



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



## ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

### Οργάνωση και Διαχείριση Θερμοκηπίου



Της φοιτήτριας  
Βεγερτζή Κοκκώνης  
Αρ. Μητρώου: 03/2320

Επιβλέπων καθηγητής  
Κ. Γιαγκουσιτίδης

Θεσσαλονίκη 2013



## ΠΕΡΙΛΗΨΗ

Σκοπός της παρούσας διπλωματικής είναι η μελέτη, η ανάλυση και η υλοποίηση μιας εφαρμογής γραφείου διαχείρισης ενός θερμοκηπίου. Η εφαρμογή έχει υλοποιηθεί με την χρήση του Visual Studio 2008. Έχει γραφτεί με γλώσσα προγραμματισμού Visual Basic ενώ οι βάσεις δεδομένων είναι σε MySql.

Μέσω της εφαρμογής ο διαχειριστής μπορεί να χειρίζεται την αποθήκη του - το θερμοκήπιο δηλαδή – αλλά και τους πελάτες του. Έμφαση έχει δοθεί στο να προσφέρεται ένα φιλικό και εύχρηστο περιβάλλον στον χρήστη και να μην απαιτεί συγκεκριμένες γνώσεις για να διαχειριστή τις εγγραφές.

Όλες οι λειτουργίες γίνονται από την κεντρική φόρμα αφού ο διαχειριστής αυθεντικοποιηθεί με όνομα και κωδικό. Του παρέχεται η δυνατότητα διαχείρισης των αποθεμάτων του σε φυτά στο θερμοκήπιο, εισαγωγή και τροποποίηση τους οποιαδήποτε στιγμή.

Με την ίδια ευκολία έχει πρόσβαση σε μια βάση δεδομένων η οποία διατηρεί τα στοιχεία των πελατών του, τις παραγγελίες τους και τις οφειλές τους. Μπορεί εύκολα να αναζητήσει, να εισάγει ή να τροποποιήσει κάποια εγγραφή του – πελάτη – χωρίς την ανάγκη κάποιων ιδιαίτερων γνώσεων.

Επίσης γίνεται μια εκτενής περιγραφή των τεχνολογιών και εργαλείων που χρησιμοποιήθηκαν για την υλοποίηση της διπλωματικής. Καταγράφονται τα στάδια υλοποίησης, οι βάσεις που δημιουργήθηκαν και η διάρθρωση του.

Τέλος περιγράφεται ο κώδικας αναλυτικά για το πώς επιλέχτηκε να υλοποιηθεί η κάθε λειτουργία της εφαρμογής όπως και μια παρουσίαση της λειτουργίας του.

## ABSTRACT

The purpose of this thesis is to study, analyze and implement a program of organizing and management of a greenhouse. The application has been implemented by using the Visual Studio 2008. Written with Visual Basic programming language and the database are in MySql server.

With this application the manager can handle the store – the greenhouse and his customer's database. Emphasis has been given to providing a user-friendly environment for the user which does not require specific knowledge to manage records.

All functions are rooted to the central form after the administrator successfully authenticated with username and password. He has been given the capability to manage his plants resources anytime, import or update his existed records.

With the same ease he has access to a database which keeps the information of his customers, their orders and their debts. Can easily search, enter or modify a record of a customer without the need for any special knowledge.

Also includes an extensive description of the technologies and tools that used to implement the thesis. We describe the stages of implementation, the way that databases were created and the structure of it.

Finally we describe the source code in detail highlighting the significantly parts of it and how we have chosen the implementation of each function. Also in the last chapter there is a user manual of the application.

## **ΕΥΧΑΡΙΣΤΙΕΣ (προαιρετικά)**

## ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ.....	3
ABSTRACT .....	4
ΕΥΧΑΡΙΣΤΙΕΣ (προαιρετικά).....	5
ΠΕΡΙΕΧΟΜΕΝΑ .....	6
Ευρετήριο σχημάτων .....	7
ΕΙΣΑΓΩΓΗ.....	9
ΚΕΦΑΛΑΙΟ 1 - Βάση Δεδομένων. Διακομιστής MySql.....	11
Εισαγωγή.....	11
1.1 Η γλώσσα SQL .....	11
1.2 MySql Server .....	11
1.3 Πλεονεκτήματα της MySql .....	12
1.4 Βασικές εντολές .....	13
1.5 Γραφικό Εργαλείο Διαχείρισης της MySql. «MySql Workbench» .....	14
1.6 Ανάλυση Σχεδιασμού Βάσεων Δεδομένων Εφαρμογής Διαχείριση Θερμοκηπίου .....	16
Επίλογος.....	20
ΚΕΦΑΛΑΙΟ 2 - Προγραμματιστικό περιβάλλον Visual Studio 2008.....	22
Εισαγωγή.....	22
2.1 Microsoft Visual Studio 2008 .....	22
2.2 Χαρακτηριστικά Γνωρίσματα .....	23
2.2.1 Συντάκτης κώδικα (Compiler) .....	23
2.2.2 Διορθωτής (Debugger) .....	24
2.2.3 Σχεδιαστής εφαρμογής (Visual Designers) .....	25
2.2.4 Λοιπά Εργαλεία .....	27
2.3 Πρόσθετες γλώσσες προγραμματισμού.....	29
2.3.1 Microsoft Visual C++.....	29
2.3.2 Microsoft Visual C# .....	30
2.3.3 Microsoft Visual Basic .....	30
2.3.4 Microsoft Visual Web Developer .....	30
2.4 Microsoft Visual Basic .....	31
2.4.1 Χαρακτηριστικά της γλώσσας .....	31
2.5 Πλαίσιο Microsoft .Net .....	33
2.5.1 Γενικά .....	33
2.5.2 Βασικά Χαρακτηριστικά .....	34

2.5.3 Χαρακτηριστικά Σχεδιασμού .....	34
2.5.4 Εκδόσεις .Net Framework .....	36
2.6 ADO.NET .....	36
Επίλογος.....	39
<b>ΚΕΦΑΛΑΙΟ 3 - Βασικές Λειτουργίες της Εφαρμογής.....</b>	<b>41</b>
Εισαγωγή.....	41
3.1 Βασικές Λειτουργίες.....	41
3.2 Καρτέλα Διαχείρισης Αποθήκης .....	43
3.3 Καρτέλα Διαχείρισης Πελατών .....	57
Επίλογος.....	70
<b>ΚΕΦΑΛΑΙΟ 4 – Εγχειρίδιο Χρήσης.....</b>	<b>71</b>
<b>ΣΥΜΠΕΡΑΣΜΑΤΑ.....</b>	<b>77</b>
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>	<b>78</b>

### **Ευρετήριο σχημάτων**

Σχήμα 1 : Το εργαλείο MySql Workbench .....	14
Σχήμα 2 : Εκκίνηση του εξυπηρετητή .....	16
Σχήμα 3 : Δημιουργία της Βάσης Δεδομένων.....	17
Σχήμα 4 : Δημιουργία πίνακα plants.....	19
Σχήμα 5 : Δημιουργία του πίνακα stoxeia .....	20
Σχήμα 6 : Visual Studio 2008.....	23
Σχήμα 7 : Στιγμιότυπο χρήσης τεχνολογίας IntelliSense του Microsoft Visual Studio 2008 .....	24
Σχήμα 8 : Παλέτα εργαλείων Visual Basic.....	26
Σχήμα 9 : Visual Studio 2008.Παρουσίαση Βασικών Εργαλείων .....	29
Σχήμα 10 : Βασική Αρχιτεκτονική ADO.Net .....	38
Σχήμα 11 : Διαφορετικοί data providers ανάλογα με την Βάση Δεδομένων .....	39
Σχήμα 12 : Φόρμα Log In.....	71
Σχήμα 13 : Μήνυμα λάθους.....	71
Σχήμα 14 : Κεντρική Φόρμα .....	72
Σχήμα 15 : Καρτέλα εισαγωγής νέου πελάτη .....	73
Σχήμα 16 : Μήνυμα επιτυχούς εισαγωγής νέου πελάτη .....	73
Σχήμα 17 : Καρτέλα ανανέωσης εγγραφής πελάτη .....	74
Σχήμα 18 : Αναζήτηση πελάτη .....	74
Σχήμα 19 : Φόρμα διαγραφής πελάτη.....	75
Σχήμα 20 : Καρτέλα χρέωσης πελάτη .....	76





## ΕΙΣΑΓΩΓΗ

Παρατηρούμε τα τελευταία χρόνια στο χώρο των νέων τεχνολογιών μια αλματώδη ανάπτυξη των Βάσεων Δεδομένων. Αυτό οφείλεται στην ολοένα αυξανόμενη ανάγκη για καταγραφή και αποθήκευση της πληροφορίας. Πληροφορία που μπορεί να μην είναι σημαντική, αλλά αναγκαία για ένα συγκεκριμένο σκοπό (π.χ. το ψευδώνυμο που χρησιμοποιεί κάποιος χρήστης σε ένα forum), αλλά και πληροφορία που είναι τόσο σημαντική που υπάρχει ανάγκη προστασίας της (π.χ. ο αριθμός λογαριασμού σε μια τράπεζα).

Αυτή η ανάγκη για μαζική αποθήκευση της πληροφορίας έχει οδηγήσει στην ανάπτυξη του κλάδου των βάσεων δεδομένων και στην ανάδειξή του σε έναν από τους σημαντικότερους και πιο διαδεδομένους κλάδους των νέων τεχνολογιών.

Εξίσου σημαντικό παραμένει και το κομμάτι της διαχείρισης αυτής της πληροφορίας εύκολα και εύχρηστα ακόμα και από απλούς χρήστες χωρίς προηγμένες γνώσεις στην επιστήμη των υπολογιστών. Ένας τρόπος να επιτευχθεί αυτό είναι μέσω έξυπνων εφαρμογών διαχείρισης μεγάλων βάσεων δεδομένων.

Στο πρώτο κεφάλαιο της διπλωματικής παρουσιάζονται τεχνολογίες βάσεων δεδομένων. Αναλυτικότερα παρουσιάζεται ο MySQL διακομιστής ο οποίος και έχει επιλεγεί για την υλοποίηση της εφαρμογής.

Στο επόμενο κεφάλαιο γίνεται εκτενής παρουσίαση του προγραμματιστικού περιβάλλοντος Microsoft Visual Studio 2008 στο οποίο έχει υλοποιηθεί η εφαρμογή. Περιγράφονται οι τεχνολογίες που χρησιμοποιούνται και η γλώσσα προγραμματισμού που έχει επιλεγεί, Visual Basic.

Στο τρίτο κεφάλαιο γίνεται ανάλυση των λειτουργιών της εφαρμογής, των βασικών κομματιών που την αποτελούν καθώς και τον τρόπο επιλογής υλοποίησης της κάθε λειτουργίας της.

Στο τέταρτο κεφάλαιο περιγράφεται ο τρόπος λειτουργίας και παρουσιάζεται ένα εγχειρίδιο χρήσης της εφαρμογής.



## ΚΕΦΑΛΑΙΟ 1 - Βάση Δεδομένων. Διακομιστής MySql

### Εισαγωγή

Στο κεφάλαιο αυτό περιγράφεται ο διακομιστής MySql και οι λειτουργίες του. Περιγράφονται τα χαρακτηριστικά που έχει και τα πλεονεκτήματα του έναντι του ανταγωνισμού. Αναλύονται οι βάσεις και οι πίνακες που χρειάστηκαν για την εφαρμογή και παρουσιάζεται η διαδικασία και ανάλυση της κατασκευής τους.

### 1.1 Η γλώσσα SQL

Η γλώσσα SQL (Structured Query Language) είναι η πιο διαδεδομένη δομημένη γλώσσα ερωτήσεων και απαντήσεων που χρησιμοποιείται για την επικοινωνία χρήστη με διάφορες σχεσιακές βάσεις δεδομένων. Είναι βασισμένη σε σχεσιακή άλγεβρα και σε πλειάδες σχεσιακού λογισμού και στόχο έχει την εισαγωγή, ερώτηση, ανανέωση και διαγραφή μιας εγγραφής, την δημιουργία και διαγραφή σχήματος βάσης και τον έλεγχο πρόσβασης στα δεδομένα. Η SQL ήταν από τις πρώτες εμπορικές γλώσσες προγραμματισμού από το 1974. Δίνοντας την δυνατότητα τροποποίησης ή αναζήτησης δεδομένων με πολύπλοκα κριτήρια σε μεγάλες βάσεις δεδομένων την έκανε διάσημη και απαραίτητη πολύ γρήγορα σε πολύ μεγάλο κοινό.

### 1.2 MySql Server

Ο MySql είναι ένα πολύ δυνατό και γρήγορο σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων ανοιχτού κώδικα που από το 2008 λειτουργεί και σαν εξυπηρετητής. Το όνομα του προέρχεται από το όνομα της κόρης του συνιδρυτή Michael Widenius My. Το SQL προέρχεται από την γλώσσα SQL. Είναι ανοιχτού λογισμικού πρόγραμμα με άδεια GNU General Public License και πολλοί διάσημοι οργανισμοί και εταιρίες που δραστηριοποιούνται στο διαδίκτυο το χρησιμοποιούν. Όπως για παράδειγμα Facebook, Youtube, Flickr, Google, Wikipedia, Twitter.

Ο MySQL διακομιστής ελέγχει την πρόσβαση στα δεδομένα του χρήστη για να μπορούν να δουλεύουν ταυτόχρονα πολλοί χρήστες πάνω στην ίδια βάση δεδομένων, ότι μόνο πιστοποιημένοι χρήστες έχουν πρόσβαση σε αυτά καθώς και καλές επιδόσεις στον χρόνο απόκρισης, Αυτό καθιστά τον MySQL έναν πολυνηματικό διακομιστή που υποστηρίζει πολλαπλούς χρήστες. Χρησιμοποιεί την SQL σαν τυπική γλώσσα ερωτημάτων για τις βάσεις δεδομένων της παγκόσμια.

### 1.3 Πλεονεκτήματα της MySQL

Βασικοί αντίπαλοι στον τομέα των βάσεων δεδομένων της MySQL είναι οι βάσεις της Oracle , Microsoft SQL και PostgreSQL. Η MySQL έχει αρκετά πλεονεκτήματα όπως χαμηλό κόστος και εύκολη διαμόρφωση και εκμάθηση αλλά το βασικό της στοιχείο είναι πως αποτελεί κομμάτι ανοιχτού κώδικα με όλα τα πλεονεκτήματα που αυτό φέρνει.

- Απόδοση

Η MySQL χωρίς αμφιβολία είναι από τις πιο γρήγορες στον τομέα της. Πολλές από τις επίσημες δοκιμές δείχνουν πως είναι αρκετά πιο γρήγορο από τον ανταγωνισμό της.

- Χαμηλό Κόστος

Η MySQL διανέμεται δωρεάν με άδεια GNU General Public License ανοιχτού κώδικα ή με πολύ μικρό κόστος αν αποκτηθεί με εμπορική άδεια.

- Ευκολία Χρήσης

Όλες οι μοντέρνες βάσεις δεδομένων χρησιμοποιούν SQL. Η χρήση του μοντέλου διαχείρισης σχεσιακών βάσεων δεδομένων είναι πολύ εύκολο στην εκμάθηση αλλά και στην χρήση του.

- Φορητότητα

Η MySQL μπορεί να μεταφερθεί σε πολλά και διαφορετικά περιβάλλοντα τόσο σε Unix όσο και σε Microsoft Windows.

- Κώδικας πηγής

Ο κώδικας του MySQL είναι διαθέσιμος δωρεάν και μπορεί να τροποποιηθεί ανάλογα με τις ανάγκες και τις απαιτήσεις του χρήστη.

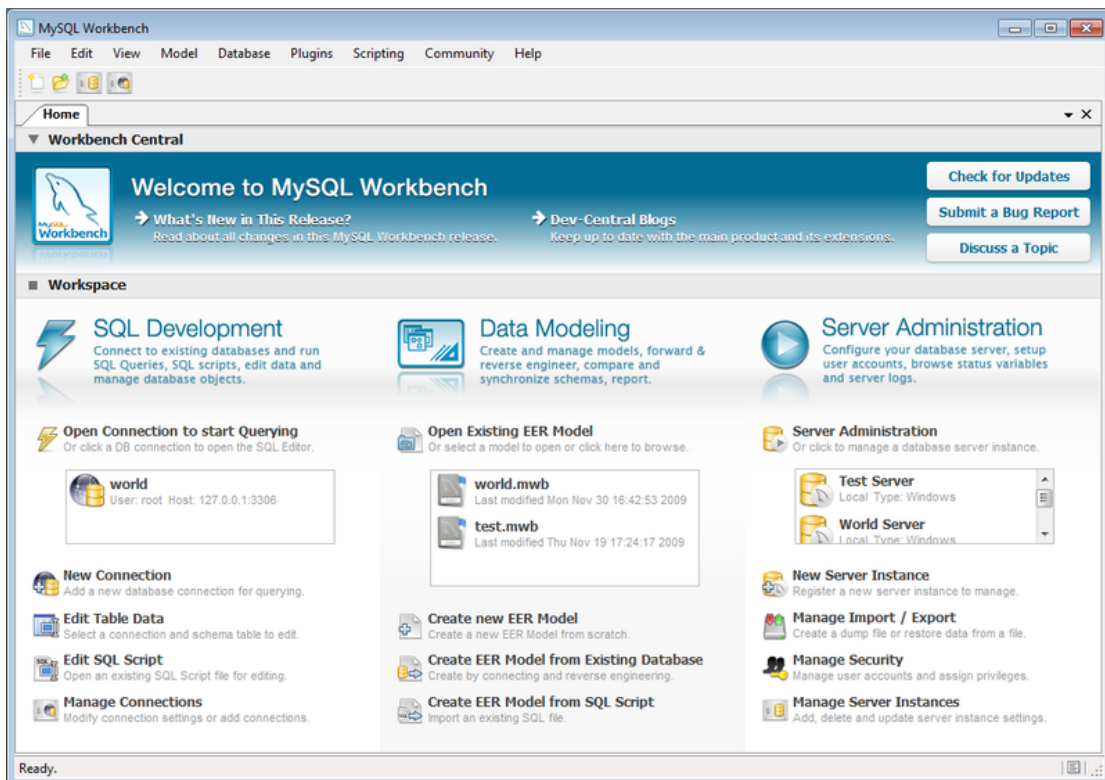
## 1.4 Βασικές εντολές

Παρακάτω παρουσιάζονται οι βασικές εντολές της SQL που χρησιμοποιούνται για την δημιουργία και χρήση ακόμα και της πιο απλής σχεσιακής βάσης δεδομένων.

- **Create:** Με την εντολή αυτή δημιουργούμε μια βάση δεδομένων ή έναν πίνακα μέσα σε κάποια βάση δεδομένων.
- **Drop:** Με την εντολή αυτή διαγράφουμε μια βάση δεδομένων.
- **Insert:** Με την εντολή αυτή εισάγουμε μια ή περισσότερες εγγραφές σε ένα πίνακα.
- **Update:** Με την εντολή αυτή ανανεώνουμε τα δεδομένα σε ένα σύνολο γραμμών ενός πίνακα.
- **Delete:** Με την εντολή αυτή διαγράφουμε ένα σύνολο εγγραφών από ένα πίνακα.
- **Select:** Πρόκειται για την πιο συχνά χρησιμοποιούμενη εντολή στην γλώσσα. Χρησιμοποιείτε για να κάνει κάποιο ερώτημα σε μια ή περισσότερες βάσεις και επιστρέφει δεδομένα ανάλογα με τα κριτήρια από μια ή περισσότερες βάσεις και πίνακες.
- **Grant:** Με την εντολή αυτή εξουσιοδοτούμε ένα χρήστη με δικαιώματα σε μια βάση δεδομένων.
- **Revoke:** Κάνει την ακριβώς αντίθετη πράξη με την Grant. Περιορίζει δικαιώματα σε κάποιο χρήστη για κάποια βάση δεδομένων.

## 1.5 Γραφικό Εργαλείο Διαχείρισης της MySql. «MySql Workbench»

Πρόκειται για ένα εργαλείο διαχείρισης βάσεων δεδομένων με γραφικό περιβάλλον το οποίο αναπτύχθηκε και διανέμεται δωρεάν από την Oracle. Το σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων MySql δεν κυκλοφορεί με κάποιο αντίστοιχο εργαλείο και οι χρήστες πρέπει να χρησιμοποιούν ένα εργαλείο σε μορφή κονσόλας εντολών.



Σχήμα 1 : Το εργαλείο MySql Workbench

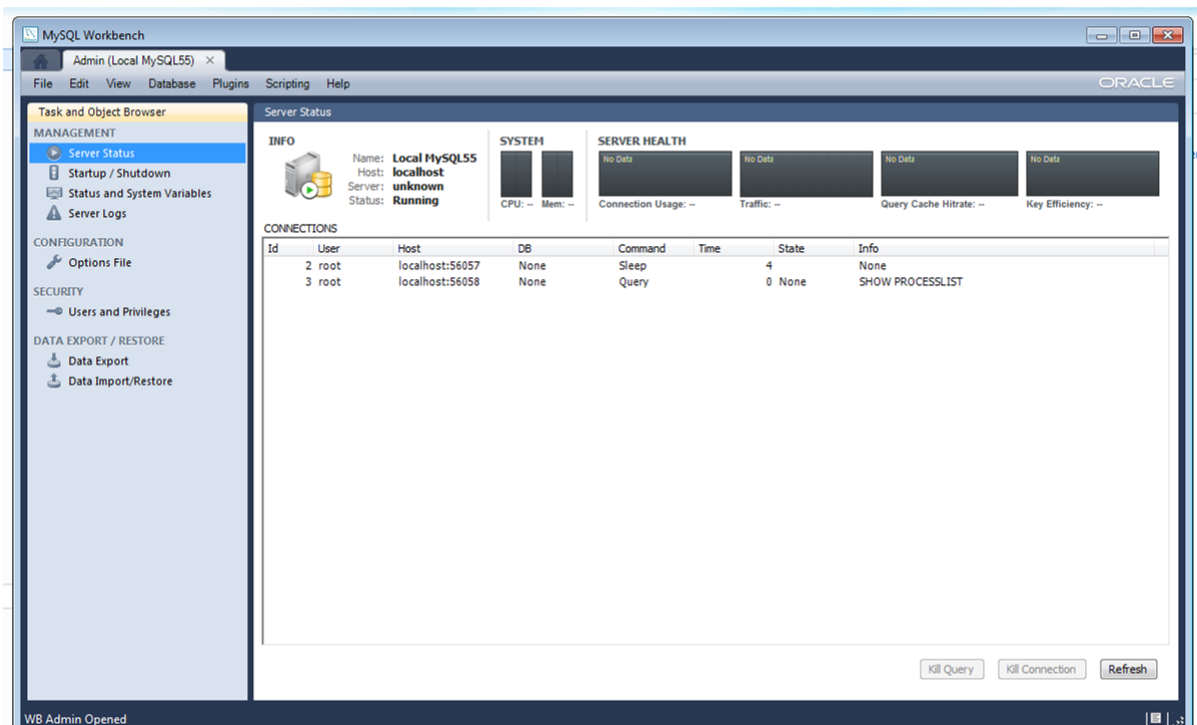
Βασικά χαρακτηριστικά του εργαλείου είναι :

- Γενικά:
  - Σύνδεση βάσης δεδομένων και διαχείριση περιστατικού
  - Χρήση οδηγού για δημιουργία εντολών
  - Υποστηρίζει διάφορα πρόσθετα
  - Έχει γραφτεί με Python και Lua

- Συντάκτης SQL:
  - Διαχειριστής αντικειμένων σχημάτων
  - Συντάκτης κειμένου και μαρκάρισμα σύνταξης
  - Πολλαπλά τροποποιήσιμα πακέτα αποτελεσμάτων
  - Έτοιμα πακέτα με SQL εντολές
  - SSH συνδέσεις
  - Υποστήριξη χαρακτήρων Unicode
  
- Μοντέλα δεδομένων:
  - Διαγράμματα ER
  - Drag n' Drop οπτικά μοντέλα
  - Συγχρονισμός σχημάτων
  - Εκτύπωση μοντέλων
  
- Διαχειριστής της Βάσης δεδομένων
  - Μπορεί να ξεκινά και να σταματά μια βάση δεδομένων
  - Ρύθμιση παραμέτρων μιας βάσης δεδομένων
  - Διαχείριση λογαριασμού μιας βάσης δεδομένων
  - Επεξεργασία του εγγράφου καταγραφής γεγονότων
  - Εξαγωγή και εισαγωγή δεδομένων
  
- Φορητότητα
  - Κάθε βάση δεδομένων που υποστηρίζει ODBC
  - Υποστηρίζει τις βάσεις : Microsoft SQL Server, PostgreSQL, Sybase ASE

## 1.6 Ανάλυση Σχεδιασμού Βάσεων Δεδομένων Εφαρμογής Διαχείριση Θερμοκηπίου

Για τις ανάγκες της εφαρμογής διαχείρισης θερμοκηπίου έχουν δημιουργηθεί μια βάση δεδομένων με το όνομα *stoixeia* και δυο πίνακες με ονόματα *plants* και *customers*. Παρακάτω παρουσιάζονται τα βήματα για την δημιουργία των πινάκων, τα χαρακτηριστικά του κάθε πίνακα και του εξυπηρετητή μας.



