.wikipedia.org/wiki/Automatic_Identification_System.
24. Ιστοσελίδα τεκμηρίωσης των GIS (Geographical Information systems) συστημάτων, http://en.wikipedia.org/wiki/Geographic_information_system.

Σχεδίαση και ανάπτυξη μίας φιλικής εφαρμογής γεωγραφικών δεδομένων με τη βοήθεια του Microsoft Visual Studio και των βιβλιοθηκών DotSpatial.

25. Ιστοσελίδα τεκμηρίωσης της γλώσσας προγραμματισμού JavaScript, <http://en.wikipedia.org/wiki/JavaScript>.
26. Επίσημη ιστοσελίδα της Google που παρέχει οδηγίες για το Google Maps Javascript API v3, <https://developers.google.com/maps/documentation/javascript/reference>.
27. Επίσημη ιστοσελίδα της Microsoft που παρέχει οδηγίες για τις βιβλιοθήκες της γλώσσας Visual Basic .Net, <http://msdn.microsoft.com/en-us/library/be9zzz62.aspx?cs-save-lang=1&cs-lang=vb#code-snippet-3>.
28. Ιστοσελίδα τεκμηρίωσης των χωρικών βάσεων δεδομένων, http://en.wikipedia.org/wiki/Spatial_database.
29. Ιστοσελίδα που παρέχει πληροφορίες σχετικά με εντολές και συναρτήσεις γεω-χωρικών δεδομένων του SQL Server, <http://weblogs.asp.net/jhallal/archive/2011/12/27/geospatial-data-in-sql-server-2008.aspx>.
30. Ιστοσελίδα που παρέχει πληροφορίες σχετικά με χωρικά δεδομένα του SQL Server, <http://www.jasonfollas.com/blog/archive/2008/03/14/sql-server-2008-spatial-data-part-1.aspx>.
31. Ιστοσελίδα που παρέχει πληροφορίες σχετικά με γεωγραφικά και γεωμετρικά δεδομένα του SQL Server, <http://gcfrench.wikidot.com/sql-2008>.
32. Ιστοσελίδα που παρέχει πληροφορίες σχετικά με τα .NET_libraries και τα frameworks, https://en.wikipedia.org/wiki/List_of_.NET_libraries_and_frameworks.
33. Ιστοσελίδα που παρέχει πληροφορίες σχετικά με τον SQL SERVER, https://el.wikipedia.org/wiki/Microsoft_SQL_Server.
34. Ιστοσελίδα που παρέχει πληροφορίες σχετικά με το Microsoft Office Interop (Excel Automation) στην C# και το VB.NET, <https://www.gemboxsoftware.com/spreadsheet/articles/c-sharp-microsoft-office-interop-Excel-automation>.
35. Ιστοσελίδα που μπορεί να κατεβάσει ένας χρήστης τα εργαλεία που παρέχονται από το Microsoft Access Database Engine 2010 Redistributable, <https://www.microsoft.com/en-us/download/details.aspx?id=13255>.
36. Ιστοσελίδα που παρέχει κώδικα για το διάβασμα αρχείων Excel με χρήση του OLEDB, <http://www.jasinskionline.com/technicalwiki/Reading-Excel-Files-with-OLEDB.ashx?AspxAutoDetectCookieSupport=1>.
37. Ιστοσελίδα που παρέχει πληροφορίες σχετικά με τα ψηφιδωτά και τα διανυσματικά δεδομένα, <http://www.fastprint.co.uk/blog/raster-vs-vector-the-easy-to-understand-guide.html>.
38. Ιστοσελίδα που παρέχει πληροφορίες για την εγκατάσταση του Microsoft.ACE.OLEDB.12.0, <https://datasavvy.me/2017/07/20/installing-the-microsoft-ace-oledb-12-0-provider-for-both-64-bit-and-32-bit-processing/>.
39. Ιστοσελίδα που παρέχει πληροφορίες σχετικά για το χωρικό σύστημα EPSG:3857 -- WGS84, <http://spatialreference.org/ref/sr-org/epsg3857-wgs84-web-mercator-auxiliary-sphere/>.
40. Ιστοσελίδα που παρέχει πληροφορίες σχετικά Ιστοσελίδα που παρέχει πληροφορίες σχετικά EPSG:4326 -- WGS84, <http://spatialreference.org/ref/epsg/wgs-84/>.
41. Ιστοσελίδα που παρέχει πληροφορίες σχετικά με τα χωρικά συστήματα, <https://help.sap.com/viewer/663ef0f75aff46ed9220ce9699973594/17.0/en-US/3c207ab56c5f1014a95ba9268e096e6a.html>.
42. Ιστοσελίδα που παρέχει πληροφορίες σχετικά για το ArcGIS, <https://en.wikipedia.org/wiki/ArcGIS>.
43. Ιστοσελίδα που παρέχει πληροφορίες σχετικά για το QGIS, <https://en.wikipedia.org/wiki/QGIS>.

