



ΤΕΙ ΗΠΕΙΡΟΥ

**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ**  
**Τμήμα Μηχανικών Πληροφορικής Τ.Ε**

**Πτυχιακή Εργασία**

**Τίτλος Εργασίας**

**Υλοποίηση πλατφόρμας  
Υποβολής Ηλεκτρονικών  
Αιτήσεων**

**Σπουδαστής**

***Αγγέλης Αλέξανδρος (ΑΜ961)***

**Επιβλέπων Καθηγητής**

**Μαργαρίτη Σπυριδούλα**

# Ευχαριστίες

Θα ήθελα να ευχαριστώ όλους όσους με βοήθησαν με τις συμβουλές τους προκειμένου να καταφέρω την εκπόνηση της παρούσας πτυχιακής μου εργασίας.

Ευχαριστώ ιδιαίτερα την εισηγήτριά μου κα Μαργαρίτη Σπυριδούλα, για την καθοδήγηση αλλά και την υπομονή της όλο αυτό το διάστημα.

Ευχαριστώ επίσης όλα μέλη της εξεταστικής επιτροπής που έκαναν την τιμή να αξιολογήσουν αυτή μου την προσπάθεια.

Τέλος ευχαριστώ την οικογένειά μου, για την στήριξη τους όλα αυτά τα χρόνια .

*Αγγέλης Αλέξανδρος*

# Σύνοψη

## **Υλοποίηση εφαρμογής υποβολής ηλεκτρονικών αιτήσεων .**

Αντικείμενο της εργασίας είναι η δημιουργία μιας web εφαρμογής μέσω της οποίας θα υποβάλλονται και θα αξιολογούνται αιτήσεις για διαθέσιμες θέσεις εργασίας.

Οι χρήστες της εφαρμογής θα μπορούν να :

- Δημιουργούν λογαριασμό μέσω web διεπαφής.
- Επεξεργάζονται το προφίλ τους μέσω web διεπαφής.
- Υποβάλουν αιτήσεις για τις διαθέσιμες θέσεις.
- Ελέγχουν την κατάσταση των αιτήσεών του.

Ο διαχειριστής της εφαρμογής θα μπορεί να :

- Δημιουργεί νέες θέσεις προς υποβολή αιτήσεων μέσω web διεπαφής.
- Διαχειρίζεται δημιουργημένες θέσεις.
- Αξιολογήσει αιτήσεις χρηστών (έγκριση / απόρριψη).
- Εξάγει σε λίστες στοιχεία χρηστών που έχουν υποβάλει αίτηση για συγκεκριμένη θέση.

Για την υλοποίηση δημιουργήθηκε μια δυναμική ιστοσελίδα για να υποστηρίξει την ηλεκτρονική πύλη υποβολής αιτήσεων για κάλυψη προσφερόμενων θέσεων εργασίας.

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε το Microsoft Visual Studio 2010 Express, ένα ολοκληρωμένο περιβάλλον ανάπτυξης (*IDE - Integrated development environment*).

Η ιστοσελίδα-εφαρμογή έχει δημιουργηθεί με την τεχνολογία LINQ του .NET (ASP.NET). Εκμεταλλευόμενοι τα Web Forms και τις λειτουργίες του Microsoft Visual Studio 2010 Express καταφέρνουμε να έχουμε καλά οργανωμένο και κατανοητό κώδικα.

Ο κώδικας που χρησιμοποιούμε είναι γραμμένος σε γλώσσα Visual Basic.NET, μια γλώσσα σχετικά εύκολη και για τους αρχάριους στον προγραμματισμό.

Η εφαρμογή στηρίζεται επίσης στη λειτουργία βάση δεδομένων σε περιβάλλον Sql Server.

Η βάση δεδομένων είναι απολύτως απαραίτητη για την λειτουργία της εφαρμογής μας, καθώς σε αυτή αποθηκεύονται όλα τα απαραίτητα στοιχεία όπως τα profiles των χρηστών, οι προσφερόμενες προς κάλυψη θέσεις, λεπτομέρειες αυτών κ.ο.κ.

Όταν ένας χρήστης πραγματοποιεί είσοδο στην ιστοσελίδα-εφαρμογή , μπορεί να περιηγηθεί σε όλες τις ανοιχτές προς αίτηση θέσεις αλλά και σε όλες τις θέσεις που πλέον δεν είναι διαθέσιμες προς υποβολή αίτησης.

Με την προϋπόθεση ότι ο χρήστης έχει δημιουργήσει τον προσωπικό του λογαριασμό στο σύστημα και έχει συνδεθεί σε αυτό με τα στοιχεία εγγραφής του, μπορεί να προβεί σε υποβολή αιτήσεων, να επεξεργαστεί το προφίλ του καθώς και να παρακολουθεί την κατάσταση των αιτήσεων που έχει υποβάλει.

Τέλος, για τους διαχειριστές του συστήματος (administrators), έχει δημιουργηθεί ξεχωριστό login account που έχει οριστεί ως admin account το οποίο παρέχει στους τελευταίους επιπλέον δυνατότητες όπως την καταχώρηση νέων θέσεων στο σύστημα, την επεξεργασία παλαιότερων θέσεων, την δυνατότητα view των profiles των χρηστών, την έγκριση και απόρριψη αιτήσεων χρηστών καθώς επίσης και την δυνατότητα εξαγωγής στοιχείων από την βάση δεδομένων με μορφή αρχείων .xls.

# Πίνακας Περιεχομένων

Ευχαριστίες	1
Σύνοψη	2
Πίνακας Περιεχομένων	4
<b>1.</b>	<b>Εισαγωγή</b>
1.1 Στόχοι και Σκοπός Εργασίας	6
1.2 Απαιτήσεις Συστήματος	6
<b>2.</b>	<b>Εργαλεία υλοποίησης Εφαρμογής</b>
2.1 Visual Studio.NET & .Net Framework	8
2.2 ASP.NET	9
2.3 Αρχιτεκτονική ASP.NET	10
2.4 Πλεονεκτήματα χρήσης ASP.NET	10
2.5 Visual Basic .NET	11
<b>3.</b>	<b>Η Βάση Δεδομένων της Εφαρμογής</b>
3.1 Βάση -Δεδομένων και Sql Server	11
3.2 Sql Server Management Studio	13
3.3 Διάγραμματα Βάση Δεδομένων Πτυχιακής Εργασίας	14
3.4 Πίνακες της βάσης δεδομένων Πτυχιακής Εργασίας	15
3.4.1 Πίνακας dbo.prolife	15
3.4.2 Πίνακας dbo.Country	15
3.4.3 Πίνακας dbo.Job	15
3.4.4 Πίνακας dbo.Education	16
3.4.5 Πίνακας dbo.Recommendation	16
3.4.6 Πίνακας dbo.position	17
3.4.7 Πίνακας dbo.position request	17
<b>4.</b>	<b>Επικυρώσεις και ασφάλεια σε ASP.NET</b>
4.1 Επικυρώσεις για Web φόρμες	19
4.2 RequiredFieldValidator	19
4.3 RegularExpressionValidator	21
4.4 CompareValidator	22
4.5 CSS (Cascading Style Sheet)	22
4.6 Σύνταξη/ Εφαρμογή CSS σε Html	24
4.7 Ασφάλεια ASP.NET και IIS	25
4.8 Ροή ασφαλείας σε μια αίτηση	25
4.9 Έλεγχος ταυτότητας	25
4.10 web.config	27
4.11 Cookies	29
4.12 Σύνδεση με βάση δεδομένων	29
<b>5.</b>	<b>Παρουσίαση της Εφαρμογής</b>
5.1 Η εφαρμογή σε επίπεδο διαχειριστή	31
5.1.1 Δημιουργία Νέας Θέσης	32
5.1.2 Επεξεργασία Θέσης	34
5.1.3 Έλεγχος/Αξιολόγηση αιτήσεων χρηστών	36
5.1.4 Έγκριση / Απόρριψη αίτησης χρήστη για συγκεκριμένη θέση	36
5.1.5 Εξαγωγή δεδομένων χρηστών σε .xls	36
5.2 Η εφαρμογή σε επίπεδο χρήστη	37
5.2.1 Δημιουργία λογαριασμού στο σύστημα (νέος χρήστης)	37
5.2.2 Σύνδεση στο σύστημα (υπάρχον χρήστης)	42
5.2.3 Άμεση υποβολή αίτησης και ταυτόχρονη υποβολή για συγκεκριμένη θέση (νέος και υπάρχον χρήστης)	41
5.2.4 Επεξεργασία Profile (βιογραφικού)	42
5.2.5 Έλεγχος κατάστασης αιτήσεων	45

<b>6.</b>	
6.1 Συνοπτικός πίνακας σελίδων της εφαρμογής	46
6.2 index.aspx	47
6.3 view-position.aspx	49
6.4 account.aspx	53
6.5 profile.aspx	57
6.6 edit-school.aspx	65
6.7 edit-jobs.aspx	69
6.8 edit-recommendations.aspx	74
6.9 applications.aspx	77
6.10 modify-position.aspx	79
<b>7.</b>	
7.1 Έντυπη βιβλιογραφία	91
7.2 Σελίδες στο Διαδύκτιο	91

## **Παράρτημα**

## **Βιβλιογραφία**

# 1. Εισαγωγή

## 1.1 Στόχοι και Σκοπός Εργασίας

Στόχος της παρούσας πτυχιακής εργασίας είναι η δημιουργία μιας διαδικτυακής εφαρμογής (web application) με λειτουργίες όπως η εγγραφή χρήστη, η δημιουργία προφίλ χρηστών, η αναζήτηση και υποβολή αιτήσεων από τους χρήστες, η δημιουργία νέων θέσεων και η αξιολόγηση των αιτήσεων των χρηστών από τους διαχειριστές της εφαρμογής.

Σκοπός της πτυχιακής εργασίας είναι η απόκτηση γνώσεων πάνω σε θέματα όπως ανάπτυξη και σχεδιασμός βάσης δεδομένων με χρήση Sql Server, κατασκευή ιστοσελίδας και διεπαφής χρήστη με χρήση ASP.NET, χρησιμοποιώντας τα προγράμματα Microsoft Visual Studio 2010 και Sql Server Management Studio.

