

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΠΟΛΥΜΕΣΩΝ



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΘΕΜΑ: « e-Projects »

Εισηγητής: Παχουλάκης Ιωάννης

Σπουδάστρια: Λιοδάκη Κατερίνα

Ευχαριστίες

Θα θέλα να ευχαριστήσω όλους όσους συμμετείχαν και συνέβαλαν στην ολοκλήρωση της παρούσας πτυχιακής εργασίας. Ειδικότερα τον εισηγητή μου, κ. Παχουλάκη Ιωάννη για την βοήθεια και την καθοδήγησή του. Τα μέλη της εξεταστικής επιτροπής που μου έκαναν την τιμή να αξιολογήσουν την προσπάθειά μου. Τέλος την οικογένεια και τους φίλους μου, που όλα αυτά τα χρόνια με στήριξαν από όλες τις πλευρές.

Σας ευχαριστώ,
Λιοδάκη Κατερίνα

ΠΕΡΙΕΧΟΜΕΝΑ

Λίστα Σχημάτων.....	5
1. Κεφάλαιο – Το λογισμικό πακέτο.....	6
1.1 Συνοπτική περιγραφή του συστήματος.....	6
1.2 Απαιτήσεις συστήματος.....	6
2. Κεφάλαιο - Εισαγωγή στις πλατφόρμες και τεχνικές ανάπτυξης.....	7
2.1 Η πλατφόρμα .NET.....	7
2.1.1 Περιγραφή της αρχιτεκτονικής .NET.....	7
2.1.2 Visual Basic.NET.....	10
2.1.3 Web Forms και ASP.NET.....	12
2.2 Βάσεις δεδομένων και SQL Server 2000.....	15
3. Κεφάλαιο – Περιγραφή της Βάσης Δεδομένων.....	16
3.1 Πίνακες της Βάσης Δεδομένων.....	16
3.2 Διάγραμμα Βάσης Δεδομένων – Συσχετίσεις πινάκων.....	20
3.3 Αποθηκευμένες Διεργασίες – Stored Procedure.....	22
4. Κεφάλαιο – Ανάλυση διαδικασιών σε ASP.NET.....	24
4.1 Επικύρωση για της Web φόρμες.....	24
4.1.1 RequiredFieldValidator.....	25
4.1.2 RegularExpressionValidator.....	26
4.2 Ασφάλεια.....	27
4.2.1 Ασφάλεια του IIS.....	27
4.2.2 Το αρχείο web.config.....	27
4.3 Σύνδεση με Βάση Δεδομένων.....	28
5. Κεφάλαιο – Ανάλυση της εφαρμογής μας.....	31
5.1 Η φόρμα logon.aspx.....	31

5.2 Η φόρμα person.aspx.....	32
5.3 Η φόρμα PersonsList.aspx.....	33
5.4 Η φόρμα project.aspx.....	34
5.5 Η φόρμα ProjectsList.aspx.....	35
5.6 Η φόρμα info.aspx.....	36
6. Κεφάλαιο – Βιβλιογραφία – Πηγές.....	37

ΛΙΣΤΑ ΣΧΗΜΑΤΩΝ

Σχήμα 2.1.1.1 Αρχιτεκτονική .NET.....	9
Σχήμα 2.1.2.1 Το Visual Studio απλοποιεί τις διαδικασίες χρησιμοποιώντας έτοιμα πρότυπα εφαρμογών.....	11
Σχήμα 2.1.3.1 Ο σχεδιασμός μιας φόρμας (Web Form) για το Internet μοιάζει με αυτόν των φορμών της VB, με τον προγραμματιστή να έχει στη διάθεσή του τη γνωστή εργαλειοθήκη.....	13
Σχήμα 3.1.1 Πίνακας User.....	16
Σχήμα 3.1.2 Πίνακας Person.....	16
Σχήμα 3.1.3 Πίνακας Sector.....	17
Σχήμα 3.1.4 Πίνακας ClassLevel.....	17
Σχήμα 3.1.5 Πίνακας ProjectPerson.....	18
Σχήμα 3.1.6 Πίνακας Organization.....	18
Σχήμα 3.1.7 Πίνακας Country.....	18
Σχήμα 3.1.8 Πίνακας Project.....	19
Σχήμα 3.1.9 Πίνακας Project.....	19
Σχήμα 3.1.10 Πίνακας ProjectOrganization.....	20
Σχήμα 3.2.1 Διάγραμμα βάσης δεδομένων.....	21
Σχήμα 5.1 logon.aspx.....	31
Σχήμα 5.2 person.aspx.....	32
Σχήμα 5.3 PersonsList.aspx.....	33
Σχήμα 5.4 project.aspx.....	34
Σχήμα 5.5 ProjectsList.aspx.....	35
Σχήμα 5.6 info.aspx.....	36

1. Κεφάλαιο – Το λογισμικό πακέτο

1.1 Συνοπτική περιγραφή του συστήματος

Το αντικείμενο της πτυχιακής εργασίας είναι μια εφαρμογή για τη διαχείριση των προγραμμάτων ενός φορέα (ερευνητικού εργαστηρίου / οργανισμού) μέσω διαδικτύου (Web Application).

Η εφαρμογή έχει αναπτυχθεί σε τεχνολογία ASP.NET, με βάση δεδομένων σε SQL Server 2000. Η βάση δεδομένων περιλαμβάνει όλα τα συστατικά που είναι απαραίτητα για την υλοποίηση μίας διαδικτυακής εφαρμογής.

Γενικότερα στο σχεδιασμό και την υλοποίηση της εφαρμογής και της βάσης δεδομένων, λάβαμε υπόψη μας τη πιθανότητα επέκτασης τους, είτε ως προς τις δυνατότητες είτε ως προς την αρχιτεκτονική τους. Όπως για παράδειγμα με τη σχεδίαση δυναμικότερων σελίδων και ασφαλέστερων λειτουργιών.

1.2 Απαιτήσεις Συστήματος

Οι απαιτούμενες τεχνικές προδιαγραφές της πλατφόρμας εγκατάστασης είναι οι ακόλουθες:

➤ Για την πλατφόρμα .NET Framework απαιτείται ένα από τα παρακάτω λειτουργικά συστήματα:

- Microsoft Windows 2000 Professional with Service Pack 2.0
- Microsoft Windows 2000 Server with Service Pack 2.0
- Microsoft Windows 2000 Advanced Server with Service Pack 2.0
- Microsoft Windows 2000 Datacenter Server with Service Pack 2.0
- Microsoft Windows XP Professional with Service Pack 2.0

➤ Για την πλατφόρμα του SQL Server 2000 ένα από τα παρακάτω:

- Microsoft Windows NT Server 4.0
- Microsoft Windows NT Server Enterprise Edition 4.0
- Microsoft Windows 2000 Server
- Microsoft Windows 2000 Advanced Server

- Microsoft Windows 2000 Data Center Server
- Microsoft Windows Server 2003
- Microsoft Windows XP Professional with Service Pack 2.0

➤ Κοινές προδιαγραφές:

- Μνήμη: 128MB τουλάχιστον (προτεινόμενη 512MB)
- Επεξεργαστής: Pentium 133 MHz ή γρηγορότερος
- Web server: Microsoft Internet Information Services (IIS) 5.0+

2. Κεφάλαιο – Εισαγωγή στις πλατφόρμες και τεχνικές ανάπτυξης

2.1 Η πλατφόρμα .NET

Η πλατφόρμα ASP. NET (Active Server Pages) που χρησιμοποιήσαμε είναι μια πλατφόρμα ανάπτυξης λογισμικού που περιλαμβάνει πολλές δυνατότητες ανάπτυξης δυναμικών εφαρμογών δικτύου και παράλληλα μεγάλη ευκολία στη σύνδεση με διάφορες βάσεις δεδομένων. Αποτελεί μέρος της Microsoft .NET Framework και είναι ένα ισχυρό εργαλείο για τη γρήγορη ανάπτυξη δυναμικών και διαδραστικών ιστοσελίδων που εκτελούνται στον web server και το αποτέλεσμα τους, σε μορφή html, εμφανίζεται στον browser του χρήστη.

2.1.1 Περιγραφή της αρχιτεκτονικής .NET

Μέχρι σήμερα, η μεταγλώττιση μιας εφαρμογής από τον αντίστοιχο compiler δημιουργούσε εκτελέσιμα αρχεία (EXE ή DLL). Αυτά περιελάμβαναν απευθείας κώδικα μηχανής για την πλατφόρμα x86 και φυσικά μπορούσαν να τρέξουν μόνο σε αυτού του είδους τα συστήματα.

Στο περιβάλλον .NET η μεταγλώττιση από έναν compiler, όπως είναι οι Visual Basic, C++ και C# στο Visual Studio, δημιουργεί έναν ενδιάμεσο κώδικα, που η Microsoft

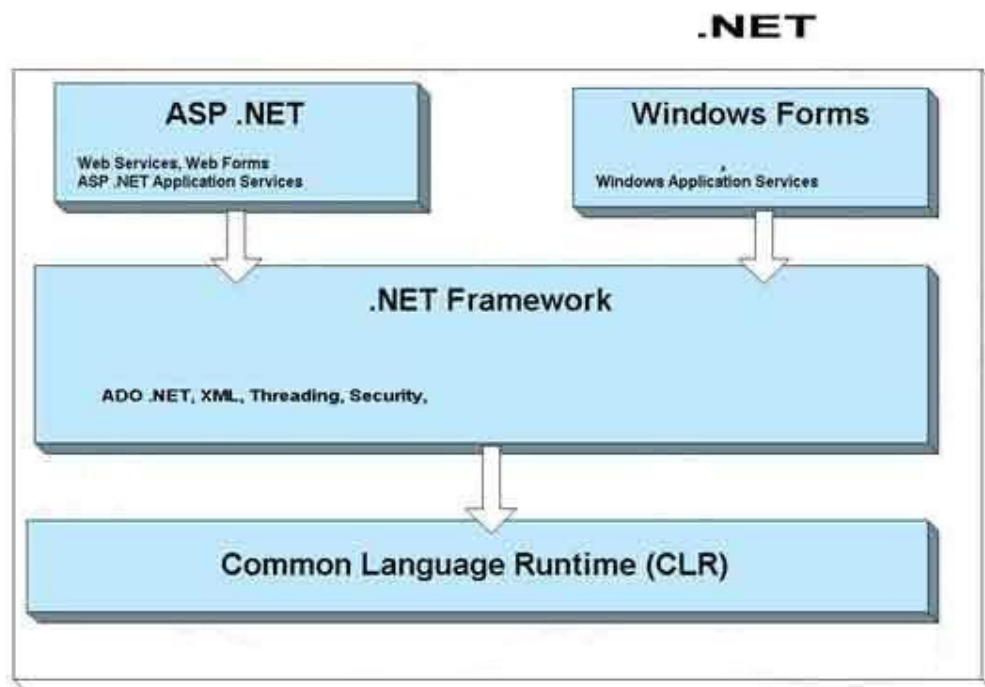
ονομάζει MSIL (Microsoft Intermediate Language). Αυτός είναι ανεξάρτητος από τον επεξεργαστή και βρίσκεται σε υψηλότερο επίπεδο από τον κώδικα μηχανής. Περιλαμβάνει μάλιστα και metadata (στοιχεία με πληροφορίες για τη χρήση τους) που χρησιμοποιεί το CLR για να φορτώσει συγκεκριμένες λειτουργίες, να τοποθετήσει αντικείμενα στην μνήμη, να ελέγξει την ασφάλεια και να πραγματοποιήσει μια σειρά από πρόσθετες λειτουργίες. Δεν πρόκειται ασφαλώς για χαζό κώδικα μηχανής, καθώς το MSIL έχει σκοπό να κάνει πιο εύκολη τη μετάβαση από τα 32μπιτα Windows στη 64μπιτη πλατφόρμα των Windows 2000, όταν αυτή κυκλοφορήσει. Φυσικά, οι εντολές του MSIL δεν μπορούν να εκτελεστούν ως έχουν από τους σημερινούς επεξεργαστές, καθώς πρέπει πρώτα να μεταφραστούν σε γλώσσα μηχανής, στοιχείο που αναλαμβάνει το CLR.