Σχήμα 2 : Εκκίνηση του εξυπηρετητή

Για να μπορέσουμε να συνδεθούμε στην βάση μας και να δημιουργήσουμε τους πίνακες πρέπει αρχικά να ξεκινήσουμε τον εξυπηρετητή μας. Αυτό γίνεται εύκολα με την εισαγωγή του ονόματος και του κωδικού που ορίσαμε στην αρχή της εγκατάστασής του.

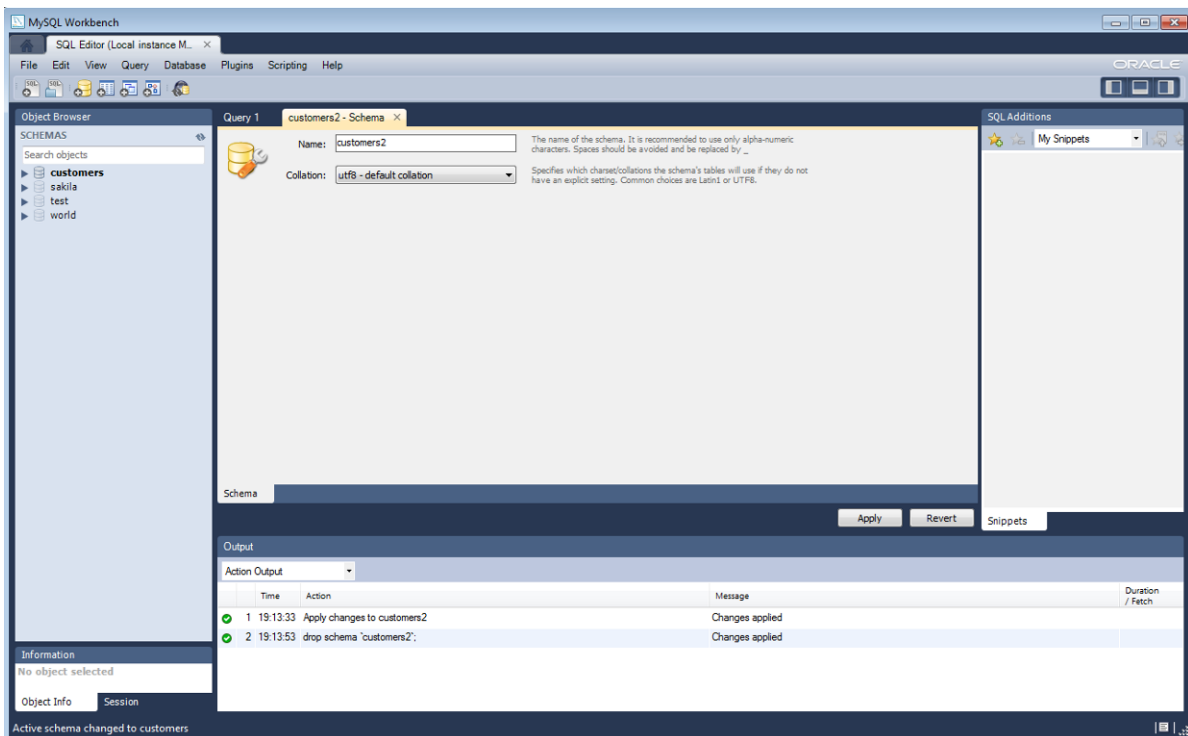
Στην συνέχεια με το ίδιο εργαλείο συνδεόμαστε στον εξυπηρετητή για να φτιάξουμε την βάση μας και τους πίνακες μας.



Με την επιλογή δημιουργίας νέου σχήματος στον συνδεδεμένο εξυπηρετητή δημιουργούμε την βάση μας με όποιο όνομα θέλουμε και ιδιότητες. Σημαντικό στοιχείο σε αυτό το βήμα είναι η κωδικοποίηση των δεδομένων που θα επιλέξουμε για τα δεδομένα μας. Επειδή τα ελληνικά αντιμετωπίζονται διαφορετικά από τους λατινικούς χαρακτήρες αν δεν επιλέξουμε σωστή κωδικοποίηση τα αποτελέσματα δεν θα είναι κατανοητά από ανθρώπους. Έτσι εδώ επιλέγουμε κωδικοποίηση utf8-default collection. Με αυτήν την κωδικοποίηση ουσιαστικά συμπεριφέρεται στα δεδομένα σαν τα γράμματα να είναι φωτογραφίες – δηλαδή εμφανίζει ότι βλέπει.

Ο κώδικας πίσω από το γραφικό περιβάλλον είναι ο εξής:

```
CREATE SCHEMA `customers` DEFAULT CHARACTER SET utf8 ;
```

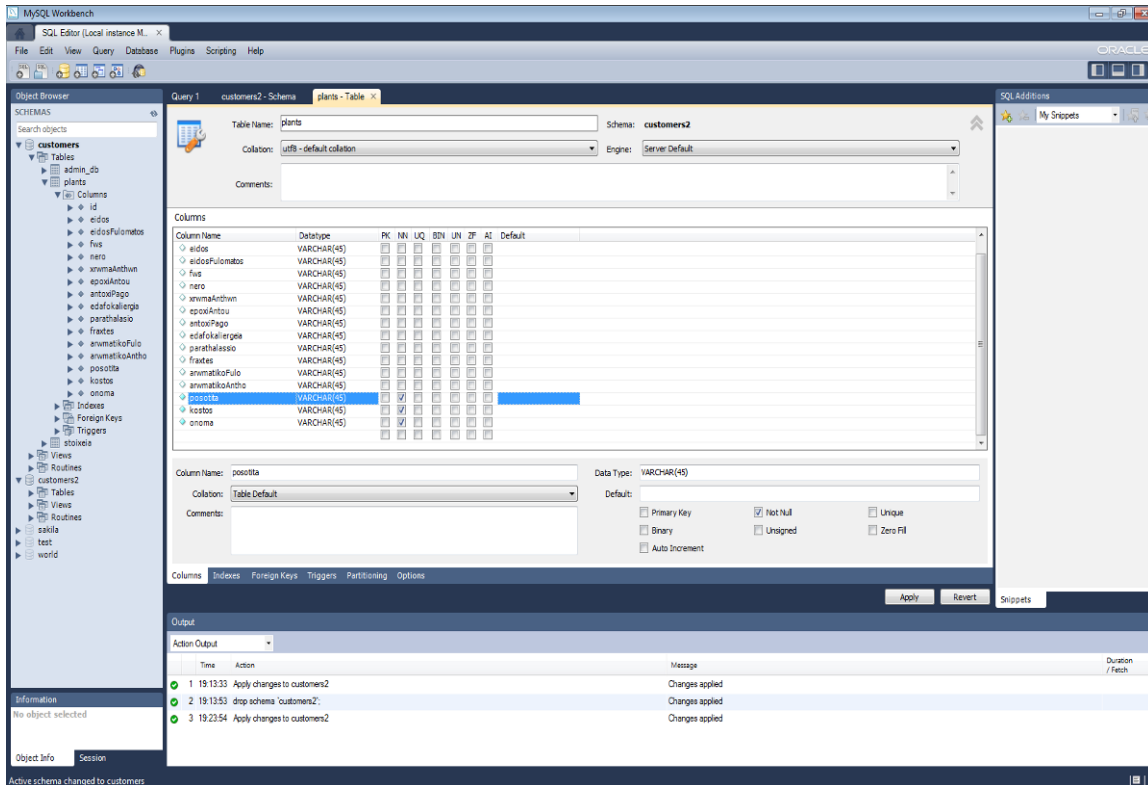


**Σχήμα 3 : Δημιουργία της Βάσης Δεδομένων**

Στην συνέχεια θα δημιουργήσουμε τους πίνακες που χρειαζόμαστε για τα δεδομένα μας. Και στο βήμα αυτό σημαντικό είναι να προσέξουμε την κωδικοποίηση των δεδομένων η οποία πρέπει να είναι παρόμοια με αυτήν της βάσης. Ο κώδικας πίσω από το εργαλείο είναι ο εξής:

```
CREATE TABLE `customers`.`plants` (  
  `id` INT NOT NULL AUTO_INCREMENT ,  
  `eidos` VARCHAR(45) NULL ,  
  `eidosFulomatos` VARCHAR(45) NULL ,  
  `fws` VARCHAR(45) NULL ,  
  `nero` VARCHAR(45) NULL ,  
  `xrwmaAnthwn` VARCHAR(45) NULL ,  
  `epoxiAntou` VARCHAR(45) NULL ,  
  `antoxiPago` VARCHAR(45) NULL ,  
  `edafokaliergeia` VARCHAR(45) NULL ,  
  `parathalassio` VARCHAR(45) NULL ,  
  `fraxtes` VARCHAR(45) NULL ,  
  `arwmatikoFulo` VARCHAR(45) NULL ,  
  `arwmatikoAntho` VARCHAR(45) NULL ,  
  `posotita` int unsigned NOT NULL ,  
  `kostos` int unsigned NOT NULL ,  
  `onoma` VARCHAR(45) NOT NULL ,  
  PRIMARY KEY (`id`))  
DEFAULT CHARACTER SET = utf8;
```

Όπως παρατηρούμε επιλέγουμε σαν πρωτεύον κλειδί το id. Αυτό ουσιαστικά είναι και ο κωδικός του φυτού στην βάση. Αυξάνεται μόνο του, δεν μπορεί να είναι κενό και δίνεται αυτόματα στο κάθε φυτό από το σύστημα. Τα υπόλοιπα πεδία είναι αλφαριθμητικά εκτός από την ποσότητα και το κόστος του κάθε προϊόντος.

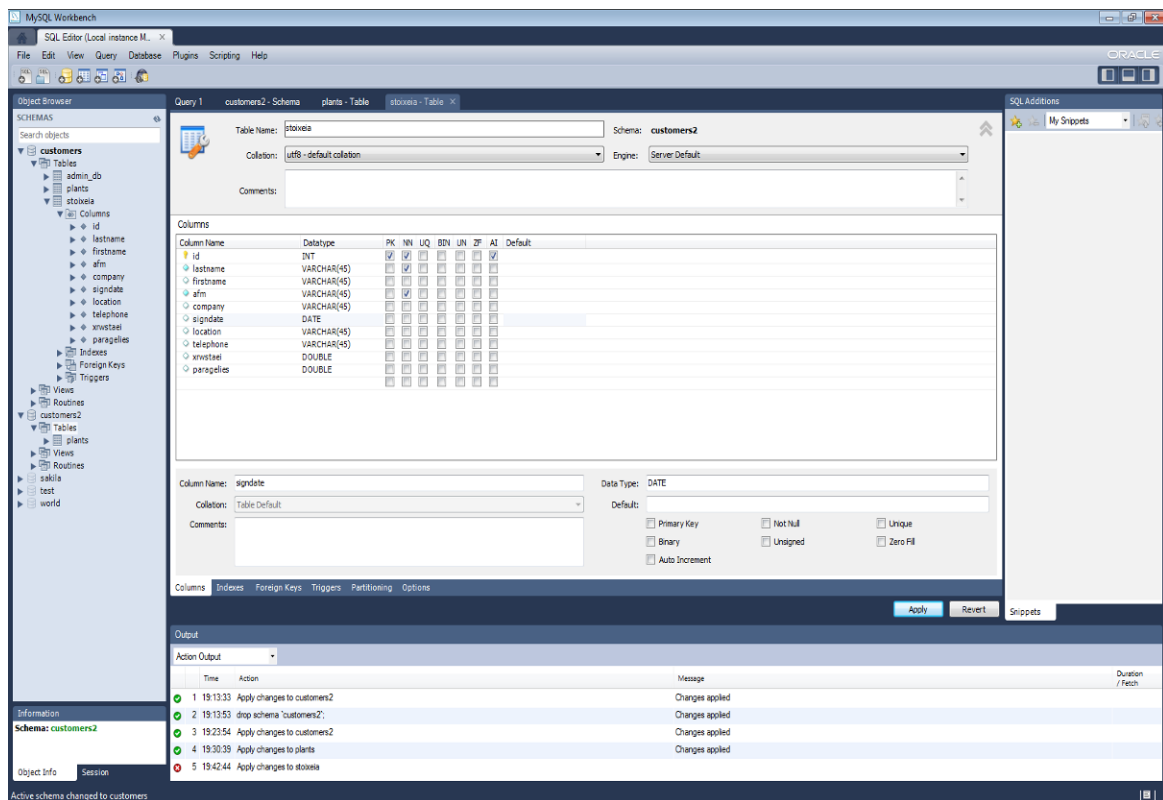


Σχήμα 4 : Δημιουργία πίνακα plants

Για την δημιουργία του πίνακα στοιχεία ο οποίος διατηρεί τα στοιχεία των πελατών ισχύουν ότι και παραπάνω. Πρέπει να δηλώσουμε για κωδικοποίηση ότι και στην βάση δεδομένων. Το πρωτεύον κλειδί είναι και πάλι το id το οποίο είναι και ο κωδικός του κάθε πελάτη, αυξάνεται αυτόματα και δεν μπορεί να είναι κενό. Τα πεδία είναι αλφαριθμητικά στο σύνολο τους εκτός από τις παραγγελίες και το υπόλοιπο. Το πεδίο παραγγελίες κρατάει το σύνολο των προϊόντων το οποίο έχει παραγγέλλει κάποιος πελάτης προσθέτοντας τα συνολικά τεμάχια από το κάθε προϊόν. Στο πεδίο υπόλοιπο προστίθεται το άθροισμα του ποσού από το σύνολο των τεμαχίων επί την τιμή τους. Η διαδικασία αυτή γίνεται αυτόματα από το σύστημα. Ο κώδικας δημιουργίας του πίνακα πίσω από το εργαλείο είναι ο εξής:

```
CREATE TABLE `customers`.`στοιχεια` (
  `id` INT NOT NULL AUTO_INCREMENT ,
  `lastname` VARCHAR(45) NOT NULL ,
  `firstname` VARCHAR(45) NULL ,
  `afm` VARCHAR(45) NOT NULL ,
```

```
`company` VARCHAR(45) NULL ,  
`signdate` DATE NULL ,  
`location` VARCHAR(45) NULL ,  
`telephone` VARCHAR(45) NULL ,  
`xrwstaei` DOUBLE NULL ,  
`paragelies` DOUBLE NULL ,  
PRIMARY KEY (`id` )  
DEFAULT CHARACTER SET = utf8;
```



Σχήμα 5 : Δημιουργία του πίνακα stoixeia

## Επίλογος

Στο κεφάλαιο αυτό αναλύσαμε γενικά τις τεχνολογίες βάσεων δεδομένων. Παραθέσαμε πληροφορίες για τον MySql διακομιστή και τον λόγο για τον οποίο τον επιλέξαμε. Στην συνέχεια παραθέσαμε το οπτικό εργαλείο δημιουργίας βάσεων δεδομένων και πινάκων, μελετήσαμε τους πίνακες που χρειάστηκαν για την εφαρμογή μας και τους δημιουργήσαμε. Στο επόμενο κεφάλαιο γίνεται λόγος για την γλώσσα προγραμματισμού

Πτυχιακή εργασία της φοιτήτριας Βεγερτζή Κοκκώνης

που χρειάστηκε για να κατασκευάσουμε την εφαρμογή και τις τεχνολογίες που χρησιμοποιήσαμε για να συνδέσουμε την εφαρμογή με την βάση δεδομένων.

## ΚΕΦΑΛΑΙΟ 2 - Προγραμματιστικό περιβάλλον Visual Studio 2008

### Εισαγωγή

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε το περιβάλλον προγραμματισμού της Microsoft Visual Studio 2008 και η γλώσσα προγραμματισμού Visual Basic. Στο κεφάλαιο αυτό αναλύεται η αρχιτεκτονική και ο τρόπος λειτουργίας του Microsoft Visual Studio 2008, της γλώσσας προγραμματισμού Visual Basic και γενικά του πακέτου .NET Framework της Microsoft. Τέλος παραθέεται μια ανάλυση της λειτουργίας ADO για την σύνδεση της εφαρμογής με την βάση δεδομένων.

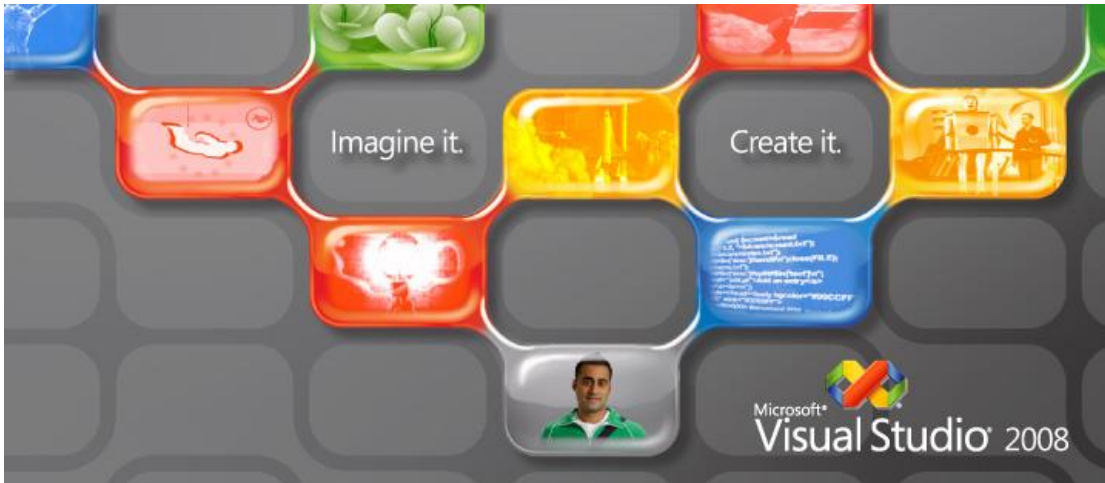
### 2.1 Microsoft Visual Studio 2008

Το Microsoft Visual Studio 2008 ανήκει στην οικογένεια των ολοκληρωμένων περιβαλλόντων ανάπτυξης λογισμικού το οποίο προσφέρεται από την Microsoft (Integrated Development Environment - IDE). Μέσα από το περιβάλλον αυτό μπορούν να αναπτυχθούν εφαρμογές για διάφορες κατηγορίες που υποστηρίζονται από την Microsoft. Εφαρμογές κονσόλας, γραφικών εφαρμογών, ιστοσελίδων και Web υπηρεσιών οι οποίες είναι συμβατές με την οικογένεια Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework και Microsoft Silverlight.

Το περιβάλλον ανάπτυξης λογισμικού Visual Studio παρέχει στον προγραμματιστή τον συντάκτη κώδικα ο οποίος υποστηρίζει IntelliSense. Ο ενσωματωμένος διορθωτής (debugger) λειτουργεί και σαν διορθωτής επιπέδου πηγής (source-level debugger) αλλά και σαν διορθωτής επιπέδου μηχανής (machine-level debugger). Επίσης στο περιβάλλον υπάρχουν ενσωματωμένα εργαλεία για σχεδιασμό των γραφικών εφαρμογών (το γνωστό GUI – Graphical User Interface), ένας σχεδιαστής Ιστού, σχεδιαστής κλάσεων καθώς και τον σχεδιαστή σχημάτων για τις βάσεις δεδομένων.

Το Visual Studio υποστηρίζει τις γλώσσες προγραμματισμού με την έννοια των γλωσσικών υπηρεσιών, οι οποίες επιτρέπουν σε οποιαδήποτε γλώσσα προγραμματισμού να υποστηριχθεί σε διαφορετικό βαθμό. Και από τον συντάκτη αλλά και από τον διορθωτή με μια προϋπόθεση, ότι υποστηρίζει γλωσσικά μια συγκεκριμένη γλωσσική υπηρεσία. Στις ενσωματωμένες γλώσσες προγραμματισμού το Visual Studio περιλαμβάνει τις C και C++ (μέσω της Visual C++), VB.NET (μέσω της Visual Basic .NET) και η C# μέσω της Visual

C#. Προσφέρεται η δυνατότητα για υποστήριξη και άλλων γλωσσών όπως είναι οι F#, Python , Ruby οι οποίες πρέπει να κατεβούν και να γίνουν εγκατάσταση χωριστά σαν ξεχωριστές γλωσσικές υπηρεσίες. Υποστηρίζονται επίσης οι γλώσσες για διαδικτυακές εφαρμογές XML/XSLT, HTML/XHTML, JavaScript και CSS.



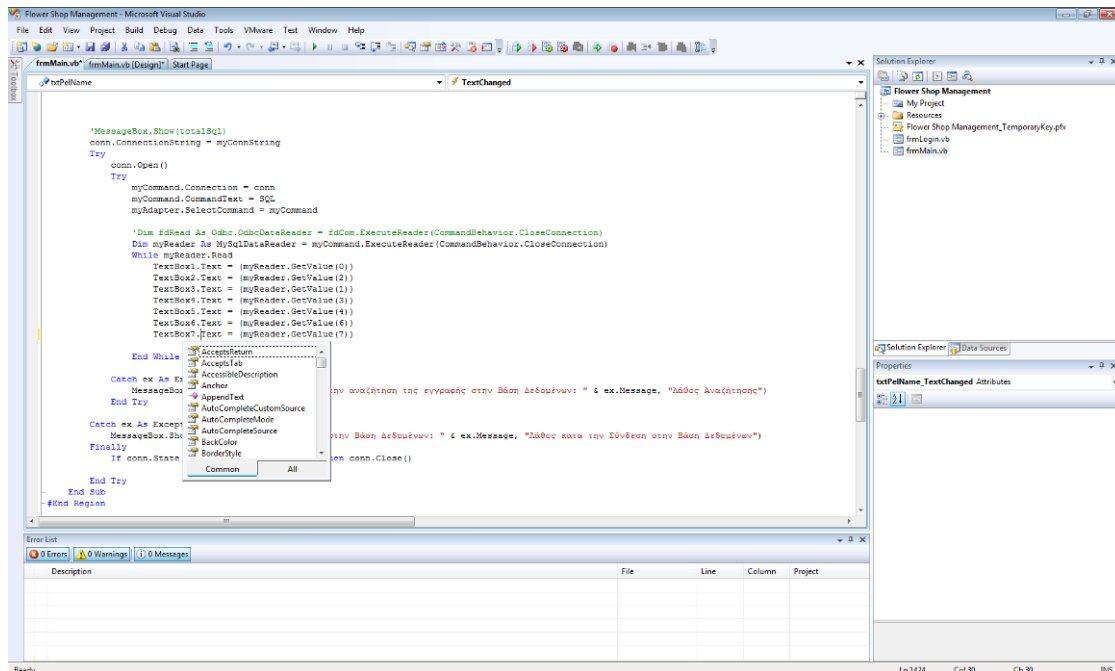
Σχήμα 6 : Visual Studio 2008

## 2.2 Χαρακτηριστικά Γνωρίσματα

### 2.2.1 Συντάκτης κώδικα (Compiler)

Όπως και κάθε εργαλείο ανάπτυξης λογισμικού έτσι και το Visual Studio περιλαμβάνει έναν συντάκτη κώδικα οποίος υποστηρίζει τον τονισμό της σύνταξης αλλά και την ολοκλήρωση κώδικα. Αυτό επιτυγχάνεται με την τεχνολογία IntelliSense, όχι μόνο για τις μεταβλητές, τις λειτουργίες και τις μεθόδους αλλά και για τα γλωσσικά κατασκευάσματα όπως τους βρόγχους και τις ερωτήσεις. Η λειτουργία IntelliSense υποστηρίζεται μόνο για τις συμπεριλαμβανόμενες γλώσσες καθώς επίσης και για τις τεχνολογίες ανάπτυξης διαδικτυακών εφαρμογών και ιστοχώρων, XML, Cascading Style Sheets (CSS) και JavaScript. Οι προτάσεις του συντάκτη για αυτόματη συμπλήρωση παρουσιάζονται με ένα παράθυρο καταλόγου, πάνω στον ίδιο συντάκτη το οποίο γίνεται ημιδιάφανο για να διευκολύνει τον προγραμματιστή να παρακολουθεί τον κώδικα.