## 1.2 Απαιτήσεις Συστήματος

Οι απαιτούμενες τεχνικές προδιαγραφές για την υλοποίηση της πλατφόρμας είναι οι εξής:

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

- Windows 7 Ultimate x86
- Windows 7 Ultimate N
- Windows 7 Ultimate x64
- Windows 7 Enterprise x86
- Windows 7 Enterprise N
- Windows 7 Enterprise x64
- Windows 7 Professional x86
- Windows 7 Professional N
- Windows 7 Professional x64
- Windows 7 Home Premium x86
- Windows 7 Home Premium N
- Windows 7 Home Premium x64
- Windows 7 Home Basic x86
- Windows 7 Home Basic N
- Windows 7 Starter x86
- Windows 7 Starter N
- Windows Vista R2 Enterprise (x64)
- Windows Vista R2 Enterprise
- Windows Vista R2 Business (x64)
- Windows Vista R2 Business
- Windows Vista R2 Ultimate (x64)
- Windows Vista R2 Ultimate
- Windows Vista R2 Home Premium (x64)
- Windows Vista R2 Home Premium
- Windows Vista R2 Home Basic
- Windows Vista R2 Starter Ed. Digital Boost (x64)
- Windows Vista R2 Starter Ed. Digital Boost
- Windows Vista R2 Starter
- Windows Vista Ultimate
- Windows Vista Ultimate x64 Edition
- Windows Vista Enterprise

- Windows Vista Enterprise x64 Edition
- Windows Vista Business
- Windows Vista Business x64 Edition
- Windows Vista Home Premium
- Windows Vista Home Premium x64 Edition
- Windows Vista Home Basic
- Windows Vista Starter Ed. Digital Boost
- Windows Vista Starter Ed. Digital Boost (x64)
- Windows Vista Starter
- Windows XP Professional
- Windows XP Professional x64 Edition
- Windows XP Home Edition

Για το SQL Server 2008 ένα από τα παρακάτω:

- Windows XP Professional x64
- Windows Server 2003 SP2 64-bit x64 Standard
- Windows Server 2003 SP2 64-bit x64 Datacenter
- Windows Server 2003 SP2 64-bit x64 Enterprise
- Windows Vista Ultimate x64
- Windows Vista Enterprise x64
- Windows Vista Business x64
- Windows Server 2008 x64 Web
- Windows Server 2008 x64 Standard, Windows Server 2008 x64 Standard without Hyper-V
- Windows Server 2008 x64 Datacenter, Windows Server 2008 x64 Datacenter without Hyper-V
- Windows Server 2008 x64 Enterprise, Windows Server 2008 x64 Enterprise without Hyper-V
- Windows Small Business Server 2008
- Windows Server 2008 for Windows Essential Server Solutions
- Windows Server 2008 without Hyper-V for Windows Essential Server Solutions
- Windows 7 64-bit x64 Ultimate
- Windows 7 64-bit x64 Enterprise
- Windows 7 64-bit x64 Professional
- Windows 2008 R2 64-bit x64 Web
- Windows 2008 R2 Foundation Server
- Windows 2008 R2 64-bit x64 Standard
- Windows 2008 R2 64-bit x64 Enterprise
- Windows 2008 R2 64-bit x64 Datacenter



## 2. Εργαλεία υλοποίησης εφαρμογής

### 2.1 Visual Studio.NET & .Net Framework

Το Visual Studio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment - IDE) που χρησιμοποιείται για την ανάπτυξη διαδικτυακών εφαρμογών καθώς και διαδικτυακών υπηρεσιών (Web Services).

Το Visual Studio παρέχει προχωρημένα εργαλεία για τη διόρθωση λαθών, για τη τεκμηρίωση και εγγραφή κώδικα, για την ανάπτυξη διεπαφών χρήστη, για τη σχεδίαση κλάσεων καθώς και για τη σχεδίαση του σχήματος μίας βάσης δεδομένων. Επίσης υποστηρίζει πολλά plug-ins που προσφέρουν προηγμένες λειτουργίες όπως unit testing και refactoring. Οι ενσωματωμένες γλώσσες προγραμματισμού του Visual Studio είναι οι Visual C++, Visual C# και Visual Basic. Επίσης, παρέχεται υποστήριξη και για άλλες γλώσσες όπως τις F#, Python, Ruby οι οποίες εγκαθίστανται ξεχωριστά μέσω των language services.



Εικόνα 1: Αρχιτεκτονική Visual Studio.NET <http://pluto.ksi.edu/~cyh/cis501/ASPDotNet.htm>

Το .Net Framework τώρα, είναι αναπόσπαστο κομμάτι των Windows και υποστηρίζει την ανάπτυξη και λειτουργία εφαρμογών και διαδικτυακών υπηρεσιών (Web Services). Το .NET Framework έχει σχεδιαστεί για να εκπληρώσει τους ακόλουθους στόχους:

- Να παράσχει ένα συνεκτικό αντικειμενοστραφή προγραμματιστικό περιβάλλον όπου ο κώδικας είτε αποθηκεύεται και εκτελείται τοπικά, είτε εκτελείται τοπικά αλλά διανέμεται στο διαδίκτυο, είτε εκτελείται από απόσταση.
- Να παράσχει ένα περιβάλλον εκτέλεσης κώδικα που αυξάνει τη παραγωγικότητα του προγραμματιστή και ελαχιστοποιεί τις πιθανές συγκρούσεις μεταξύ διαφορετικών εκδόσεων του .Net Framework.
- Να παράσχει ένα περιβάλλον εκτέλεσης κώδικα που να προάγει την ασφαλή εκτέλεση του κώδικα, περιλαμβανομένου του κώδικα που δημιουργήθηκε από άγνωστο ή μη έμπιστο τρίτο μέρος.
- Να παρέχει ένα σταθερό περιβάλλον ανάπτυξης για όλα τα είδη εφαρμογών (Windows-based ή Web-based εφαρμογές).
- Να εξασφαλίσει ότι ο κώδικας που βασίζεται στο .NET Framework μπορεί να ενσωματωθεί σε οποιονδήποτε άλλο κώδικα.

Το .NET Framework έχει δύο κύρια στοιχεία: Το Common Language Runtime (CLR) και τη βιβλιοθήκη κλάσεων του .NET Framework. Το CLR καθορίζει το περιβάλλον για την εκτέλεση κώδικα του

προγράμματος και οι προγραμματιστές που χρησιμοποιούν το CLR γράφουν κώδικα σε μια γλώσσα όπως τη Visual Basic ή τη C#. Κατά τη διάρκεια της μεταγλώττισης του προγράμματος ο .NET μεταγλωττιστής μετατρέπει τον εν λόγω κώδικα σε ενδιάμεσο κώδικα (byte code) που ονομάζεται Common Intermediate Language (CIL), ο οποίος ουσιαστικά ορίζει οδηγίες για το (CLR) που λειτουργεί ως εικονική μηχανή (virtual machine). Κατόπιν κατά τη διάρκεια της εκτέλεσης ο μεταγλωττιστής του CLR μετατρέπει τον CIL κώδικα σε κώδικα μηχανής. Το CLR επιτρέπει στους προγραμματιστές να αγνοήσουν πολλές λεπτομέρειες της συγκεκριμένης CPU που θα εκτελέσει το πρόγραμμα και παρέχει και άλλες σημαντικές υπηρεσίες, όπως:

- Διαχείριση μνήμης
- Διαχείριση νημάτων (threads)
- Διαχείριση εξαιρέσεων
- Συλλογή απορριμμάτων (garbage collection)
- Ασφάλεια

Το ADO.NET (ActiveX Data Object) είναι μοντέλο της Microsoft για το .NET Framework. Δεν είναι απλά η μεταφορά του γνωστού ADO μοντέλου σε περιβάλλον διαχείρισης, αλλά ένα τελείως καινούριο παράδειγμα για πρόσβαση σε δεδομένα.

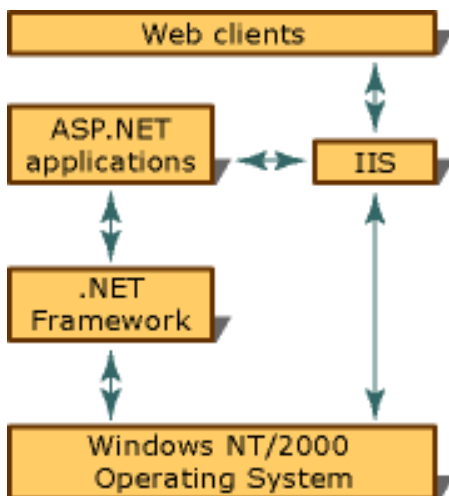
Το τέταρτο επίπεδο του framework αποτελείται από το μοντέλο εφαρμογής Windows και παράλληλα το μοντέλο εφαρμογής Web. Οι υπηρεσίες Web είναι λύσεις λογισμικού διαμοιρασμένες από το διαδίκτυο σε οποιαδήποτε συσκευή. Το CLS είναι μια συλλογή από κανόνες που περιορίζουν την κάθε γλώσσα προγραμματισμού και πρέπει να τους ακολουθήσει, έτσι ώστε να μην έχουμε γλώσσες που τρέχουν και να δημιουργούν δικές τους παραμέτρους και νέους τύπους δεδομένων. Το CLR και το .NET Framework είναι σχεδιασμένα με τέτοιο τρόπο που ο κώδικας που έχει γραφτεί σε μια γλώσσα να μην μπορεί απρόσκοπτα να χρησιμοποιηθεί από άλλη γλώσσα.