Τα πλεονεκτήματα αυτού του τρόπου εκτέλεσης των εφαρμογών είναι πολλά. Πρώτα από όλα προσφέρεται ένας ενιαίος τρόπος για τη χρήση των υπηρεσιών του λειτουργικού συστήματος, σε αντίθεση με την σημερινή κατάσταση, κατά την οποία μερικές υπηρεσίες βρίσκονται σε DLL και άλλες σε αρχεία τύπου COM. Αυτό σημαίνει επίσης ότι παύει η δυσχρηστία με τα DLL, δηλαδή το φαινόμενο κατά το οποίο η εγκατάσταση μιας εφαρμογής αντικαθιστά κάποιο DLL μιας άλλης εφαρμογής με απρόβλεπτα, όπως πάντα, αποτελέσματα. Στην πλατφόρμα .NET υπάρχει πια ο διαχωρισμός των στοιχείων της κάθε εφαρμογής, έτσι ώστε όταν αυτή εκτελείται, να φορτώνονται συγχρόνως και τα στοιχεία με τα οποία αναπτύχθηκε και ελέγχθηκε. Η δυνατότητα της εύκολης μεταφοράς μεταξύ διαφορετικών επεξεργαστών και λειτουργικών συστημάτων και η ευελιξία στον προγραμματισμό με τη χρήση πολλαπλών γλωσσών συμπληρώνουν με μια πρώτη ματιά τα πιο αξιόλογα πλεονεκτήματα της πλατφόρμας .NET. Σήμερα υπάρχουν αρκετές εκδόσεις των Windows, τα Windows 95, Windows 98, Windows 98 SE, Windows Millennium, Windows CE, Windows NT 4.0, Windows 2000 και σύντομα πρόκειται να κυκλοφορήσουν και τα 64μπιτα Windows 2000. Τα περισσότερα από αυτά τρέχουν σε επεξεργαστές x86. Η εξαίρεση αυτού του κανόνα έρχεται με τα Windows CE και τα 64μπιτα Windows. Υπάρχουν φυσικά και αρκετές νέες συσκευές στα άμεσα πλάνα των κατασκευαστών που αναμένονται να έχουν κάποιο, διαφορετικό ίσως, τρόπο επικοινωνίας με τον χρήστη. Θεωρητικά, μια εφαρμογή .NET μπορεί να εκτελείται σε λειτουργικά συστήματα διαφορετικά από τα Windows, αρκεί να υπάρχει ένα CLR για κάθε διαφορετική συσκευή, λύνοντας έτσι τα χέρια του προγραμματιστή.

Ένα ακόμα πλεονέκτημα που αφορά στους προγραμματιστές είναι ότι το Visual Studio

.NET επιτρέπει την ενσωμάτωση περισσότερων γλωσσών προγραμματισμού. Μπορεί δηλαδή να δημιουργήσει κανείς μια κλάση σε C++ που να προέρχεται από μια άλλη κλάση σε Visual Basic ή το αντίστροφο.

Η εγκατάσταση μιας εφαρμογής αποτελεί, σε όλες ανεξαιρέτως εκδόσεις των Windows, ένα από τα πιο δύστροπα σημεία. Συνήθως, ο κατασκευαστής συνοδεύει την εφαρμογή με αρκετά πρόσθετα αρχεία, εκτελέσιμα DLL και OCX, εγγραφές στο μητρώο και συντομεύσεις. Τα Windows 2000 έφεραν το Windows Installer που βελτίωσε κάπως την κατάσταση. Στην πλατφόρμα .NET η Microsoft αποσύνδεσε επιτέλους τα στοιχεία των εφαρμογών από το μητρώο του συστήματος. Στην πράξη, το .NET υπόσχεται να κάνει την εγκατάσταση τόσο απλή όσο μια αντιγραφή των αρχείων από τον ένα κατάλογο στον άλλο και την απεγκατάσταση τόσο εύκολη όσο τη διαγραφή αυτών των αρχείων.



Σχήμα 2.1.1.1 Αρχιτεκτονική .NET

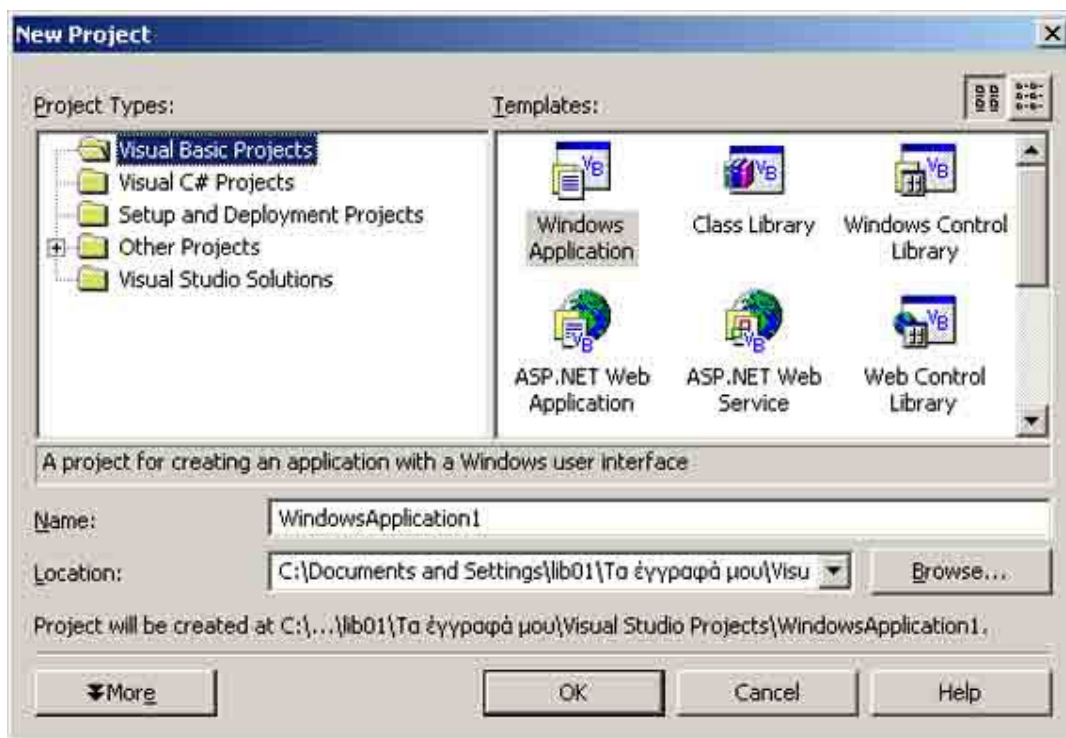
Στο παραπάνω σχεδιάγραμμα βλέπουμε ένα απλουστευμένη αναπαράσταση της αρχιτεκτονικής της πλατφόρμας .NET. Στην καρδιά της πλατφόρμας βρίσκεται το CLR (Common Language Runtime) που επιτρέπει τη μεταγλώττιση και την εκτέλεση του κώδικα MSIL (Microsoft Intermediate Language). Στο μεσαίο επίπεδο υπάρχουν τα προγράμματα υποδομής του .NET Framework. Στο ανώτερο επίπεδο βρίσκονται τα εργαλεία ανάπτυξης εφαρμογών.

2.1.2 Visual Basic .NET

Η Visual Basic ακολούθησε εντυπωσιακή εξέλιξη την τελευταία δεκαετία, από την έκδοση 1.0 στην σημερινή έκδοση .NET, ή 7.0, όπως είναι η εσωτερική αρίθμηση του προϊόντος. Οι νέες εντολές, οι βελτιώσεις στον τρόπο ανίχνευσης των λαθών αλλά και όσα εργαλεία προστέθηκαν μέχρι τη σημερινή έκδοση, δεν άλλαξαν τον αντικειμενικό σκοπό, που είναι η εύκολη και γρήγορη ανάπτυξη εφαρμογών.

Έτσι, η Visual Basic, μέσα από το Visual Studio, παραμένει ένα εύκολο και ιδιαίτερα γρήγορο εργαλείο για την ανάπτυξη αυτόνομων εφαρμογών αλλά και υπηρεσιών στο Web. Επιλέγοντας μια νέα λύση (Solution), μέσα από το περιβάλλον της Visual Basic, μπορείτε να δημιουργήσετε μια αυτόνομη εφαρμογή (Windows Application), μια εφαρμογή στο Internet (Web Application) ή μια υπηρεσία στο Web (Web Service). Και στις τρεις περιπτώσεις όλα γίνονται εύκολα, με το σύστημα να κάνει για εσάς την περισσότερη δουλειά.

Ωστόσο, η νέα έκδοση Visual Basic.NET φέρνει αρκετές αλλαγές στην ίδια τη γλώσσα, στο περιβάλλον προγραμματισμού και κατ' επέκταση στη μεταφορά μιας υπάρχουσας εφαρμογής. Οι βελτιώσεις στη γλώσσα περιλαμβάνουν νέες εντολές και σχήματα που φέρνουν πιο κοντά τη VB στις υπόλοιπες αντικειμενοστρεφείς γλώσσες, όπως η C++ και η Java. Από τις πιο σημαντικές βελτιώσεις σε αυτό τον τομέα είναι η δυνατότητα της κληρονομικότητας (inheritance), της δημιουργίας κλάσεων (class) αλλά και της ενσωμάτωσης πιο αυστηρών κανόνων για την αντίδραση σε λάθη (error trapping). Ο προγραμματιστής έχει στη διάθεσή του την εντολή "inherits", που επιτρέπει τη μεταφορά όλων των ιδιοτήτων ενός αντικειμένου ή μιας κλάσης. Υπάρχει επίσης και το function overloading, που προσφέρει τη δυνατότητα να χρησιμοποιούμε τις ίδιες λειτουργίες με διαφορετικές παραμέτρους. Επιπλέον, κανείς είναι σε θέση να δημιουργήσει ένα νέο νήμα (thread) με την εντολή New Thread. Με αυτό τον τρόπο η εκτέλεση μιας εφαρμογής μπορεί να μην είναι σύγχρονη, να μην χρειάζεται δηλαδή να περιμένει μια εντολή το τέλος της προηγούμενης.