44.Ιστοσελίδα που παρέχει πληροφορίες σχετικά για τα χωρικά συστήματα και τα χωρικά δεδομένα, <https://www.techopedia.com/definition/871/spatial-data>.

45.Ιστοσελίδα που παρέχει πληροφορίες για τη δημιουργία αρχείου εικόνας, <https://social.msdn.microsoft.com/Forums/vstudio/en-US/f9e39595-04ca-42ae-a353-eb1a08602631/resolved-creating-image-file-parameter-is-not-valid?forum=netfxbcl>.

46.Ιστοσελίδα που παρέχει πληροφορίες σχετικά με τις βιβλιοθήκες του DotSpatial, <https://dotspatial.codeplex.com/wikipage?title=DotSpatial%20Tutorials&referringTitle=Documentation#DotSpatial1.0>.

47.Ιστοσελίδα που παρέχει πληροφορίες σχετικά με τους περιορισμούς του αρχείου .DBF, <https://support.esri.com/en/technical-article/000007920>.

48.Ιστοσελίδα που παρέχει πληροφορίες σχετικά για τα ψηφιδωτά και τα διανυσματικά δεδομένα, <https://www.ordnancesurvey.co.uk/support/understanding-gis/raster-vector.html>.

49.Ιστοσελίδα που αναλύει τις διαφορές μεταξύ των ψηφιδωτών και των διανυσματικών δεδομένων, <https://www.ppsprint.com/resources/difference-between-raster-vector/>.

50.Ιστοσελίδα που παρέχει πληροφορίες σχετικά με τα χωρικά συστήματα πληροφοριών, <https://www.slideshare.net/ssinghb4u/introduction-to-geographic-information-system-gis>.

51.Ιστοσελίδα που παρέχει πληροφορίες σχετικά για τα αρχεία shapefile, https://en.wikipedia.org/wiki/Shapefile#cite_note-limitations-4.

52.PDF που παρέχει πληροφορίες σχετικά με τα χωρικά συστήματα πληροφοριών, <http://www.esri.com/library/bestpractices/what-is-gis.pdf>.

53.Ιστοσελίδα που παρέχει πληροφορίες για την εμφάνιση tooltip, <http://www.c-sharpcorner.com/uploadfile/mahesh/tooltip-in-C-Sharp/>.

54.Ιστοσελίδα που παρέχει πληροφορίες σχετικά για την εισαγωγή δεδομένων από ένα αρχείο .dbf σε μία βάση δεδομένων SQL Server, <https://forums.asp.net/t/1221734.aspx?code+for+import+dbf+file+into+sqlserver+using+C+> .

55.Ιστοσελίδα που παρέχει πληροφορίες σχετικά με την εύκολη διαγραφή των δεδομένων ενός πίνακα, <https://stackoverflow.com/questions/11103181/a-fast-way-to-delete-all-rows-of-a-datatable-at-once>.

56.Ιστοσελίδα που παρέχει πληροφορίες για τον χειρισμό του SqlBulkCopyColumnMapping, [https://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlbulkcopycolumnmapping\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlbulkcopycolumnmapping(v=vs.110).aspx).

57.Ιστοσελίδα που παρέχει πληροφορίες για την εισαγωγή δεδομένων από ένα Excel σε μία βάση δεδομένων SQL server, <https://stackoverflow.com/questions/43295145/insert-data-into-sql-server-database-from-an-excel-file>.