## **2.2 ASP.NET**

Η ASP.NET έχει αναπτυχθεί και διατεθεί εμπορικά από τη Microsoft Corporation. Δημιουργήθηκε για να επιτρέπει στους προγραμματιστές τη δημιουργία δυναμικών ιστοσελίδων, καθώς επίσης και web εφαρμογών. Η ASP.NET ανήκει στην κατηγορία γλωσσών ανάπτυξης web εφαρμογών που περιλαμβάνει τεχνολογίες όπως η PHP, JSP, Ruby on Rails. Τέτοιου είδους γλώσσες είναι σχεδιασμένες να τρέχουν στο server και όχι στον browser του πελάτη (χρήστη). Ο κώδικας που ο προγραμματιστής γράφει σε ASP.NET εκτελείται στο server και παράγει HTML σελίδες με δυναμικό τρόπο. Ανάλογα με τη λογική της εφαρμογής μια συγκεκριμένη σελίδα μπορεί να μοιάζει διαφορετική π.χ. για διαφορετικούς χρήστες, ή σε διαφορετικές χρονικές στιγμές. Αυτό γίνεται αυτόματα χωρίς να χρειάζεται τροποποίηση της σελίδας όπως θα είχαμε στην περίπτωση των HTML σελίδων. Οι σελίδες γραμμένες σε ASP.NET ονομάζονται Web Forms και τέτοιου είδους web sites ονομάζονται εφαρμογές web (web applications). Τα αρχεία αυτής της μορφής έχουν κατάληξη «.aspx».

## 2.3 Αρχιτεκτονική ASP.NET

Παρακάτω γίνεται μια σύντομη επισκόπηση της υποδομής της ASP.NET και των σχέσεων του υποσυστήματος, που σχετίζονται με το θέμα της ασφάλειας. Στην παρακάτω εικόνα φαίνονται οι σχέσεις μεταξύ των συστημάτων ασφάλειας στην ASP.NET.



Εικόνα 2: Αρχιτεκτονική ASP.NET [http://msdn.microsoft.com/en-us/library/yedba920\(v=vs.71\).aspx](http://msdn.microsoft.com/en-us/library/yedba920(v=vs.71).aspx)

Όλοι οι Web Clients (χρήστες) επικοινωνούν με τις ASP.NET εφαρμογές μέσω του Microsoft Internet Information Services (IIS). Το IIS αποκρυπτογραφεί και επίσης επικυρώνει το αίτημα. Αν έχει επιλεγεί η επιλογή Allow Anonymous, δε γίνεται έλεγχος ταυτότητας. Το IIS βρίσκει την πηγή που ζητήθηκε (πχ εφαρμογή ASP.NET) και αν ο πελάτης (χρήστης) έχει λάβει την άδεια, επιστρέφει την κατάλληλη πηγή (απάντηση). Εκτός από τα ενσωματωμένα χαρακτηριστικά ασφάλειας, μια εφαρμογή ASP.NET μπορεί να χρησιμοποιήσει τα χαμηλού επιπέδου χαρακτηριστικά ασφάλειας του .Net Framework. Στην ASP.NET ο compiler μεταγλωττίζει γλώσσες όπως οι Visual Basic, C++, C# στο Visual Studio και στη συνέχεια δημιουργείται ένας ενδιάμεσος κώδικας, MSIL (Microsoft Intermediate Language) στον οποίο περιέχονται metadata. Τα metadata χρησιμοποιούνται από το CLR για την εκτέλεση συγκεκριμένων λειτουργιών. Το προτέρημα αυτής της λειτουργία είναι ότι προσφέρει έναν ενιαίο τρόπο για την αξιοποίηση των υπηρεσιών του λειτουργικού συστήματος. Καθώς άλλες υπηρεσίες βρίσκονται σε αρχεία DLL και άλλες σε αρχεία COM, το αποτέλεσμα είναι να υπάρχει πλέον διευκόλυνση, καθώς δεν απαιτείται η αντικατάσταση ενός αρχείου DLL μιας X εφαρμογής για τη εγκατάσταση μιας νέας εφαρμογής που πιθανότατα θα είχε απρόβλεπτα αποτελέσματα για το σύστημά μας.

## 2.4 Πλεονεκτήματα χρήσης ASP.NET

Μερικά από τα βασικά πλεονεκτήματα που καθιέρωσαν την ASP.NET ως μια από τις σημαντικότερες γλώσσες ανάπτυξης web εφαρμογών είναι τα παρακάτω :

- Χρησιμοποιεί το .NET framework, που αναφέραμε παραπάνω, δηλαδή ένα πακέτο που προσφέρει έτοιμη λειτουργικότητα στον προγραμματιστή για την διευκόλυνσή του με βασικές διαδικασίες.
- Επιτρέπει την γραφή κώδικα σε μια από περισσότερες από 40 γλώσσες προγραμματισμού μεταξύ των οποίων και οι πιο διαδεδομένες όπως C# και Visual Basic.

- Βάση σχεδιασμού επιτρέπει την ανεξαρτητοποίηση της λογικής μιας σελίδας από το παρουσιαστικό της. Αλλαγές που κάνει ενδεχομένως κάποιος στην εμφάνιση μιας web εφαρμογής δεν επηρεάζουν απαραίτητα και τη λογική της εφαρμογής.
- Υπάρχουν κορυφαία εργαλεία ανάπτυξης εφαρμογών με ASP.NET όπως είναι το Visual Studio.

## 2.5 Visual Basic .NET

Η γλώσσα Visual Basic έκανε την εμφάνιση της το 1991 και είναι μια συνέχεια της κλασσικής γλώσσας προγραμματισμού Basic. Η Visual Basic προσφέρει πολύ ισχυρά χαρακτηριστικά όπως τα γραφικά περιβάλλοντα χρηστών, τον χειρισμό συμβάντων και εξαιρέσεων και κυρίως τον αντικειμενοστραφή προγραμματισμό. Από την στιγμή που η Microsoft εισήγαγε την πλατφόρμα .NET, σχεδίασε μια έκδοση της Visual Basic για την συγκεκριμένη πλατφόρμα. Η ανανεωμένη γλώσσα αυτή προσφέρει βελτιωμένη αντικειμενοστρέφεια, συμπεριλαμβανομένης μιας βιβλιοθήκης συστατικών και επιτρέποντας στους προγραμματιστές να αναπτύσσουν εφαρμογές ακόμα πιο γρήγορα και εύκολα. Η Visual Basic .NET δίνει ακόμα την δυνατότητα διάφορα συστατικά ενός λογισμικού που έχουν αναπτυχθεί με διάφορες γλώσσες προγραμματισμού, να αλληλεπιδρούν τέλεια μεταξύ τους.

# 3. Η Βάση Δεδομένων της εφαρμογής

## 3.1 Βάση Δεδομένων και Sql Server

Μια βάση δεδομένων είναι ένα εργαλείο συλλογής και οργάνωσης πληροφοριών. Παρότι θα μπορούσαν να δοθούν αρκετοί ορισμοί, μια βάση δεδομένων μπορεί να περιγραφεί ως μια οργανωμένη συλλογή δεδομένων για γρήγορη αναζήτηση και ανάκτηση αυτών από έναν υπολογιστή. Επίσης μολονότι στο παρελθόν υπήρξαν διάφορα μοντέλα για βάσεις δεδομένων, αυτό που τελικά επικράτησε είναι το σχεσιακό μοντέλο βάσεων δεδομένων.

Μια βάση δεδομένων αποτελείται από το ΣΔΒΔ (Σύστημα Διαχείρισης Βάσης Δεδομένων) καθώς και ένα σύνολο εσωτερικών αρχείων που αποθηκεύονται τα δεδομένα. Ωστόσο η πρόσβαση στα δεδομένα γίνεται αποκλειστικά μέσω του ΣΔΒΔ και όχι άμεσα στα αρχεία. Η επικοινωνία με το ΣΔΒΔ γίνεται μέσω της SQL. Δηλαδή τα διάφορα προγράμματα εφαρμογών που χρειάζονται δεδομένα από τη βάση, επικοινωνούν με το ΣΔΒΔ μέσω της SQL και τελικά παρουσιάζουν στον χρήστη τις ζητούμενες πληροφορίες.

Ο SQL Server είναι ένα σύστημα διαχείρισης βάσεων δεδομένων (Database Management System) που έχει αναπτυχθεί και προωθείται από τη Microsoft. Ως σύστημα διαχείρισης βάσης δεδομένων υποστηρίζει τα εξής χαρακτηριστικά:

- Περιγραφή δεδομένων
- Ανεξαρτησία δεδομένων και λειτουργιών
- Αποδοτικότερη διαχείριση δεδομένων,

- Προστασία δεδομένων και δικαιώματα χρηστών
- Μηχανισμούς ταυτόχρονης προσπέλασης και επεκτασιμότητα.

Η σχεσιακή γλώσσα που χρησιμοποιεί ο SQL Server για την επικοινωνία των εφαρμογών με το ΣΔΒΔ ονομάζεται Transact – SQL (T – SQL). Είναι η «διάλεκτος» της πιο σημαντικής γλώσσας βάσεων δεδομένων που υπάρχει σήμερα, της Δομημένης Γλώσσας Ερωτημάτων SQL (Structured Query Language). Ο SQL Server παρέχει επίσης μια σειρά πρόσθετων υπηρεσιών οι οποίες, αν και δεν είναι απαραίτητες για τη λειτουργία του συστήματος βάσης δεδομένων, παρέχουν παρόλα αυτά υπηρεσίες προστιθέμενης αξίας πάνω από το κεντρικό σύστημα διαχείρισης βάσης δεδομένων. Οι υπηρεσίες αυτές λειτουργούν είτε ως μέρος κάποιου στοιχείου του SQL Server, είτε αυτόνομα ως υπηρεσία των Windows η οποία χρησιμοποιεί το δικό της API για να αλληλεπιδρά με αυτές.