Σχήμα 2.1.2.1 Το Visual Studio απλοποιεί τις διαδικασίες χρησιμοποιώντας έτοιμα πρότυπα εφαρμογών.

Οι εντολές On Error και Resume Next έχουν αντικατασταθεί με πιο σταθερές και συνεπείς λύσεις, όπως οι εντολές Try, Catch και Final, επιτρέποντας τη συγγραφή πιο καθαρού κώδικα. Υπάρχει επίσης η δυνατότητα της αρχικοποίησης μιας μεταβλητής κατά τη δήλωσή της (Dim Y as Integer = 1). Θα βρείτε αλλαγές και σε πολλά ακόμα σημεία που αφορούν σε μεμονωμένα στοιχεία ελέγχου, όπως το μήνυμα MsgBox. Επιπλέον, μια αλλαγή που πιθανόν θα δημιουργήσει προβλήματα με τον κώδικα της προηγούμενης έκδοσης, είναι ο τρόπος αντιμετώπισης των παραμέτρων. Στην έκδοση 6 οι παράμετροι σε μια διαδικασία (function DoSomething(X As Integer)) περνούσαν ως ByRef, που σημαίνει ότι οι τιμές τους μπορούσαν να αλλαχτούν μέσα στη συγκεκριμένη διαδικασία.

Στην έκδοση .NET οι παράμετροι περνούν ως ByVal, που σημαίνει ότι πρέπει να αλλαχτεί ο κώδικας (function DoSomething(ByRef X As Integer)) για να περάσει η σωστή τιμή στον κορμό του προγράμματος. Μια αλλαγή που προβλέπεται επίσης να δημιουργήσει αρκετά προβλήματα είναι τα χαρακτηριστικά των αντικειμένων σε μια φόρμα.

Όπως θα έχετε καταλάβει, οι αλλαγές είναι πολλές και η μεταφορά από την έκδοση 6 απαιτεί την πληκτρολόγηση αρκετού κώδικα. Για αυτόν το λόγο η Microsoft ενσωμάτωσε

ένα βοηθητικό εργαλείο (Upgrade Tool), που αναλαμβάνει την αναβάθμιση της εφαρμογής στο νέο περιβάλλον, αλλάζοντας αυτόματα τις εντολές στο νέο φορμά. Αυτό το εργαλείο ενεργοποιείται αυτόματα αν ανοίξετε κάποια παλιότερη εφαρμογή μέσα από το Visual Studio .NET. Το αποτέλεσμα είναι να δημιουργηθεί ένα νέο έργο, αφήνοντας τα αρχικά αρχεία της έκδοσης 6.0 όπως έχουν. Δύσκολα όμως θα αποφύγετε την επέμβαση στον κώδικα, αφού τις περισσότερες φορές η αυτόματη αναβάθμιση θα δημιουργήσει αρκετές ασυμβατότητες.

Ένα ακόμα ενδιαφέρον χαρακτηριστικό στις φόρμες της Visual Basic .NET είναι η μεγαλύτερη ακρίβεια. Χαρακτηριστικά όπως το Menu Designer, Control Anchoring, Control Docking και μια σειρά από νέα στοιχεία ελέγχου επιτρέπουν να αναπτύξει κανείς πιο καλαίσθητες φόρμες. Σε αυτό έρχεται να προστεθεί και η υποστήριξη του GDI+, του διαδόχου του GDI (Graphic Device Interface) που συναντούσαμε στις προηγούμενες εκδόσεις των Windows, το οποίο προσφέρει πιο πλούσια χαρακτηριστικά. Έτσι, μπορεί κανείς εύκολα να σχεδιάσει οθόνες που εμφανίζονται σε διαφανή και πολυεπίπεδα παράθυρα, χαρακτηριστικό που επιτρέπουν τα Windows 2000 και τα Windows XP. Άλλο ένα ενδιαφέρον στοιχείο της Visual Basic .NET είναι η πλήρης υποστήριξη του Unicode, ακόμα και στις φόρμες.

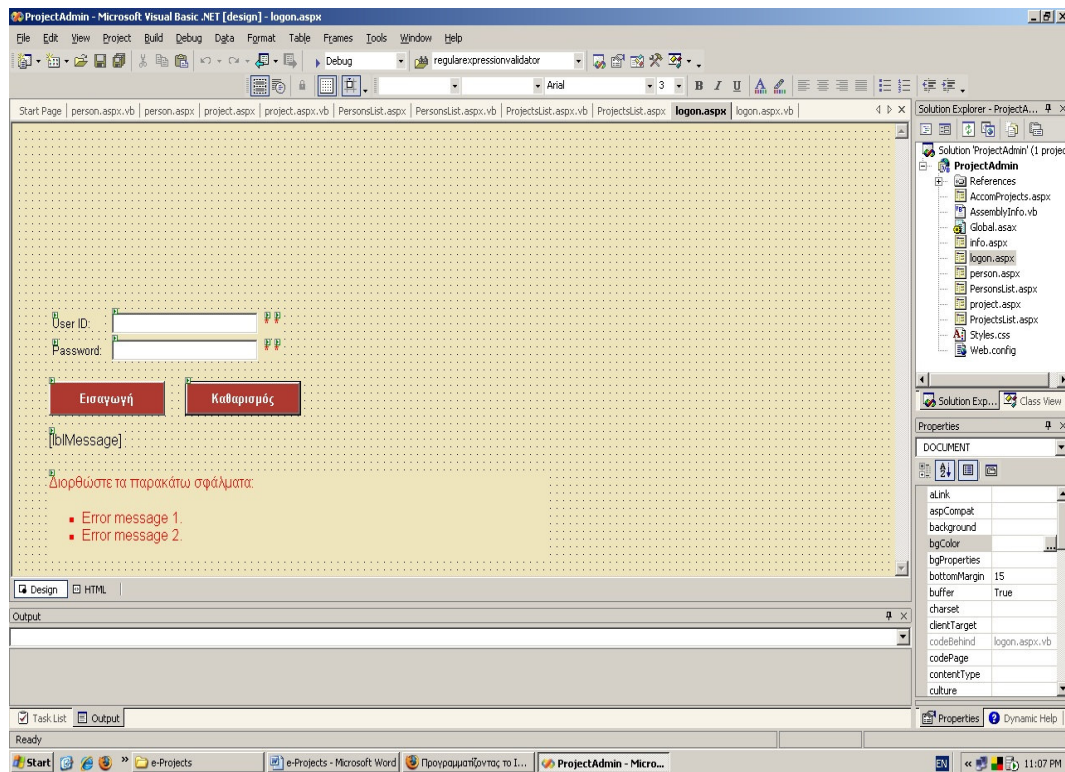
Μαζί με το Visual Studio έρχεται και μια έκδοση του Crystal Reports, της γνωστής εφαρμογής για εύκολη και γρήγορη κατασκευή εκτυπώσεων, με τις ανάλογες επεκτάσεις για την υποστήριξη εκτυπώσεων σε εφαρμογές για το Internet.

2.1.3 Web Forms και ASP .NET

Η Visual Basic ακολούθησε εντυπωσιακή εξέλιξη την τελευταία δεκαετία, από την έκδοση 1.0 στην σημερινή έκδοση .NET, ή 7.0, όπως είναι η εσωτερική αρίθμηση του προϊόντος. Οι νέες εντολές, οι βελτιώσεις στον τρόπο ανίχνευσης των λαθών αλλά και όσα εργαλεία προστέθηκαν μέχρι τη σημερινή έκδοση, δεν άλλαξαν τον αντικειμενικό σκοπό, που είναι η εύκολη και γρήγορη ανάπτυξη εφαρμογών.

Τα Web Form (φόρμες για το Web) αποτελούν μια καινούρια μέθοδο, μια νέα προσέγγιση στον τρόπο δημιουργίας εφαρμογών για το Internet. Πρακτικά γεφυρώνει το κενό μεταξύ της ανάπτυξης αυτόνομων εφαρμογών για τα Windows και για το Internet, καθώς χρησιμοποιεί τα ίδια εργαλεία, κενές φόρμες, μια εργαλειοθήκη και κώδικα σε

Visual Basic (ή οποιαδήποτε άλλη γλώσσα). Απαραίτητη προϋπόθεση είναι η ύπαρξη του αντίστοιχου διακομιστή της πλατφόρμας .NET (Windows 2000 και το .NET FrameWork).



Σχήμα 2.1.3.1 Ο σχεδιασμός μιας φόρμας (Web Form) για το Internet μοιάζει με αυτόν των φορμών της VB, με τον προγραμματιστή να έχει στη διάθεσή του τη γνωστή εργαλειοθήκη.

Η δημιουργία μιας εφαρμογής για το Internet γίνεται με τον ίδιο τρόπο, μέσα από το Visual Studio .NET, επιλέγοντας το ASP .NET Web Application αντί του Windows Application, το οποίο χρησιμοποιείται για τις "πατροπαράδοτες" παραθυρικές εφαρμογές.

Στη συνέχεια, πρέπει να επιλέξει κανείς τον τόπο όπου θα "φιλοξενηθεί" η εφαρμογή, ο οποίος μπορεί να είναι ένας απομακρυσμένος διακομιστής ή ένας διακομιστής στο τοπικό δίκτυο. Στην πρώτη περίπτωση το Visual Studio θα χρησιμοποιήσει τις επεκτάσεις του FrontPage και το πρωτόκολλο HTTP για να επικοινωνήσει με το διακομιστή και να δημιουργήσει τα τμήματα που αποτελούν την εφαρμογή. Οι διαφορές στο κατασκευαστικό κομμάτι της εφαρμογής είναι παρ' όλα αυτά λίγες, αφού ο χρήστης έχει στη διάθεσή του μια κενή φόρμα και την εργαλειοθήκη με την οποία έχει συνηθίσει να δουλεύει.

Η δημιουργία μιας απλής εφαρμογής, στην οποία ο τελικός χρήστης θα μπορεί να εισαγάγει το όνομα και τον κωδικό πρόσβασης, είναι υπόθεση λίγων λεπτών. Αρκεί να "τραβήξει" κανείς δύο αντικείμενα τύπου Label για τις περιγραφές, δύο άλλα αντικείμενα

TextBox και ένα κουμπί εντολών. Το σκεπτικό που ακολουθούν οι φόρμες για το Web είναι το ίδιο όπως και στον κλασικό προγραμματισμό. Με διπλό κλικ, για παράδειγμα, πάνω στο κουμπί εντολών μπορείτε να γράψετε τον κώδικα που θέλετε και να αλλάξετε οποιοδήποτε από τα χαρακτηριστικά των αντικειμένων που βρίσκονται πάνω στη φόρμα. Αν χρησιμοποιήσετε ελληνικούς χαρακτήρες στις επιλογές, τα μενού και τα πεδία κειμένου, καλό είναι να επιλέξετε, με δεξί κλικ πάνω στη φόρμα, το ελληνικό σετ χαρακτήρων.