Ένα από τα χαρακτηριστικά του Microsoft Visual Studio είναι η σύνταξη υποβάθρου (background compilation) η οποία υποστηρίζει. Καθώς ο προγραμματιστής προσθέτει γραμμές κώδικα στην εφαρμογή το Microsoft Visual Studio ελέγχει την ορθή σύνταξη της γλώσσας προγραμματισμού στο υπόβαθρο. Τα λάθη σύνταξης υπογραμμίζονται με κόκκινη κυματιστή γραμμή ενώ οι προειδοποιήσεις με μια πράσινη κυματιστή γραμμή. Η σύνταξη υποβάθρου δεν παράγει το τελικό εκτελέσιμο κομμάτι κώδικα, δεδομένου ότι χρησιμοποιεί έναν διαφορετικό μεταγλωττιστή από αυτόν που χρησιμοποιείται για την παραγωγή του τελικού εκτελέσιμου κώδικα. Η λειτουργία της σύνταξης υποβάθρου είχε εισαχθεί αρχικά στην Microsoft Visual Basic αλλά έχει πλέον επεκταθεί σε όλες τις συμπεριλαμβανόμενες γλώσσες προγραμματισμού του περιβάλλοντος Microsoft Visual Studio.



Σχήμα 7 : Στιγμιότυπο χρήσης τεχνολογίας IntelliSense του Microsoft Visual Studio 2008

## 2.2.2 Διορθωτής (Debugger)

Το Microsoft Visual Studio περιλαμβάνει έναν διορθωτή (debugger) ο οποίος χρησιμοποιείται και σαν διορθωτής πηγής (source level debugger) και ως διορθωτής μηχανής (machine-level debugger). Ο διορθωτής χρησιμοποιείται για την διόρθωση εφαρμογών γραμμένες από όλες τις γλώσσες που υποστηρίζει το περιβάλλον του Visual



Studio. Επιπλέον μπορεί να συνδέεται με τις τρέχουσες διεργασίες, να τις επιτηρεί όπως και να τις επιδιορθώνει.

Ο διορθωτής επιτρέπει την εισαγωγή σημείων παύσης (breakpoints) τα οποία επιτρέπουν στο πρόγραμμα να σταματήσει προσωρινά στην θέση που θα ορίσει ο προγραμματιστής κάνοντας την διαδικασία της διόρθωσης της εφαρμογής πιο εύκολη. Κατά την παύση ο προγραμματιστής μπορεί να ελέγξει μια πληθώρα επιλογών όπως τιμές και μεταβλητές καθώς η εκτέλεση προχωράει βήμα- βήμα. Τα σημεία παύσης μπορούν να είναι και υπό όρους που σημαίνει ότι προκαλούνται αφού ελέγθη ο όρος και προκύψει ότι ικανοποιείται. Ο διορθωτής τέλος περιλαμβάνει την λειτουργία της τροποποίησης και συνέχειας (Edit and Continue) το οποίο επιτρέπει στον προγραμματιστή να τροποποιεί τον κώδικα καθώς αυτός εκτελείται στην διαδικασία της διόρθωσης.

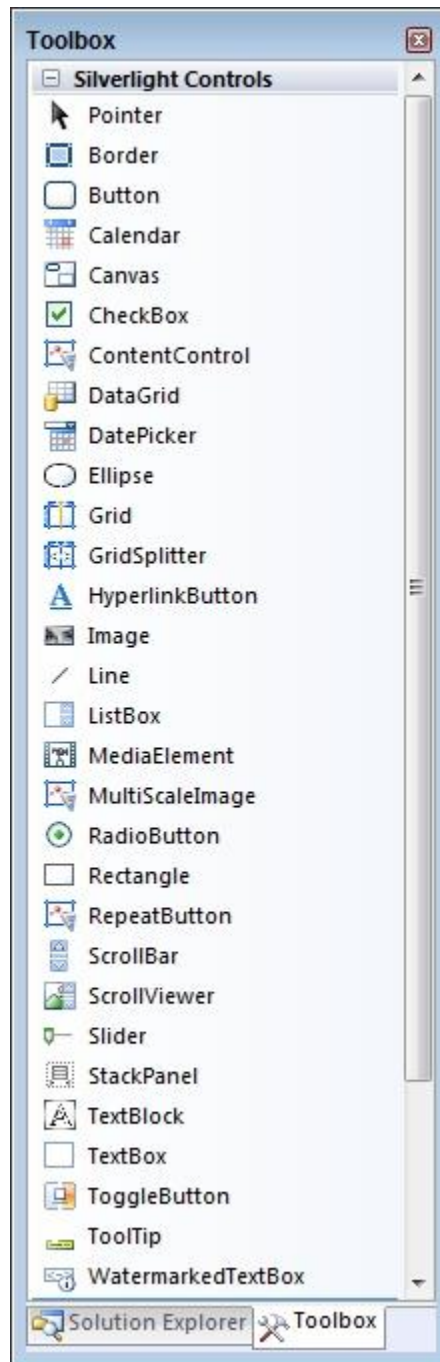
### **2.2.3 Σχεδιαστής εφαρμογής (Visual Designers)**

Το Microsoft Visual Studio περιλαμβάνει έναν πλήθος από οπτικούς σχεδιαστές οι οποίοι βοηθούν τον προγραμματιστή στην ανάπτυξη εφαρμογών. Αυτά τα εργαλεία περιλαμβάνουν:

- Σχεδιαστής WPF στο Microsoft Visual Studio 2008
- Visual Studio σχεδιαστής Ιστού (Web Designer) Code Editor View
- Microsoft Visual Studio 2005 Class Designer View
- Microsoft Visual Studio 2005 Designer View

Σχεδιαστής φορμών των Windows:

Ο σχεδιαστής φορμών Windows (WFD – Windows Form Designer ) χρησιμοποιείται από τον προγραμματιστή για την δημιουργία γραφικών εφαρμογών. Περιλαμβάνει μια παλέτα που αποτελείται από αντικείμενα του User Interface (UI) και διάφορους άλλους ελεγκτές όπως κουμπιά, κουμπιά επιλογής, ετικετών και άλλων. Όλα αυτά μπορούν να προστεθούν στην φόρμα με απλό Drag n' Drop. Πολλά από τα στοιχεία ελέγχου όπως τα textboxes ή τα DataGridViews μπορούν να παίρνουν στοιχεία από διάφορες πηγές όπως αρχεία εγγράφων ή βάσεις δεδομένων.



Σχήμα 8 : Παλέτα εργαλείων Visual Basic

Σχεδιαστής WPF:

Ο σχεδιαστής αυτός εισήχθη με την έκδοση Microsoft Visual Studio 2008. Όπως και στον σχεδιαστή φορμών των Windows έτσι και εδώ υποστηρίζεται η μεταφορά αντικειμένων στο User Interface (UI) με την μέθοδο του Drag n' Drop από μια παλέτα με εργαλεία. Χρησιμοποιείται για την δημιουργία και πάλι γραφικών εφαρμογών απευθυνόμενος στο ίδρυμα παρουσίασης των Windows (Windows Presentation Foundation -WPF).

Υποστηρίζει όλη την λειτουργία WPF συμπεριλαμβανομένης και της αυτόματης διαχείρισης σχεδιαγράμματος. Παράγει τον κώδικα XAML για το User Interface ενώ το παραγόμενο αρχείο είναι απόλυτα συμβατό με τον σχεδιαστή έκφρασης της Microsoft (Microsoft Expression Design). Ο κώδικας του XAML συνδέεται με τον κώδικα αυτόν μέσω του προτύπου Code-Behind.

Σχεδιαστής Ιστού:

Το περιβάλλον σχεδιασμού λογισμικού περιλαμβάνει επίσης συντάκτη και τον αντίστοιχο σχεδιαστή ιστοσελίδας. Αυτός με την σειρά του υποστηρίζει την μεταφορά αντικειμένων ελέγχου από μια παλέτα με εργαλεία στην φόρμα σχεδιασμού User Interface με την μέθοδο και πάλι του Drag n' Drop. Αυτός ο σχεδιαστής χρησιμοποιείται για την ανάπτυξη ASP.NET εφαρμογών και υποστηρίζει τις τεχνολογίες HTML5, CSS και JavaScript. Χρησιμοποιεί ένα πρότυπο Code-Behind για να συνδέσει την φόρμα με τον κώδικα ενώ από το Visual Studio 2008 και μετά η μηχανή σχεδιασμού που χρησιμοποιείται από τον σχεδιαστή Ιστού μοιράζεται και το Microsoft Web Expression.

Σχεδιαστής κλάσεων:

Ο σχεδιαστής κλάσεων χρησιμοποιείται από τον συντάκτη και τροποποιεί τις κλάσεις (συμπεριλαμβανομένων και των μελών τους και της πρόσβασης τους) χρησιμοποιώντας τη διαμόρφωση UML. Ο σχεδιαστής κατηγορίας είναι υπεύθυνος για να παράγει τις περιλήψεις κώδικα C# και VB.NET για τις κατηγορίες και τις μεθόδους τους. Μπορεί επίσης να παράγει και τα διαγράμματα κλάσεων από τις κλάσεις που έχει ο προγραμματιστής.

Σχεδιαστής στοιχείων:

Με τον σχεδιαστή στοιχείων μπορούν να τροποποιηθούν γραφικά τα διαγράμματα των βάσεων δεδομένων (Data Base Schemas) συμπεριλαμβανομένων των πινάκων της βάσης δεδομένων όπως και τα πρωτεύοντα και δευτερεύοντα κλειδιά και περιορισμοί.

#### **2.2.4 Λοιπά Εργαλεία**

- Ανοιχτή μηχανή αναζήτησης ετικετών

Η ανοιχτή μηχανή αναζήτησης ετικετών επιτρέπει την αναζήτηση όλων των ανοιχτών ετικετών του προγράμματος και επιτρέπει την εναλλαγή μεταξύ αυτών με την χρήση των πλήκτρων CTRL+TAB.

- Συντάκτης ιδιοτήτων

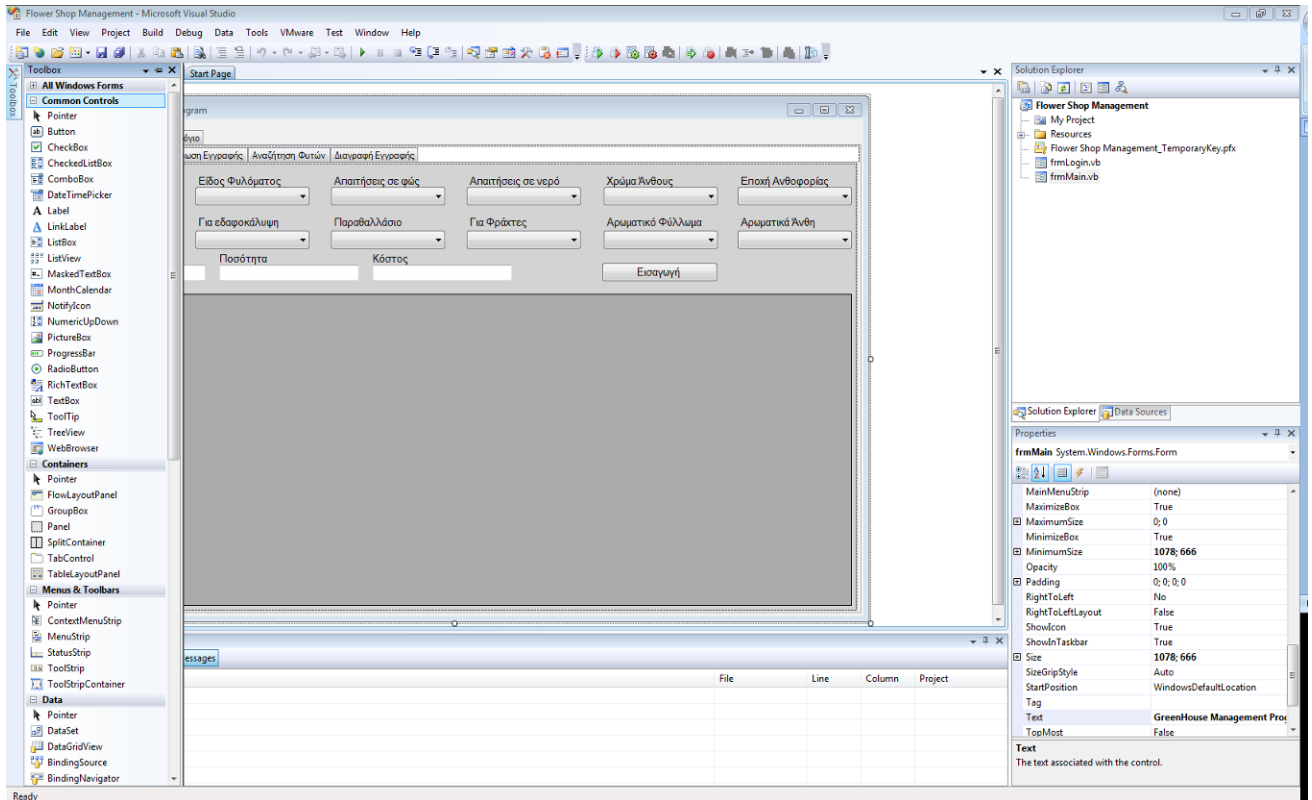
Το εργαλείο συντάκτης ιδιοτήτων δημιουργεί μια λίστα με όλες τις διαθέσιμες ιδιότητες των αντικειμένων, των συμπεριλαμβανομένων κλάσεων, των αντικειμένων μιας φόρμας, των ιστοσελίδων και των άλλων στοιχείων της εφαρμογής.

- Εξερευνητής λύσης

Με τον όρο εξερευνητή λύσης αναφερόμαστε στο πλαίσιο το οποίο διατηρεί το σύνολο των αρχείων και των πόρων για την υλοποίηση μιας εφαρμογής. Τα αρχεία σε μια λύση τακτοποιούνται ιεραρχικά και απεικονίζει την οργάνωση στο σύστημα αρχείων. Ο εξερευνητής χρησιμοποιείται για να διαχειριστεί και να ταξινομή τα αρχεία μέσα στην λύση.

- Εξερευνητής κεντρικών υπολογιστών

Με το εργαλείο εξερεύνησης κεντρικών υπολογιστών μπορεί ο προγραμματιστής να διαχειριστεί τις συνδέσεις βάσεων δεδομένων με έναν υπολογιστή που παρέχει πρόσβαση. Μπορεί να χρησιμοποιηθεί επίσης και για την ταξινόμηση τρεχουσών υπηρεσιών των Windows, τους μετρητές απόδοσης, τα αρχεία καταγραφής γεγονότων καθώς και διάφορα άλλα μηνύματα της εφαρμογής.



Σχήμα 9 : Visual Studio 2008.Παρουσίαση Βασικών Εργαλείων

## 2.3 Πρόσθετες γλώσσες προγραμματισμού

### 2.3.1 Microsoft Visual C++

Με την Microsoft Visual C++ γίνεται η προσθήκη του μεταγλωττιστή της Microsoft C και C++ στο Visual Studio IDE καθώς επίσης και άλλων εργαλείων και γλωσσικών υπηρεσιών. Μπορεί να χρησιμοποιηθεί για την δημιουργία προγραμμάτων C και C++ . Σαν C ακολουθεί το τα πρότυπα του ISO C με τα κομμάτια προδιαγραφής C 99 καθώς επίσης και με συγκεκριμένες προσθήκες υπό μορφή βιβλιοθηκών. Σαν C++ ακολουθεί την προδιαγραφή της ANSI C++ μαζί με μερικά χαρακτηριστικά της C++ 0x. Η Visual C++ μπορεί να χρησιμοποιήσει τον σχεδιαστή φορμών Windows για την κατασκευή γραφικών εφαρμογών με User Interface. Επίσης υποστηρίζει την χρήση εγγενών λειτουργιών που είναι λειτουργίες που αναγνωρίζονται από τον ίδιο τον μεταγλωττιστή και που δεν εφαρμόζονται ως βιβλιοθήκες. Οι εγγενείς λειτουργίες χρησιμοποιούνται για να παρουσιάσουν το σύνολο οδηγιών SSE των σύγχρονων κεντρικών μονάδων επεξεργασίας.

### **2.3.2 Microsoft Visual C#**

Η C # (προφέρεται "C-Sharp") είναι μια γλώσσα προγραμματισμού που έχει σχεδιαστεί για τη δημιουργία μια ποικιλίας από εφαρμογές που τρέχουν στο πλαίσιο .NET . Η C # είναι απλή, ισχυρή, ασφαλή γλώσσα , και αντικειμενοστραφής. Οι πολλές καινοτομίες της, επιτρέπουν στη C# ταχεία ανάπτυξη εφαρμογών διατηρώντας την εκφραστικότητα και την κομψότητα των γλωσσών C-style. Η Visual C# είναι μια υλοποίηση της γλώσσας C# από τη Microsoft. Το Visual Studio υποστηρίζει τη Visual C# με ένα πλήρες πρόγραμμα επεξεργασίας κώδικα, διορθωτή, πρότυπα του έργου, σχεδιαστές, κωδικοποιημένους οδηγούς, ένα ισχυρό και εύκολο στη χρήση πρόγραμμα εντοπισμού σφαλμάτων, και άλλα εργαλεία. Το .NET είναι μια κλάση βιβλιοθήκης που παρέχει πρόσβαση σε πολλές υπηρεσίες του λειτουργικού συστήματος και άλλες χρήσιμες , καλά σχεδιασμένες κλάσεις που επιταχύνουν τον κύκλο ανάπτυξης σημαντικά. Η σύνταξη της C# είναι πολύ εκφραστική, αλλά είναι επίσης απλή και εύκολη να τη μάθει ένας προγραμματιστής.

### **2.3.3 Microsoft Visual Basic**

Η γλώσσα προγραμματισμού Visual Basic είναι η προσθήκη της γλώσσας VB .NET και των σχετικών εργαλείων και υπηρεσιών στο περιβάλλον ανάπτυξης εφαρμογών της Microsoft. Εισήχθη σαν πρώτη φορά με το Visual Studio .NET το 2002 και θεωρείται από την Microsoft σαν την γρήγορη εφαρμογή ανάπτυξης εφαρμογών(Rapid Application Development - RAD). Η Visual Basic μπορεί να χρησιμοποιηθεί τόσο για εφαρμογές κονσόλας (command prompt) όσο και για εφαρμογές με γραφικό περιβάλλον (User Interface). Όπως και η Visual C++ και η Visual C# έτσι και η Visual Basic χρησιμοποιεί τον σχεδιαστή φορμών, σχεδιαστή κλάσεων και στοιχείων. Όπως και η Visual C# ο μεταγλωττιστής της γλώσσας VB.NET είναι διαθέσιμος ως μέρος του .Net Framework.

### **2.3.4 Microsoft Visual Web Developer**

Ο Microsoft Visual Web Developer χρησιμοποιείται για την ανάπτυξη εφαρμογών και υπηρεσιών Ιστού (Web Applications and Web services) με την χρήση της γλώσσας προγραμματισμού ASP.NET. Μπορούν να χρησιμοποιηθούν σαν γλώσσες

προγραμματισμού και οι γλώσσες Visual Basic και Visual C#. Τέλος ο Microsoft Visual Web Developer μπορεί να σχεδιάσει γραφικά τα σχεδιαγράμματα της ιστοσελίδας.

## 2.4 Microsoft Visual Basic

Η Visual Basic είναι μια γλώσσα προγραμματισμού τρίτης γενιάς που ανήκει στην κατηγορία οδηγούμενη από συμβάντα(event driven) και έχει ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) από την Microsoft για το μοντέλο προγραμματισμού Component Object Model –COM η οποία πρωτοεμφανίστηκε το 1991. Η Visual Basic είναι σχεδιασμένη να μαθαίνεται αλλά και να χρησιμοποιείται εύκολα αφού ανήκει στην οικογένεια BASIC και έχει γραφικό περιβάλλον ανάπτυξης της φόρμας. Έχει παρόμοια χαρακτηριστικά με την BASIC και εμπλέκει την τεχνολογία άμεσης ανάπτυξης εφαρμογών (RAD) μέσω του γραφικού περιβάλλοντος της. Παρέχει πρόσβαση σε βάσεις δεδομένων μέσω των αντικειμένων Data Access Objects, Remote Data Objects, ActiveX Data Objects και δημιουργίας ActiveX αντικειμένων και κουμπιών. Η γλώσσα VBA είναι αποτελεσματικά παρόμοια με την Visual Basic αλλά με την διαφορά ότι είναι δεμένη με το περιβάλλον Microsoft Office περιορίζοντας την να λειτουργεί αυστηρά μέσα σε αυτήν και μόνο. Η γλώσσα VBScript είναι μια γλώσσα scripting και είναι υποσύνολο της Visual Basic. Ένας προγραμματιστής μπορεί να υλοποιήσει μια εφαρμογή χρησιμοποιώντας τα αντικείμενα που παρέχει η Visual Basic και μόνο με αυτά. Παρέχεται όμως η δυνατότητα χρήσης και του Windows API(Application Programming Interface) απλά με την δήλωση κάποιων εξωτερικών συναρτήσεων. Αν και αρχικά η γλώσσα είχε αρνητικές κριτικές για τα λάθη που παρουσίαζε από την έκδοση 3 και μετά και αφού έγινε μια εμπορική επιτυχία πολλές εταιρίες ασχολήθηκαν με την δημιουργία λειτουργιών επεκτείνοντας έτσι την λειτουργικότητά της. Η τελική της έκδοση ήταν η έκδοση 6 το 1998 και συνέχισε με την τωρινή Visual Basic .NET γνωστή πλέον σαν Visual Basic.

### 2.4.1 Χαρακτηριστικά της γλώσσας

Όπως και η γλώσσα προγραμματισμού BASIC έτσι και η Visual Basic έχει σχεδιαστεί για να είναι εύκολη τόσο στην εκμάθηση όσο και στην χρήση της. Το περιβάλλον προσφέρει στον προγραμματιστή την δυνατότητα δημιουργίας τόσο απλών GUI εφαρμογών όσο και πιο πολύπλοκων. Ο προγραμματισμός σε Visual Basic συνδυάζει την οπτική δημιουργία μιας φόρμας με κουμπιά και αντικείμενα ελέγχου, τον ορισμό

ιδιοτήτων και ενεργειών των αντικειμένων καθώς και την προσθήκη επιπλέον γραμμών κώδικα για αύξηση της λειτουργικότητας της εφαρμογής. Ο προγραμματιστής μπορεί να δημιουργήσει ένα απλό πρόγραμμα χωρίς την προσθήκη πολλών γραμμών κώδικα γιατί πολλά χαρακτηριστικά και ενέργειες έχουν προεπιλεγεί με την μεταφορά τους στην φόρμα. Σε παλαιότερες εκδόσεις της Visual Basic υπήρχε έντονα το θέμα της απόδοσης στην επίδοση κάτι το οποίο πλέον δεν είναι κάτι το σημαντικό αφού η επεξεργαστική ισχύς των υπολογιστών έχει αυξηθεί δραματικά καθώς και ο τρόπος μεταγλώττισης του εγγενούς κώδικα. Αν και οι εφαρμογές μπορούν να μετατραπούν σε ατόφιο εκτελέσιμο κώδικα από την έκδοση 5 και μετά δεν παύουν να εξαρτώνται από την παρουσία βιβλιοθηκών εκτέλεσης με μέγεθος τουλάχιστον 1 MB. Αν και οι βιβλιοθήκες αυτές από την έκδοση Windows 2000 και μετά περιέχεται αυτόματα στο λειτουργικό σε παλαιότερες εκδόσεις των Windows (95/98/NT) χρειαζόταν η διανομή μαζί με το εκτελέσιμο.