Οι υπηρεσίες αυτές είναι οι εξής:

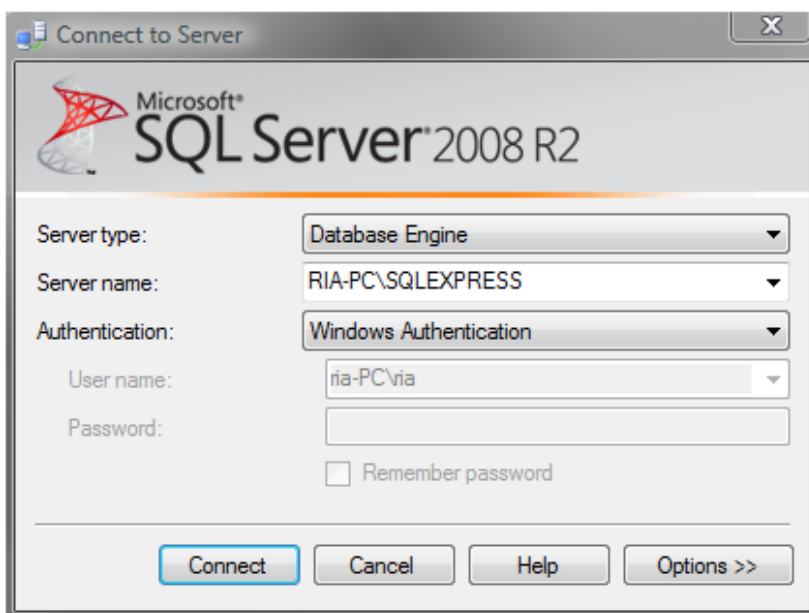
- **Ο Μεσίτης Υπηρεσιών (Service Broker)**  
Παρέχει μια αξιόπιστη πλατφόρμα ανταλλαγής δεδομένων για SQL Server εφαρμογές. Για τις cross instance εφαρμογές χρησιμοποιεί το πρωτόκολλο TCP/IP ώστε να μπορέσουν τα διαφορετικά στοιχεία (components) να συγχρονιστούν μεταξύ τους, μέσω της ανταλλαγής μηνυμάτων.
- **Οι υπηρεσίες συγχρονισμού (Replication)**  
Χρησιμοποιούνται για την αντιγραφή και τον συγχρονισμό αντικειμένων της βάσης δεδομένων, είτε εξ ολοκλήρου είτε ως υποσύνολο των υπάρχοντων αντικειμένων.
- **Οι υπηρεσίες ανάλυσης (Analysis Services)**  
Προσθέτουν τις ικανότητες OLAP και περίπλοκης ανάκτησης δεδομένων για τις βάσεις δεδομένων SQL, με περίπλοκη ανάλυση χωρίς την ανάγκη για δύσκολες SQL queries. Υποστηρίζεται το MOLAP, ROLAP και HOLAP για τα στοιχεία.
- **Reporting Υπηρεσίες (Reporting Services)**  
Ο SQL Server έχει υπηρεσίες Reporting για στοιχεία που συγκεντρώνονται από τις βάσεις δεδομένων SQL Server. Με αυτόν τον τρόπο είναι δυνατό να παραχθεί μια έκθεση Ιστού (Web Report) σχεδόν αυτόματα χωρίς προγραμματισμό HTML, CSS ή JavaScript, μέσα από το Visual Studio. Επιβλέπει, μέσω μιας διαδικτυακής διεπαφής (web interface), δεδομένα που έχουν συγκεντρωθεί από μία SQL Server βάση δεδομένων.
- **Οι υπηρεσίες ανακοίνωσης (Notification Services)**  
Δίνουν την δυνατότητα στον χρήστη να παράγει μηνύματα ηλεκτρονικού ταχυδρομείου με πολύ λίγο κόπο.
- **Οι υπηρεσίες ενσωμάτωσης (Integration Services)**  
Χρησιμοποιούνται για να ενσωματώσουν στοιχεία από διαφορετικές πηγές δεδομένων. Χρησιμοποιεί τις ικανότητες ETL. Οι Integration Services περιλαμβάνουν εργαλεία GUI για να χτίσουν διάφορες λειτουργίες ροών δεδομένων, όπως η εξαγωγή στοιχείων από διάφορες πηγές, η συζήτηση στοιχείων, ο μετασχηματισμός των στοιχείων συμπεριλαμβανομένης της συνάθροισης, του διπλασιασμού και η συγχώνευση.
- **Η υπηρεσία αναζήτησης πλήρους κειμένου (Full Text Search Service)**  
Προσφέρει εξειδικευμένη αναζήτηση κειμένου που είναι αποθηκευμένο σε μια SQL Server βάση δεδομένων.

## 3.2 Sql Server Management Studio

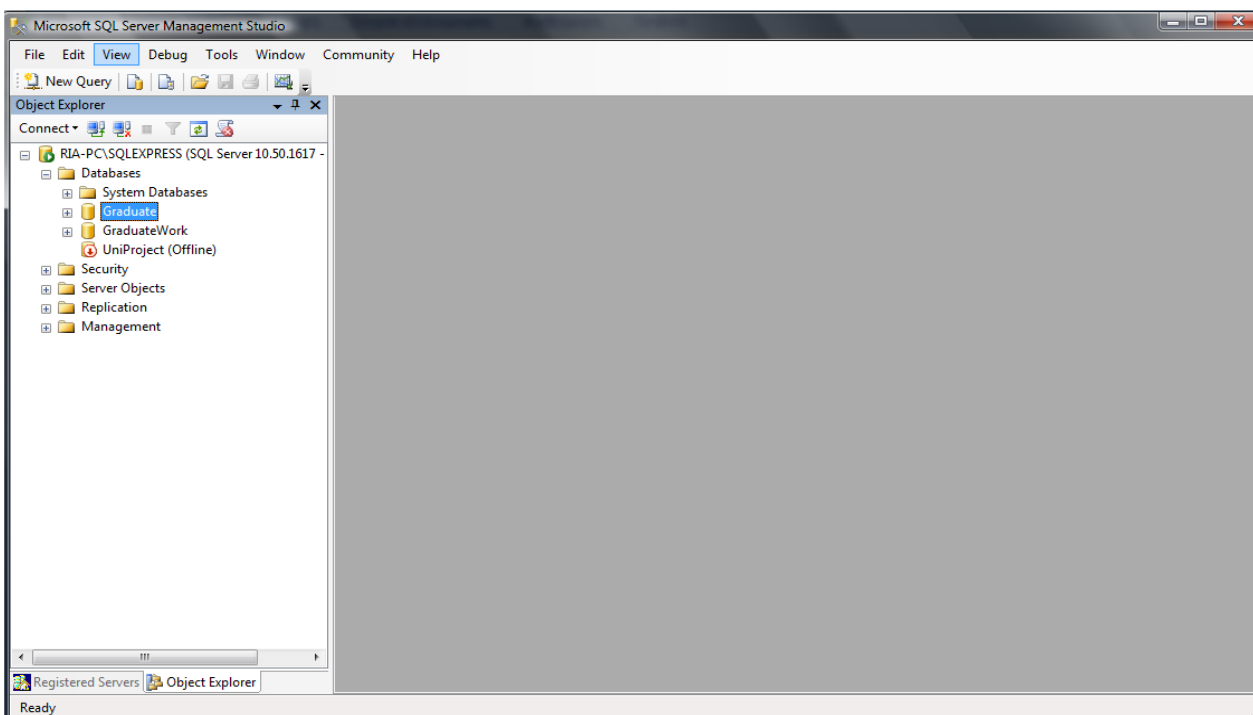
Ο Sql Server Management Studio είναι το βασικό εργαλείο που χρησιμοποιήσαμε για την διαχείριση της βάσης δεδομένων της εφαρμογής μας στην παρούσα πτυχιακή εργασία.

Το Sql Server Management Studio είναι ένα δωρεάν εργαλείο που περιλαμβάνεται στο Microsoft Sql server 2005 και σε μεταγενέστερες εκδόσεις του προϊόντος. Το Sql Server Management χρησιμοποιείται για να μπορείς να τροποποιείς, να διαχειρίζεσαι και γενικότερα για να πραγματοποιήσεις οτιδήποτε ενέργεια χρειαστεί σχετικά με την βάση μας η στα συστατικά αυτής χωρίς να επέμβουμε απευθείας στον Sql Server.

Μία βασική δυνατότητα που παρέχει το Sql Server Management Studio είναι το object Explorer, το οποίο επιτρέπει στον χρήστη να πλοηγηθεί, να επιλέξει και να επιδράσει πάνω σε οποιοδήποτε αντικείμενο μέσα στον Server.