Μια φόρμα Web Form αποτελείται από δύο μέρη: το αρχείο HTML που αντιπροσωπεύει την οπτική απεικόνιση της σελίδας, έτσι όπως θα τη δει ο τελικός χρήστης στο PC, και ένα αρχείο με κώδικα σε VB που χειρίζεται τα γεγονότα, τα αντικείμενα και τα αποτελέσματα στη συγκεκριμένη σελίδα. Ο διακομιστής ανταλλάσσει στοιχεία και δεδομένα σε μορφή XML μέσα από τις φόρμες Web Form και εμφανίζει τα αποτελέσματα σε μορφή HTML 3.2. Αυτό σημαίνει ότι τα αποτελέσματα είναι δυνατόν να εμφανιστούν μέσα από οποιοδήποτε πρόγραμμα αναζήτησης. Τα αρχεία HTML έχουν τη μορφή .aspx, η οποία είναι η νέα έκδοση των σελίδων .asp, που επιτρέπει την πολύ καλύτερη διαχείριση της VB .NET μέσα στην ίδια την ιστοσελίδα. Θα δείτε ότι μια φόρμα με το όνομα WebForm, για παράδειγμα, αποτελείται από δύο αρχεία, το WebForm.aspx και το WebForm.vb με κώδικα HTML και VB αντίστοιχα.

Φυσικά, μπορείτε να γράψετε απλό κώδικα HTML και VB μέσα στην ίδια ιστοσελίδα από το FrontPage ή έναν άλλο editor, αρκεί να την αποθηκεύσετε σε ένα διακομιστή που διαθέτει την υποστήριξη του .NET Framework και το συγκεκριμένο αρχείο να έχει την κατάληξη .aspx. Δουλεύοντας με τη νέα δομή των ιστοσελίδων ASP .NET, θα δείτε ότι υπάρχουν πολύ μεγάλες αλλαγές και για αυτό δεν ενδείκνυται καθόλου να προσπαθήσετε να μεταφέρετε τον παλιό σας κώδικα στη νέα μορφή. Αλλωστε, ένα διακομιστής Windows 2000 με το .NET Framework υποστηρίζει και τις δύο εκδόσεις.

2.2 Βάσεις Δεδομένων και SQL Server 2000

Για την ολοκλήρωση της εφαρμογής μας χρησιμοποιήσαμε μια βάση δεδομένων, έτσι ώστε τα δεδομένα να βρίσκονται αποθηκευμένα σε ένα κεντρικό σύστημα. Μ' αυτόν τον τρόπο εξασφαλίζεται η άμεση προσβασιμότητα τους από τον χρήστη, ώστε να τα διαχειριστεί και να τα επεξεργαστεί διαμέσου της web εφαρμογής.

Η επιλογές για τη χρήση μιας βάσης δεδομένων είναι αρκετές όπως Oracle, mySQL, Access, SQL Server 2000 κ.α. Η τελική επιλογή μας ήταν η χρήση SQL Server 2000 λόγω του ότι πληρούσε τα περισσότερα κριτήρια όπως, να είναι τύπου server, να υπάρχει η κατάλληλη τεχνογνωσία, να υποστηρίζει πολλούς χρήστες ταυτόχρονα, κ.α.

Ο Microsoft SQL Server είναι ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων (relational database management system - RDBMS), σχεδιασμένο για εφαρμογές επεξεργασίας συναλλαγών εντός επικοινωνίας (online transaction processing - OLTP), αποθηκών δεδομένων (data warehousing), και ηλεκτρονικού εμπορίου (e-commerce) με υψηλές απαιτήσεις.

Ο Microsoft SQL Server 2000 είναι σχεδιασμένος για να υποστηρίζει βάσεις δεδομένων μεγάλου όγκου και κρίσιμης σημασίας σε πολλά και διαφορετικά πεδία εφαρμογών, στα οποία περιλαμβάνονται η επεξεργασία συναλλαγών εντός επικοινωνίας (online transaction processing), οι αποθήκες δεδομένων (data warehousing) και το ηλεκτρονικό εμπόριο (e-commerce). Για την υποστήριξη αυτών των λειτουργιών, ο SQL Server διαθέτει ορισμένα εργαλεία, στα οποία περιλαμβάνονται κάποια βοηθητικά προγράμματα γραμμής διαταγών όπως το bcp.exe, το οποίο αντιγράφει δεδομένα μεταξύ του SQL Server και ενός αρχείου του λειτουργικού συστήματος, και τον Enterprise Manager, ένα προηγμένο παραστατικό εργαλείο για τη διαχείριση πολλών βάσεων δεδομένων και του ίδιου του SQL Server.

3. Κεφάλαιο – Περιγραφή της Βάσης Δεδομένων

3.1 Πίνακες της Βάσης Δεδομένων

Η βάση δεδομένων αποτελείται από τους παρακάτω πίνακες:

➤ Πίνακας User

tblUser			
	Column Name	Data Type	Length
	UserID	tinyint	1
	FirstName	varchar	50
	LastName	varchar	50
	Email	varchar	50
	Password	varchar	50
	Role	varchar	10

Σχήμα 3.1.1 Πίνακας User

Ο παραπάνω πίνακας (Σχήμα 3.1.1) περιέχει όλα τα απαραίτητα στοιχεία για την εισαγωγή ενός χρήστη στην εφαρμογή. Περιλαμβάνει τα εξής πεδία: όνομα, επίθετο, e-mail, κωδικό πρόσβασης, είδος χρήστη (administrator ή user), ενώ το πεδίο UserID παίρνει αυτόματα ένα αύξοντα αριθμό από τη βάση. Τέλος, παρατηρούμε ότι δεν έχει οριστεί πρωτεύων κλειδί και αυτό διότι δεν είναι χρήσιμο.

➤ Πίνακας Person

tblPerson			
	Column Name	Data Type	Length
	PersonID	int	4
	SectorID	int	4
	OrganizationID	int	4
	LastName	nvarchar	100
	FirstName	nvarchar	100
	ClassLevelID	int	4
	Email	nvarchar	50
	OfficeTelephone	nvarchar	50
	HouseTelephone	nvarchar	50
	Cell	nvarchar	50
	Fax	nvarchar	50
	CV_Link	nvarchar	200

Σχήμα 3.1.2 Πίνακας Person

Στον παραπάνω πίνακα (Σχήμα 3.1.2) γίνεται η εισαγωγή των στοιχείων του ερευνητή. Ο πίνακας Person περιλαμβάνει τον τομέα, τον φορέα και την βαθμίδα του ερευνητή, καθώς επίσης και το επίθετο, το όνομα, το e-mail, το φαξ και τα τηλέφωνα: εργασίας, οικίας και κινητό. Επίσης, υπάρχει το πεδίο CV_Link για το βιογραφικό του ερευνητή. Τέλος παρατηρούμε πως πρωτεύων κλειδί είναι το πεδίο PersonID το οποίο παίρνει αυτόματα ένα αύξοντα αριθμό από την βάση.

➤ Πίνακας Sector

tblSector			
	Column Name	Data Type	Length
🔑	SectorID	int	4
	Description	nvarchar	150

Σχήμα 3.1.3 Πίνακας Sector

Ο παραπάνω πίνακας (Σχήμα 3.1.3) περιέχει δυο πεδία, το ID του τομέα και την περιγραφή του.

➤ Πίνακας ClassLevel

tblClassLevel			
	Column Name	Data Type	Length
🔑	ClassLevelID	int	4
	Title	nvarchar	50

Σχήμα 3.1.4 Πίνακας ClassLevel

Ο παραπάνω πίνακας (Σχήμα 3.1.4) περιέχει δυο πεδία, το ID της βαθμίδας ερευνητή και την περιγραφή του.

➤ Πίνακας ProjectPerson

tblProjectPerson			
	Column Name	Data Type	Length
	ProjectID	int	4
	PersonID	int	4
	IsCoordinator	bit	1

Σχήμα 3.1.5 Πίνακας ProjectPerson

Ο παραπάνω πίνακας (Σχήμα 3.1.5) περιέχει τρία πεδία, το ID του προγράμματος, το ID του ερευνητή και ένα πεδίο για το αν ο ερευνητής είναι συντονιστής ή όχι. Ο πίνακας αυτός χρησιμοποιείται για την αντιστοίχιση του κάθε προγράμματος με τα μέλη της ομάδας έργου και τον επιστημονικό υπεύθυνο.

➤ Πίνακας Organization

tblOrganization			
	Column Name	Data Type	Length
	OrganizationID	int	4
	Name	nvarchar	200
	CountryID	nvarchar	50

Σχήμα 3.1.6 Πίνακας Organization

Ο παραπάνω πίνακας (Σχήμα 3.1.6) περιέχει όλα τα απαραίτητα στοιχεία για τον κάθε οργανισμό. Αποτελείται από τρία πεδία, το ID του οργανισμού, το όνομα του και το ID της χώρας προέλευσης του. Παρατηρούμε πως πρωτεύων κλειδί είναι το πεδίο OrganizationID το οποίο παίρνει αυτόματα ένα αύξοντα αριθμό από την βάση.

➤ Πίνακας Country

tblCountry			
	Column Name	Data Type	Length
	CountryID	nvarchar	50
	CountryName	nvarchar	200

Σχήμα 3.1.7 Πίνακας Country

Ο παραπάνω πίνακας (Σχήμα 3.1.7) περιέχει δυο πεδία, το ID της χώρας και το όνομα της.

➤ Πίνακας Project

tblProject			
	Column Name	Data Type	Length
🔑	ProjectID	int	4
	Acronym_en	nvarchar	200
	Acronym_el	nvarchar	200
	Title_en	nvarchar	200
	Title_el	nvarchar	200
	Description_en	nvarchar	4000
	Description_el	nvarchar	4000
	Budget	float	8
	TotalBudget	float	8
	DateStart	datetime	8
	DateEnd	datetime	8
	Financer	nvarchar	200
	ProjectCategoryID	int	4
	HomeSite	nvarchar	50
	IsGranted	bit	1

Σχήμα 3.1.8 Πίνακας Project

Στο παραπάνω πίνακα (Σχήμα 3.1.8) γίνεται η εισαγωγή των στοιχείων του προγράμματος. Ο πίνακας περιλαμβάνει το ακρόνυμο, τον τίτλο, την περιγραφή, τον συνολικό προϋπολογισμό, τον προϋπολογισμό του ΚΤΕ, τις ημερομηνίες έναρξης και λήξης υλοποίησης του προγράμματος, τον φορέα χρηματοδότησης, την κατηγορία προγράμματος, την ηλεκτρονική διεύθυνση και ένα πεδίο για να δηλώνει αν το πρόγραμμα είναι υλοποιημένο ή όχι. Παρατηρούμε πως πρωτεύων κλειδί είναι το πεδίο ProjectID το οποίο παίρνει αυτόματα ένα αύξοντα αριθμό από την βάση.

➤ Πίνακας ProjectCategory

tblProjectCategory			
	Column Name	Data Type	Length
🔑	ProjectCategoryID	int	4
	Title	nvarchar	50

Σχήμα 3.1.9 Πίνακας Project

Ο παραπάνω πίνακας (Σχήμα 3.1.9) περιέχει την περιγραφή της κατηγορίας προγράμματος και το ID το οποίο παίρνει αυτόματα ένα αύξοντα αριθμό από την βάση και που είναι το πρωτεύων κλειδί.