58.Ιστοσελίδα που παρέχει πληροφορίες σχετικά με την εξαγωγή δεδομένων από μία βάση δεδομένων SQL σε ένα αρχείο Excel, <https://stackoverflow.com/questions/8580591/export-sql-to-excel>.

59.Ιστοσελίδα που παρέχει πληροφορίες για την ανάγνωση και μεταφορά δεδομένων από ένα αρχείο .DBF, <https://forums.asp.net/t/1697341.aspx?Transfer%20Read%20Column%20Names%20in%20a%20FoxPro%20dbf%20file%20in%20C>.

60.Η μεταπτυχιακή διατριβή «Σχεδίαση και Ανάπτυξη της Εφαρμογής Vessel Tracking System με χρήση Spatial Database» του Ιωάννη Καραγουστή του Νικολάου στο Πανεπιστήμιο Πειραιώς, με επιβλέποντες τον καθηγητή Ιωάννη Θεοδωρίδη και τον διαδοκτορικό Μάριο Βόντα του Προγράμματος Μεταπτυχιακών Σπουδών «Προηγμένα Συστήματα Πληροφορικής».

61.Η μεταπτυχιακή διατριβή «Δημιουργία ενός Ολοκληρωμένου Συστήματος παροχής χωρικής πληροφορίας, βασισμένο σε τεχνολογίες ανοιχτού κώδικα και ελεύθερα δεδομένα» του

Δεληγιάννη Ιωάννη του Σπύρου στο Πανεπιστήμιο Πειραιώς με επιβλέποντα Καθηγητή τον κ.Ιωάννη Θεοδωρίδη.

62.Ιστοσελίδα που παρέχει πληροφορίες για τα χωρικά συστήματα πληροφοριών, <http://gisgeography.com/free-gis-software/>.

63.Ιστοσελίδα που παρέχει πληροφορίες για την εφαρμογή MapWindow, https://en.wikipedia.org/wiki/MapWindow_GIS.

63.Ιστοσελίδα που παρέχει πληροφορίες για την εφαρμογή MapWindow, [https://translate.google.gr/?hl=el#en/el/The%20MapWindow%20GIS%20project%20includes%20a%20free%20and%20open%20source%20desktop%20geographic%20information%20system%20\(GIS\)%20with%20an%20extensible%20plugin%20architecture%2C%20a%20GIS%20ActiveX%20control%2C%20and%20C%23%20GIS%20programmer%20library%20called%20DotSpatial](https://translate.google.gr/?hl=el#en/el/The%20MapWindow%20GIS%20project%20includes%20a%20free%20and%20open%20source%20desktop%20geographic%20information%20system%20(GIS)%20with%20an%20extensible%20plugin%20architecture%2C%20a%20GIS%20ActiveX%20control%2C%20and%20C%23%20GIS%20programmer%20library%20called%20DotSpatial).

64.Ιστοσελίδα που παρέχει πληροφορίες για την εφαρμογή HydroDesktop, <http://www.ce.utexas.edu/prof/maidment/giswr2010/Ex1/Ex1HydroDesktop.htm>.

65.Ιστοσελίδα που παρέχει πληροφορίες για την βιβλιοθήκη DotSpatial, <https://archive.codeplex.com/?p=dotspatial>.

66.Ιστοσελίδα που παρέχει πληροφορίες για την εφαρμογή HydroDesktop, https://www.researchgate.net/figure/Extensible-ribbon-based-layout-design-of-HydroDesktop_fig2_260182619.

67.Ιστοσελίδα που παρέχει πληροφορίες για την εφαρμογή MapWindow, <http://www.softpedia.com/get/Programming/Other-Programming-Files/MapWindow-GIS.shtml>.

68.Ιστοσελίδα που παρέχει πληροφορίες για την εφαρμογή Microsoft Excel, https://el.wikipedia.org/wiki/Microsoft_Excel.

69.Ιστοσελίδα που παρέχει πληροφορίες για την αυτοματοποίηση του Microsoft Excel, <https://support.microsoft.com/el-gr/help/302084/how-to-automate-microsoft-excel-from-microsoft-visual-c-net>.