Οι φόρμες μπορούν να δημιουργηθούν με την παλέτα εργαλείων που περιέχει το προγραμματιστικό περιβάλλον με την τεχνική Drag n' Drop. Με ένα εργαλείο τοποθετούνται τα αντικείμενα ελέγχου (κουμπιά, textboxes) στην φόρμα. Τα αντικείμενα ελέγχου έχουν συμπεριφορές και οι χειριστές γεγονότων είναι υπεύθυνοι να αλληλεπιδρούν με αυτά. Προεπιλεγμένες τιμές επιλέγονται για τα αντικείμενα ελέγχου από την δημιουργία τους κιόλας τα οποία φυσικά αλλάζουν από τον προγραμματιστή. Πολλές τιμές στις ιδιότητες τους μπορούν να αλλάξουν κατά την εκτέλεση ανάλογα με τις επιλογές που κάνει ο χρήστης της εφαρμογής ή από αλλαγές στο περιβάλλον προσφέροντας μια δυναμική εφαρμογή. Για παράδειγμα ο προγραμματιστής μπορεί να εισάγει στην φόρμα τον ελεγκτή γεγονότων αλλαγή θέσης ενός αντικειμένου ελέγχου κάνοντας έτσι το κουμπί στην φόρμα να παραμένει σταθερό ή να γεμίζει όλη την φόρμα κτλ. Με την εισαγωγή κώδικα στον χειριστή γεγονότων της ιδιότητας key press ενός textbox το πρόγραμμα μπορεί αυτόματα να μεταφράζει το κείμενο που δίνεται από τον χρήστη ή να απαγορεύει την εισαγωγή ορισμένων χαρακτήρων σε αυτό.

Η Visual Basic δημιουργεί εκτελέσιμα αρχεία .exe, στοιχεία ελέγχου Active X ή αρχεία DLL αλλά η κύρια χρήση της είναι για δημιουργία εφαρμογών Windows και της σύνδεσης της με βάσεις δεδομένων. Προσφέρονται πλαίσια διαλόγου με αισθητά λιγότερες λειτουργίες για την χρήση pop-up δυνατοτήτων. Με τα στοιχεία ελέγχου επιτυγχάνονται οι βασικές λειτουργίες της εφαρμογής αλλά οι προγραμματιστές μπορούν να εισάγουν διάφορες μεθόδους μέσα στους κατάλληλους χειριστές γεγονότων. Για παράδειγμα ένα αντικείμενο ελέγχου drop down combination list θα εμφανίζει αυτόματα



μια λίστα με επιλογές για τον χρήστη και αυτός θα μπορεί να επιλέγει όποιο αντικείμενο επιθυμεί. Ένας χειριστής γεγονότων καλείται όταν κάποιο αντικείμενο επιλεγθεί και στην συνέχεια εκτελείται πρόσθετος κώδικας ο οποίος έχει εισαχθεί από τον προγραμματιστή για να εκτελείται όταν επιλεγεί το συγκεκριμένο αντικείμενο.

Ακόμα ένα αντικείμενο της Visual Basic μπορεί να μην έχει γραφικό περιβάλλον αλλά να παρέχει υπηρεσίες σε κάποιο άλλο αντικείμενο Active X μέσω του μοντέλου Component Object Model – COM. Αυτό επιτρέπει την επεξεργασία του στην μεριά του εξυπηρετητή ή την δημιουργία πρόσθετων διεργασιών. Η γλώσσα παρέχει αυτόματη διαχείριση μνήμης τύπου συλλέκτη σκουπιδιών χρησιμοποιώντας τον υπολογισμό αναφορών, παρέχει μια βιβλιοθήκη με βοηθητικά αντικείμενα και παρέχει και μια βασική αντικειμενοστραφή υποστήριξη. Αντίθετα με άλλες γλώσσες προγραμματισμού η Visual Basic δεν είναι case sensitive δηλαδή δεν ξεχωρίζει πεζούς και κεφαλαίους χαρακτήρες. Βέβαια μετατρέπει τις λέξεις κλειδιά αυτόματα στην μορφή που πρέπει να είναι. Επίσης η σύγκριση των αλφαριθμητικών είναι case sensitive σαν προεπιλογή του συστήματος. Ο μεταγλωττιστής της Visual Basic είναι κοινός με άλλες γλώσσες του Visual Studio C,C++ αλλά οι περιορισμοί στο IDE δεν επιτρέπουν την δημιουργία ορισμένων τελικών (Windows model DLL) και μοντέλων νημάτων.

## 2.5 Πλαίσιο Microsoft .Net

### 2.5.1 Γενικά

Η τεχνολογία .Net Framework υποστηρίζει την κατασκευή και εκτέλεση της επόμενης γενιάς προγραμμάτων καθώς επίσης και XML διαδικτυακών υπηρεσιών. Το .Net Framework έχει σχεδιαστεί για να καλύψει τους παρακάτω τομείς:

- Να παρέχει ένα συμπαγές αντικειμενοστραφές περιβάλλον προγραμματισμού στο οποίο ο κώδικας των αντικειμένων θα εκτελείται τοπικά, τοπικά αλλά θα κατανέμεται διαδικτυακά ή θα εκτελείται από απόσταση.
- Να παρέχει ένα περιβάλλον εκτέλεσης κώδικα το οποίο θα ελαχιστοποιεί την πιθανότητα συγκρούσεων λόγω διαφορετικών εκδόσεων ή διαφορετικού λογισμικού.

- Να παρέχει ένα περιβάλλον εκτέλεσης κώδικα το οποίο υποστηρίζει την ασφαλή εκτέλεση κώδικα όπως επίσης και την εκτέλεση κώδικα από άγνωστο πάροχο ή τρίτο ο οποίος δεν είναι και πολύ έμπιστος.
- Να παρέχει ένα περιβάλλον εκτέλεσης κώδικα το οποίο να ελαχιστοποιεί τα προβλήματα επιδόσεων από διαφορετικά περιβάλλοντα.
- Να κάνει την εμπειρία του προγραμματισμού συμπαγή σε μια μεγάλη γκάμα προγραμμάτων όπως βασισμένα σε Windows ή βασισμένα στο διαδίκτυο.
- Να εγκαθίδρυση μια μορφή πρότυπου στην βιομηχανία κατασκευής λογισμικού διασφαλίζοντας πως τα προγράμματα που βασίζονται σε αυτό μπορούν να αλληλεπιδράσουν με οποιοδήποτε άλλο κώδικα.

## 2.5.2 Βασικά Χαρακτηριστικά

Το .Net Framework έχει 2 βασικά χαρακτηριστικά στοιχεία: Το Common Language Runtime (CLR) και την βιβλιοθήκη κλάσεων του. Τα προγράμματα που είναι γραμμένα για το πλαίσιο αυτό εκτελούνται σε ένα περιβάλλον λογισμικού το Common Language Runtime (CLR) μια εικονική μηχανή της εφαρμογής η οποία προσφέρει υπηρεσίες όπως ασφάλεια, διαχείριση της μνήμης και διαχείριση αποτυχιών της εφαρμογής. Η βιβλιοθήκη κλάσεων του .Net Framework προσφέρει το γραφικό περιβάλλον, την διαχείριση δεδομένων, την συνδεσιμότητα στις βάσεις δεδομένων, την κρυπτογράφηση, την ανάπτυξη διαδικτυακών εφαρμογών, του αλγορίθμους αριθμών και την επικοινωνία μεταξύ δικτύων. Οι προγραμματιστές παράγουν λογισμικό συνδυάζοντας τον κώδικα τους και το .Net Framework μαζί με άλλες βιβλιοθήκες.

## 2.5.3 Χαρακτηριστικά Σχεδιασμού

- Διαλειτουργικότητα

Επειδή τα υπολογιστικά συστήματα απαιτούν συνήθως την αλληλεπίδραση μεταξύ νέων και παλαιότερων εφαρμογών το .Net Framework παρέχει τα μέσα για πρόσβαση στις λειτουργίες που υλοποιούνται σε νεότερα και παλαιότερα προγράμματα τα οποία

εκτελούνται εκτός περιβάλλοντος .Net Framework. Πρόσβαση στα αντικείμενα COM παρέχεται μέσω των System.Runtime.InteropServices και System.EnterpriseServices namespaces ενώ πρόσβαση στις λειτουργίες έχει μέσω των χαρακτηριστικών P/Invoke.

- Common Language Runtime engine

Το Common Language Runtime (CLR) εξυπηρετεί σαν την μηχανή εκτέλεσης του .Net Framework. Όλα τα προγράμματα .Net εκτελούνται κάτω από την επίβλεψη του CLR δίνοντας εγγύηση για συγκεκριμένες ιδιότητες και συμπεριφορές όσον αφορά την διαχείριση της μνήμης, την ασφάλεια και την διαχείριση αποτυχίας της εφαρμογής.

- Γλωσσική Ανεξαρτησία

Το .Net Framework έχει το Common Type System-CTS. Το CTS ορίζει όλους τους δυνατούς τύπους δεδομένων αλλά και προγραμματιστικών δομών που υποστηρίζονται από το CLR και πως αυτά μπορούν ή δεν μπορούν να αντιδράσουν μεταξύ τους σύμφωνα με τα χαρακτηριστικά του CLI. Εξαιτίας αυτού του χαρακτηριστικού το .Net Framework υποστηρίζει την ανταλλαγή τύπων και αντικειμένων μεταξύ διαφορετικών βιβλιοθηκών και εφαρμογών γραμμένες σε οποιαδήποτε γλώσσα που υποστηρίζει το .Net.

- Base Class Library – BCL

Η Base Class Library η οποία αποτελεί μέρος της βιβλιοθήκης FCL είναι μια βιβλιοθήκη λειτουργιών η οποία είναι διαθέσιμη σε όλες τις γλώσσες προγραμματισμού που χρησιμοποιούν το .Net Framework. Παρέχει κλάσεις οι οποίες ενσωματώνουν έναν αριθμό από κοινές συναρτήσεις, όπως εγγραφή και ανάγνωση αρχείου, δια-δραστικότητα με βάσεις δεδομένων, απόδοση γραφικών, διαχείριση XML αρχείων κτλ. Αποτελείται από κλάσεις, διεπαφές επαναχρησιμοποιούμενων τύπων που αλληλεπιδρούν με το CLR.

- Απλοποιημένη εγκατάσταση

Το . Net Framework περιλαμβάνει σχεδιαστικά χαρακτηριστικά και εργαλεία τα οποία βοηθούν στην διαχείριση της εγκατάστασης λογισμικού τους υπολογιστές για να αποφευχθούν τυχόν παρεμβολές από προηγούμενες εκδόσεις.

- Ασφάλεια και Φορητότητα

Το μοντέλο περιγράφει κάποιες από τις βασικές αδυναμίες λογισμικού όπως είναι το buffer overflow τα οποία εκμεταλλεύονται ορισμένα κακόβουλα λογισμικά. Επίσης

προσφέρει ένα κοινό μοντέλο ασφάλειας για όλες τις εφαρμογές. Τέλος αν και η Microsoft δεν έχει κατασκευάσει το ολοκληρωμένο πακέτο του .Net Framework για άλλες πλατφόρμες πέρα από τα Windows, έχει πάρει και ISO και ECMA για την κατασκευή του και κάνοντας το διαθέσιμο επίσημα σε όποιον τρίτο το θέλει για την κατασκευή .Net Framework για άλλες πλατφόρμες.

#### **2.5.4 Εκδόσεις .Net Framework**

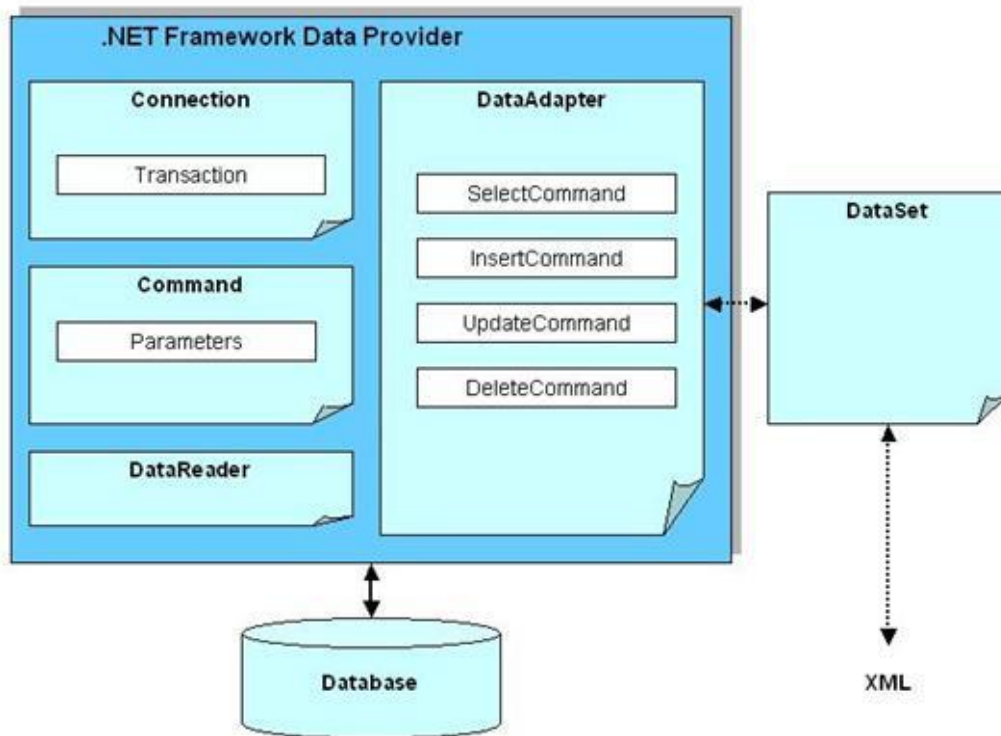
Η έκδοση του .Net Framework 3 κυκλοφόρησε το 2007 και συμπεριλαμβάνεται και στα Windows Server 2008 και στα Vista. Η έκδοση 3.5 περιλαμβάνεται στα Windows 7 και μπορεί να εγκατασταθεί στα Windows XP και στα Windows Server 2003. Τον Απρίλιο του 2010 κυκλοφόρησε η έκδοση 4.0 μαζί με το Visual Studio 2010. Η οικογένεια .Net Framework περιλαμβάνει επίσης εκδόσεις για κινητές και ενσωματωμένες συσκευές. Μια πιο περιορισμένη έκδοση του πακέτου το .Net Compact Framework είναι διαθέσιμο για τις πλατφόρμες Windows CE και τις κινητές συσκευές τύπου Smartphone.

#### **2.6 ADO.NET**

Το μοντέλο ADO.NET το οποίο χρησιμοποιείται και στην παρούσα διπλωματική παρέχει στον προγραμματιστή ένα API – Application Programming Interface για πρόσβαση σε σχεσιακά συστήματα βάσεων δεδομένων αλλά και για την πρόσβαση σε δεδομένα μη σχεσιακών βάσεων δεδομένων. Σχεδιάστηκε για το .Net Framework και είναι η τελευταία τεχνολογία της Microsoft για επικοινωνία των Active X Data Objects - (ADO) με τα Component Object Model - (COM) της Microsoft. Οι βασικές κλάσεις του ADO.NET είναι:

- **Connection:** Με το αντικείμενο αυτό ο προγραμματιστής εγκαθιδρύει την σύνδεση με τον προορισμό των δεδομένων παρέχοντας τις απαραίτητες πληροφορίες.
- **Command:** Με το αντικείμενο αυτό ο προγραμματιστής μπορεί να εκτελέσει τα απαιτούμενα για αυτόν SQL ερωτήματα καθώς και τις αποθηκευμένες διαδικασίες.

- **Datareader:** Το αντικείμενο αυτό δημιουργείται πάντα μετά το αντικείμενο **Command**. Παρέχει έναν σειριακό τρόπο ανάγνωσης των δεδομένων πάντα προς την μια κατεύθυνση και χωρίς την δυνατότητα τροποποίησης τους.
- **Dataset:** Το αντικείμενο αυτό είναι υπεύθυνο για την συλλογή των δεδομένων προσωρινά σε μια μνήμη εκτελώντας καθήκοντα προσωρινής μνήμης για την εφαρμογή. Μπορεί να συνδεθεί με οποιαδήποτε αριθμό **Datatables** αφού η δομή του είναι παρόμοια με αυτή μιας σχεσιακής βάσης δεδομένων. Το μεγάλο πλεονέκτημα που προσφέρει το αντικείμενο αυτό είναι ότι παραμένει αποσυνδεδεμένο από την πηγή. Συνδέεται μόνο για να πάρει τις εγγραφές από την βάση δεδομένων, αποσυνδέεται για να μπορέσει ο χρήστης να τροποποιήσει τα δεδομένα ή το πρόγραμμα και περιμένει πάλι να συνδεθεί όταν αυτό τελειώσει για να ενημερώσει την βάση με τις αλλαγές.
- **Dataadapter:** Το αντικείμενο αυτό μπορεί και εκτελεί δύο λειτουργίες: Αρχικά είναι υπεύθυνο να συμπληρώσει με δεδομένα το **Dataset** από την βάση δεδομένων. Και δεύτερον είναι υπεύθυνος για την εφαρμογή των αλλαγών στις βάσεις ανάλογα με τις τροποποιήσεις που έχει κάνει ο χρήστης στο **Dataset**.



Σχήμα 10 : Βασική Αρχιτεκτονική ADO.Net

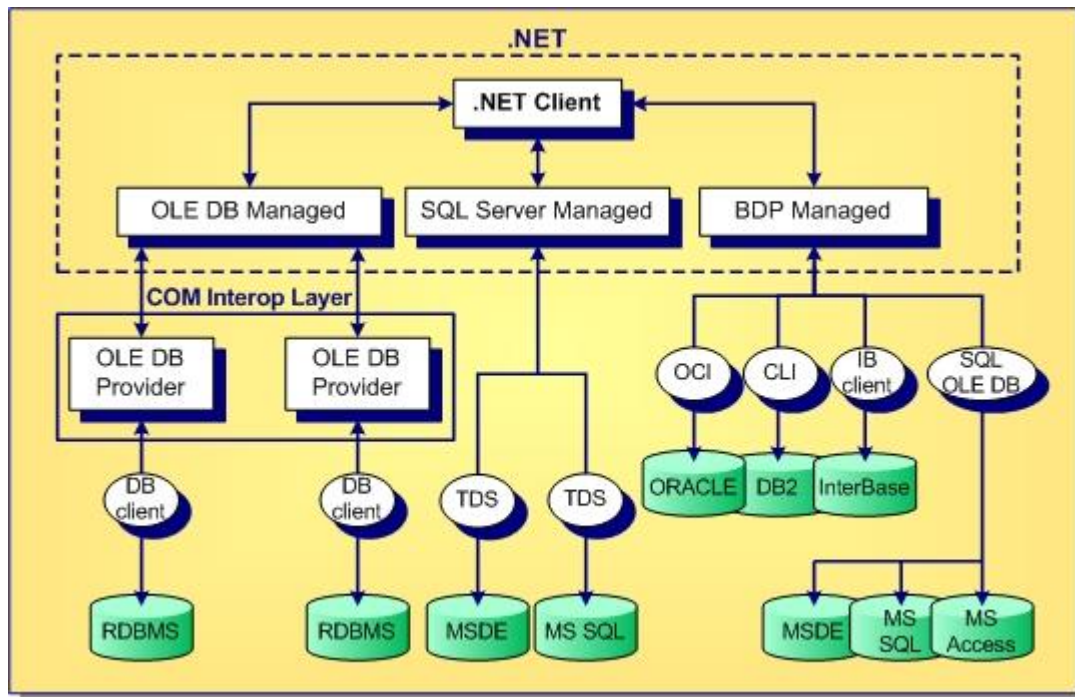
Βασικό χαρακτηριστικό της τεχνολογίας ADO.Net είναι η ποικιλία σε παρόχους που υποστηρίζει(Data Providers). Σαν πάροχος δεδομένων ορίζεται ένα σύνολο από κλάσεις και αντικείμενα τα οποία μπορεί να χρησιμοποιήσει ο προγραμματιστής για να πετύχει πρόσβαση στην βάση δεδομένων και να εκτελέσει τα SQL ερωτήματα του. Ουσιαστικά αποτελεί μια γέφυρα ανάμεσα στην εφαρμογή και στην βάση δεδομένων. Οι κυριότερες κλάσεις που απαρτίζουν έναν πάροχο δεδομένων είναι οι εξής:

- Connection
- Command
- DataReader
- DataAdapter

Το ADO.NET παρέχει υποστήριξη για διαφορετικούς παρόχους δεδομένων ειδικά σχεδιασμένους για διαφορετικές προελεύσεις δεδομένων. Κάθε πάροχος έχει με διαφορετικούς τρόπους υλοποιημένες τις κλάσεις έτσι ώστε να είναι βελτιστοποιημένες για το συγκεκριμένο σχεσιακό μοντέλο βάσης δεδομένων.

Το .Net Framework υποστηρίζει τους εξής παρόχους δεδομένων:

- Sql Server provider: Για Sql Server βάση δεδομένων.
- OLE DB provider: Παρέχει πρόσβαση σε οποιαδήποτε βάση δεδομένων διαθέτει απλά έναν OLE DB οδηγό.
- Oracle provider: Για Oracle βάσεις δεδομένων.
- ODBC provider: Παρέχει πρόσβαση σε οποιαδήποτε βάση δεδομένων διαθέτει απλά έναν ODBC οδηγό.



Σχήμα 11 : Διαφορετικοί data providers ανάλογα με την Βάση Δεδομένων

## Επίλογος

Στο παραπάνω κεφάλαιο αναλύσαμε το Visual Studio 2008, αναφέροντας τα βασικά του χαρακτηριστικά και τις γλώσσες τις οποίες υποστηρίζει. Μελετήσαμε την τεχνολογία .Net Framework της Microsoft σχετικά με τις υπηρεσίες που προσφέρει και αναφερθήκαμε

στην νέα τεχνολογία ADO.Net για την σύνδεση της εφαρμογής με την βάση δεδομένων. Στο επόμενο κεφάλαιο θα μελετήσουμε τις βασικές λειτουργίες της εφαρμογής μας και πως χρησιμοποιήσαμε τις παραπάνω τεχνολογίες για την υλοποίηση της.



## ΚΕΦΑΛΑΙΟ 3 - Βασικές Λειτουργίες της Εφαρμογής

### Εισαγωγή

Βασικό κομμάτι σε μια εφαρμογή είναι και η τεκμηρίωση της. Δηλαδή οι τρόποι με τους οποίους έχει υλοποιηθεί η κάθε λειτουργία της. Αυτό κάνει εύκολο το έργο του μελλοντικού προγραμματιστή που θα προσπαθήσει να το βελτιώσει όπως και του ίδιου του χρήστη στο κομμάτι της ασφάλειας. Στο κεφάλαιο αυτό θα παρουσιαστούν οι βασικές λειτουργίες της εφαρμογής καθώς και οι τρόπος που επιλέχθηκε να υλοποιηθεί η κάθε μια. Θα γίνει μια ανάλυση των σημαντικότερων βημάτων της υλοποίησης με παραπομπές στον κώδικα.

### 3.1 Βασικές Λειτουργίες

Οι βασικές λειτουργίες της εφαρμογής είναι η ανάκτηση, τροποποίηση και αποθήκευση δεδομένων – πληροφοριών σε μια βάση δεδομένων. Η εφαρμογή ξεκινά με μια φόρμα (frmLogin) όπου ο χρήστης – διαχειριστής καλείται να αυθεντικοποιηθεί με όνομα και κωδικό προκειμένου να έχει πρόσβαση στην εφαρμογή. Αν η διαδικασία της αυθεντικοποίησης είναι επιτυχής τότε προχωράει στην επόμενη φόρμα, την κεντρική (frmMain).

Βασικό κομμάτι της φόρμας εισόδου είναι η σωστή αυθεντικοποίηση του χρήστη και η μεταφορά των παραμέτρων σύνδεσης στην βάση δεδομένων και στην κεντρική φόρμα. Για την αυθεντικοποίηση του ο διαχειριστής έχει τα ίδια συνθηματικά που ρυθμίζονται κατά την εγκατάσταση του διακομιστή MySQL. Όνομα χρήστη root και κωδικό 12345. Τα συνθηματικά είναι σκόπιμα επιλεγμένα αφού ουσιαστικά ο χρήστης της εφαρμογής θα είναι ένας, ο διαχειριστής και δεν χρειάζεται η ύπαρξη νέου πίνακα για τον έλεγχο ενός συνθηματικού. Το ίδιο ισχύει και για την επιλογή της βάσης δεδομένων από τον διακομιστή. Ο διαχειριστής δεν χρειάζεται να επιλέξει την βάση δεδομένων που θα συνδεθεί αφού ουσιαστικά σε μια βάση δεδομένων υπάρχουν οι απαιτούμενοι πίνακες.