➤ Πίνακας ProjectOrganization

tblProjectOrganization			
	Column Name	Data Type	Length
	ProjectID	int	4
	OrganizationID	int	4
	IsProjectManager	bit	1

Σχήμα 3.1.10 Πίνακας ProjectOrganization

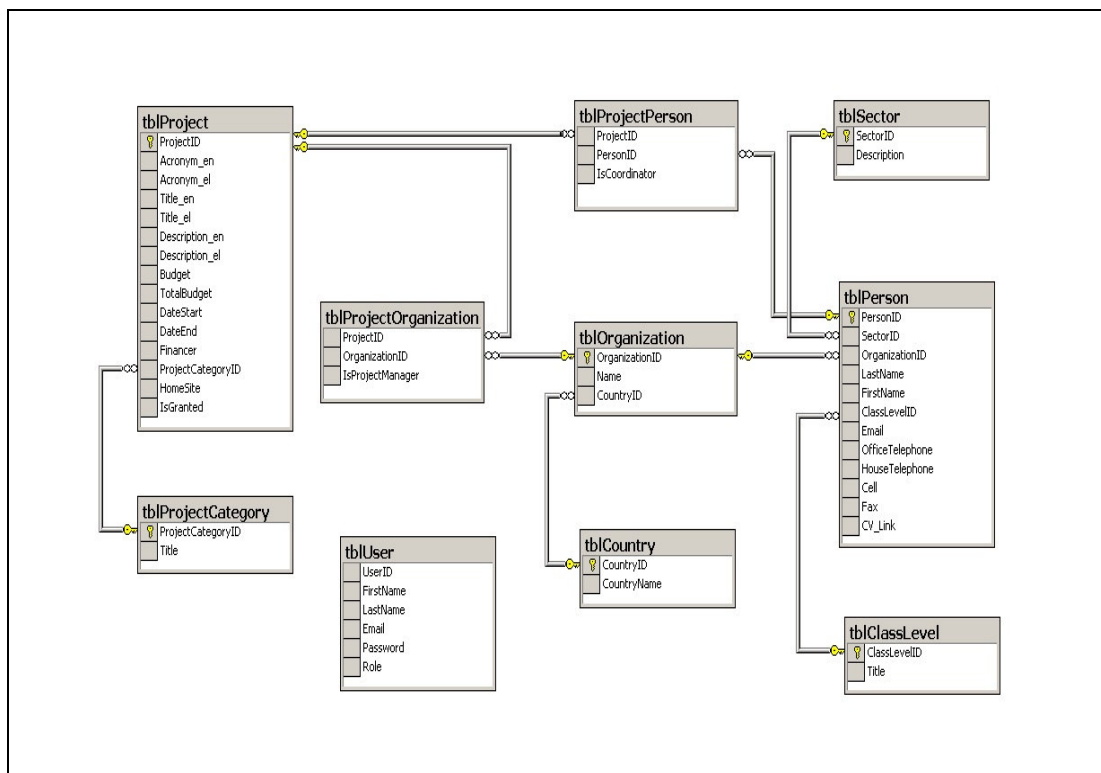
Ο παραπάνω πίνακας (Σχήμα 3.1.10) περιέχει τρία πεδία, το ID του προγράμματος, το ID του οργανισμού και ένα πεδίο για το αν ο οργανισμός είναι συντονιστής ή όχι. Ο πίνακας αυτός χρησιμοποιείται για την αντιστοίχιση του κάθε προγράμματος με τους συμμετέχοντες οργανισμούς και τον συντονιστή οργανισμό.

3.2 Διάγραμμα βάσης δεδομένων – Συσχετίσεις πινάκων

Το διάγραμμα της βάσης δεδομένων είναι μια γραφική απεικόνιση των πινάκων και των μεταξύ τους σχέσεων. Η σχεδίασή του διαγράμματος αποτελεί ένα χρήσιμο για εμάς εργαλείο τόσο για τον έλεγχο της δομής, όσο και των συσχετίσεων των πινάκων της βάσης. Τέλος, αξίζει να σημειωθεί ότι εμφανίζει αυτόματα και τις όποιες αλλαγές που γίνονται στη βάση δεδομένων.

Παρακάτω βλέπουμε στο Σχήμα 3.2.1 το διάγραμμα της βάσης δεδομένων όπου μπορούμε να εξετάσουμε τις σχέσεις μεταξύ των πινάκων. Ο πίνακας ProjectCategory συνδέεται με τον πίνακα Project στο πεδίο ProjectCategoryID με σχέση «σχέση ένα προς πολλά» δηλαδή σε μια κατηγορία προγράμματος μπορούν να ανήκουν πολλά προγράμματα. Ο πίνακας Project συνδέεται με τον πίνακα ProjectOrganization στο πεδίο ProjectID με σχέση «σχέση ένα προς πολλά» δηλαδή σ' ένα πρόγραμμα μπορούν να συμμετέχουν πολλοί οργανισμοί. Ο πίνακας Project συνδέεται με τον πίνακα ProjectPerson στο πεδίο ProjectID με σχέση «σχέση ένα προς πολλά» δηλαδή σ' ένα πρόγραμμα

μπορούν να συμμετέχουν πολλοί ερευνητές. Ο πίνακας Organization συνδέεται με τον πίνακα ProjectOrganization στο πεδίο OrganizationID με σχέση «σχέση ένα προς πολλά» δηλαδή ένας οργανισμός μπορεί να συμμετέχει σε πολλά προγράμματα. Ο πίνακας Country συνδέεται με τον πίνακα Organization στο πεδίο CountryID με σχέση «σχέση ένα προς πολλά» δηλαδή από μια χώρα μπορούν να προέρχονται πολλοί οργανισμοί. Ο πίνακας Organization συνδέεται με τον πίνακα Person στο πεδίο OrganizationID με σχέση «σχέση ένα προς πολλά» δηλαδή σε έναν οργανισμό μπορούν να ανήκουν πολλά άτομα. Ο πίνακας Sector συνδέεται με τον πίνακα Person στο πεδίο SectorID με σχέση «σχέση ένα προς πολλά» δηλαδή σ' ένα τομέα μπορούν να ανήκουν πολλοί ερευνητές. Ο πίνακας Person συνδέεται με τον πίνακα ProjectPerson στο πεδίο PersonID με σχέση «σχέση ένα προς πολλά» δηλαδή ένας ερευνητής μπορεί να συμμετέχει σε πολλά προγράμματα. Ο πίνακας ClassLevel συνδέεται με τον πίνακα Person στο πεδίο ClassLevelID με σχέση «σχέση ένα προς πολλά» δηλαδή σε μια βαθμίδα μπορούν να ανήκουν πολλοί ερευνητές.



Σχήμα 3.2.1 Διάγραμμα βάσης δεδομένων

3.3 Αποθηκευμένες Διεργασίες - Stored Procedure

Κατά τη διάρκεια υλοποίησης της εφαρμογής δημιουργήσαμε αποθηκευμένες διεργασίες ή αλλιώς stored procedures, μέσω του SQL SERVER στον οποίο και αποθηκεύτηκαν. Τα πλεονεκτήματα των stored procedures είναι πολλά, όπως το ότι μπορούν να κληθούν και να εκτελεστούν οποιαδήποτε στιγμή και όσες φορές είναι απαραίτητο κατά τη διάρκεια της εφαρμογής, ενώ και η αλλαγή/ διόρθωση τους είναι πολύ εύκολη. Έτσι όλοι οι χρήστες της Βάσης Δεδομένων μπορούν να χρησιμοποιούν τις ίδιες αποθηκευμένες διεργασίες χωρίς τον μεγάλο κίνδυνο λάθους.

Μάλιστα, όταν ένα stored procedure τροποποιηθεί στον SQL Server, εάν το όνομά αυτού και των μεταβλητών του παραμείνουν ίδια, δεν είναι αναγκαία καμία επέμβαση στον κώδικα ASP.NET.

Επιπρόσθετα, εκτός από την ευκολία οι αποθηκευμένες διεργασίες προσφέρουν και μεγαλύτερη ταχύτητα στην εφαρμογή. Πιο συγκεκριμένα, λόγω του ότι οι παράμετροι των stored procedures περνιούνται στις υπορουτίνες από την πρώτη φορά που καλούνται, επιτρέπουν στις διεργασίες και κατ' επέκταση στην εφαρμογή να εκτελεστεί ταχύτερα.

Ένα ακόμα πλεονέκτημα και άμεσο επακόλουθο των προαναφερθέντων, είναι η σταθερότητα και καλύτερη χρήση των πόρων του συστήματος. Αφού οι αποθηκευμένες διεργασίες εκτελούνται στον SQL Server, το σύστημα των χρηστών και το δίκτυο γενικότερα δεν επιβαρύνεται. Επομένως, και με τη χρήση αξιόπιστου υλικού κατά την υλοποίηση του SQL Server μπορούμε να επιτύχουμε και τα ανάλογα αποτελέσματα.

Ενδεικτικά μερικά από τα stored procedures που χρησιμοποιήθηκαν είναι:

- CREATE PROCEDURE dbo.tblPerson_ins
(
 @SectorID int,
 @OrganizationID int,
 @LastName nvarchar(100),
 @FirstName nvarchar(100),
 @ClassLevelID int,
 @email nvarchar(50),
 @OfficeTelephone nvarchar(50),
 @HouseTelephone nvarchar(50),
 @Cell nvarchar(50),

```

        @Fax nvarchar(50),
        @CV_Link nvarchar(200)
    )
AS
    SET NOCOUNT OFF;
INSERT INTO tblPerson(SectorID, OrganizationID, LastName, FirstName,
ClassLevelID, Email, OfficeTelephone, HouseTelephone, Cell, Fax, CV_Link)
VALUES (@SectorID, @OrganizationID, @LastName, @FirstName,
@ClassLevelID, @Email, @OfficeTelephone, @HouseTelephone, @Cell,
@Fax, @CV_Link);
SELECT PersonID, SectorID, OrganizationID, LastName, FirstName,
ClassLevelID, Email, OfficeTelephone, HouseTelephone, Cell, Fax, CV_Link
FROM tblPerson WHERE (PersonID = @@identity)
GO

```

- CREATE PROCEDURE dbo.tblPerson_sel

```

(
    @PersonID int
)
AS
    SET NOCOUNT ON;
SELECT PersonID, SectorID, OrganizationID, LastName, FirstName,
ClassLevelID, Email, OfficeTelephone, HouseTelephone, Cell, Fax, CV_Link
FROM tblPerson WHERE (PersonID = @PersonID)
GO

```
- CREATE PROCEDURE dbo.tblPerson_upd

```

(
    @PersonID int,
    @SectorID int,
    @OrganizationID int,
    @LastName nvarchar(100),
    @FirstName nvarchar(100),
    @ClassLevelID int,
    @Email nvarchar(50),
    @OfficeTelephone nvarchar(50),
    @HouseTelephone nvarchar(50),
    @Cell nvarchar(50),
    @Fax nvarchar(50),
    @CV_Link nvarchar(200)
)
AS
    SET NOCOUNT OFF;
UPDATE tblPerson SET SectorID = @SectorID, OrganizationID =
@OrganizationID, LastName = @LastName, FirstName = @FirstName,
ClassLevelID = @ClassLevelID, Email = @Email, OfficeTelephone =
@OfficeTelephone, HouseTelephone = @HouseTelephone, Cell = @Cell, Fax =
@Fax, CV_Link = @CV_Link WHERE (PersonID = @PersonID);