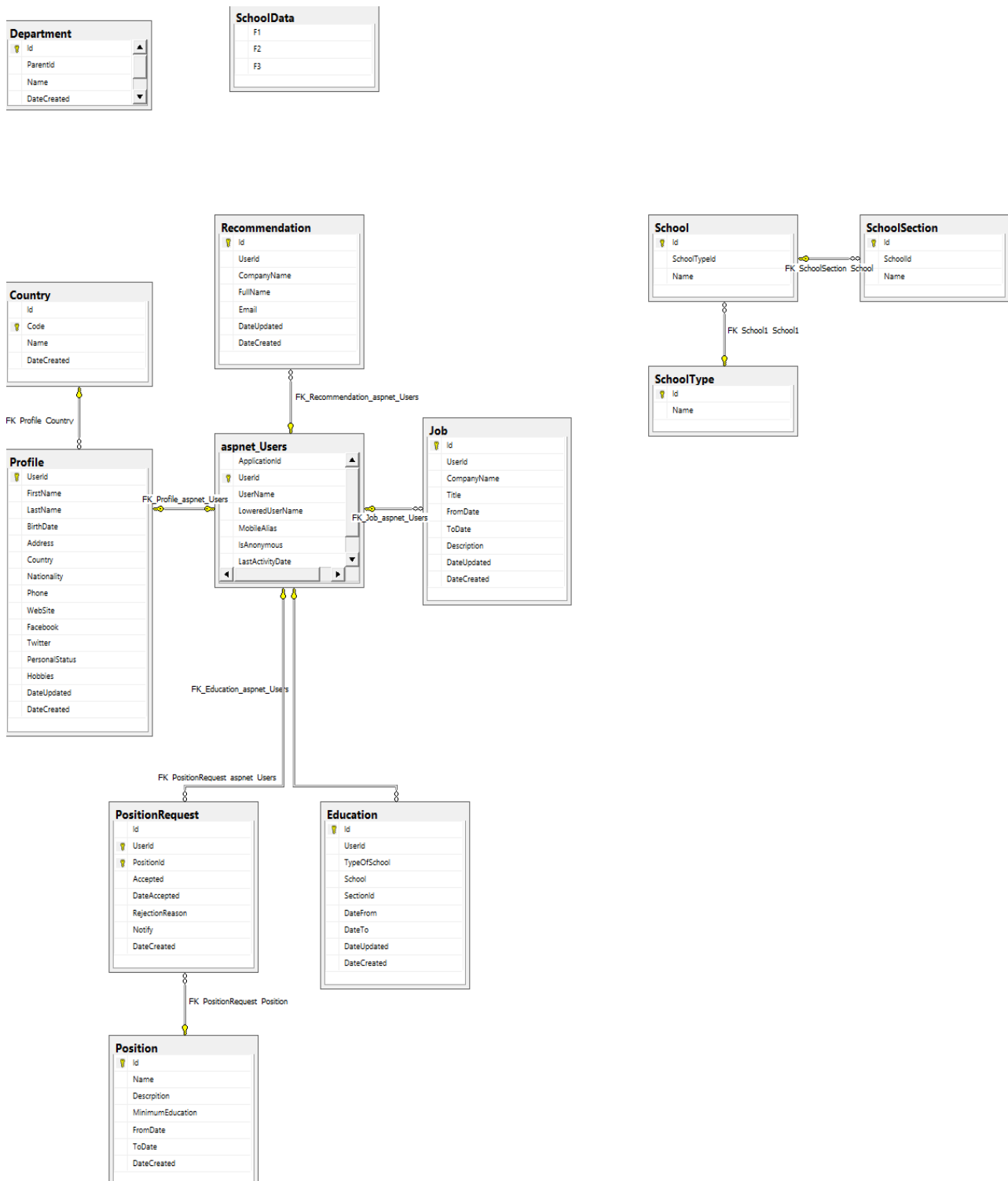
Εικόνα 3: Sql Server Management Studio 2008 παράθυρο αυθεντικοποίησης χρήστη



Εικόνα 4: Σύνδεση με το Management Studio ως Administrator.

### 3.3 Διαγράμματα Βάσης Δεδομένων Πτυχιακής Εργασίας

Παρακάτω θα δούμε με την βοήθεια του Sql Server Management Studio, ένα διάγραμμα της Βάσης Δεδομένων που σχεδιάστηκε, προκειμένου να εξυπηρετεί τις ανάγκες της εφαρμογής μας. Παράλληλα παρουσιάζονται και οι συσχετίσεις μεταξύ των tables της βάσης μας.

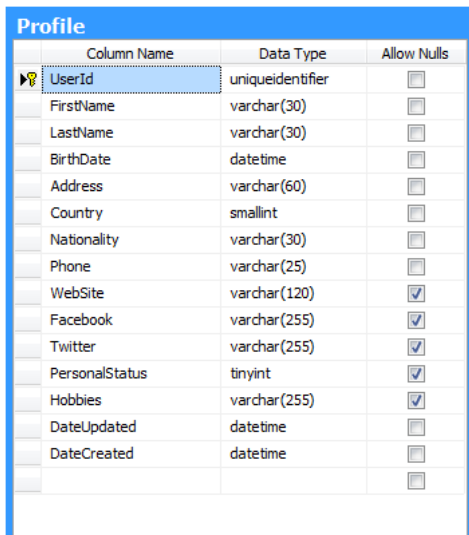


Εικόνα 5: Διάγραμμα Βάσης Δεδομένων

### 3.4 Πίνακες της βάσης δεδομένων Πτυχιακής Εργασίας

Σε αυτό το κεφάλαιο θα προσδιορίσουμε τα χαρακτηριστικά και τις ιδιότητες των tables της βάσης δεδομένων μας καθώς και τις συσχετίσεις τους.

#### 3.4.1 Πίνακας *dbo.prolife*

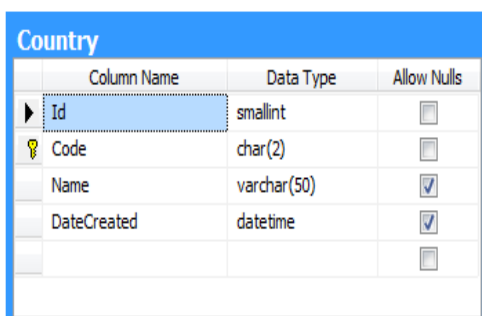


Column Name	Data Type	Allow Nulls
UserId	uniqueidentifier	<input type="checkbox"/>
FirstName	varchar(30)	<input type="checkbox"/>
LastName	varchar(30)	<input type="checkbox"/>
BirthDate	datetime	<input type="checkbox"/>
Address	varchar(60)	<input type="checkbox"/>
Country	smallint	<input type="checkbox"/>
Nationality	varchar(30)	<input type="checkbox"/>
Phone	varchar(25)	<input type="checkbox"/>
WebSite	varchar(120)	<input checked="" type="checkbox"/>
Facebook	varchar(255)	<input checked="" type="checkbox"/>
Twitter	varchar(255)	<input checked="" type="checkbox"/>
PersonalStatus	tinyint	<input checked="" type="checkbox"/>
Hobbies	varchar(255)	<input checked="" type="checkbox"/>
DateUpdated	datetime	<input type="checkbox"/>
DateCreated	datetime	<input type="checkbox"/>

Εικόνα 6: Πίνακας *dbo.prolife*

Στον πίνακα *prolife* αποθηκεύονται τα στοιχεία του χρήστη όπως το ονοματεπώνυμό του, την ημερομηνία γέννησης, την διεύθυνσή του, την πόλη διαμονής, τηλέφωνο, και άλλα προσωπικά στοιχεία. Η εισαγωγή των στοιχείων γίνεται μέσα από την καρτέλα *aspxnet\_Users* όπου ο ίδιος ο χρήστης εισάγει τα συγκεκριμένα δεδομένα. Η τροποποίηση των δεδομένων από τον χρήστη, γίνεται πάλι από την ίδια καρτέλα *aspxnet\_Users*

#### 3.4.2 Πίνακας *dbo.Country*



Column Name	Data Type	Allow Nulls
Id	smallint	<input type="checkbox"/>
Code	char(2)	<input type="checkbox"/>
Name	varchar(50)	<input checked="" type="checkbox"/>
DateCreated	datetime	<input checked="" type="checkbox"/>

Εικόνα 7: Πίνακας *dbo.Country*

Ο πίνακας *country* συνδέεται με τον πίνακα *prolife* καθώς εκεί υπάρχουν καταχωρημένες όλες οι χώρες και ο χρήστης κάνει την επιλογή της Εθνικότητάς του από drop down menu.

#### 3.4.3 Πίνακας *dbo.Job*



Job			
Column Name	Data Type	Allow Nulls	
Id	int		<input type="checkbox"/>
UserId	uniqueidentifier		<input checked="" type="checkbox"/>
CompanyName	varchar(120)		<input type="checkbox"/>
Title	varchar(120)		<input type="checkbox"/>
FromDate	datetime		<input type="checkbox"/>
ToDate	datetime		<input checked="" type="checkbox"/>
Description	varchar(255)		<input checked="" type="checkbox"/>
DateUpdated	datetime		<input checked="" type="checkbox"/>
DateCreated	datetime		<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Εικόνα 8: Πίνακας dbo.Job

Μέσω του συγκεκριμένου πίνακα *Job* καταχωρούνται όλες οι εγγραφές εκπαίδευσης του χρήστη. Η εισαγωγή γίνεται μέσα από την καρτέλα *aspxnet\_Users* όπως και στον πίνακα *Profile*. Ο χρήστης αφού περιηγηθεί στην σελίδα *profile.aspx*, επιλέγει στην εκπαίδευση "Προσθήκη" και μεταφέρεται στην σελίδα *edit-job.aspx* όπου αποθηκεύει τις εγγραφές τις εργασιακής του εμπειρίας.

### 3.4.4 Πίνακας **dbo.Education**

Education			
Column Name	Data Type	Allow Nulls	
Id	int		<input type="checkbox"/>
UserId	uniqueidentifier		<input type="checkbox"/>
School	varchar(255)		<input type="checkbox"/>
DateFrom	datetime		<input type="checkbox"/>
DateTo	datetime		<input type="checkbox"/>
DateUpdated	datetime		<input type="checkbox"/>
DateCreated	datetime		<input type="checkbox"/>
			<input type="checkbox"/>

Εικόνα 9: Πίνακας dbo.Education

Ο πίνακας *Education* χρησιμοποιείται για να αποθηκεύσουμε στην βάση όλες τις εγγραφές που κάνει ο χρήστης στο *profile* του και αφορούν στην εκπαίδευσή του. Ο χρήστης αφού περιηγηθεί στην σελίδα *profile.aspx*, επιλέγει στην Εργασιακή εμπειρία "Προσθήκη" και μεταφέρεται στην σελίδα *edit-school.aspx* όπου αποθηκεύει την σχολή/ύδρημα που έχει φοιτήσει ή ακόμη φοιτεί.