1. 'Login Button. Με αυτό το κουμπί γίνεται ο έλεγχος ταυτοποίησης του χρήστη για το πρόγραμμα.
2. 'Για να αυθεντικοποιηθεί πρέπει να βάλει τους κωδικούς για την βάση δεδομένων..

```
3. 'Οι κωδικοί είναι: username: root
4. '           password: 12345
5. Private Sub btnOk_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnOk.Click
6. 'Εδώ αποθηκεύονται οι παράμετροι για την σύνδεση
7. Dim myConnString As String
8. conn = New MySqlConnection()
9. myConnString = "user id=" & txtUsername.Text & ";" & "password=" &
    txtPassword.Text & ";" & "database=customers"
10.     conn.ConnectionString = myConnString
11.     Try
12.         'Συνδεόμαστε και γίνεται ο έλεγχος αν μπορούμε να συνεχίσουμε
13.         'Σε περίπτωση λάθους έχει σχετικό μήνυμα
14.         'Αν γίνει σωστά η σύνδεση μεταφέρει την μεταβλητή στην βασική
            φόρμα(Main Form) και αυτή κλείνει
15.         conn.Open()
16.         conn.Close()
17.         Dim MainForm As New frmMain
18.         MainForm.connectionString = myConnString
19.         MainForm.Show()
20.         Me.Hide()
21.         Me.Close()
22.         Catch myerror As MySqlException
23.             MessageBox.Show("Λάθος κατά την σύνδεση: " & myerror.Message,
                " Λάθος Σύνδεσης ", MessageBoxButtons.OK,
                MessageBoxIcon.Exclamation)
24.         conn.Dispose()
25.         End Try
26. End Sub
```

Μετά την επιτυχή είσοδο του διαχειριστή στο σύστημα δεν γίνεται περαιτέρω έλεγχος για την ταυτοποίηση του χρήστη εφόσον δεν προκύπτει τέτοια ανάγκη.

Στην συνέχεια βασική λειτουργία της εφαρμογής είναι η συνάρτηση που περνάει την σύνδεση στις άλλες φόρμες της εφαρμογής. Αυτό γίνεται με μια public συνάρτηση η οποία μπορεί να προσφέρει την τροποποίηση της private μεταβλητής που αποθηκεύει τις παραμέτρους της σύνδεσης.

1. 'Πρέπει να δημιουργήσουμε μια μεταβλητή για να περάσουμε την σύνδεση στην βάση μας απο την φόρμα LogIn στην φόρμα Main.
2. 'Γι αυτό τον λόγο θα δημιουργήσουμε μια private μεταβλητή η οποία θα κρατάει την σύνδεση και μια συνάρτηση public
3. 'η οποία θα χρησιμοποιείται για να αλλάξουμε την private τιμή

```
4. Private myConnString As String
5. 'Η public συνάρτηση η οποία μας δίνει την δυνατότητα αλλαγής της
   private μεταβλητής
6. Public WriteOnly Property connectionString() As String
7. Set(ByVal value As String)
8. myConnString = value
9. End Set
10.      End Property
```

### 3.2 Καρτέλα Διαχείρισης Αποθήκης

Στην βασική φόρμα ο διαχειριστής μπορεί να επιλέξει ουσιαστικά με ποια βάση επιθυμεί να εργαστεί. Στην φόρμα υπάρχουν χωρισμένες διαφορετικά οι λειτουργίες που αφορούν την αποθήκη του θερμοκηπίου και τα φυτά και οι λειτουργίες που αφορούν τους πελάτες. Στην πρώτη καρτέλα εμφανίζονται οι λειτουργίες που μπορεί ο διαχειριστής να κάνει με την βάση των φυτών. Αυτές είναι :

- Εισαγωγή νέου προϊόντος
- Διαγραφή προϊόντος
- Τροποποίηση του προϊόντος
- Καθώς και αναζήτηση κάποιου μέσα στην βάση δεδομένων.

#### **Συνάρτηση εισαγωγής νέου προϊόντος:**

```
1. Private Sub btnEisagwgiFutou_Click(ByVal sender As System.Object,
   ByVal e As System.EventArgs) Handles btnEisagwgiFutou.Click
2. 'Στην συνέχεια δηλώνουμε την MySqlCommand η οποία μπορεί να κρατήσει το
   SQL ερώτημα μας.
3. 'Την SQL η οποία θα κρατήσει το ερώτημα μας αφού το διαμορφώσουμε.
4. 'Θα ελέγξουμε αν ο διαχειριστής έχει συμπληρώσει όλες τις
   παραμέτρους
5. 'Και αν ναι θα κάνουμε την εισαγωγή
6. Dim conn As New MySqlConnection
7. Dim myCommand As New MySqlCommand
8. Dim SQL As String = ""
```

```

9. 'Για να κάνουμε το query πρέπει να ελέγξουμε αν ο διαχειριστής
έβαλε όλες τις παραμέτρους
10. If (cmbEidosFutou.SelectedItem IsNot Nothing) And
(cmbEidosFulomatos.SelectedItem IsNot Nothing) And
(cmbApaitisisFws.SelectedItem IsNot Nothing) And
(cmbApaitisisNero.SelectedItem) IsNot Nothing And
(cmbXrwmaAnthous.SelectedItem) IsNot Nothing And
(cmbEpoxiAnthos.SelectedItem) IsNot Nothing And
(cmbAntoxiPago.SelectedItem) IsNot Nothing And
(cmbEdafos.SelectedItem) IsNot Nothing And
(cmbParathalasio.SelectedItem) IsNot Nothing And
(cmbFraxtes.SelectedItem) IsNot Nothing And
(cmbArwmatikoFilwma.SelectedItem) IsNot Nothing And
(cmbArwmatikaAnthi.SelectedItem) IsNot Nothing And txtCost.Text <>
"" And txtPosotita.Text <> "" And txtPlantName.Text <> "" Then
11. 'SQL ερώτημα που πρέπει να κάνουμε για να μπει η εγγραφή στην
βάση
12. SQL = "INSERT INTO plants VALUES (NULL,'" &
cmbEidosFutou.SelectedItem.ToString & "',' ' _
13. & cmbEidosFulomatos.SelectedItem.ToString & "',' ' &
cmbApaitisisFws.SelectedItem.ToString _
14. & "',' ' & cmbApaitisisNero.SelectedItem.ToString & "',' ' &
cmbXrwmaAnthous.SelectedItem.ToString & "',' ' _
15. & cmbEpoxiAnthos.SelectedItem.ToString & "',' ' &
cmbAntoxiPago.SelectedItem.ToString & "',' ' _
16. & cmbEdafos.SelectedItem.ToString & "',' ' &
cmbParathalasio.SelectedItem.ToString & "',' ' _
17. & cmbFraxtes.SelectedItem.ToString & "',' ' &
cmbArwmatikoFilwma.SelectedItem.ToString & "',' ' _
18. & cmbArwmatikaAnthi.SelectedItem.ToString & "',' ' &
txtPosotita.Text.Trim & "',' ' & txtCost.Text.Trim & "',' ' &
txtPlantName.Text.Trim & "'"")"
19. 'Παίρνουμε τα στοιχεία για την σύνδεση
20. conn.ConnectionString = myConnString
21. Try
22. 'Ανοίγουμε την σύνδεση
23. conn.Open ()
24. Try
25. 'Το επόμενο βήμα είναι να παραμετροποιήσουμε το αντικείμενο
command.

```

```
26. 'Για ένα απλό query σαν το δικό μας θα πρέπει να φτιάξουμε το
    κείμενο στο query και να πούμε στο αντικείμενο
27. 'command πια σύνδεση να χρησιμοποιήση
28. myCommand.Connection = conn
29. myCommand.CommandText = SQL
30. 'Το εκτελούμε
31. myCommand.ExecuteNonQuery()
32. Catch ex As MySqlException
33. MessageBox.Show("Κατι πήγε στραβά με την εισαγωγή της
    εγγραφής στην Βάση Δεδομένων: " & ex.Message & " " & ex.Number & "
    " & ex.ErrorCode, "Λάθος εισαγωγής")
34. End Try
35. 'Σχετικά μηνύματα σε περίπτωση λάθους
36. Catch ex As MySqlException
37. MessageBox.Show("Σφάλμα κατά την σύνδεση στην Βάση Δεδομένων:
    " & ex.Message, "Λάθος κατά την Σύνδεση στην Βάση Δεδομένων")
38. Finally
39. 'Κλείνουμε την σύνδεση και ενημερώνουμε για το πως πήγε η
    εισαγωγή της εγγραφής
40. If conn.State <> ConnectionState.Closed Then conn.Close()
41. MessageBox.Show("Η εισαγωγή του νέου φυτού ολοκληρώθηκε με
    επιτυχία.", "Εισαγωγή νέου φυτού", MessageBoxButtons.OK,
    MessageBoxIcon.Information)
42. 'Καθαρισμός της φόρμας εισαγωγής
43. cmbEidosFutou.SelectedItem = Nothing
44. cmbEidosFulomatos.SelectedItem = Nothing
45. cmbApaitisisFws.SelectedItem = Nothing
46. cmbApaitisisNero.SelectedItem = Nothing
47. cmbXrwmaAnthous.SelectedItem = Nothing
48. cmbEpoxiAnthos.SelectedItem = Nothing
49. cmbAntoxiPago.SelectedItem = Nothing
50. cmbEdafos.SelectedItem = Nothing
51. cmbParathalasio.SelectedItem = Nothing
52. cmbFraxtes.SelectedItem = Nothing
53. cmbArwmatikoFilwma.SelectedItem = Nothing
54. cmbArwmatikaAnthi.SelectedItem = Nothing
55. txtPosotita.Text = ""
56. txtCost.Text = ""
57. txtPlantName.Text = ""
58. End Try
59. Else
```

```
60.     MessageBox.Show("Παρακαλώ συμπληρώστε όλες τις παραμέτρους  
για την εγγραφή.", "Λάθος εισαγωγής", MessageBoxButtons.OK,  
        MessageBoxIcon.Information)  
61.     End If  
62. End Sub
```

Τα περισσότερα κομμάτια της συνάρτησης δηλώνονται με παρόμοιο τρόπο και στις άλλες λειτουργίες. Βασικά κομμάτια του κώδικα εδώ είναι τα εξής.

**Γραμμές 1-8 :** Στο κομμάτι αυτό δηλώνονται οι μεταβλητές που θα χρειαστούν για την συνάρτηση. Πρόκειται για τις μεταβλητές οι οποίες αποθηκεύουν τα στοιχεία της σύνδεσης με την βάση και του αντικειμένου που είναι υπεύθυνο για την εκτέλεση του SQL υπό-ερωτήματος.

**Γραμμές 9-11 :** Στο κομμάτι αυτό ο κώδικας ελέγχει αν ο διαχειριστής έχει επιλέξει όλες τις παραμέτρους για την νέα εισαγωγή. Ο διαχειριστής καλείται να επιλέξει από τα αντικείμενα combobox και μέσα από προεπιλεγμένα χαρακτηριστικά τα κατάλληλα για την εισαγωγή του νέου φυτού. Το μόνο που πρέπει να πληκτρολογήσει είναι το όνομα, τα τεμάχια και το κόστος του.

**Γραμμές 12-18:** Στο κομμάτι αυτό κατασκευάζουμε το SQL ερώτημα για την εισαγωγή μιας νέας εγγραφής στην βάση δεδομένων.

**Γραμμές 19-31:** Εδώ γίνεται η σύνδεση με την βάση δεδομένων και η εκτέλεση του ερωτήματος εισαγωγής νέας εγγραφής στην βάση δεδομένων. Στο σημείο αυτό πρέπει να τονιστεί ότι τόσο η διαδικασία σύνδεσης με την βάση δεδομένων όσο και η διαδικασία εκτέλεσης του SQL ερωτήματος γίνεται μέσα στην διαδικασία Try – Catch. Με τον τρόπο αυτό καταφέρνουμε να πιάσουμε πιθανά λάθη της εφαρμογής και να στείλουμε μήνυμα κατανοητό από το χρήστη για το τι πήγε στραβά. Σε διαφορετική περίπτωση ο χρήστης θα έβλεπε ένα μήνυμα το οποίο θα περιείχε πολύ περιορισμένες πληροφορίες και πιθανόν το πρόγραμμα να μην συμπεριφερόταν σωστά.

**Γραμμές 32-41:** Εδώ παρουσιάζονται τα μηνύματα που θα εμφανιστούν στον χρήστη αν κάποια από τις διαδικασίες δεν πάνε σωστά. Επίσης ολοκληρώνεται το κομμάτι του Try – Catch.

**Γραμμές 42-62:** Στις γραμμές εδώ κλείνουμε την σύνδεση με την βάση δεδομένων αν είναι ανοιχτή και μηδενίζουμε τις επιλογές του διαχειριστή για την εισαγωγή του φυτού. Τέλος εμφανίζουμε σχετικό μήνυμα στον διαχειριστή – χρήστη για την σωστή ολοκλήρωση της εισαγωγής.

**Συνάρτηση διαγραφής μιας εγγραφής στην βάση φυτά:**

```
1. 'Με την συνάρτηση αυτή γίνεται διαγραφή μιας εγγραφής στην βάση
2. Private Sub btnDelDel_Click(ByVal sender As System.Object, ByVal e
   As System.EventArgs) Handles btnDelDel.Click
3. 'Δηλώνουμε και πάλι τις μεταβλητές μας
4. Dim conn As New MySqlConnection
5. Dim myCommand As New MySqlCommand
6. Dim myAdapter As New MySqlDataAdapter
7. Dim myData As New DataTable
8. Dim SQL As String = ""
9. Dim selectedRow As Integer

10.      'Αποθήκευση της σειράς που επέλεξε ο διαχειριστής.
11.      If dtgrvwResultsDel.Rows.Count > 0 Then
12.      If dtgrvwResultsDel.SelectedRows.Count > 0 Then

13.      selectedRow = dtgrvwResultsDel.CurrentRow.Index

14.      'Για να κάνουμε αναζήτηση στην βάση πρέπει να κατασκευάσουμε
      αρχικά το SQL ερώτημα
15.      'Ελέγχουμε τις επιλογές έχει βάλει ο διαχειριστής και
      κατασκευάζουμε σταδιακά το query.

16.      If MessageBox.Show("Η εγγραφή αυτή θα διαγραφεί απο την Βάση
      Δεδομένων.Πατήστε Ναι για συνέχεια ή Όχι για ακύρωση", "Διαγραφή
      Εγγραφής", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) =
      DialogResult.Yes Then
17.      SQL = "DELETE FROM plants WHERE id='" &
      Integer.Parse(dtgrvwResultsDel.Item(0, selectedRow).Value.ToString)
      & "'"

18.      conn.ConnectionString = myConnString
19.      Try
20.      conn.Open()
21.      Try
22.      myCommand.Connection = conn
23.      myCommand.CommandText = SQL
24.      myCommand.ExecuteNonQuery()

25.      Catch ex As Exception
26.      MessageBox.Show("Κατι πήγε στραβά με την αναζήτηση της
      εγγραφής στην Βάση Δεδομένων: " & ex.Message, "Λάθος Αναζήτησης")
```

```
27.     End Try

28.     Catch ex As Exception
29.     MessageBox.Show("Σφάλμα κατά την σύνδεση στην Βάση Δεδομένων:
" & ex.Message, "Λάθος κατά την Σύνδεση στην Βάση Δεδομένων")
30.     Finally
31.     If conn.State <> ConnectionState.Closed Then conn.Close()

32.     End Try

33.     Else

34.     'Αν ο χρήστης πατήσει όχι γίνεται εκ νέου εμφάνιση όλης της
βάσης.
35.     SQL = "SELECT id As Κωδικός, ειδος As 'Είδος Φυτού',
ειδοςFulomatos As 'Είδος Φυλλόματος', fws As 'Απαιτήσεις Φώς', nero
As 'Απαιτήσεις Νερό', xrwmaAnthwn As 'Χρώμα Ανθους', eroxiAntou As
'Εποχή Άνθησης', antoxiPago As 'Αντοχή Παγωνιά', edafokaliergia As
'Για εαυδοκάλυψη', parathalasio As Παραθαλάσσιο, fraxtes As
Φράχτες, arwmatikoFulo As 'Αρωματικό Φύλλο', arwmatikoAntho As
'Αρωματικό Ανθό', posotita As Ποσότητα, kostos As Κόστος, onoma As
Όνομα FROM plants WHERE id <> '-1'"
36.     conn.ConnectionString = myConnString
37.     Try
38.     conn.Open()
39.     Try
40.     myCommand.Connection = conn
41.     myCommand.CommandText = SQL
42.     myAdapter.SelectCommand = myCommand
43.     myAdapter.Fill(myData)
44.     dtgrvwResultsDel.DataSource = myData

45.     Catch ex As Exception
46.     MessageBox.Show("Κατι πήγε στραβά με την αναζήτηση της
εγγραφής στην Βάση Δεδομένων: " & ex.Message, "Λάθος Αναζήτησης")
47.     End Try

48.     Catch ex As Exception
49.     MessageBox.Show("Σφάλμα κατά την σύνδεση στην Βάση Δεδομένων:
" & ex.Message, "Λάθος κατά την Σύνδεση στην Βάση Δεδομένων")
50.     Finally
51.     If conn.State <> ConnectionState.Closed Then conn.Close()

52.     End Try

53.     End If
54.     'Εδώ κλείνει ο έλεγχος για το αν ο χρήστης έχει επιλέξει
κάποια σειρά για να σβήσει.
55.     Else
56.     MessageBox.Show("Πρέπει να επιλέξετε μια σειρά για να
διαγραφη η εγγραφή", "Διαγραφή εγγραφής", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation)
57.     End If
```



```
58. Else
59.     MessageBox.Show("Πρέπει να επιλέξετε μια σειρά για να
        διαγραφεί η εγγραφή", "Διαγραφή εγγραφής", MessageBoxButtons.OK,
        MessageBoxIcon.Exclamation)
60. End If
61. End Sub
```

Η συνάρτηση αυτή λειτουργεί με την διαδικασία της αναζήτησης μιας εγγραφής και της διαγραφής της από τον διαχειριστή αν αυτός το επιθυμεί. Η διαδικασία της αναζήτησης θα περιγραφεί παρακάτω εφόσον είναι παρόμοια με την συνάρτηση αναζήτησης ενός προϊόντος στην βάση δεδομένων.

**Γραμμές 1-9:** Στο κομμάτι αυτό δηλώνονται και πάλι οι μεταβλητές οι οποίες θα χρειαστούν για την συνάρτηση μας. Είναι παρόμοιες με αυτές τις προηγούμενης συνάρτησης.

**Γραμμές 10-15:** Στο κομμάτι αυτό ελέγχουμε αν ο διαχειριστής έχει επιλέξει κάποια εγγραφή για να διαγράψει και αποθηκεύουμε τον αριθμό της.

**Γραμμές 15-18:** Οι γραμμές αυτές είναι οι πιο βασικές στην συνάρτηση. Αν ο διαχειριστής πατήσει το κουμπί της διαγραφής εμφανίζεται ένας διάλογος ο οποίος τον ενημερώνει ότι πρόκειται να διαγράψει μια εγγραφή οριστικά από τον πίνακα. Αν ο διαχειριστής πατήσει Yes τότε το SQL ερώτημα διαμορφώνεται σύμφωνα με την διαδικασία της διαγραφής μιας εγγραφής. Σε αντίθετη περίπτωση ο κώδικας που λειτουργεί είναι από την σειρά 33 και μετά και οποίος κάνει και πάλι εμφάνιση όλο τον πίνακα με τις εγγραφές.

**Γραμμές 36-44:** Στο κομμάτι αυτό εκτελείται το SQL ερώτημα και χρησιμοποιώντας το αντικείμενο DataAdapter γεμίζουμε με τα δεδομένα από το αποτέλεσμα του ερωτήματος τον πίνακα μας (DataGridView).

Στα υπόλοιπα κομμάτια του ο κώδικας είναι παρόμοιος με αυτόν στην εισαγωγή νέας εγγραφής. Δηλαδή εκτελείται μέσα σε διαδικασία Try – Catch για να πιάσουμε τυχόν λάθη, κλείνουμε την σύνδεση με την βάση αφού τελειώσουμε και εμφανίζουμε σχετικά μηνύματα ανάλογα με το λάθος που έχει παρουσιαστεί.

**Συνάρτηση τροποποίησης μιας εγγραφής στην βάση δεδομένων:**

1. 'Εδώ έχουμε το αντικείμενο το οποίο είναι υπεύθυνο να μας δείχνει τα αποτελέσματα της αναζήτησης σε πίνακα.
2. 'Για να γίνει η ανανέωση ο διαχειριστής θα μπορεί να πατήσει μέσα στον πίνακα και να κάνει την αλλαγή του.
3. `Private Sub dtgwRefresh_CellValueChanged(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles dtgwRefresh.CellValueChanged`
4. 'Κρατάει τον αριθμό της σειράς που επέλεξε να κάνει ανανέωση ο διαχειριστής.
5. 'Εμείς στην συνέχεια όταν τελειώσει με την αλλαγή θα αλλάζουμε τα δεδομένα όλης της σειράς.
6. `Dim selectedRow As Integer`
7. 'Αποθήκευση της σειράς που επέλεξε ο διαχειριστής.
8. `selectedRow = dtgwRefresh.CurrentRow.Index`
9. `MessageBox.Show(dtgwRefresh.Item(0, selectedRow).Value)`
10. 'Δηλώνουμε και πάλι τις μεταβλητές μας
11. `Dim conn As New MySqlConnection`
12. `Dim myCommand As New MySqlCommand`
13. `Dim myAdapter As New MySqlDataAdapter`
14. `Dim myData As New DataTable`
15. `Dim SQL As String = "", SQL2 As String = ""`
16. 'Για να κάνουμε αναζήτηση στην βάση πρέπει να κατασκευάσουμε αρχικά το SQL ερώτημα
17. 'Ελέγχουμε τις επιλογές έχει βάλει ο διαχειριστής και κατασκευάζουμε σταδιακά το query.
18. `SQL = "SELECT id As Κωδικός, ειδos As 'Είδος φυτού', ειδosFulomatos As 'Είδος φυλλόματος', fws As 'Απαιτήσεις Φώς', nero As 'Απαιτήσεις Νερό', xrwmaAnthwn As 'Χρώμα Άνθους', epoxiAntou As 'Εποχή Άνθησης', antoxiPago As 'Αντοχή Παγωνιά', edafokaliergia As 'Για εαδφοκάλυψη', parathalasio As Παραθαλάσσιο, fraxtes As Φράχτες, arwmatikoFulo As 'Αρωματικό Φύλλο', arwmatikoAntho As 'Αρωματικό Ανθό', posotita As Ποσότητα, kostos As Κόστος, onoma As Όνομα FROM plants WHERE id <> '-1'" & Integer.Parse(i) & ""`
19. 'Αρχικά πρέπει να ελέγξουμε αν όλα τα πεδία που θα αλλάξει ο διαχειριστής δεν είναι κενά.
20. 'Αν είναι κενά δεν μπορούμε να κάνουμε ανανέωση και εμφανίζεται σχετικό μήνυμα στον διαχειριστή
21. `If dtgwRefresh.Item(0, selectedRow).Value.ToString.Length = 0 Or dtgwRefresh.Item(1, selectedRow).Value.ToString.Length = 0 Or dtgwRefresh.Item(2, selectedRow).Value.ToString.Length = 0 Or dtgwRefresh.Item(3, selectedRow).Value.ToString.Length = 0 Or dtgwRefresh.Item(4, selectedRow).Value.ToString.Length = 0 Or dtgwRefresh.Item(5, selectedRow).Value.ToString.Length = 0 _`

```

22.         Or dtgwRefresh.Item(6, selectedRow).Value.ToString.Length = 0
Or dtgwRefresh.Item(7, selectedRow).Value.ToString.Length = 0 Or
dtgwRefresh.Item(8, selectedRow).Value.ToString.Length = 0 Or
dtgwRefresh.Item(9, selectedRow).Value.ToString.Length = 0 Or
dtgwRefresh.Item(10, selectedRow).Value.ToString.Length = 0 Or
dtgwRefresh.Item(11, selectedRow).Value.ToString.Length = 0 _
23.         Or dtgwRefresh.Item(12, selectedRow).Value.ToString.Length =
0 Or dtgwRefresh.Item(13, selectedRow).Value.ToString.Length = 0 Or
dtgwRefresh.Item(14, selectedRow).Value.ToString.Length = 0 Or
dtgwRefresh.Item(15, selectedRow).Value.ToString.Length = 0 Then
24.         'MessageBox.Show("UPDATE plants SET id='" &
dtgwRefresh.Item(0, selectedRow).Value & "'")

25.         MessageBox.Show("Λάθος κατα την εισαγωγή τιμής για την
ανανέωση.Δεν μπορεί να είναι κενό.", "Λάθος εισαγωγής",
MessageBoxButtons.OK, MessageBoxIcon.Warning)

26.         Else
27.         'Αν στις κατηγορίες κόστος και ποσότητα ο χρήστης βάλει
γράμματα εμφανίζεται σχετικό μήνυμα λάθους
28.         'και η αναζήτηση γίνεται εκ νέου
29.         If IsNumeric(dtgwRefresh.Item(13, selectedRow).Value) And
IsNumeric(dtgwRefresh.Item(14, selectedRow).Value) Then

30.         'MessageBox.Show(dtgwRefresh.Item(1, selectedRow).Value)
31.         conn.ConnectionString = myConnString
32.         Try
33.         'Εδώ φτιάχνουμε το query για την ανανέωση της βάσης μας.
34.         SQL2 = "UPDATE plants SET eidος='" & dtgwRefresh.Item(1,
selectedRow).Value.ToString & "',eidοςFulomatos='" &
dtgwRefresh.Item(2, selectedRow).Value.ToString & "',fws='" _
35.         & dtgwRefresh.Item(3, selectedRow).Value.ToString &
"',nero='" & dtgwRefresh.Item(4, selectedRow).Value.ToString &
"',xrwmaAnthwn='" & dtgwRefresh.Item(5, selectedRow).Value.ToString
& "',epoxiAntou='" _
36.         & dtgwRefresh.Item(6, selectedRow).Value.ToString &
"',antoxiPago='" & dtgwRefresh.Item(7, selectedRow).Value.ToString
& "',edafokaliergia='" & dtgwRefresh.Item(8,
selectedRow).Value.ToString & "',parathalasio='" _
37.         & dtgwRefresh.Item(9, selectedRow).Value.ToString &
"',fraxtes='" & dtgwRefresh.Item(10, selectedRow).Value.ToString &
"',arwmatikoFulo='" & dtgwRefresh.Item(11,
selectedRow).Value.ToString & "',arwmatikoAntho='" _
38.         & dtgwRefresh.Item(12, selectedRow).Value.ToString &
"',posotita='" & dtgwRefresh.Item(13, selectedRow).Value.ToString &
"',kostos='" & dtgwRefresh.Item(14, selectedRow).Value.ToString &
"',onoma='" & dtgwRefresh.Item(15, selectedRow).Value.ToString &
"" _
39.         & "WHERE id='" & Integer.Parse(dtgwRefresh.Item(0,
selectedRow).Value.ToString) & "'")
40.         conn.Open()
41.         Try
42.         myCommand.Connection = conn