```

```

SELECT PersonID, SectorID, OrganizationID, LastName, FirstName,
ClassLevelID, Email, OfficeTelephone, HouseTelephone, Cell, Fax, CV_Link
FROM tblPerson WHERE (PersonID = @PersonID)
GO

```

- CREATE PROCEDURE dbo.tblProjectPerson_del
 (
 @ProjectID int
)
 AS
 SET NOCOUNT ON;
 DELETE FROM tblProjectPerson WHERE (ProjectID = @ProjectID)
 GO
- CREATE PROCEDURE dbo.tblPerson_sel_byEmail
 (
 @Email varchar(50)
)
 AS
 SET NOCOUNT ON;
 SELECT PersonID, SectorID, OrganizationID, LastName, FirstName,
 ClassLevelID, Email, OfficeTelephone, HouseTelephone, Cell, Fax, CV_Link
 FROM tblPerson WHERE (Email = @Email)
 GO

4. Κεφάλαιο – Ανάλυση Διαδικασιών σε ASP.NET

4.1 Επικύρωση για τις Web φόρμες

Στο ASP.NET, μπορείτε να προσθέσετε στοιχεία ελέγχου επικύρωσης σε όλα σχεδόν τα στοιχεία ελέγχου δεδομένων εισόδου της σελίδας, που είναι είτε HTML στοιχεία ελέγχου διακομιστή, είτε στοιχεία ελέγχου διακομιστή μιας Web φόρμας. Μπορούν να ορισθούν περισσότερα από ένα είδη επικύρωσης σε ένα συγκεκριμένο στοιχείο ελέγχου εισόδου, κάνοντας δυνατή την επικύρωση με διάφορα κριτήρια. Όταν εκτελείται η ASP.NET σελίδα με τα στοιχεία ελέγχου επικύρωσης, οι τιμές των στοιχείων ελέγχου εισόδου που επηρεάζονται από τα στοιχεία ελέγχου επικύρωσης υπόκεινται σε επεξεργασία σε σχέση με την αντίστοιχη λογική που τα διέπει, όπως αναφέρεται στα στοιχεία ελέγχου επικύρωσης.. Μετά τους υπολογισμούς, οι ιδιότητες των στοιχείων ελέγχου επικύρωσης παίρνουν την τιμή True ή

False, ανάλογα με το αποτέλεσμα. Μόλις γίνει η επεξεργασία όλων των στοιχείων ελέγχου επικύρωσης, η ίδια η σελίδα ορίζει μια ιδιότητα με βάση τις τιμές των ιδιοτήτων επικύρωσης των στοιχείων ελέγχου. Αν κάποιο από τα στοιχεία ελέγχου δεν περάσει την επικύρωση, η ιδιότητα της σελίδας δίνει αυτόματα την τιμή False στην ιδιότητα επικύρωσης. Όταν δημιουργείτε τον κώδικα επικύρωσης, η ιδιότητα της σελίδας θα είναι μια σημαντική ιδιότητα που πρέπει να ελέγχετε για να είστε σίγουροι ότι όλα είναι σωστά. Αφού ελεγχθεί η ιδιότητα της σελίδας και δείτε ότι η επικύρωση απέτυχε, η σελίδα συνήθως επιστέφει στο χρήστη και θα εμφανιστεί κάποιο μήνυμα σφάλματος, αν έχει ορισθεί ένα τέτοιο μήνυμα στα στοιχεία ελέγχου που απέτυχαν να περάσουν την επικύρωση.

Το ASP.NET έχει έξι είδη στοιχείων ελέγχου επικύρωσης τα: RequiredFieldValidator, RegularExpressionValidator, CompareValidator, RangeValidator, CustomValidator και το ValidationSummary. Όλα αυτά τα στοιχεία ελέγχου έχουν ένα κοινό σύνολο από ιδιότητες και μεθόδους, οι περισσότερες από τις οποίες, κληρονομούνται από την κλάση BaseValidator και από την κλάση WebControl που βρίσκονται στο χώρο ονομάτων System.UI.WebControls και την κλάση Control που μπορεί να βρεθεί στο χώρο ονομάτων System.Web.UI. Μια εξαίρεση σε αυτόν τον κανόνα είναι η ιδιότητα Text, η οποία κληρονομείται από την κλάση Label.

4.1.1 RequiredFieldValidator

Το στοιχείο ελέγχου RequiredFieldValidator ελέγχει αν ο χρήστης έδωσε δεδομένα ή αν επέλεξε μια τιμή στο στοιχείο ελέγχου εισόδου. Μόλις γίνει η αντιστοίχιση με ένα στοιχείο ελέγχου εισόδου, αυτό το στοιχείο ελέγχου επικύρωσης διασφαλίζει ότι εισάγεται μια τιμή στο πεδίο του στοιχείου ελέγχου. Μαζί με τις κοινές ιδιότητες επικύρωσης των στοιχείων ελέγχου, το RequiredFieldValidator έχει μια μοναδική ιδιότητα: την ιδιότητα InitialValue. Αυτή καθορίζει μια αρχική τιμή για το στοιχείο ελέγχου επικύρωσης. Αν η τιμή στο στοιχείο ελέγχου είναι ίση με την τιμή της ιδιότητας InitialValue, η επικύρωση αποτυγχάνει.

Η σύνταξη για αυτό το στοιχείο ελέγχου είναι η παρακάτω:

```
<asp:requiredfieldvalidator  
    id="rvfEmail"
```

```

runat="server"
CssClass="VALIDATOR"
ControlToValidate="txtEmail"
ErrorMessage="Λείπει: Διεύθυνση Email">
</asp:requiredfieldvalidator>

```

Αυτό το παράδειγμα δημιουργεί ένα στοιχείο ελέγχου RequiredFieldValidator με όνομα rvfEmail που επικυρώνει το στοιχείο ελέγχου txtEmail. Αν δεν υπάρχει τιμή στο στοιχείο ελέγχου θα εμφανιστεί το μήνυμα σφάλματος «Λείπει: Διεύθυνση Email». Παρατηρούμε ότι λείπει η ιδιότητα InitialValue που από default έχει τιμή το κενό.

4.1.2 RegularExpressionValidator

Το στοιχείο ελέγχου RegularExpressionValidator ελέγχει αν συμπίπτει ένα στοιχείο με ένα μοτίβο που καθορίζεται από μια κανονική παράσταση (regular expression). Μια *κανονική παράσταση* είναι ένα μοτίβο κειμένου, το οποίο αποτελείται από συνηθισμένους και ειδικούς χαρακτήρες, που είναι γνωστοί (*μετα-χαρακτήρες ή metacharacter*). Το μοτίβο χρησιμεύει ως πρότυπο, καθώς περιγράφει μία ή περισσότερες συμβολοσειρές τις οποίες πρέπει να βρει όταν κάνει αναζήτηση σε ένα κείμενο.

Αυτό το είδος της επικύρωσης σας επιτρέπει να ψάχνετε για προβλεπόμενες ακολουθίες χαρακτήρων, όπως εκείνες που χρησιμοποιούνται για τις διευθύνσεις ηλεκτρονικού ταχυδρομείου, για αριθμούς τηλεφώνου κτλ. Υπάρχει μια μοναδική ιδιότητα που σχετίζεται με τα στοιχεία ελέγχου RegularExpressionValidator: η ιδιότητα ValidationExpression. Προσδιορίζει την κανονική παράσταση που ορίζει τα κριτήρια επικύρωσης.

```

<asp:RegularExpressionValidator
id="rvEmail" runat="server"
CssClass="VALIDATOR"
ValidationExpression="\w+([-+.]\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*"
ControlToValidate="txtEmail"
ErrorMessage="Email: Invalid Format">
</asp:RegularExpressionValidator>

```

Αυτό το παράδειγμα δημιουργεί ένα στοιχείο ελέγχου `RegularExpressionValidator` με όνομα `revEmail` που επικυρώνει το στοιχείο ελέγχου `txtEmail`. Όταν εισάγεται μια τιμή συγκρίνεται με την ιδιότητα `ValidationExpression` που στην συγκεκριμένη περίπτωση επιβεβαιώνει την σωστή εισαγωγή ενός Email.

4.2 Ασφάλεια

Η αρχιτεκτονική ασφάλειας του ASP.NET αποτελείται από δύο μέρη. Το πρώτο μέρος είναι η ενσωμάτωσή της στον IIS. Το δεύτερο βρίσκεται στο βασικό κατάλογο της εφαρμογής ASP.NET, σε ένα αρχείο που ονομάζεται `Web.config`, το οποίο περιέχει διάφορες ρυθμίσεις διαμόρφωσης για την εφαρμογή μας, όπως ρυθμίσεις ασφαλείας.

4.2.1 Ασφάλεια του IIS (Internet Information Server)

Το πρώτο βήμα για την ασφάλεια της εφαρμογής μας ξεκινά από τον Web διακομιστή ο οποίος φιλοξενεί την τοποθεσία μας. Στις περισσότερες περιπτώσεις, αυτός ο Web διακομιστής είναι ο Internet Information Server 5.0+, ο οποίος εγκαθίσταται αυτόματα με τα Windows 2000+ Server και μπορεί να προστεθεί με μη αυτόματο τρόπο στα Windows 2000+ Professional από τα Windows Components. Ο IIS μας δίνει αρκετές μεθόδους για να διασφαλίσουμε τη Web τοποθεσία μας. Οι επιλογές αυτές βρίσκονται στην καρτέλα `Directory Security` (ασφάλεια καταλόγου) του παραθύρου `Properties` (ιδιότητες) της τοποθεσίας μας.

4.2.2 Το αρχείο Web.config

Το `web.config` βρίσκεται αποθηκευμένο ως ένα `xml` αρχείο στον κατάλογο ρίζα και περιγράφει διάφορες ρυθμίσεις διαμόρφωσης. Στο αρχείο αυτό υπάρχει ένας κόμβος που ονομάζεται `<security>` ο οποίος περιέχει πληροφορίες διαμόρφωσης ασφαλείας για την ASP.NET εφαρμογή μας, καθώς και διάφορες άλλες ιδιότητες. Μέσω των ιδιοτήτων αυτών

μπορούμε και να ορίσουμε την προσβασιμότητα των χρηστών, ανάλογα το επίπεδο τους, στα δεδομένα που ζητούν.