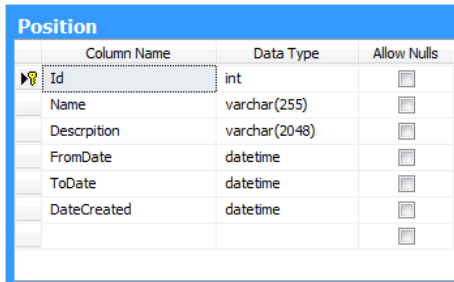
### 3.4.5 Πίνακας **dbo.Recommendation**

Recommendation			
Column Name	Data Type	Allow Nulls	
Id	int		<input type="checkbox"/>
UserId	uniqueidentifier		<input checked="" type="checkbox"/>
CompanyName	varchar(120)		<input type="checkbox"/>
FullName	varchar(120)		<input type="checkbox"/>
Email	varchar(120)		<input type="checkbox"/>
DateUpdated	datetime		<input type="checkbox"/>
DateCreated	datetime		<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Εικόνα 10: Πίνακας dbo.Recommendation

Στον πίνακα αυτόν αποθηκεύουμε στοιχεία σχετικά με τις συστάσεις που προτείνει ο χρήστης να ζητηθούν σχετικά με την αίτησή του. Ο χρήστης αφού περιηγηθεί στην σελίδα *profile.aspx*, επιλέγει στις Συστάσεις "Προσθήκη" και μεταφέρεται στην σελίδα *edit-recommendation.aspx* όπου αποθηκεύει τα στοιχεία συστάσεων.

### 3.4.6 Πίνακας *dbo.position*

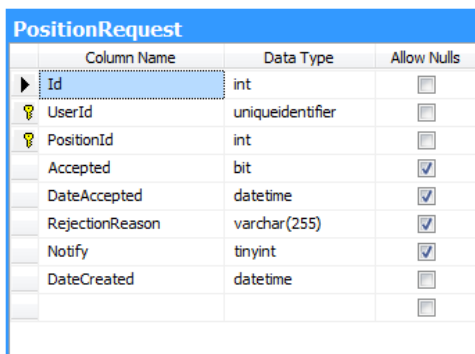


Column Name	Data Type	Allow Nulls
Id	int	<input type="checkbox"/>
Name	varchar(255)	<input type="checkbox"/>
Description	varchar(2048)	<input type="checkbox"/>
FromDate	datetime	<input type="checkbox"/>
ToDate	datetime	<input type="checkbox"/>
DateCreated	datetime	<input type="checkbox"/>

Εικόνα 11: Πίνακας *dbo.position*

Ο πίνακας *position* χρησιμοποιείται για να αποθηκεύσουμε στην βάση τις θέσεις που δημιουργεί ο διαχειριστής της εφαρμογής . Ο διαχειριστής μέσω της σελίδας *modify-position.aspx* επιλέγοντας δημιουργία νέας θέσης καταχωρεί την προσφερόμενη θέση.

### 3.4.7 Πίνακας *dbo.position request*



Column Name	Data Type	Allow Nulls
Id	int	<input type="checkbox"/>
UserId	uniqueidentifier	<input type="checkbox"/>
PositionId	int	<input type="checkbox"/>
Accepted	bit	<input checked="" type="checkbox"/>
DateAccepted	datetime	<input checked="" type="checkbox"/>
RejectionReason	varchar(255)	<input checked="" type="checkbox"/>
Notify	tinyint	<input checked="" type="checkbox"/>
DateCreated	datetime	<input type="checkbox"/>

Εικόνα 12: Πίνακας *bo.position request*

Ο συγκεκριμένος πίνακας *dbo.position request* χρησιμοποιείται για να αποθηκεύονται οι αιτήσεις του χρήστη. Ο χρήστης αφού επιλέξει την διαθέσιμη θέση μεταφέρεται στην σελίδα *view-position.aspx*. Αν δεν έχει αίτηση για την συγκεκριμένη σε εξέλιξη μπορεί να υποβάλει την αίτησή του για την θέση. Ο πίνακας *dbo.position request* συνδέεται τόσο με τον πίνακα *dbo.prolife* μέσω του πεδίου κλειδιού *UsedId* για να γνωρίζουμε ποιόν χρήση αφορά η αίτηση αλλά και με τον πίνακα *position* μέσω του πεδίου κλειδιού *PositionId* προκειμένου να γνωρίζουμε κατά την αποθήκευση σε ποια θέση αφορά η αίτηση του συγκεκριμένου χρήστη-*profile*.

# 4. Επικυρώσεις και ασφάλεια σε ASP.NET

## 4.1 Επικυρώσεις για τις Web φόρμες

Κάθε πρόγραμμα μπορεί να αντιμετωπίσει διάφορα προβλήματα λόγω λανθασμένων δεδομένων, όπως αριθμούς αντί χαρακτήρες, κενό αντί μιας συμβολοσειράς κοκ. Για να αποφύγουμε αποθήκευση τυχόν λανθασμένων δεδομένων, χρησιμοποιούμε τα στοιχεία ελέγχου επικύρωσης που μας προσφέρει η ASP.NET. Τα στοιχεία ελέγχου επικύρωσης (ή validator) καθορίζουν αν τα δεδομένα σε ένα στοιχείο ελέγχου είναι στην κατάλληλη μορφή. Στην ASP.NET μπορούμε να προσθέσουμε στοιχεία ελέγχου σε όλα σχεδόν τα στοιχεία δεδομένων εισόδου της σελίδας μας. Τα στοιχεία ελέγχου επικύρωσης παρέχουν επίσης ένα μηχανισμό για την επικύρωση της εισόδου χρήστη στην εφαρμογή. Όταν δημιουργηθεί η HTML για τη σελίδα μας, το στοιχείο επικύρωσης μετατρέπεται σε ECMAScript που εκτελεί την επικύρωση. ECMAScript είναι μια γλώσσα scripting που βελτιώνει τη λειτουργικότητα και την εμφάνιση των σελίδων Web.

Για παράδειγμα, στην παρακάτω φόρμα *account.aspx* έχουμε εφαρμόσει έλεγχο επικύρωσης στο πεδίο του κωδικού. Αν δώσουμε μόνο τρία ψηφία, η επικύρωση θα αποτύχει και η σελίδα μας επιστρέφει ένα μήνυμα σφάλματος και μας ενημερώνει ότι τα στοιχεία που δώσαμε δεν είναι στην αποδεκτή μορφή που αναμένει βάση ορισμού από τον προγραμματιστή.

**ΠΥΛΗ ΗΛΕΚΤΡΟΝΙΚΩΝ ΑΙΤΗΣΕΩΝ** [ σύνδεση ]

Αρχική

**ΝΕΟΣ ΧΡΗΣΤΗΣ ;**

Συμπληρώστε τη φόρμα που ακολουθεί για να δημιουργήσετε νέο λογαριασμό.

Ο κωδικός πρόσβασης πρέπει να είναι τουλάχιστον 6 χαρακτήρες.

**Ο κωδικός πρόσβασης δεν είναι αποδεκτός. Πρέπει να είναι τουλάχιστον 6 χαρακτήρες**

Account Information

E-mail:  
user4@teiep.gr

Κωδικός πρόσβασης:  
...

Επιβεβαίωση κωδικού πρόσβασης:  
... \*

εγγραφή

**ΣΥΝΔΕΣΗ**

Παρακαλώ εισάγετε το e-mail και τον κωδικό πρόσβασης σας.

Στοιχεία Λογαριασμού

E-mail:  
[input field]

Κωδικός πρόσβασης:  
[input field]

Να με θυμάσαι

σύνδεση

Εικόνα 13: Έλεγχος κωδικού ορθότητας κωδικού πρόσβασης

Η ASP.NET έχει τα εξής είδη στοιχείων ελέγχου επικύρωσης: το *RequiredFieldValidator*, το *RegularExpressionValidator*, το *CompareValidator*, το *RangeValidator*, το *CustomValidator* και το *ValidationSummary*. Όλα αυτά τα στοιχεία έχουν μερικές κοινές ιδιότητες και μεθόδους. Κληρονομούνται από την κλάση *BaseValidator*, από την κλάση *WebControl* του χώρου

*System.UI.WebControls* και από την κλάση *Controls* που βρίσκεται στο *System.Web.UI*. Μοναδική εξαίρεση αποτελεί η ιδιότητα *Text*, η οποία κληρονομείται από την κλάση *Label*.

## 4.2 RequiredFieldValidator

Το στοιχείο ελέγχου *RequiredFieldValidator* είναι το στοιχείο το οποίο ελέγχει αν ο χρήστης έχει δώσει μια τιμή δεδομένων στο στοιχείο ελέγχου εισόδου. Η ιδιότητα *ControlToValidate* παίρνει τιμή το ID του στοιχείου εισόδου που θέλουμε. Μόλις γίνει η αντιστοίχιση αυτή διασφαλίζεται ότι στο συγκεκριμένο πεδίο εισόδου θα εισαχτεί μια τιμή και δε θα παραμείνει κενό.

Το *RequiredFieldValidator* λειτουργεί με την ιδιότητα *InitialValue*. Το *InitialValue* είναι η αρχική τιμή του στοιχείου ελέγχου επικύρωσης. Αν η τιμή του *Initialvalue* είναι ίση με την τιμή του πεδίου εισόδου τότε η επικύρωση αποτυγχάνει και μας εμφανίζει στη σελίδα το κείμενο που υπάρχει στην ιδιότητα του *RequiredFieldValidator*, *ErrorMessage*.

The screenshot shows a web application interface with a dark blue header. The header contains the text "ΠΥΛΗ ΗΛΕΚΤΡΟΝΙΚΩΝ ΑΙΤΗΣΕΩΝ" on the left and "[ σύνδεση ]" on the right. Below the header is a dark blue bar with the text "Αρχική".

The main content area is divided into two columns. The left column is titled "ΝΕΟΣ ΧΡΗΣΤΗΣ ;" and contains the text "Συμπληρώστε τη φόρμα που ακολουθεί για να δημιουργήσετε νέο λογαριασμό." Below this is a yellow box with a red bullet point and the text "το e-mail είναι υποχρεωτικό". Underneath is a form titled "Account Information" with three input fields: "E-mail:" (with a red asterisk below it), "Κωδικός πρόσβασης:", and "Επιβεβαίωση κωδικού πρόσβασης:". A "εγγραφή" button is at the bottom of the form.

The right column is titled "ΣΥΝΔΕΣΗ" and contains the text "Παρακαλώ εισάγετε το e-mail και τον κωδικό πρόσβασης σας." Below this is a form titled "Στοιχεία Λογαριασμού" with two input fields: "E-mail:" and "Κωδικός πρόσβασης:". There is a checkbox labeled "Να με θυμάσαι" and a "σύνδεση" button at the bottom of the form.