```

```
43. myCommand.CommandText = SQL2
44. myAdapter.SelectCommand = myCommand
45. myAdapter.Fill(myData)
46. dtgwRefresh.DataSource = myData
47. Catch ex As Exception
48. MessageBox.Show("Κατι πήγε στραβά με την αναζήτηση της
εγγραφής στην Βάση Δεδομένων: " & ex.Message, "Λάθος Αναζήτησης")
49. End Try

50. Catch ex As Exception
51. MessageBox.Show("Σφάλμα κατα την σύνδεση στην Βάση Δεδομένων:
" & ex.Message, "Λάθος κατα την Σύνδεση στην Βάση Δεδομένων")
52. Finally
53. If conn.State <> ConnectionState.Closed Then conn.Close()
54. 'Μήνυμα επιτυχίας.
55. MessageBox.Show("Η ανανέωση της βάσης σας ολοκληρώθηκε με
επιτυχία", "Ανανέωση τιμής", MessageBoxButtons.OK,
MessageBoxIcon.Information)
56. End Try
57. 'Θα ξανακάνουμε αναζήτηση με όλα τα αποτελέσματα απλά για να
γεμίσουμε με δεδομένα την οθόνη του διαχειριστή
58. conn.ConnectionString = myConnString
59. Try
60. conn.Open()
61. Try
62. myCommand.Connection = conn
63. myCommand.CommandText = SQL
64. myAdapter.SelectCommand = myCommand
65. myAdapter.Fill(myData)
66. dtgwRefresh.DataSource = myData
67. Catch ex As Exception
68. MessageBox.Show("Κατι πήγε στραβά με την αναζήτηση της
εγγραφής στην Βάση Δεδομένων: " & ex.Message, "Λάθος Αναζήτησης")
69. End Try

70. Catch ex As Exception
71. MessageBox.Show("Σφάλμα κατα την σύνδεση στην Βάση Δεδομένων:
" & ex.Message, "Λάθος κατα την Σύνδεση στην Βάση Δεδομένων")
72. Finally
73. If conn.State <> ConnectionState.Closed Then conn.Close()

74. End Try

75. Else
76. 'Μήνυμα γιατί ο χρήστης έδωσε τιμές αλφαριθμητικά στην θέση
αριθμών.
77. 'Η αναζήτηση γίνεται αυτόματα και ο χρήστης λαμβάνει ένα
μήνυμα
78. MessageBox.Show("Προσοχή! Οι τιμές πρέπει να είναι αριθμοί.",
"Λάθος εισαγωγής τιμής", MessageBoxButtons.OK,
MessageBoxIcon.Error)
79. conn.ConnectionString = myConnString
80. Try
```

```

81.     conn.Open ()
82.     Try
83.     myCommand.Connection = conn
84.     myCommand.CommandText = SQL
85.     myAdapter.SelectCommand = myCommand
86.     myAdapter.Fill(myData)
87.     dtgwRefresh.DataSource = myData
88.     Catch ex As Exception
89.     MessageBox.Show("Κατι πήγε στραβά με την αναζήτηση της
εγγραφής στην Βάση Δεδομένων: " & ex.Message, "Λάθος Αναζήτησης")
90.     End Try

91.     Catch ex As Exception
92.     MessageBox.Show("Σφάλμα κατά την σύνδεση στην Βάση Δεδομένων:
" & ex.Message, "Λάθος κατά την Σύνδεση στην Βάση Δεδομένων")
93.     Finally
94.     If conn.State <> ConnectionState.Closed Then conn.Close()

95.     End Try
96.     End If
97.     End If
98. End Sub

```

Βασικό κομμάτι της διαχείρισης της βάσης δεδομένων είναι η τροποποίηση της και στα ήδη υπάρχοντα δεδομένα. Η τροποποίηση των δεδομένων γίνεται από τον διαχειριστή με την αλλαγή κάποιου χαρακτηριστικού απευθείας. Ο τρόπος υλοποίησης της ανανέωσης που έχει επιλεγεί γίνεται αρχικά καλώντας την συνάρτηση αναζήτησης η οποία θα περιγραφεί στην συνέχεια. Αμέσως μετά και αφού ο διαχειριστής βρει στον πίνακα αποτελεσμάτων την εγγραφή που θέλει να τροποποιήσει αλλάζει το χαρακτηριστικό - κελί που επιθυμεί και η αποθήκευση γίνεται αυτόματα.

**Γραμμές 1-20:** Στο κομμάτι αυτό δηλώνονται και πάλι οι μεταβλητές που θα χρησιμοποιηθούν στην συνάρτηση ανανέωσης της εγγραφής και περιέχουν τα στοιχεία της σύνδεσης με την βάση και τα αντικείμενα εκτέλεσης του SQL ερωτήματος και της παρουσίασης των αποτελεσμάτων σε πίνακα.

**Γραμμές 20-25:** Στο κομμάτι αυτό η διαδικασία της αναζήτησης μιας εγγραφής έχει προχωρήσει και ο διαχειριστής έχει βρει την εγγραφή την οποία επιθυμεί να τροποποιήσει. Επειδή δεν θέλουμε κανένα κενό στην εισαγωγή στοιχείων γίνεται έλεγχος πριν την ανανέωση για τυχόν κενά κελιά. Και αυτό γίνεται γιατί στην υλοποίηση της συνάρτησης αναζήτησης έχει επιλεγεί ο διαχειριστής να μπορεί να αναζητήσει με όλα τα χαρακτηριστικά ένα φυτό. Σε περίπτωση που κάποιο από αυτά είναι κενό υπάρχει περίπτωση κάποια εγγραφή να μην εμφανιστεί.

**Γραμμές 28-30:** Σε αυτές τις γραμμές γίνεται έλεγχος για το αν οι τιμές που εισάγει ο διαχειριστής είναι αριθμοί και όχι γράμματα. Το event το οποίο καλείται για να ελέγξει το τμήμα αυτό ενεργοποιείται με το που πατάει ο διαχειριστής το πλήκτρο(keypress).

**Γραμμές 31-56:** Στο κομμάτι αυτό εκτελείται το τμήμα της ανανέωσης της εγγραφής. Τα νέα δεδομένα αποθηκεύονται στην βάση και η εγγραφή αλλάζει ολόκληρη. Στο σημείο αυτό πρέπει να τονίσουμε ότι όποιο πεδίο και να αλλάξει ο χρήστης η εγγραφή ανανεώνεται ολόκληρη. Το κομμάτι περικλείεται και πάλι με την διαδικασία Try – Catch για να πιαστεί πιθανόν λάθος και στην συνέχεια η σύνδεση κλείνει. Τέλος εμφανίζονται τα σχετικά μηνύματα αν πήγε κάτι λάθος.

**Γραμμές 57-74:** Στις γραμμές αυτές γίνεται αναζήτηση και εμφάνιση όλης της βάσης μετά από σχετικό μήνυμα ότι το κομμάτι της ανανέωσης πήγε όπως έπρεπε.

**Γραμμές 75-98:** Στις γραμμές αυτές εμφανίζονται μηνύματα στον διαχειριστή σχετικά με το αν κάτι έχει πάει στραβά με την διαδικασία της ανανέωσης και επιστρέφει και πάλι όλο τον πίνακα αν έχει συμβεί κάτι λάθος.

#### **Συνάρτηση αναζήτησης κάποιας εγγραφής στην βάση δεδομένων φυτών:**

1. 'Συνάρτηση αναζήτησης μιας εγγραφής φυτού στην βάση μας.
2. 'Μπορούμε να αναζητήσουμε με οποιοδήποτε κριτήριο θέλουμε αλλά τουλάχιστον με ένα.
3. 'Σε περίπτωση που ο διαχειριστής πατήσει αναζήτηση χωρίς καμία παράμετρο θα επιστρέφεται όλη η βάση
4. `Private Sub btnSearch_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSearch.Click`
5. 'Δηλώνουμε και πάλι τις μεταβλητές μας
6. `Dim conn As New MySqlConnection`
7. `Dim myCommand As New MySqlCommand`
8. `Dim myAdapter As New MySqlDataAdapter`
9. `Dim myData As New DataTable`
10. `Dim SQL As String = "", totalSQL As String = ""`
11. `Dim one As String = "", two As String = "", three As String = "", four As String = "", five As String = "", six As String = ""`
12. `Dim seven As String = "", eight As String = "", nine As String = "", ten As String = "", eleven As String = "", twelve As String = ""`
13. `Dim thirteen As String = "", fourteen As String = ""`
14. 'Για να κάνουμε αναζήτηση στην βάση πρέπει να κατασκευάσουμε αρχικά το SQL ερώτημα
15. 'Ελέγχουμε τις επιλογές έχει βάλει ο διαχειριστής και κατασκευάζουμε σταδιακά το query.

```

16.      SQL = "SELECT id As Κωδικός, ειδος As 'Είδος Φυτού',
            ειδοςFulomatos As 'Είδος Φυλλόματος', fws As 'Απαιτήσεις Φώς', nero
            As 'Απαιτήσεις Νερό', xrwmaAnthwn As 'Χρώμα Άνθους', epoxiAntou As
            'Εποχή Άνθησης', antoxiPago As 'Άντοχή Παγωνιά', edafokaliergia As
            'Για εαδφοκάλυψη', parathalasio As Παραθαλλάσιο, fraxtes As
            Φράχτες, arwmatikoFulo As 'Αρωματικό Φύλλο', arwmatikoAntho As
            'Αρωματικό Ανθό', posotita As Ποσότητα, kostos As Κόστος, onoma As
            Όνομα FROM plants WHERE id <> '-1' "

17.      If cmbEidos.SelectedItem IsNot Nothing Then
18.      one = " AND ειδος = '" & cmbEidos.SelectedItem.ToString & "'"
19.      End If
20.      If cmbFula.SelectedItem IsNot Nothing Then
21.      two = " AND ειδοςFulomatos = '" &
            cmbFula.SelectedItem.ToString & "'"
22.      End If
23.      If cmbFws.SelectedItem IsNot Nothing Then
24.      three = " AND fws = '" & cmbFws.SelectedItem.ToString & "'"
25.      End If
26.      If cmbNero.SelectedItem IsNot Nothing Then
27.      four = " AND nero = '" & cmbNero.SelectedItem.ToString & "'"
28.      End If
29.      If cmbAnthos.SelectedItem IsNot Nothing Then
30.      five = " AND xrwmaAnthwn = '" &
            cmbAnthos.SelectedItem.ToString & "'"
31.      End If

32.      If cmbEpoxi.SelectedItem IsNot Nothing Then
33.      six = " AND epoxiAntou = '" & cmbEpoxi.SelectedItem.ToString
            & "'"
34.      End If
35.      If cmbPagos.SelectedItem IsNot Nothing Then
36.      seven = " AND antoxiPago = '" &
            cmbPagos.SelectedItem.ToString & "'"
37.      End If
38.      If cmbEdafokalipsi.SelectedItem IsNot Nothing Then
39.      eight = " AND edafokaliergia = '" &
            cmbEdafokalipsi.SelectedItem.ToString & "'"
40.      End If
41.      If cmbThalasa.SelectedItem IsNot Nothing Then
42.      nine = " AND parathalasio = '" &
            cmbThalasa.SelectedItem.ToString & "'"
43.      End If
44.      If Fraxtes.SelectedItem IsNot Nothing Then
45.      ten = " AND fraxtes = '" & Fraxtes.SelectedItem.ToString &
            "'"
46.      End If
47.      If cmbAromFul.SelectedItem IsNot Nothing Then
48.      eleven = " AND arwmatikoFulo = '" &
            cmbAromFul.SelectedItem.ToString & "'"
49.      End If
50.      If cmbAromAnth.SelectedItem IsNot Nothing Then

```

```

51.         twelve = " AND arwmatikoAntho = '" &
           cmbAromAnth.SelectedItem.ToString & "'"
52.     End If
53.     If txtOnoma.Text <> "" Then
54.         thirteen = " AND onoma = '" & txtOnoma.Text & "'"
55.     End If
56.     If txtKwdikos.Text <> "" Then
57.         fourteen = " AND id = '" & txtKwdikos.Text & "'"
58.     End If
59.     'Το τελικό μας SQL αφού το κατασκευάσαμε σύμφωνα με τα
           κριτήρια αναζήτησης του διαχειριστή
60.     totalsQL = SQL + one + two + three + four + five + six +
           seven + eight + nine + ten + eleven + twelve + thirteen + fourteen
61.     conn.ConnectionString = myConnString
62.     Try
63.         conn.Open()
64.     Try
65.         myCommand.Connection = conn
66.         myCommand.CommandText = totalsQL
67.         myAdapter.SelectCommand = myCommand
68.         myAdapter.Fill(myData)
69.         dtgvwResults.DataSource = myData
70.     Catch ex As Exception
71.         MessageBox.Show("Κατι πήγε στραβά με την αναζήτηση της
           εγγραφής στην Βάση Δεδομένων: " & ex.Message, "Λάθος Αναζήτησης")
72.     End Try

73.     Catch ex As Exception
74.         MessageBox.Show("Σφάλμα κατά την σύνδεση στην Βάση Δεδομένων:
           " & ex.Message, "Λάθος κατά την Σύνδεση στην Βάση Δεδομένων")
75.     Finally
76.         If conn.State <> ConnectionState.Closed Then conn.Close()
77.     End Try
78. End Sub

```

Τελευταία συνάρτηση της καρτέλας διαχείρισης της αποθήκης είναι η αναζήτηση μιας εγγραφής στην βάση. Ο τρόπος που έχει επιλεγεί να γίνεται η αναζήτηση από τον διαχειριστή είναι με βάση όποιο χαρακτηριστικό αυτός επιθυμεί. Δηλαδή πέρα από το όνομα ή κωδικό κάποιου φυτού, ο διαχειριστής μπορεί να αναζητήσει και με οποιοδήποτε άλλο χαρακτηριστικό αυτός επιθυμεί επιλέγοντας από τα προεπιλεγμένα στοιχεία των αντικειμένων ComboBox.

**Γραμμές 1-15:** Στις γραμμές αυτές η συνάρτηση είναι παρόμοια με τις προηγούμενες. Δηλώνουμε τις μεταβλητές οι οποίες θα μας χρειαστούν για την συνάρτηση.

**Γραμμές 16-60:** Στις γραμμές αυτές είναι το βασικότερο κομμάτι της συνάρτησης αναζήτησης. Εδώ πρέπει να κατασκευαστεί σωστά το SQL ερώτημα ανάλογα με τα



κριτήρια που θα επιλέξει ο διαχειριστής να αναζητήσει ένα προϊόν. Για το λόγο αυτό αν ο διαχειριστής επιλέξει κάποιο κριτήριο προστίθεται στο τέλος του ερωτήματος το κομμάτι AND "" και το κομμάτι που αναλογεί στο ερώτημα. Για παράδειγμα αν επιλέξει αναζήτηση ανάλογα με το είδος το ερώτημα θα διαμορφωθεί σε: `one = " AND ειδος = "" & cmbEidos.SelectedItem.ToString` το οποίο προσθέτει στο ερώτημα και το κομμάτι του είδους από το αντικείμενο `cmbEidos`.

**Γραμμές 61-78:** Στις γραμμές αυτές η διαδικασία είναι παρόμοια με τις υπόλοιπες συναρτήσεις. Μέσα σε μια διαδικασία Try – Catch προσπαθούμε να κάνουμε σύνδεση με την βάση και να εκτελέσουμε το SQL ερώτημα μας. Με το αντικείμενο `DataAdapter` το εμφανίζουμε στον πίνακα που έχουμε για εμφάνιση αποτελεσμάτων. Τέλος ελέγχουμε αν η σύνδεση είναι ανοιχτή και την κλείνουμε και παρουσιάζουμε τυχόν μηνύματα σε περίπτωση λάθους.

### 3.3 Καρτέλα Διαχείρισης Πελατών

Η επόμενη βασική λειτουργία της εφαρμογής είναι η διαχείριση της βάσης δεδομένων των πελατών. Η λειτουργία αυτή βρίσκεται στην δεύτερη καρτέλα της φόρμας και περιλαμβάνει όλες τις δυνατές λειτουργίες που μπορεί να κάνει ο διαχειριστής σχετικά με την βάση δεδομένων των πελατών. Στην καρτέλα αυτή μπορεί να επιλέξει μια από τις τέσσερις βασικές λειτουργίες.

- Εισαγωγή νέου πελάτη
- Διαγραφή πελάτη
- Ανανέωση εγγραφής πελάτη
- Αναζήτηση πελάτη
- Χρέωση πελάτη

**Συνάρτηση εισαγωγής νέου πελάτη:**

## Πτυχιακή εργασία της φοιτήτριας Βεγερτζή Κοκκώνης

```
1. 'Συνάρτηση εισαγωγής νέας εγγραφής στην βάση στοιχεία όπου είναι η
   βάση η οποία κρατάει τα στοιχεία των πελατών μας
2. Private Sub btnPelEis_Click(ByVal sender As System.Object, ByVal e
   As System.EventArgs) Handles btnPelEis.Click
3. 'Δήλωση μεταβλητών
4. Dim conn As New MySqlConnection
5. Dim myCommand As New MySqlCommand
6. Dim SQL As String = Nothing
7. Dim formattedDate As String

8. Dim now As Date
9. now = System.DateTime.Today
10. 'Η ημερομηνιά εγγραφής του χρήστη σε μορφή dd-MM-yy
11. formattedDate = now.ToString("dd-MM-yy")
12. If txtPelEis1.Text = Nothing Or txtPelEis2.Text = Nothing Or
   txtPelEis3.Text = Nothing Or txtPelEis4.Text = Nothing _
13. Or txtPelEis5.Text = Nothing Or txtPelEis6.Text = Nothing
   Then
14.     MessageBox.Show("Πρέπει να συμπληρωθούν όλα τα πεδία για την
   εισαγωγή νέου πελάτη.", "Λάθος εισαγωγής", MessageBoxButtons.OK,
   MessageBoxIcon.Information)
15. Else
16.     SQL = "INSERT INTO stoiceia VALUES (NULL, '" & txtPelEis2.Text
   & "', '" & txtPelEis1.Text & "', '" & txtPelEis3.Text & "', '" &
   txtPelEis4.Text _
17. & "', '" & formattedDate & "', '" & txtPelEis5.Text & "', '" &
   txtPelEis6.Text & "', NULL, NULL) "
18.     conn.ConnectionString = myConnString
19.     Try
20.         conn.Open()
21.     Try
22.         myCommand.Connection = conn
23.         myCommand.CommandText = SQL
24.         myCommand.ExecuteNonQuery()
25.         txtPelEis1.Text = ""
26.         txtPelEis2.Text = ""
27.         txtPelEis3.Text = ""
28.         txtPelEis4.Text = ""
29.         txtPelEis5.Text = ""
30.         txtPelEis6.Text = ""

31.     MessageBox.Show("Η εισαγωγή νέου πελάτη ολοκληρώθηκε με
   επιτυχία.", "Εισαγωγή νέου πελάτη", MessageBoxButtons.OK,
   MessageBoxIcon.Information)
32.     Catch ex As Exception
33.         MessageBox.Show("Κάτι πήγε στραβά στην σύνδεση με την βάση: "
   & ex.Message, "Λάθος εγγραφής στην βάση", MessageBoxButtons.OK,
   MessageBoxIcon.Error)
34.     End Try

35.     Catch ex As Exception
```

```
36.     MessageBox.Show("Κάτι πήγε στραβά στην σύνδεση με την βάση: "  
    & ex.Message, "Λάθος σύνδεσης στην βάση", MessageBoxButtons.OK,  
    MessageBoxIcon.Error)  
37.     Finally  
38.     If conn.ConnectionTimeout <> ConnectionState.Closed Then  
    conn.Clone()  
39.     End Try  
40.     End If  
41. End Sub
```

Η συνάρτηση αυτή είναι υπεύθυνη για την εισαγωγή νέου πελάτη στην βάση δεδομένων πελατών. Η διαδικασία αυτής της συνάρτησης γίνεται με την εισαγωγή των στοιχείων από τον διαχειριστή και μετά με την εκτέλεση του ερωτήματος ολοκληρώνεται η εισαγωγή στην βάση.

**Γραμμές 1-14:** Στις γραμμές αυτές δηλώνονται αρχικά οι μεταβλητές που είναι απαραίτητες για την υλοποίηση της διαδικασίας εισαγωγή. Έχουμε το αντικείμενο σύνδεσης με την βάση το οποίο περιέχει τις πληροφορίες σχετικά με τον πίνακα και την βάση στα οποία επιθυμούμαι να έχουμε πρόσβαση. Γίνεται έλεγχος αν όλα τα στοιχεία είναι συμπληρωμένα και σε αντίθετη περίπτωση εμφανίζουμε σχετικό μήνυμα. Ο έλεγχος είναι σημαντικός λόγω του τρόπου υλοποίησης της συνάρτησης αναζήτησης. Ο διαχειριστής μπορεί να αναζητήσει κάποιο πελάτη με όλα τα χαρακτηριστικά τα οποία κάνει την αποθήκευση. Σε περίπτωση που υπάρχει κάποια κενή εγγραφή η αναζήτηση θα επιστρέψει ημιτελή αποτελέσματα.

**Γραμμές 15-41:** Στις γραμμές αυτές κατασκευάζουμε αρχικά το SQL ερώτημα το οποίο θέλουμε να εκτελέσουμε. Δημιουργούμε και πάλι διαδικασίες Try – Catch και μέσα σε αυτές προσπαθούμε να ανοίξουμε την σύνδεση και να εκτελέσουμε το ερώτημα. Εμφανίζουμε τα μηνύματα σε περίπτωση λάθους είτε της εκτέλεσης του ερωτήματος είτε της σύνδεσης με την βάση. Σε αντίθετη περίπτωση εμφανίζουμε το μήνυμα επιτυχίας στον διαχειριστή. Τέλος κλείνουμε την σύνδεση με την βάση.