Απαραίτητο στοιχείο για τη σύνδεση της εφαρμογής μας με τη βάση δεδομένων είναι το connection string. Προκειμένου να χρησιμοποιείται από όλες τις φόρμες με ευκολία και να μην φαίνεται στον HTML κώδικα. Ο κώδικας που προσθέτουμε στο Web.config για να ορίσουμε το connection string είναι ο παρακάτω:

```
<appSettings>
    <add key="CONNECTION_STRING" value="workstation id=localhost;packet
size=4096;integrated security=SSPI;data source=localhost;persist security
info=False;initial catalog=ProjectAdmin" />
    <add key="PAGE_EXPIRED_STRING" value="Page has expired. Please log
in again" />
</appSettings>
```

4.3 Σύνδεση με Βάση Δεδομένων

Όπως προαναφέραμε απαραίτητη είναι η δημιουργία ενός connection string για την σύνδεση της εφαρμογής μας με τη βάση. Παρακάτω βλέπουμε τις εντολές που χρησιμοποιήσαμε για την σύνδεση, με τη δήλωση ενός SqlConnection του οποίου το connection string το διαβάζουμε από το web.config αρχείο.

```
Dim con As New SqlClient.SqlConnection
Dim cmd As New SqlClient.SqlCommand

Try
    con.ConnectionString =
System.Configuration.ConfigurationSettings.AppSettings("CONNECTION_STRING
")
    con.Open()
    cmd.CommandType = CommandType.StoredProcedure
    cmd.CommandText = "tblProjectOrganization_sel2"
    cmd.Connection = con

    With cmd
        .Parameters.Add("@ProjectID", PID.Trim)
    End With

Catch ex As Exception
    Session("MESSAGE") = ex.Message.ToString
    Session("PAGE") = "project.aspx"
    Response.Redirect("info.aspx")
Finally
```

```

        dr.Close()
        cmd.Dispose()
        con.Close()
    End Try

End Sub

```

Στο επόμενο παράδειγμα βλέπουμε τον τρόπο με τον οποίο εκτελούμε ένα stored procedure: αφού του έχουμε δώσει προηγουμένως τις απαραίτητες παραμέτρους και με την βοήθεια ενός datareader, που ψάχνει και διαβάζει μέσα από τη βάση τον ερευνητή με το συγκεκριμένο ID, επιστρέφει όλα τα στοιχεία του συγκεκριμένου ερευνητή.

Παρατηρούμε την μέθοδο ExecuteReader() η οποία στέλνει το CommandText (cmd) στο Connection (con) και δημιουργεί ένα SqlDataReader που χρησιμοποιείται για να επιστρέψει τα στοιχεία του ερευνητή βάσει του ID του.

```

Private Sub PersonRead()
    Dim con As New SqlClient.SqlConnection
    Dim cmd As New SqlClient.SqlCommand
    Dim dr As SqlClient.SqlDataReader
    Dim st As String
    Dim o As Object

    con.ConnectionString =
System.Configuration.ConfigurationSettings.AppSettings("CONNECTION_STRING
")
    cmd.CommandType = CommandType.StoredProcedure
    cmd.CommandText = "tblPerson_sel"
    cmd.Connection = con
    With cmd
        .Parameters.Add("@PersonID", GetPersonID())
    End With

    Try
        con.Open()
        dr = cmd.ExecuteReader(CommandBehavior.SequentialAccess)
        If dr.Read Then
            st = dr.GetValue(0).ToString.Trim ' PersonID

            o = dr.GetValue(1)
            If IsNothing(o) Then
                drpSector.SelectedIndex = -1
            Else
                drpSector.SelectedValue = o.ToString.Trim
            End If
            drpOrganization.SelectedValue = dr.GetValue(2).ToString.Trim
            txtLastName.Text = dr.GetValue(3).ToString.Trim
            txtFirstName.Text = dr.GetValue(4).ToString.Trim

            o = dr.GetValue(5)
            If IsNothing(o) Then
                drpClassLevel.SelectedIndex = -1
            Else
                drpClassLevel.SelectedValue = o.ToString.Trim
            End If
        End If
    End Try
End Sub

```

```

    End If
    txtEmail.Text = dr.GetValue(6).ToString.Trim
    txtOfficeTelephone.Text = dr.GetValue(7).ToString.Trim
    txtHouseTelephone.Text = dr.GetValue(8).ToString.Trim
    txtCell.Text = dr.GetValue(9).ToString.Trim
    txtFax.Text = dr.GetValue(10).ToString.Trim
    txtCV.Text = dr.GetValue(11).ToString.Trim
    lblResult2.Text = ""
Else
    lblResult2.Text = "Person Not Found!"
End If
Catch ex As Exception
    Session("MESSAGE") = ex.Message.ToString
    Session("PAGE") = "person.aspx"
    Response.Redirect("info.aspx")
Finally
    dr.Close()
    cmd.Dispose()
    con.Close()
End Try

End Sub

```

Στο επόμενο βλέπουμε πως εκτελείται ένα stored procedure μέσα απο την Visual Basic.NET όπου αφού ορίσουμε τις τιμές των παραμέτρων του stored procedure το εκτελούμε με την μέθοδο ExecuteNonQuery().

```

Dim con As New SqlConnection
Dim cmd As New SqlCommand

con.ConnectionString =
System.Configuration.ConfigurationSettings.AppSettings("CONNECTION_STRING")
)
cmd.CommandType = CommandType.StoredProcedure
cmd.CommandText = "tblProjectOrganization_ins"
cmd.Connection = con
With cmd
    .Parameters.Add("@ProjectID", PID.Trim)
    .Parameters.Add("@OrganizationID", OID.Trim)
    .Parameters.Add("@IsProjectManager", IsPM)
End With

Try
    con.Open()
    cmd.ExecuteNonQuery()

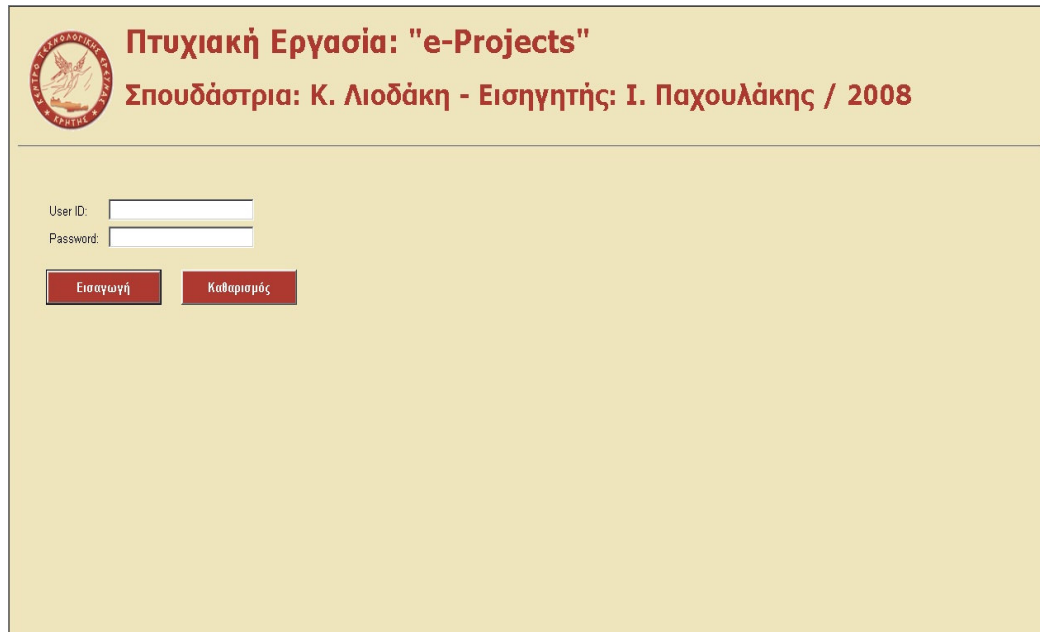
Catch ex As Exception
    Session("MESSAGE") = ex.Message.ToString
    Session("PAGE") = "project.aspx"
    Response.Redirect("info.aspx")
Finally
    cmd.Dispose()
    con.Close()
End Try

End Sub

```

5. Κεφάλαιο – Ανάλυση της εφαρμογής μας

5.1 Η φόρμα logon.aspx



Πτυχιακή Εργασία: "e-Projects"
Σπουδάστρια: Κ. Λιοδάκη - Εισηγητής: Ι. Παχουλάκης / 2008

User ID:
Password:

Σχήμα 5.1 logon.aspx

Η παραπάνω φόρμα logon.aspx (Σχήμα 5.1) είναι η φόρμα στην οποία ο χρήστης, κάνει login, εισάγοντας το e-mail και το password του. Μόλις ολοκληρωθεί η διαδικασία με επιτυχία, ο χρήστης μπορεί να προχωρήσει στην επόμενη φόρμα στην οποία έχει πρόσβαση στα στοιχεία των μελών του ΚΤΕ. Ανάλογα με το επίπεδο του χρήστη (π.χ. administrator, μέλος ομάδας έργου κλπ) θα περιορίζεται και η δυνατότητα προβολής και τροποποίησης των στοιχείων. Τέλος, σε κάθε ένα από τα πεδία υπάρχει ένα RequiredFieldValidator και ένα RegularExpressionValidator με το οποίο υποδεικνύουμε στον χρήστη να συμπληρώσει και με στη σωστή μορφή το κάθε πεδίο ενώ παράλληλα γίνονται και οι αντίστοιχοι έλεγχοι.

5.2 Η φόρμα person.aspx

The screenshot displays the 'KTE Κρήτης - Administration' interface. At the top right, it says 'Αποσύνδεση Χρήστη'. Below this, there are tabs for 'Προσωπικό' and 'Προγράμματα'. The main section is titled 'Κριτήρια Αναζήτησης' (Search Criteria) and includes a sub-header 'Επιλέξτε Φορέα, Τομέα και Βαθμίδα Έρευνας'. There are three dropdown menus for 'Φορέας' (set to 'Center for Technological Research of Crete'), 'Τομέας', and 'Βαθμίδα'. To the right, a list of 26 members is shown, starting with 'Dimitris Giveth'. Below the search criteria, there is a section titled 'ΣΤΟΙΧΕΙΑ ΜΕΛΟΥΣ' (Member Information) with input fields for 'Email', 'Όνομα', 'Επίθετο', 'Τηλέφωνο Γραφείου', 'Τηλέφωνο Οικίας', 'Τηλέφωνο Κλήσης', 'Fax', and 'CV'. There are two buttons, 'Εύρεση' and 'Καθαρισμός', and a 'Βαθμολογία' button at the bottom.

Σχήμα 5.2 person.aspx

Η παραπάνω φόρμα (Σχήμα 5.2) είναι η δεύτερη της web εφαρμογής στην οποία εμφανίζεται η σελίδα αναζήτησης και διαχείρισης των μελών του ΚΤΕ. Πιο συγκεκριμένα, στο πάνω μέρος της φόρμας ο χρήστης μπορεί να θέσει όσα κριτήρια επιθυμεί ή γνωρίζει για την αναζήτηση κάποιου μέλους ορίζοντας τον φορέα, τον τομέα και την βαθμίδα, ενώ τα αποτελέσματα εμφανίζονται στη δεξιά λίστα. Εφόσον δεν τεθούν κριτήρια η λίστα εμφανίζει αλφαβητικά όλα τα μέλη του ΚΤΕ, αφού τον έχουμε ορίσει στην εφαρμογή ως προεπιλεγμένο φορέα. Στη συνέχεια επιλέγοντας ένα μέλος από τη προαναφερθείσα λίστα (κάνοντας κλικ πάνω στο όνομα) εμφανίζονται τα στοιχεία του ατόμου αυτού στο κάτω μέρος της φόρμας.

Επιπρόσθετα, η αναζήτηση ενός ερευνητή μπορεί να επιτευχθεί και από το κάτω πλαίσιο της φόρμας, στα στοιχεία μέλους, εισάγοντας είτε το email, είτε το όνομα είτε το επίθετο του ερευνητή προς αναζήτηση. Τα αποτελέσματα του search εμφανίζονται σε νέα φόρμα, η οποία περιγράφεται παρακάτω.