Εικόνα 14: Έλεγχος ύπαρξης στοιχείου στο πεδίο e-mail

Αρχική

ΝΕΟΣ ΧΡΗΣΤΗΣ ;

Συμπληρώστε τη φόρμα που ακολουθεί για να δημιουργήσετε νέο λογαριασμό.

Ο κωδικός πρόσβασης πρέπει να είναι τουλάχιστον 6 χαρακτήρες.

- το e-mail είναι υποχρεωτικό
- ο κωδικός πρόσβασης είναι υποχρεωτικός.
- ο κωδικός επιβεβαίωσης είναι υποχρεωτικός

Account Information

E-mail:

\*  
 Κωδικός πρόσβασης:

\*  
 Επιβεβαίωση κωδικού πρόσβασης:

\*

ΣΥΝΔΕΣΗ

Παρακαλώ εισάγετε το e-mail και τον κωδικό πρόσβασης σας.

Στοιχεία Λογαριασμού

E-mail:

Κωδικός πρόσβασης:

Να με θυμάσαι

Εικόνα 15: Έλεγχος ύπαρξης στοιχείων εγγραφής

Αρχική

ΝΕΟΣ ΧΡΗΣΤΗΣ ;

Συμπληρώστε τη φόρμα που ακολουθεί για να δημιουργήσετε νέο λογαριασμό.

Ο κωδικός πρόσβασης πρέπει να είναι τουλάχιστον 6 χαρακτήρες.

- ο κωδικός πρόσβασης είναι υποχρεωτικός.
- ο κωδικός επιβεβαίωσης είναι υποχρεωτικός

Account Information

E-mail:

Κωδικός πρόσβασης:

\*  
 Επιβεβαίωση κωδικού πρόσβασης:

\*

ΣΥΝΔΕΣΗ

Παρακαλώ εισάγετε το e-mail και τον κωδικό πρόσβασης σας.

Στοιχεία Λογαριασμού

E-mail:

Κωδικός πρόσβασης:

Να με θυμάσαι

Εικόνα 16: Έλεγχος ύπαρξης στοιχείου κωδικού πρόσβασης

Σε περίπτωση που ο χρήστης δεν εισάγει κάποια από τα απαραίτητα δεδομένα στη σελίδα, μόλις επιχειρήσει σύνδεση εμφανίζονται τα μηνύματα των παραπάνω παραδειγμάτων.

Στο παραπάνω κομμάτι κώδικα έχουμε δημιουργήσει ένα RequiredFieldValidator με όνομα



ΝΕΟΣ ΧΡΗΣΤΗΣ ;

Συμπληρώστε τη φόρμα που ακολουθεί για να δημιουργήσετε νέο λογαριασμό.

Ο κωδικός πρόσβασης πρέπει να είναι τουλάχιστον 6 χαρακτήρες.

- συμπληρώστε σωστά το e-mail

Account Information

E-mail:  
abcd@mail \*

Κωδικός πρόσβασης:  
.....

Επιβεβαίωση κωδικού πρόσβασης:  
.....

εγγραφή

ΣΥΝΔΕΣΗ

Παρακαλώ εισάγετε το e-mail και τον κωδικό πρόσβασης σας.

Στοιχεία Λογαριασμού

E-mail:  
.....

Κωδικός πρόσβασης:  
.....

Να με θυμάσαι

σύνδεση

Εικόνα 17: Έλεγχος ορθότητας e-mail

Αυτό ελέγχει το πεδίο κειμένου UserName. Για το αν η συμβολοσειρά που περιέχει είναι της μορφής που ορίζουμε Αυτό σημαίνει ότι η συμβολοσειρά πρέπει να περιέχει στην αρχή της αλφαριθμητικά μέχρι να συναντήσει το χαρακτήρα @. Μετά περιμένει να βρει στη συνέχεια της συμβολοσειράς χαρακτήρες μόνο μικρούς και αριθμούς μέχρι να συναντήσει την τελεία. Τέλος, πρέπει να βρει στο τέλος της συμβολοσειράς δυο ή τρεις χαρακτήρες. Αν η συμβολοσειρά που υπάρχει στο UserName δεν έχει την κατάλληλη οριζόμενη μορφή τότε μας επιστρέφει το μήνυμα «συμπληρώστε σωστά το e-mail».

#### 4.4 CompareValidator

Μέσω της συγκεκριμένης λειτουργίας ελέγχετε ότι το πεδία Κωδικός πρόσβασης και Επιβεβαίωση κωδικού πρόσβασης είναι ίδια .Ο συγκεκριμένος έλεγχος είναι σημαντικός για να διασφαλιστεί ότι ο χρήστης έχει πληκτρολογήσει σωστά τον κωδικό που επιθυμεί να έχει για την εισόδου του στο σύστημα. Αυτό επιτυγχάνετε μέσω της διπλή (επιβεβαιωτική) εισαγωγής και του έλεγχου.

Στο παραπάνω κομμάτι κώδικα έχουμε δημιουργήσει ένα RegularExpressionValidator με όνομα PasswordCompare.

```
<asp:CompareValidator ID="PasswordCompare" runat="server" ControlToCompare="Password"
ControlToValidate="ConfirmPassword"
CssClass="failureNotification" Display="Dynamic" ErrorMessage="ο κωδικός πρόσβασης
πρέπει να συμφωνεί με τον κωδικό επιβεβαίωσης."
ValidationGroup="RegisterUserValidationGroup">*</asp:CompareValidator>
```

Ο κώδικας συγκρίνει τις τιμές των πεδίων Password & ConfirmPassword . Στην περίπτωση που οι δύο τιμές δεν είναι απόλυτα όμοιες επιστρέφει το Μήνυμα «ο κωδικός πρόσβασης πρέπει να συμφωνεί με τον κωδικό επιβεβαίωσης».

Αρχική

ΝΕΟΣ ΧΡΗΣΤΗΣ ;

Συμπληρώστε τη φόρμα που ακολουθεί για να δημιουργήσετε νέο λογαριασμό.

Ο κωδικός πρόσβασης πρέπει να είναι τουλάχιστον 6 χαρακτήρες.

- ο κωδικός πρόσβασης πρέπει να συμφωνεί με τον κωδικό επιβεβαίωσης.

Account Information

E-mail:  
user5@teiep.gr

Κωδικός πρόσβασης:  
.....

Επιβεβαίωση κωδικού πρόσβασης:  
.....  
\*

εγγραφή

ΣΥΝΔΕΣΗ

Παρακαλώ εισάγετε το e-mail και τον κωδικό πρόσβασης σας.

Στοιχεία Λογαριασμού

E-mail:

Κωδικός πρόσβασης:

Να με θυμάσαι

σύνδεση

Εικόνα 18: Έλεγχος επιβεβαίωσης κωδικού

Στον κώδικα μας έχει λοιπόν δημιουργηθεί ένα `RegularExpressionValidator` για το validation του email ,ένα `RegularExpressionValidator` για τον έλεγχο των χαρακτηριστικών των εισαγομένων κωδικών πρόσβασης καθώς και ένα `CompareValidator` για τον έλεγχο της ορθότητας των εισαγομένων κωδικών πρόσβασης.

## 4.5 CSS (Cascading Style Sheet)

Τα Διαδοχικά Φύλλα Στυλ (CSS, Cascading Style Sheets) αποτελούν το εργαλείο με τα οποία μπορούμε να αλλάζουμε την εμφάνιση και τη διάταξη των ιστοσελίδων μας. Μπορούν να μας εξοικονομήσουν πολύ χρόνο και να μας γλυτώσουν σημαντικό κόπο καθώς μας δίνουν τη δυνατότητα να σχεδιάζουμε τις ιστοσελίδες μας με μια εντελώς καινούργια φιλοσοφία.

*Τι μπορούμε να κάνουμε με τα CSS;*

Θα πρέπει να γνωρίζουμε ότι τα CSS είναι μια γλώσσα στυλ (style language) που ορίζουν τη διάταξη (layout) των HTML εγγράφων. Για παράδειγμα, τα CSS έχουν να κάνουν με γραμματοσειρές (fonts), με χρώματα (colours), με περιθώρια (margins), με εικόνες φόντου (background images) και με πολλά άλλα. Με την HTML θα δυσκολευτούμε να αλλάξουμε τη διάταξη των ιστοσελίδων μας, αλλά τα CSS προσφέρουν πολλές επιλογές και είναι πολύ πιο συγκεκριμένα στις λεπτομέρειες. Επιπλέον, υποστηρίζονται απ' όλους τους φυλλομετρητές.



*Ποια η διαφορά ανάμεσα στα CSS και την HTML;*

Μπορούμε να πούμε ότι η HTML χρησιμοποιείται για να δομήσει το περιεχόμενο (content), ενώ τα CSS χρησιμοποιούνται για τη διαμόρφωση ή μορφοποίηση (formatting) του δομημένου περιεχομένου. Από το ξεκίνημά της, η HTML χρησιμοποιείτο μόνο για να μπορούμε να προσθέτουμε δομή στο κείμενο, πχ. για να επισημαίνουμε ποια είναι η κεφαλίδα (headline) ή ποια είναι παράγραφος (paragraph) με τα γνωστά tags (ετικέτες) της HTML. Όμως, καθώς η δημοτικότητα του Web αύξανε συνεχώς, οι σχεδιαστές των ιστοσελίδων άρχισαν να ψάχνουν για επιπλέον δυνατότητες, όπως προσθήκη διάταξης (layout) στα έγγραφα. Για να γίνει αυτό, οι φυλλομετρητές επινόησαν καινούργια HTML tags, όπως για παράδειγμα το <font>, τα οποία διέφεραν από τα αρχικά HTML tags καθώς όριζαν τη διάταξη και όχι τη δομή μιας ιστοσελίδας.