#### **Συνάρτηση διαγραφής μιας εγγραφής πελάτη στην βάση δεδομένων πελατών:**

```
1. Private Sub btnPelDelete_Click(ByVal sender As System.Object, ByVal  
    e As System.EventArgs) Handles btnPelDelete.Click  
2. 'Δηλώνουμε και πάλι τις μεταβλητές μας  
3. Dim conn As New MySqlConnection  
4. Dim myCommand As New MySqlCommand
```

```
5. Dim myAdapter As New MySqlDataAdapter
6. Dim myData As New DataTable
7. Dim SQL As String = ""
8. Dim selectedRow As Integer
9. 'Αποθήκευση της σειράς που επέλεξε ο διαχειριστής.
10.     If dtgvwPelDel.Rows.Count > 0 Then
11.     If dtgvwPelDel.SelectedRows.Count > 0 Then
12.         selectedRow = dtgvwPelDel.CurrentRow.Index
13.         'Για να κάνουμε αναζήτηση στην βάση πρέπει να κατασκευάσουμε
           αρχικά το SQL ερώτημα
14.         'Ελέγχουμε τις επιλογές έχει βάλει ο διαχειριστής και
           κατασκευάζουμε σταδιακά το query.
15.     If MessageBox.Show("Η εγγραφή αυτή θα διαγραφεί απο την Βάση
           Δεδομένων.Πατήστε Ναι για συνέχεια ή Όχι για ακύρωση", "Διαγραφή
           Εγγραφής", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) =
           DialogResult.Yes Then
16.         SQL = "DELETE FROM stoixeia WHERE id='" &
           Integer.Parse(dtgvwPelDel.Item(0, selectedRow).Value.ToString) &
           "'"
17.         conn.ConnectionString = myConnString
18.         Try
19.             conn.Open()
20.         Try
21.             myCommand.Connection = conn
22.             myCommand.CommandText = SQL
23.             myCommand.ExecuteNonQuery()
24.         Catch ex As Exception
25.             MessageBox.Show("Κατι πήγε στραβά με την αναζήτηση της
           εγγραφής στην Βάση Δεδομένων: " & ex.Message, "Λάθος Αναζήτησης")
26.         End Try
27.         Catch ex As Exception
28.             MessageBox.Show("Σφάλμα κατα την σύνδεση στην Βάση Δεδομένων:
           " & ex.Message, "Λάθος κατα την Σύνδεση στην Βάση Δεδομένων")
29.         Finally
30.             If conn.State <> ConnectionState.Closed Then conn.Close()
31.         End Try
32.         Else
33.             'Αν ο χρήστης πατήσει όχι γίνεται εκ νέου εμφάνιση όλης της
           βάσης.
34.             SQL = "SELECT id As Κωδικος, lastname As Επίθετο, firstname
           As Όνομα, afm As ΑΦΜ, company As Εταιρία, signdate As 'Ημερομηνία
```

```
καταχώρησης', location As Περιοχή, telephone As Τηλέφωνο, xrwstaei
As Υπόλοιπο, paragelies As Παραγγελίες FROM stoiceia WHERE id <> '-
1'"
35.     conn.ConnectionString = myConnString
36.     Try
37.     conn.Open()
38.     Try
39.     myCommand.Connection = conn
40.     myCommand.CommandText = SQL
41.     myAdapter.SelectCommand = myCommand
42.     myAdapter.Fill(myData)
43.     dtgvwPelDel.DataSource = myData
44.     Catch ex As Exception
45.     MessageBox.Show("Κατι πήγε στραβά με την αναζήτηση της
εγγραφής στην Βάση Δεδομένων: " & ex.Message, "Λάθος Αναζήτησης")
46.     End Try
47.     Catch ex As Exception
48.     MessageBox.Show("Σφάλμα κατα την σύνδεση στην Βάση Δεδομένων:
" & ex.Message, "Λάθος κατα την Σύνδεση στην Βάση Δεδομένων")
49.     Finally
50.     If conn.State <> ConnectionState.Closed Then conn.Close()
51.     End Try
52.     End If
53.     'Εδώ κλείνει ο έλεγχος για το αν ο χρήστης έχει επιλέξει
κάποια σειρά για να σβήσει.
54.     Else
55.     MessageBox.Show("Πρέπει να επιλέξετε μια σειρά για να
διαγραφή η εγγραφή", "Διαγραφή εγγραφής", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation)
56.     End If
57.     Else
58.     MessageBox.Show("Πρέπει να επιλέξετε μια σειρά για να
διαγραφή η εγγραφή", "Διαγραφή εγγραφής", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation)
59.     End If
60. End Sub
```

Η διαδικασία διαγραφής ενός πελάτη από την βάση είναι παρόμοια με αυτήν στην καρτέλα διαχείρισης της αποθήκης. Οι αλλαγές οι οποίες έχουν γίνει είναι απλά στον πίνακα τον οποίο έχουμε πρόσβαση για να διαγράψουμε τις αντίστοιχες εγγραφές. Αρχικά

ο διαχειριστής μπορεί να αναζητήσει την εγγραφή την οποία επιθυμεί να διαγράψει από την καρτέλα διαγραφής πελάτη. Στην συνέχεια εμφανίζονται τα αποτελέσματα της αναζήτησης στον πίνακα αποτελεσμάτων της καρτέλας. Ο διαχειριστής πρέπει να επιλέξει μια εγγραφή και να πατήσει το πλήκτρο διαγραφής. Στην συνέχεια εμφανίζεται ένας διάλογος με τον οποίο το σύστημα προειδοποιεί τον διαχειριστή για την διαγραφή. Αν αυτός πατήσει Yes η εγγραφή διαγράφεται και εμφανίζεται σχετικό μήνυμα από το σύστημα. Σε αντίθετη περίπτωση ο πίνακας απλά ξαναγεμίζει με όλη την βάση.

### **Συνάρτηση ανανέωσης μιας εγγραφής πελάτη στην βάση δεδομένων πελατών:**

1. 'Η συνάρτηση αυτή καλείται κάθε φορά που ο διαχειριστής θα αλλάξει κάποια τιμή στις εγγραφές που εμφανίζονται μετά την ανανέωση
2. `Private Sub dtgwPelAnanewsi_CellValueChanged(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles dtgwPelAnanewsi.CellValueChanged`
3. `Dim conn As New MySqlConnection`
4. `Dim myCommand As New MySqlCommand`
5. `Dim myAdapter As New MySqlDataAdapter`
6. `Dim myData As New DataTable`
7. `Dim SQL As String = Nothing`
8. 'Αποθηκεύουμε τον αριθμό της γραμμής την οποία επέλεξε ο διαχειριστής για να ανανεώσει
9. `Dim selectedRow As Integer = 0`
10. `selectedRow = dtgwPelAnanewsi.CurrentRow.Index`
11. 'Αν ο διαχειριστής προσπαθήσει να ανανεώσει με κενό κάποια τιμή του εμφανίζεται σχετικό μήνυμα
12. `If dtgwPelAnanewsi.Item(1, selectedRow).Value.ToString.Length = 0 Or dtgwPelAnanewsi.Item(2, selectedRow).Value.ToString.Length = 0 Or dtgwPelAnanewsi.Item(3, selectedRow).Value.ToString.Length = 0 Or dtgwPelAnanewsi.Item(4, selectedRow).Value.ToString.Length = 0 Or dtgwPelAnanewsi.Item(5, selectedRow).Value.ToString.Length = 0 Or dtgwPelAnanewsi.Item(6, selectedRow).Value.ToString.Length = 0 Or dtgwPelAnanewsi.Item(7, selectedRow).Value.ToString.Length = 0 Then`
13. `MessageBox.Show("Δεν πρέπει να υπάρχει κενή εγγραφή για την ανανέωση.", "Λάθος κατά την ανανέωση", MessageBoxButtons.OK, MessageBoxIcon.Information)`

```
14.     Else
15.         'Κάνουμε έλεγχο αν στις τιμές ΑΦΜ και Τηλέφωνο ο διαχειριστής
           βάζει μόνο αριθμό
16.         If IsNumeric(dtgwPelAnanewsi.Item(3, selectedRow).Value) And
           IsNumeric(dtgwPelAnanewsi.Item(7, selectedRow).Value) Then
17.             SQL = "UPDATE stoiceia SET lastname='" &
           dtgwPelAnanewsi.Item(1, selectedRow).Value.ToString &
           "',firstname='" & dtgwPelAnanewsi.Item(2,
           selectedRow).Value.ToString & "',afm='" & dtgwPelAnanewsi.Item(3,
           selectedRow).Value.ToString & "',company='" _
18.             & dtgwPelAnanewsi.Item(4, selectedRow).Value.ToString &
           "',signdate='" & dtgwPelAnanewsi.Item(5,
           selectedRow).Value.ToString & "',location='" &
           dtgwPelAnanewsi.Item(6, selectedRow).Value.ToString &
           "',telephone='" & dtgwPelAnanewsi.Item(7,
           selectedRow).Value.ToString & "'" _
19.             & "WHERE id='" & Integer.Parse(dtgwPelAnanewsi.Item(0,
           selectedRow).Value.ToString) & "'"
20.             conn.ConnectionString = myConnString
21.             Try
22.                 conn.Open()
23.             Try
24.                 myCommand.Connection = conn
25.                 myCommand.CommandText = SQL
26.                 myAdapter.SelectCommand = myCommand
27.                 myAdapter.Fill(myData)
28.                 dtgwPelAnanewsi.DataSource = myData
29.             Catch ex As Exception
30.                 MessageBox.Show("Κατι πήγε στραβά με την αναζήτηση της
           εγγραφής στην Βάση Δεδομένων: " & ex.Message, "Λάθος Αναζήτησης")
31.             End Try

32.             Catch ex As Exception
33.                 MessageBox.Show("Σφάλμα κατα την σύνδεση στην Βάση Δεδομένων:
           " & ex.Message, "Λάθος κατα την Σύνδεση στην Βάση Δεδομένων")
34.             Finally
35.                 If conn.State <> ConnectionState.Closed Then conn.Close()
36.             End Try
37.             Else
38.                 'Αν κάνει λάθος και προσπαθήσει να εισάγει γράμματα εκεί που
           υπάρχουν μόνο αριθμοί
```

```
39.     MessageBox.Show("Λάθος κατα την ειδαγωγή της τιμής.Μόνο  
        αριθμούς", "Λάθος παραμετροποίησης", MessageBoxButtons.OK,  
        MessageBoxIcon.Error)  
40.     End If  
41.     End If  
42. End Sub
```

Η διαδικασία της ανανέωσης των στοιχείων στην βάση δεδομένων είναι παρόμοια με την διαδικασία ανανέωσης ενός προϊόντος στην αποθήκη φυτών. Ο διαχειριστής πρέπει να αναζητήσει μέσα από την καρτέλα ανανέωσης την εγγραφή που θέλει να ανανεώσει. Τα αποτελέσματα εμφανίζονται στον πίνακα αποτελεσμάτων της καρτέλας. Ο διαχειριστής στην συνέχεια μπορεί να πατήσει σε όποιο χαρακτηριστικό θελήσει και να το τροποποιήσει. Μετά το τέλος της τροποποίησης το αποτέλεσμα αποθηκεύεται εκ νέου στην βάση.

**Γραμμές 1-14:** Στις γραμμές αυτές δηλώνονται οι μεταβλητές οι οποίες θα χρειαστούν στην συνάρτηση. Εκεί αποθηκεύουμε τις τιμές για το αντικείμενο της σύνδεσης και της εκτέλεσης του SQL ερωτήματος. Επίσης ελέγχουμε αν κατά την ανανέωση ο διαχειριστής έχει αφήσει κάποιο κενό κελί. Σε αυτήν την περίπτωση εμφανίζουμε σχετικό μήνυμα.

**Γραμμές 14-16:** Στις γραμμές αυτές καλείται μια εξωτερική συνάρτηση ελέγχου αν οι τιμές που δίνει ο διαχειριστής είναι αριθμοί. Στα πεδία ΑΦΜ, τηλέφωνο ο διαχειριστής δεν μπορεί να δώσει γράμματα σαν τιμή αλλά μόνο αριθμούς.

**Γραμμές 17-42:** Στις γραμμές αυτές κατασκευάζεται αρχικά το SQL ερώτημα για την ανανέωση της εγγραφής. Στην συνέχεια μέσα στην διαδικασία Try – Catch προσπαθούμε να κάνουμε την σύνδεση και να εκτελέσουμε το ερώτημα. Σε περίπτωση λάθους εμφανίζονται σχετικά μηνύματα. Τέλος ελέγχουμε αν η σύνδεση με την βάση είναι ακόμα ανοιχτή και τελειώνοντας την κλείνουμε.

#### ***Συνάρτηση αναζήτησης μιας εγγραφής στην βάση δεδομένων πελατών:***

```
1. 'Αναζήτηση εγγραφής πελάτη απο την βάση δεδομένων stoixeia  
2. 'Η αναζήτηση μπορεί να γίνει με οποιαδήποτε κριτήρια  
3. Private Sub btnPelSearch_Click(ByVal sender As System.Object, ByVal  
   e As System.EventArgs) Handles btnPelSearch.Click  
4. Dim conn As New MySqlConnection
```



```
5. Dim myCommand As New MySqlCommand
6. Dim myAdapter As New MySqlDataAdapter
7. Dim myData As New DataTable
8. Dim SQL As String = "", totalSql As String = ""
9. Dim one As String = "", two As String = "", three As String = "",
    four As String = "", five As String = "", six As String = "", seven
    As String = ""
10.     'Αυτό είναι το βασικό μας SQL ερώτημα για το οποίο πρέπει να
        προσθέσουμε τις επιλογές του χρήστη για να δώσει το σωστό
        αποτέλεσμα
11.     SQL = "SELECT id As Κωδικός, lastname As Επίθετο, firstname
        As Όνομα, afm As ΑΦΜ, company As Εταιρία, signdate As 'Ημερομηνία
        καταχώρησης', location As Περιοχή, telephone As Τηλέφωνο, xrwstaei
        As Υπόλοιπο, paragelies As Παραγγελίες FROM stoixeia WHERE id <> '-
        1'"

12.     If txtPelSear2.Text <> "" Then
13.         one = "AND lastname = '" & txtPelSear2.Text & "'"
14.     End If
15.     If txtPelSear1.Text <> "" Then
16.         two = "AND firstname = '" & txtPelSear1.Text & "'"
17.     End If
18.     If txtPelSear3.Text <> "" Then
19.         three = "AND afm = '" & txtPelSear3.Text & "'"
20.     End If
21.     If txtPelSear4.Text <> "" Then
22.         four = "AND company = '" & txtPelSear4.Text & "'"
23.     End If
24.     If txtPelSear5.Text <> "" Then
25.         five = "AND location = '" & txtPelSear5.Text & "'"
26.     End If
27.     If txtPelSear6.Text <> "" Then
28.         six = "AND telephone = '" & txtPelSear6.Text & "'"
29.     End If
30.     If txtPelSea7.Text <> "" Then
31.         seven = "AND id = '" & txtPelSea7.Text & "'"
32.     End If
33.     totalSql = SQL + one + two + three + four + five + six +
        seven
34.     'MessageBox.Show(totalSql)
35.     conn.ConnectionString = myConnString
```

```
36. Try
37. conn.Open()
38. Try
39. myCommand.Connection = conn
40. myCommand.CommandText = totalSql
41. myAdapter.SelectCommand = myCommand
42. myAdapter.Fill(myData)
43. dtgvwPelSearch.DataSource = myData
44. Catch ex As Exception
45. MessageBox.Show("Κατι πήγε στραβά με την αναζήτηση της
εγγραφής στην Βάση Δεδομένων: " & ex.Message, "Λάθος Αναζήτησης")
46. End Try
47. Catch ex As Exception
48. MessageBox.Show("Σφάλμα κατά την σύνδεση στην Βάση Δεδομένων:
" & ex.Message, "Λάθος κατά την Σύνδεση στην Βάση Δεδομένων")
49. Finally
50. If conn.State <> ConnectionState.Closed Then conn.Close()
51. End Try
52. End Sub
```

Προτελευταία συνάρτηση στην καρτέλα διαχείρισης των πελατών είναι η αναζήτηση ενός πελάτη στην βάση. Κατά την διαδικασία της αναζήτησης ισχύει ότι και στην αναζήτηση ενός προϊόντος στην βάση των φυτών. Ο διαχειριστής μπορεί να αναζητήσει μια εγγραφή με όποιο κριτήριο επιθυμεί. Σημαντικό στοιχείο στην συνάρτηση αυτή είναι η κατασκευή του ερωτήματος στην βάση ανάλογα με τα κριτήρια που επιθυμεί.

**Γραμμές 11-34:** Στις γραμμές αυτές ελέγχουμε ποια πεδία (TextBoxes) έχει συμπληρώσει ο διαχειριστής. Αν κάποιο από αυτά έχει κάποια εγγραφή το προσθέτουμε στο τέλος του SQL ερωτήματος με την μορφή `one = "AND παράμετρος = " & αντίστοιχο textbox & ""`. Έτσι επιτυγχάνεται το σταδιακό χτίσιμο του ερωτήματος ανάλογα με τα κριτήρια που επιθυμεί ο διαχειριστής. Σε περίπτωση που δεν συμπληρωθεί κανένα συγκεκριμένο χαρακτηριστικό για αναζήτηση και πατηθεί το κουμπί αναζήτηση επιστρέφεται όλη η βάση δεδομένων στον διαχειριστή.

#### **Συνάρτηση χρέωσης πελάτη:**

Η συνάρτηση χρέωσης πελάτη είναι αποτέλεσμα συνδυασμού τριών συναρτήσεων. Αρχικά ο διαχειριστής πρέπει να αναζητήσει τον πελάτη στον οποίο θέλει να κάνει την χρέωση. Μπορεί να τον αναζητήσει είτε με όνομα, είτε με επίθετο είτε με τον κωδικό του.

Η διαδικασία της αναζήτησης εδώ είναι διαφορετική από τις συναρτήσεις στις προηγούμενες καρτέλες. Εδώ καλούμε τον event handler του key press μέσα στο text box. Δηλαδή όποτε ο διαχειριστής πατήσει ένα πλήκτρο μέσα στο text box γίνεται αναζήτηση στην βάση με ότι κείμενο έχει εκείνη την στιγμή και επιστρέφεται η πρώτη εγγραφή που συναντάται. Για τον σκοπό αυτό η κατασκευή του ερωτήματος αναζήτησης της SQL χρησιμοποιεί τον τελεστή LIKE. Με το LIKE το ερώτημα ψάχνει για εγγραφές που μοιάζουν με αυτό και όχι που είναι ακριβώς ίδιο. Για τον λόγο αυτό μπορεί να πάρουμε αρχικά πολύ διαφορετικά αποτελέσματα από αυτά που θα πάρουμε αν ολοκληρώσουμε την μεταβλητή αναζήτησης.

Στην συνέχεια και αφού ο διαχειριστής έχει ελέγξει την ορθότητα του αποτελέσματος αναζήτησης του πρέπει να επιλέξει τα προϊόντα – φυτά τα οποία θέλει να χρεώσει στον πελάτη. Αυτό γίνεται αναζητώντας τα είτε με κωδικό είτε με όνομα. Η αναζήτηση γίνεται με τον ίδιο τρόπο που γίνεται και η αναζήτηση του πελάτη. Στα αποτελέσματα εμφανίζονται τα διαθέσιμα αποθέματα του προϊόντος και το κόστος του κάθε τεμαχίου. Ο διαχειριστής μπορεί να τροποποιήσει και το κόστος το οποίο θέλει να χρεώσει το κάθε τεμάχιο αλλά και τα τεμάχια τα οποία θέλει να χρεώσει. Στην συνέχεια πρέπει να πατήσει το πλήκτρο προσθήκη για να μπει στην φόρμα χρέωσης. Μπορεί να χρεώσει μέχρι 20 διαφορετικά προϊόντα την φορά, να αφαιρέσει όποια θέλει και στην συνέχεια να πατήσει το κουμπί για ολοκλήρωση της χρέωσης. Με το κουμπί ολοκλήρωσης της αγοράς ενημερώνονται και οι δυο βάσεις τόσο της αποθήκης όσο και αυτής που διατηρεί τους πελάτες. Από την βάση δεδομένων των φυτών αφαιρεί τα τεμάχια που θέλει να χρεώσει ενώ σε αυτήν των πελατών χρεώνει τα τεμάχια και το συνολικό κόστος της παραγγελίας. Το αποτέλεσμα της χρέωσης γίνεται αυτόματα από το σύστημα.

### **Κομμάτι της συνάρτησης αναζήτησης ενός πελάτη:**

1. Αυτό είναι το βασικό μας SQL ερώτημα για το οποίο πρέπει να προσθέσουμε τις επιλογές του χρήστη για να δώσει το σωστό αποτέλεσμα
2. 

```
SQL = "SELECT id As Κωδικος, lastname As Επίθετο, firstname As Όνομα, afm As ΑΦΜ, company As Εταιρία, signdate As 'Ημερομηνία καταχώρησης', location As Περιοχή, telephone As Τηλέφωνο, xrwstaei As Υπόλοιπο, paragelies As Παραγγελίες FROM stoixeia WHERE id = '" & txtPelCode.Text & "'"
```
3. 

```
MessageBox.Show(totalSql)
```

```
4. conn.ConnectionString = myConnString
5. Try
6. conn.Open()
7. Try
8. myCommand.Connection = conn
9. myCommand.CommandText = SQL
10. myAdapter.SelectCommand = myCommand
11. Dim myReader As MySqlDataReader =
    myCommand.ExecuteReader(CommandBehavior.CloseConnection)
12. While myReader.Read
13.     TextBox1.Text = (myReader.GetValue(0))
14.     TextBox2.Text = (myReader.GetValue(2))
15.     TextBox3.Text = (myReader.GetValue(1))
16.     TextBox4.Text = (myReader.GetValue(3))
17.     TextBox5.Text = (myReader.GetValue(4))
18.     TextBox6.Text = (myReader.GetValue(6))
19.     TextBox7.Text = (myReader.GetValue(7))
20. End While
```

### ***Κομμάτι της συνάρτησης προσθήκη προϊόντος:***

```
1. If numberOfRows = 0 Then
2.     'TextBox12.Text = TextBox8.Text & " " & onomaFutou & " " &
    TextBox10.Text & " " & TextBox11.Text & Chr(10)
3. DataGridView1.Item(0, 0).Value = TextBox8.Text
4. DataGridView1.Item(1, 0).Value = onomaFutou
5. DataGridView1.Item(2, 0).Value = TextBox10.Text
6. DataGridView1.Item(3, 0).Value = TextBox11.Text
7. 'Υπολογίζει το συνολικό κόστος ανάλογα με τον αριθμό των προϊόντων
    και τα τεμάχια
8. DataGridView1.Item(4, numberOfRows).Value =
    Integer.Parse(DataGridView1.Item(2, numberOfRows).Value) *
    Integer.Parse(DataGridView1.Item(3, numberOfRows).Value)
9. numberOfRows += 1
10.     ' DataGridView1.RowCount += 1
11.     'Για τις υπόλοιπες εγγραφές στο πλέγμα
12. Else
13.     'TextBox12.AppendText(TextBox8.Text & " " & onomaFutou & " "
    & TextBox10.Text & " " & TextBox11.Text & Chr(10))
14.     DataGridView1.Item(0, numberOfRows).Value = TextBox8.Text
15.     DataGridView1.Item(1, numberOfRows).Value = onomaFutou
16.     DataGridView1.Item(2, numberOfRows).Value = TextBox10.Text
17.     DataGridView1.Item(3, numberOfRows).Value = TextBox11.Text
18.     'Υπολογίζει το συνολικό κόστος ανάλογα με τον αριθμό των
    προϊόντων και τα τεμάχια
19.     DataGridView1.Item(4, numberOfRows).Value =
    Integer.Parse(DataGridView1.Item(2, numberOfRows).Value) *
    Integer.Parse(DataGridView1.Item(3, numberOfRows).Value)
```

20.           numberOfRow += 1

**Κομμάτι της συνάρτησης αναζήτηση προϊόντος:**

```

1. 'Αυτό είναι το βασικό μας SQL ερώτημα για την αναζήτηση
2. SQL = "SELECT * FROM plants WHERE onoma LIKE '" & TextBox9.Text &
   "%'"
3. 'MessageBox.Show(totalSql)
4. conn.ConnectionString = myConnString
5. Try
6. conn.Open()
7. Try
8. myCommand.Connection = conn
9. myCommand.CommandText = SQL
10.       ' myAdapter.SelectCommand = myCommand

11.       'Dim fdRead As Odbc.OdbcDataReader =
   fdCom.ExecuteReader(CommandBehavior.CloseConnection)
12.       Dim myReader As MySqlDataReader =
   myCommand.ExecuteReader(CommandBehavior.CloseConnection)
13.       While myReader.Read
14.           TextBox8.Text = (myReader.GetValue(0))
15.           TextBox10.Text = (myReader.GetValue(13))
16.           TextBox11.Text = (myReader.GetValue(14))
17.           onomaFutou = (myReader.GetValue(15))
18. End While
    
```