Τέλος, σε όποιο πεδίο είναι απαραίτητο υπάρχει ένα `RequiredFieldValidator` και ένα `RegularExpressionValidator`, ενώ ο χρήστης αναλόγως του επιπέδου του, `administrator` ή `user`, του επιτρέπεται ή όχι η τροποποίηση των στοιχείων.

5.3 Η φόρμα `PersonsList.aspx`



Σχήμα 5.3 `PersonsList.aspx`

Η παραπάνω φόρμα (Σχήμα 5.3) είναι η φόρμα που εμφανίζει τα αποτελέσματα αναζήτησης της πρώτης φόρμας της web εφαρμογής. Πιο συγκεκριμένα, τα αποτελέσματα της αναζήτησης μέσω email, ονόματος ή επιθέτου, που πραγματοποιείται από το δεύτερο πλαίσιο, παρουσιάζονται σε αυτήν τη φόρμα σε μορφή λίστας. Έτσι, ο χρήστης μπορεί πιο εύκολα να κάνει την επιλογή του.

5.4 Η φόρμα project.aspx

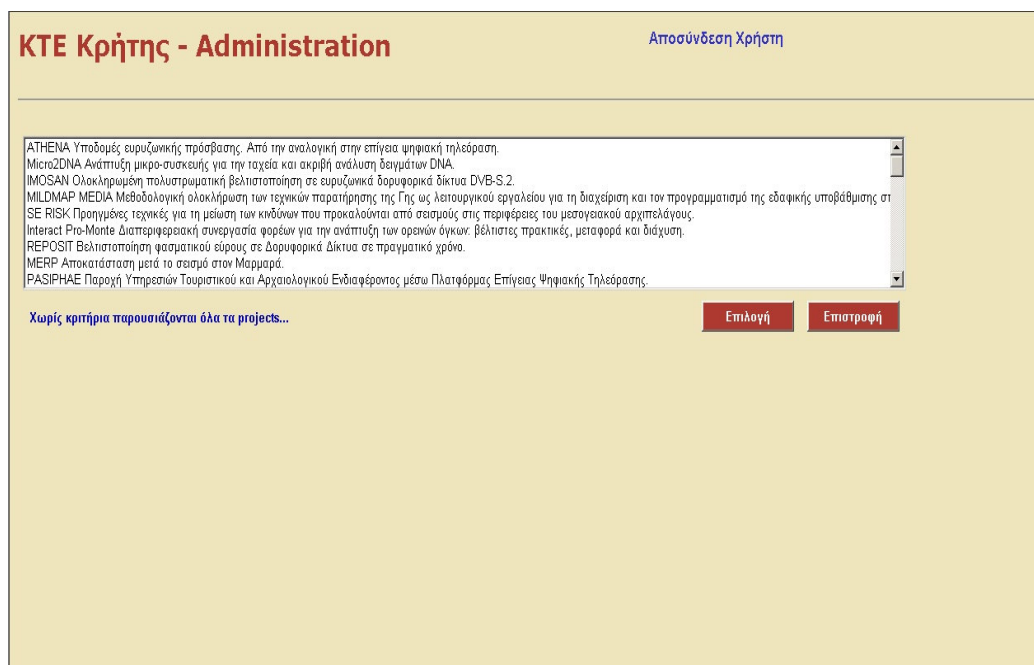
The screenshot shows the 'Κατηγορία Προγράμματος' (Program Category) form in the KTE Crete Administration system. The form is titled 'Κατηγορία Προγράμματος' and has two tabs: 'Ελληνικά' (selected) and 'English'. The form contains several input fields for program details, including 'Τίτλος/Title', 'Αριθμός/ Ακρίβεια', and 'Περιγραφή/ Description'. There are also dropdown menus for 'Στοιχεία Οργανισμού' and 'Μόλις Ουδείς Έγγραφο', and a 'Κατηγορία' dropdown. At the bottom, there are checkboxes for 'Υπογραφή' and 'Υπογραφή Προγράμματος', and a 'Καταχώρηση' button.

Σχήμα 5.4 project.aspx

Η παραπάνω φόρμα (Σχήμα 5.4) είναι η τρίτη φόρμα της web εφαρμογής και σε αυτήν εμφανίζεται η σελίδα αναζήτησης και διαχείρισης των προγραμμάτων που πραγματοποιεί ή συμμετέχει το ΚΤΕ. Πιο συγκεκριμένα, στη φόρμα αυτή ο χρήστης έχει τη δυνατότητα να καταχωρήσει κάποιο πρόγραμμα, να τροποποιήσει τα στοιχεία ενός προγράμματος ή να αναζητήσει ένα πρόγραμμα που τον ενδιαφέρει, συμπληρώνοντας τον ελληνικό ή τον αγγλικό τίτλο. Τα αποτελέσματα του search εμφανίζονται σε νέα φόρμα, η οποία περιγράφεται παρακάτω.

Φυσικά, ο χρήστης αναλόγως του επιπέδου του, administrator ή user, του επιτρέπεται ή όχι η τροποποίηση των στοιχείων.

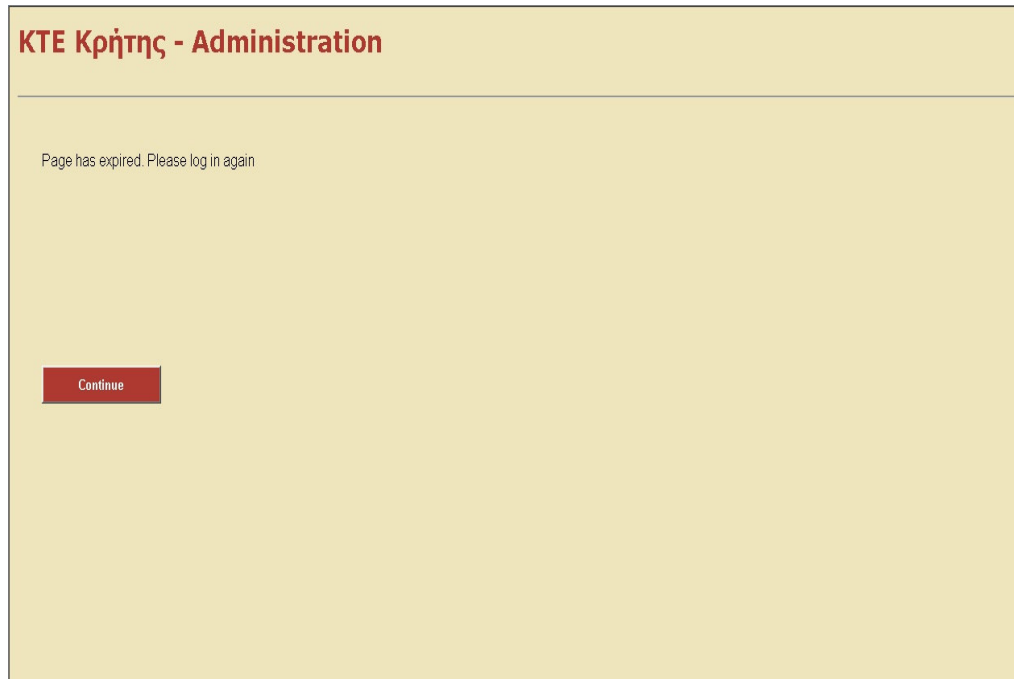
5.5 Η φόρμα ProjectsList.aspx



Σχήμα 5.5 ProjectsList.aspx

Η παραπάνω φόρμα (Σχήμα 5.5) είναι η φόρμα που εμφανίζει τα αποτελέσματα αναζήτησης προγράμματος της web εφαρμογής. Πιο συγκεκριμένα, τα αποτελέσματα αναζήτησης μέσω του ελληνικού ή αγγλικού τίτλου, παρουσιάζονται στη φόρμα αυτήν σε μορφή λίστας. Εδώ ο χρήστης μπορεί να δει αναλυτικά το ακρώνυμο και τον τίτλο του κάθε προγράμματος και να διακρίνει ευκολότερα αυτό που τον ενδιαφέρει.

5.6 Η φόρμα info.aspx



Σχήμα 5.6 info.aspx

Η φόρμα Info.aspx (Σχήμα 5.6) είναι η φόρμα στην οποία, εμφανίζονται τα διάφορα μηνύματα λάθους, σε περίπτωση που ο χρήστης κάνει έναν λάθος χειρισμό ή υπάρχει κάποιο πρόβλημα στον κώδικα της εφαρμογής, το οποίο θα πρέπει να επιλύσει ο administrator του οργανισμού.

6. Βιβλιογραφία – Πηγές

- 📖 SQL Server 2000 Web Application Developer's Guide, *C. Utley, McGraw-Hill/OsborneMedia, 2000*
- 📖 Προγραμματισμός στο Microsoft SQL Server 2000, *R. M. Riordan, Κλειδάριθμος, 2001*
- 📖 Πλήρες Εγχειρίδιο Της Visual Basic .Net, *Πετρούτσος, Γκιούρδας Μ., 2002*
- 📖 Microsoft ASP.NET Programming with Microsoft Visual Basic .NET Version 2003 Step By Step, *G. A. Duthie, Microsoft Press, 2003*
- 📖 Microsoft Visual Basic .NET 2003 Step by Step, *M. Halvorson, Microsoft Press, 2003*
- 📖 ASP.NET Website Programming: Problem - Design - Solution, Visual Basic .NET Edition, *M. Bellinaso, K. Hoffman, Wrox, 2003*
- 📖 Advanced Visual Basic.NET: Programming Web and Desktop Applications in ADO.NET and ASP.NET, *D. Gefen, C. Govindarajulu, Prentice Hall, 2003*
- 📖 Introduction To Programming with Visual Basic .net, *G. J. Bronson, D. Rosenthal, Microsoft Press, 2004*
- 📖 Programming Microsoft Visual Basic .NET Version 2003, *F. Balena, Microsoft Press, 2004*
- 📖 Beginning ASP.NET 1.1 Databases: From Novice to Professional, *D. Foggon, D. Maharry, Apress, 2004*
- 📖 Beginning ASP.NET 1.1 with VB.NET 2003, *C. Ullman, J. Kauffman, C. Hart, D. Sussman, Wrox, 2004*
- 📖 Beginning VB.NET 2003, *T. Willis, J. Crossland, R. Blair, Wrox, 2004*
- 📖 Beginning ASP.NET 2.0, *C. Hart, J. Kauffman, D. Sussman, C. Ullman, Wrox, 2005*
- 📖 Programming Microsoft ASP.NET 2.0 Core Reference, *D. Esposito, Microsoft Press, 2006*
- 📖 Programming Microsoft ASP.NET 2.0 Applications: Advanced Topics, *D. Esposito, Microsoft Press, 2006*
- 📖 Beginning ASP.NET 2.0 and Databases, *J. Kauffman, B. Millington, Wrox, 2006*
- 📖 Microsoft SQL Server 2000 Unleashed, *R. Rankins, P. Jensen, P. Bertucci, Sams, 2006*
- 📖 Professional ASP.NET 2.0 Databases, *T. Thangarathinam, Wrox, 2007*