**Κομμάτι της συνάρτησης ολοκλήρωσης της παραγγελίας:**

```

1. If TextBox1.Text <> "" And numberOfRow > 0 Then
2. 'Περιμένουμε ο διαχειριστής να πατήσει ΝΑΙ ή ΟΧΙ για την ολοκλήρωση
   της παραγγελίας
3. If MessageBox.Show("Θέλετε να κάνετε την παρακάτω αγορά?",
   "Ολοκλήρωση παραγγελίας", MessageBoxButtons.YesNo,
   MessageBoxIcon.Question) = Windows.Forms.DialogResult.Yes Then
4. 'Κάνουμε άθροισμα της ποσότητας και του συνολικού κόστους
5. For i = 0 To numberOfRow - 1
6. If DataGridView1.Item(0, numberOfRow - 1).Value <> "" Then
7. sumUp += DataGridView1.Item(4, i).Value
8. posotita += DataGridView1.Item(2, i).Value

9. 'MessageBox.Show(DataGridView1.Item(2, numberOfRow - 1).Value)

10.       End If
11.       Next
12.       'Και εδώ ανανεώνουμε την καρτέλα πελάτη σύμφωνα με τις αγορές
   του
13.       Dim conn As New MySqlConnection
14.       Dim myCommand As New MySqlCommand
15.       Dim myAdapter As New MySqlDataAdapter
16.       Dim myData As New DataTable
17.       Dim SQL As String = ""
18.       'Αυτό είναι το βασικό μας SQL ερώτημα
19.       SQL = "UPDATE stoxeia SET xrwstaei='" & sumUp &
   "' ,paragelies='" & posotita & "' WHERE id='" & TextBox1.Text & ""
20.       'MessageBox.Show(totalSql)
21.       conn.ConnectionString = myConnString
    
```

```
22.     Try
23.     conn.Open()
24.     Try
25.     myCommand.Connection = conn
26.     myCommand.CommandText = SQL
27.     myCommand.ExecuteNonQuery()
28.     Catch ex As Exception
29.     MessageBox.Show("Κατι πήγε στραβά με την αναζήτηση της
εγγραφής στην Βάση Δεδομένων: " & ex.Message, "Λάθος Αναζήτησης")
30.     End Try
31.     Catch ex As Exception
32.     MessageBox.Show("Σφάλμα κατα την σύνδεση στην Βάση Δεδομένων:
" & ex.Message, "Λάθος κατα την Σύνδεση στην Βάση Δεδομένων")
33.     Finally
34.     If conn.State <> ConnectionState.Closed Then conn.Close()

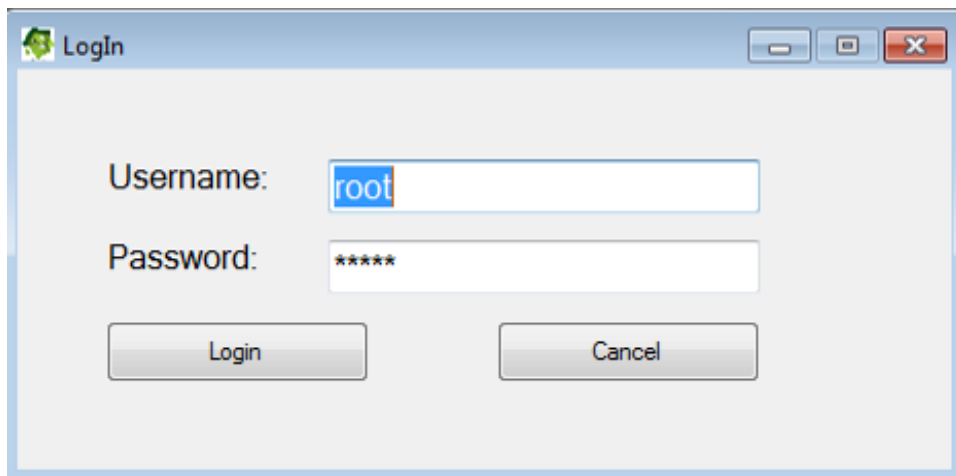
35.     End Try
36.     End If
37.     Else
38.     MessageBox.Show("Δεν έχετε επιλέξει πελάτη ή προϊόν", "Λάθος
κατα την ολοκλήρωση της παραγγελίας", MessageBoxButtons.OK,
MessageBoxIcon.Information)
39. End If
```

## Επίλογος

Στο κεφάλαιο αυτό μελετήσαμε τις βασικές λειτουργίες της εφαρμογής. Επίσης είδαμε τον τρόπο που επιλέχτηκε να υλοποιηθεί το κάθε κομμάτι της υλοποίησης. Έγινε παράθεση του κώδικα και επεξήγηση της λογικής και των σημαντικών κομματιών του κώδικα. Στο επόμενο κεφάλαιο παρουσιάζεται ένα εγχειρίδιο χρήσης με τις βασικές λειτουργίες που μπορεί να εκτελέσει ο διαχειριστής.

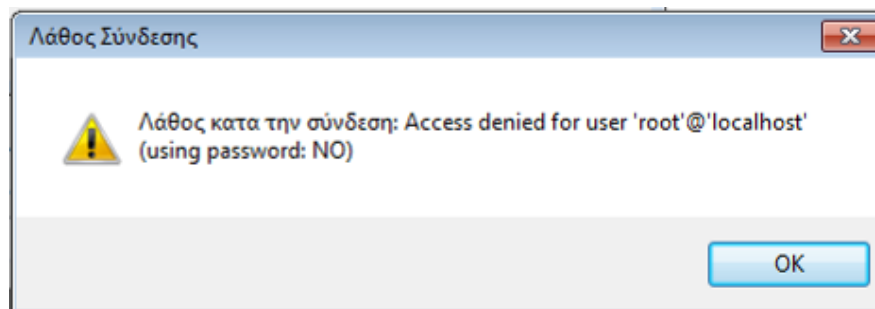
## ΚΕΦΑΛΑΙΟ 4 – Εγχειρίδιο Χρήσης

Κατά την εκκίνηση της εφαρμογής εμφανίζεται στον διαχειριστή η φόρμα Log In. Είναι η φόρμα η οποία πιστοποιεί ότι ο διαχειριστής γνωρίζει τα συνθηματικά και είναι αυτός που υποστηρίζει πως είναι. Τα συνθηματικά ξεκινούν σαν default επιλογές στην εφαρμογή μας οπότε απλά πατάμε Log In. Αυτό έχει γίνει απλά για χάριν ευκολίας στην διόρθωση των λαθών της άσκησης.



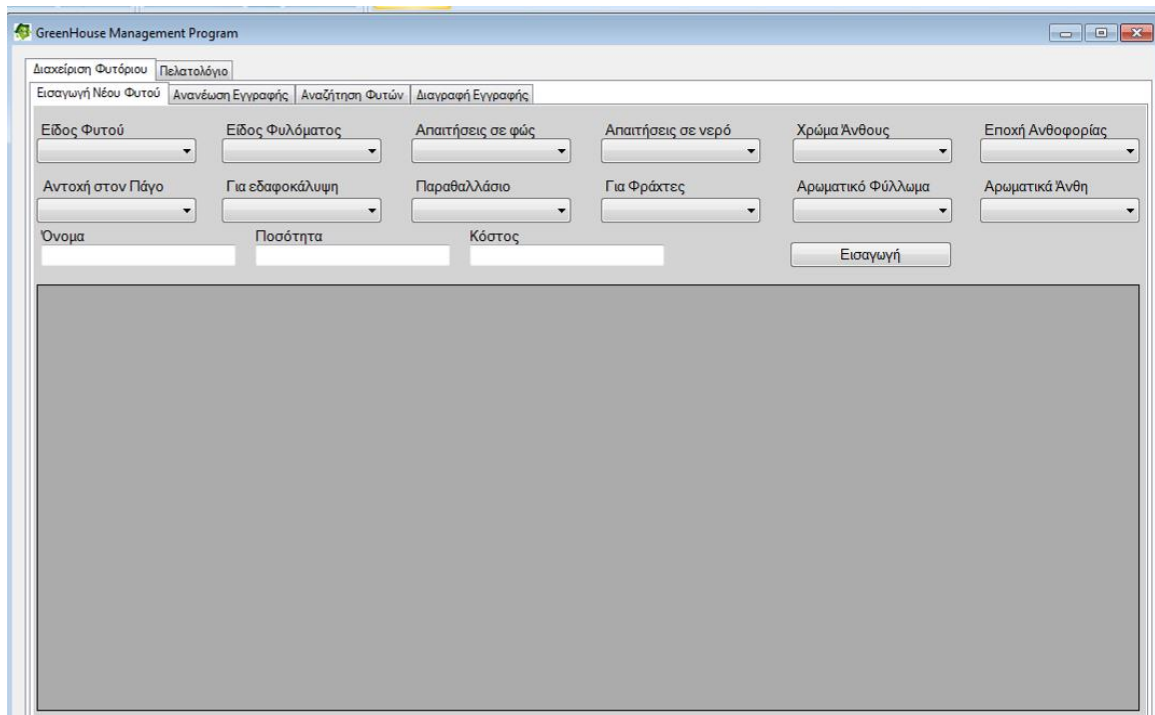
Σχήμα 12 : Φόρμα Log In

Σε περίπτωση λάθους παίρνουμε το σχετικό μήνυμα.



Σχήμα 13 : Μήνυμα λάθους

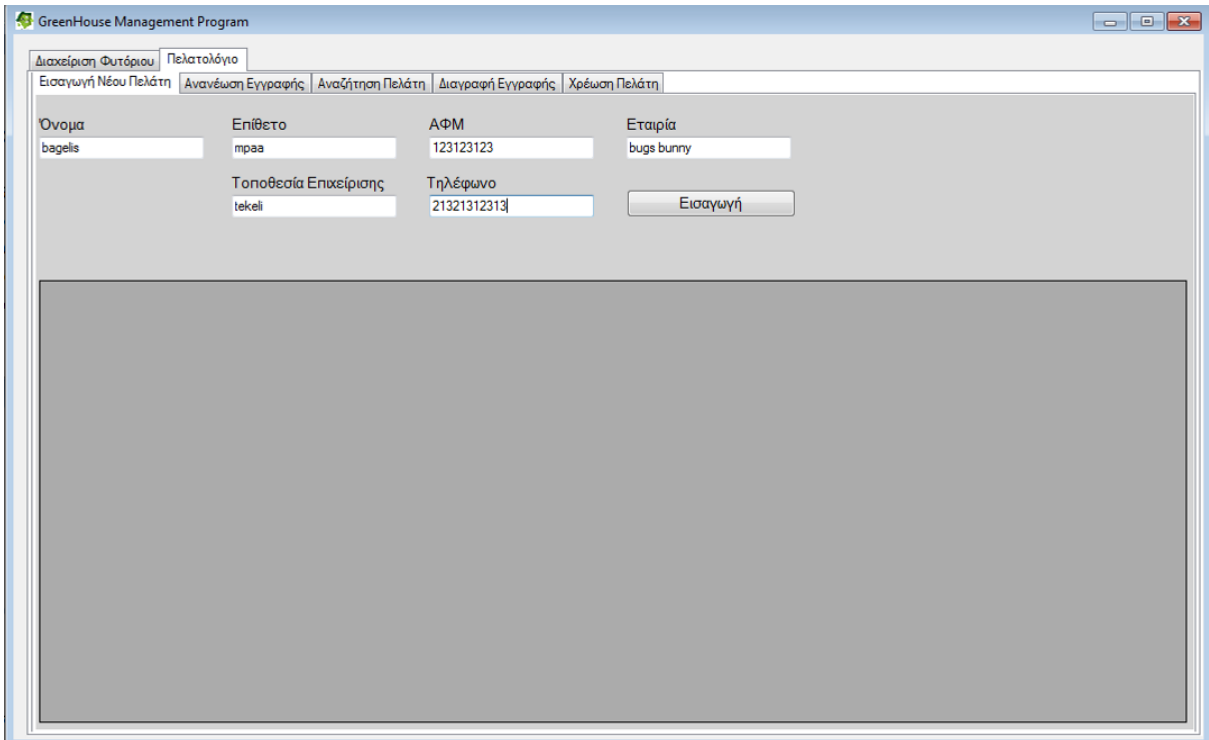
Σε διαφορετική περίπτωση προχωράμε στην δεύτερη φόρμα, την Main. Στην φόρμα αυτήν υπάρχουν οι καρτέλες διαχείρισης της βάσης δεδομένων φυτών και πελατών.



Σχήμα 14 : Κεντρική Φόρμα

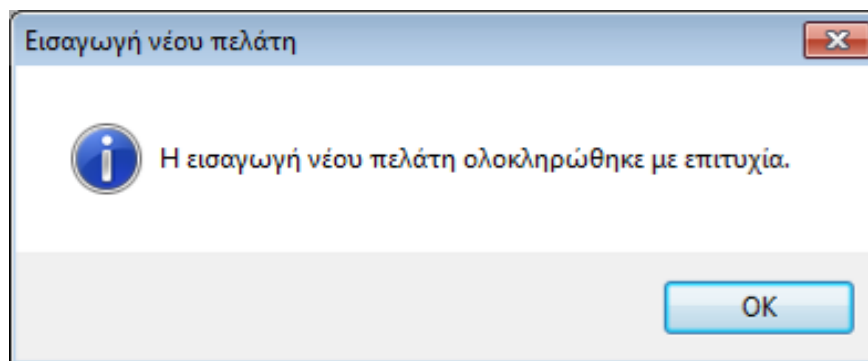
Οι φόρμες τόσο της διαχείρισης πελατών όσο και της διαχείρισης φυτών – αποθήκης είναι παρόμοιες σε λειτουργίες εκτός από την λειτουργία χρέωσης πελάτη. Για τον λόγο αυτό θα παρουσιαστεί η λειτουργία χρήσης της καρτέλας πελάτες και τα ίδια ισχύουν και για τις καρτέλες φυτών.





Σχήμα 15 : Καρτέλα εισαγωγής νέου πελάτη

Στην παραπάνω φόρμα κάνουμε εισαγωγή νέου πελάτη. Αν κάτι πάει στραβά εμφανίζεται σχετικό μήνυμα. Σε αντίθετη περίπτωση το πρόγραμμα μας ενημερώνει για την εισαγωγή.



Σχήμα 16 : Μήνυμα επιτυχούς εισαγωγής νέου πελάτη

Στην επόμενη φόρμα παρουσιάζεται η διαδικασία ανανέωσης μιας εγγραφής. Ο διαχειριστής πρέπει να εισάγει τα κριτήρια με τα οποία θα κάνει την αναζήτηση και να πατήσει το κουμπί της αναζήτησης. Στην συνέχεια από τον πίνακα αποτελεσμάτων μπορεί να τροποποιήσει ορισμένα χαρακτηριστικά που του επιτρέπονται.

Κωδικός	Επίθετο	Όνομα	ΑΦΜ	Εταιρία	Ημερομηνία καταχώρησης	Περιοχή	Τηλέφωνο	Υπόλοιπο	Παραγγελίες
4	mpak	baggelis	123125	company	25-01-13	kdaskdad	21313	60	5

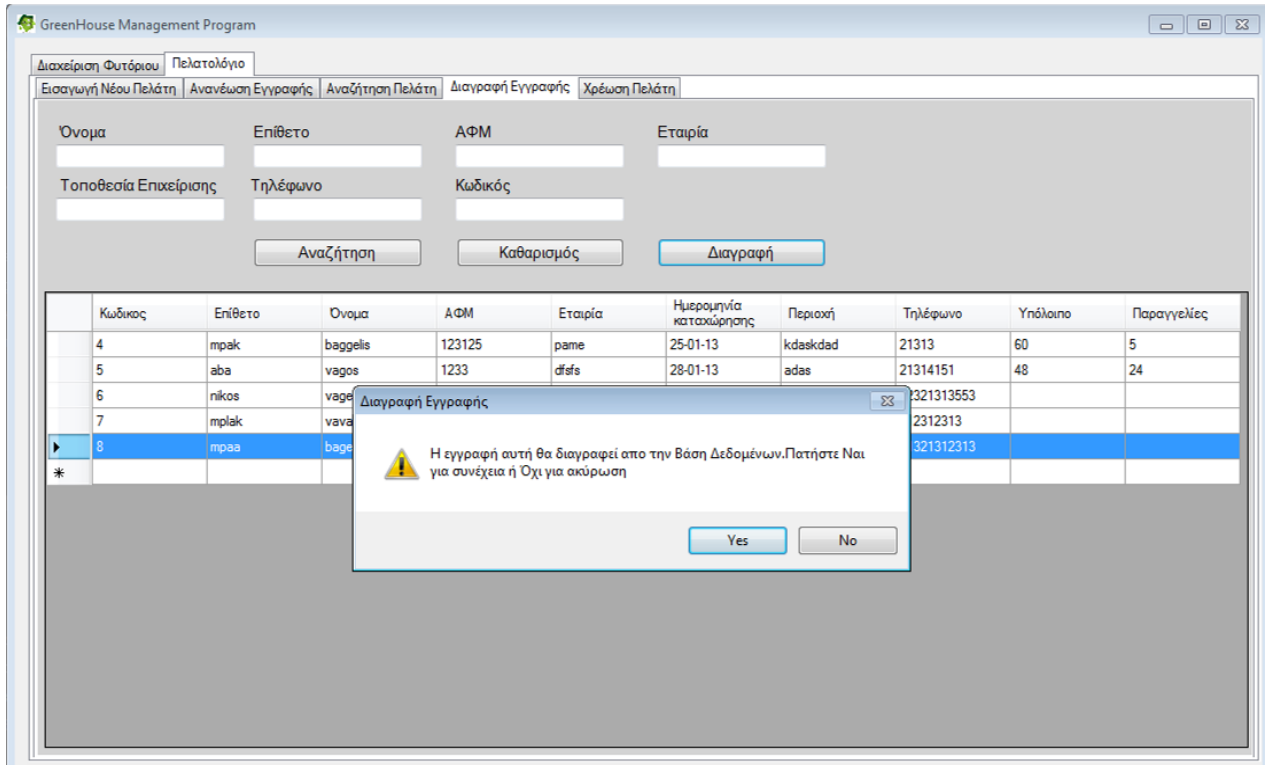
Σχήμα 17 : Καρτέλα ανανέωσης εγγραφής πελάτη

Στο επόμενο σχήμα ο διαχειριστής μπορεί να αναζητήσει με οποιοδήποτε κριτήριο θέλει έναν πελάτη. Τα αποτελέσματα εμφανίζονται στον πίνακα αποτελεσμάτων.

Κωδικός	Επίθετο	Όνομα	ΑΦΜ	Εταιρία	Ημερομηνία καταχώρησης	Περιοχή	Τηλέφωνο	Υπόλοιπο	Παραγγελίες
4	mpak	baggelis	123125	rame	25-01-13	kdaskdad	21313	60	5
5	aba	vagos	1233	dfsf	28-01-13	adas	21314151	48	24
6	nikos	vagelis	123123	kjsdkfjsdfjsaf	30-01-13	tekeli	12321313553		
7	mpiak	vava	1213	asdad	30-01-13	asdad	312312313		
8	mpaa	bagelis	123123123	bugs bunny	13-02-13	tekeli	21321312313		

Σχήμα 18 : Αναζήτηση πελάτη

Στο παρακάτω σχήμα ο διαχειριστής μπορεί να διαγράψει έναν πελάτη. Στην καρτέλα διαγραφή πρέπει να αναζητήσει με όποιο κριτήριο επιθυμεί τον πελάτη που θέλει. Στην συνέχεια πρέπει να επιλέξει την εγγραφή και να πατήσει το κουμπί της διαγραφής. Εκείνη την στιγμή θα του παρουσιαστεί ένα μήνυμα ερώτησης αν θέλει να συνεχίσει ή όχι.



Σχήμα 19 : Φόρμα διαγραφής πελάτη

Αν ο διαχειριστής πατήσει όχι τότε θα ξαναγεμίσει ο πίνακας με την βάση δεδομένων. Σε διαφορετική περίπτωση θα γίνει η διαγραφή και ο διαχειριστής θα λάβει μήνυμα επιβεβαίωσης.

Στην καρτέλα χρέωση πελάτη αρχικά ο διαχειριστής πρέπει να αναζητήσει τον πελάτη τον οποίο επιθυμεί να χρεώσει. Στην συνέχεια πρέπει να επιλέξει πάλι με αναζήτηση από την καρτέλα τα προϊόντα που θα χρεώσει. Μπορεί να επιλέξει το πόσα τεμάχια θα χρεώσει και το κόστος του κάθε τεμαχίου. Στην συνέχεια κάνει προσθήκη στον πίνακα παραγγελίας και για να ολοκληρωθεί η χρέωση πρέπει να πατήσει ολοκλήρωση της χρέωσης.

GreenHouse Management Program

Αποθήκη Φυτών > Πελατολόγιο

Εισαγωγή Νέου Πελάτη | Αναζήτηση Πελάτη | Διαγραφή Πελάτη | Χρέωση Πελάτη

Αναζήτηση Πελάτη με Κωδικό, Επώνιμο ή Όνομα: Παραγγελία:

Κωδικός:   
 Επώνιο:   
 Όνομα:

Κωδικός:     Όνομα:     Έμβλημα:     Κόστος:

Στοιχεία πελάτη:

Κωδικός:   
 Όνομα:   
 Επώνιο:   
 ΑΦΜ:   
 Εταιρία:   
 Περιοχή:   
 Τηλ.:

Κωδικός	Όνομα	Ποσότητα	Κόστος	Σύνολο
2	ραακ	2	12	24

Σχήμα 20 : Καρτέλα χρέωσης πελάτη

## ΣΥΜΠΕΡΑΣΜΑΤΑ

Βασικό κομμάτι της σύγχρονης κοινωνίας της πληροφορίας αποτελεί η διαχείριση, αποθήκευση και διανομή της όλο και αυξανόμενης και διαθέσιμης πληροφορίας. Για το λόγο αυτό παρατηρείται ραγδαία ανάπτυξη στον τομέα των βάσεων δεδομένων, στην ασφαλή αποθήκευση των δεδομένων, στον γρήγορο τρόπο ανάκτησης της πληροφορίας και στον τρόπο διαχείρισης της.

Αυτό καθιστά την κατασκευή εφαρμογών οι οποίες έχουν πρόσβαση σε αυτές τις βάσεις κρίσιμες για την επίτευξη της εύκολης πρόσβασης και από χρήστες χωρίς μεγάλη εμπειρία ή τεχνική γνώση στην επιστήμη των υπολογιστών. Με την κατασκευή αυτής της εφαρμογής έγινε προσπάθεια δημιουργίας ενός φιλικού περιβάλλοντος για τον διαχειριστή ώστε να μπορεί εύκολα και χωρίς πολλές γνώσεις να μπορεί να διαχειρίζεται την αποθήκη και το πελατολόγιο του. Με την χρήση του MySQL και της Visual Basic, της γλώσσας η οποία είναι από τις καλύτερες για τέτοιες εφαρμογές δημιουργήθηκε ένα πρόγραμμα το οποίο δίνει την δυνατότητα παρακολούθησης της αποθήκης όσο και των πωλήσεων.

Μελετήθηκαν πιθανά σενάρια τα οποία θα έχει απαίτηση ο διαχειριστής από το πρόγραμμα να αποκρίνεται ορθά. Ο σχεδιασμός της εφαρμογής κινήθηκε στο να μην βασιστεί καθόλου στην θέληση του διαχειριστή να χειριστεί σωστά το πρόγραμμα και προσπάθησε να χειριστεί όλα τα πιθανά λάθη.

Σε μελλοντικό επίπεδο το πρόγραμμα θα μπορούσε να διαχειριστεί ταυτόχρονα και από ένα μόνο εκτελέσιμο και άλλες αποθήκες φυτών. Έτσι θα έδινε την δυνατότητα στον διαχειριστή ο οποίος έχει παραπάνω από ένα θερμοκήπια να τα διαχειρίζεται με ένα μόνο πρόγραμμα.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1]. Mark Maslakowski, Teach yourself MySql in 21 Days, Sams, 2000
- [2]. Tim Patrick, Programming Visual Basic 2008 O'Reilly May 2008
- [3]. Evangelos Petroutsos, Mark Ridgeway, Mastering Visual Basic 2008, Wiley Publishing, 2008
- [4]. I. Sommerville, *Software Engineering (8th Edition)*. Addison-Wesley, 2006.
- [5]. <http://en.wikipedia.org/wiki/MySQL>
- [6]. <http://msdn.microsoft.com/en-US/>