*Τι έχουν να μου προσφέρουν τα CSS;*

Εδώ μπορούμε να πούμε ότι τα CSS αποτέλεσαν μια από τις επαναστάσεις στον κόσμο του Web Design και τα σημαντικότερα πλεονεκτήματά τους είναι τα εξής :

- Δυνατότητα για τον ταυτόχρονο έλεγχο της διάταξης (layout) πολλών εγγράφων από ένα μόνο φύλλο στυλ (style sheet).
- Πιο ακριβής έλεγχος της διάταξης (layout) των ιστοσελίδων.
- Εφαρμογή διαφορετικής διάταξης σε διαφορετικές εξόδους, όπως screen, print κ.ά.
- Αναριθμητές προηγμένες και εξεζητημένες τεχνικές.

## **4.6 Σύνταξη/ Εφαρμογή CSS σε HTML**

Η CSS έχει απλή σύνταξη και χρησιμοποιεί μερικές αγγλικές λέξεις-κλειδιά, για να καθορίσει τα ονόματα διαφόρων ιδιοτήτων στυλ. Ένα style sheet απαρτίζεται από μια λίστα με κανόνες. Κάθε κανόνας περιέχει ένα ή περισσότερους selectors (επιλογείς) και ένα μπλοκ δήλωσης. Κάθε μπλοκ έχει μια σειρά από δηλώσεις τοποθετημένες μέσα σε αγκύλες ({...}). Κάθε δήλωση έχει τη μορφή ιδιότητα-άνω και κάτω τελεία (:)-τιμή. Αν υπάρχουν πάνω από μια δήλωση στο μπλοκ, ξεχωρίζουν με το ελληνικό ερωτηματικό (;). Στη CSS οι επιλογείς χρησιμοποιούνται για να δηλώσουν σε ποιο στοιχείο markup θα εφαρμοστούν.

Ένα απλό παράδειγμα CSS είναι το εξής:

```
body
{
background-color: #ffffff;
background-image: url('../Images/bodybg.png');
background-repeat: no-repeat;
}
```

Ονομάζουμε τον επιλογέα "body" και στη συνέχεια δίνουμε την ιδιότητα το χρώμα στο υπόβαθρο να είναι άσπρο, να πάρει μια εικόνα για υπόβαθρο από αρχείο και να μην επαναλαμβάνεται αυτή η εικόνα. Η CSS γράφεται σε ξεχωριστό αρχείο και ενώνεται με την HTML με σύνδεσμο. Για παράδειγμα, αν έχουμε το αρχείο Style.css, στο head του HTML εγγράφου μας θα το καλέσουμε κάπως έτσι:

```
<link href="Styles/Style.css" rel="Stylesheet" type="text/css" />
```

Εφ' όσον έχουμε συνδέσει CSS και HTML μπορούμε να δώσουμε σε ένα στοιχείο HTML τις ιδιότητες του "body", δηλαδή <div class="body" />

Παρακάτω παρουσιάζεται η CSS σελίδα που έχουμε δημιουργήσει για την εφαρμογή μας

## **4.7 Ασφάλεια ASP.NET και IIS**

Η ASP.NET προσφέρει έλεγχο και ασφάλεια στην εφαρμογή μας. Η ασφάλεια του ASP λειτουργεί σε συνδυασμό με τις υπηρεσίες Microsoft Internet Information Services (IIS) και περιλαμβάνει υπηρεσίες έλεγχου ταυτότητας και εξουσιοδότησης. Ο IIS είναι ένας ευέλικτος, ασφαλής και εύκολα στη χρήση εξυπηρετητής ιστού (Web Server) που αναπτύχθηκε από τη Microsoft για τη φιλοξενία διαδικτυακών εφαρμογών. Παρόλο που ο IIS παρέχει κυρίως φιλοξενία σε εφαρμογές που έχουν αναπτυχθεί με .NET τεχνολογία, μπορεί να χρησιμοποιηθεί και από εφαρμογές που έχουν αναπτυχθεί σε PHP, JSP ή Perl. Τα πρωτόκολλα που υποστηρίζονται από τον IIS7 είναι τα εξής: FTP, FTPS, SMTP, NNTP, και HTTP/HTTPS.

Το μοντέλο ασφάλειας του ASP.NET περιλαμβάνει επίσης μια βασισμένη σε ρόλους δυνατότητα ασφάλειας, που μπορούν να εφαρμοστούν σε συνδυασμό με το Microsoft Windows και τον λογαριασμό χρήστη των Windows.

## **4.8 Ροή ασφαλείας σε μια αίτηση**

Παρακάτω περιγράφεται η ακολουθία συμβάντων, όταν ένα πρόγραμμα-πελάτης υποβάλει μια αίτηση προς τον ISS:

Ένας υπολογιστής-πελάτης ζητά μια σελίδα .aspx που βρίσκεται σε διακομιστή IIS.  
Τα διαπιστευτήρια του πελάτη διαβιβάζονται στις υπηρεσίες IIS.

Ο IIS ελέγχει την ταυτότητα του προγράμματος-πελάτη και προωθεί το διακριτικό με έλεγχο ταυτότητας μαζί με την αίτηση του προγράμματος-πελάτη για τη διαδικασία εργασίας ASP.NET. Βάσει του ελέγχου ταυτότητας διακριτικού από τις υπηρεσίες του IIS και τις ρυθμίσεις παραμέτρων τις Web εφαρμογής ASP, ο ISS αποφασίζει αν επιτρέψει ένα χρήστη και αν θα προχωρήσει η επεξεργασία της αίτησης.

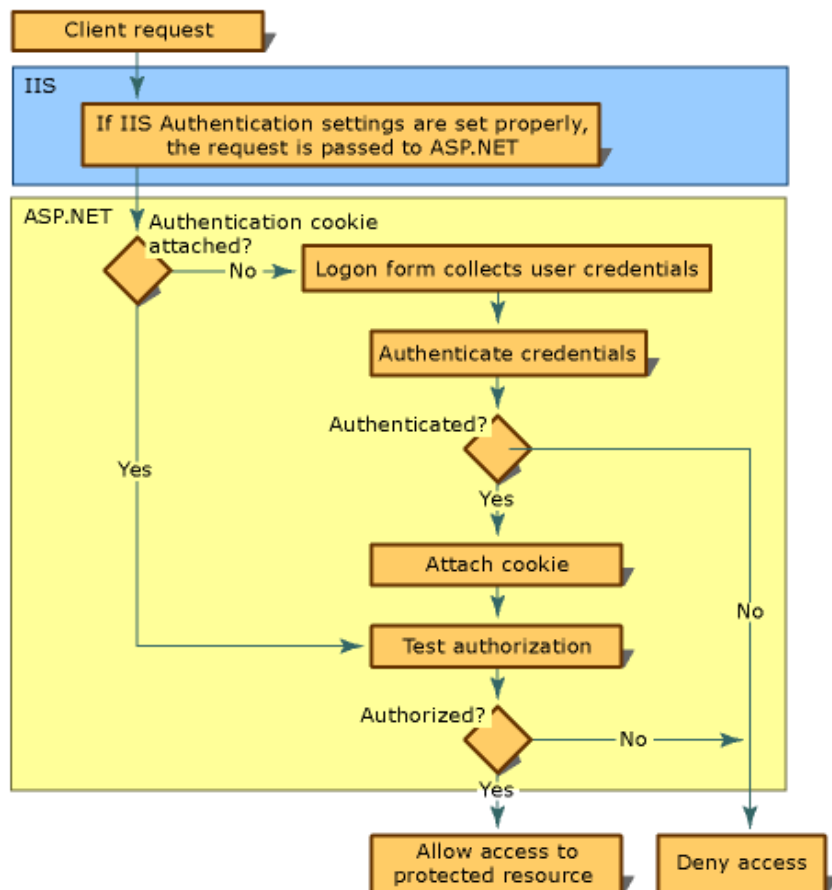
Ο IIS διατηρεί τις ρυθμίσεις παραμέτρων που σχετίζονται με την ασφάλεια στο IIS μετάβασης. Ωστόσο, η ASP.NET διατηρεί το δίκτυο ασφαλείας στα αρχεία παραμέτρων Extensible Markup Language (XML). Παρόλο που αυτό γενικά απλοποιεί την ανάπτυξη της εφαρμογής από μια άποψη ασφαλείας, το μοντέλο ασφαλείας που θεσπίζει η εφαρμογή απαιτεί την σωστή ρύθμιση παραμέτρων μεταξύ IIS και της ASP.NET εφαρμογής μέσω του αρχείου ρύθμισης παραμέτρων (Web.config).

## **4.9 Έλεγχος ταυτότητας**

Είναι η διαδικασία μέσω της οποίας μπορούμε να αποκτήσουμε αναγνώριση πιστοποίησης, όπως το όνομα χρήστη και κωδικό πρόσβασης καθώς και να επικυρώσουμε τις πιστοποιήσεις μας. Η ASP.NET παρέχει τέσσερις υπηρεσίες ελέγχους ταυτότητας που είναι οι παρακάτω:

### ❖ Φόρμες έλεγχος ταυτότητας

Έλεγχος ταυτότητας φορμών αναφέρεται σε ένα σύστημα στο οποίο χωρίς έλεγχο ταυτότητας οι αιτήσεις ανακατευθύνονται σε μια φόρμα Hypertext Markup Language (HTML) που οι χρήστες πληκτρολογούν τις πιστοποιήσεις τους. Αφού ο χρήστης παρέχει τα διαπιστευτήρια του και υποβάλλει στη φόρμα-εφαρμογή, πιστοποιεί την αίτηση και ζητήματα συστήματος ένα δελτίο εξουσιοδότησης με τη μορφή ενός cookie. Με αυτό το cookie που περιέχεται οι επόμενες αιτήσεις περιλαμβάνουν αυτόματα το πρόγραμμα περιήγησης.



Εικόνα 19: Διάγραμμα Ροής με Φόρμες έλεγχου ταυτότητας  
[http://msdn.microsoft.com/en-us/library/xa68twcb\(v=vs.71\).aspx](http://msdn.microsoft.com/en-us/library/xa68twcb(v=vs.71).aspx)

#### ❖ Windows έλεγχος ταυτότητας

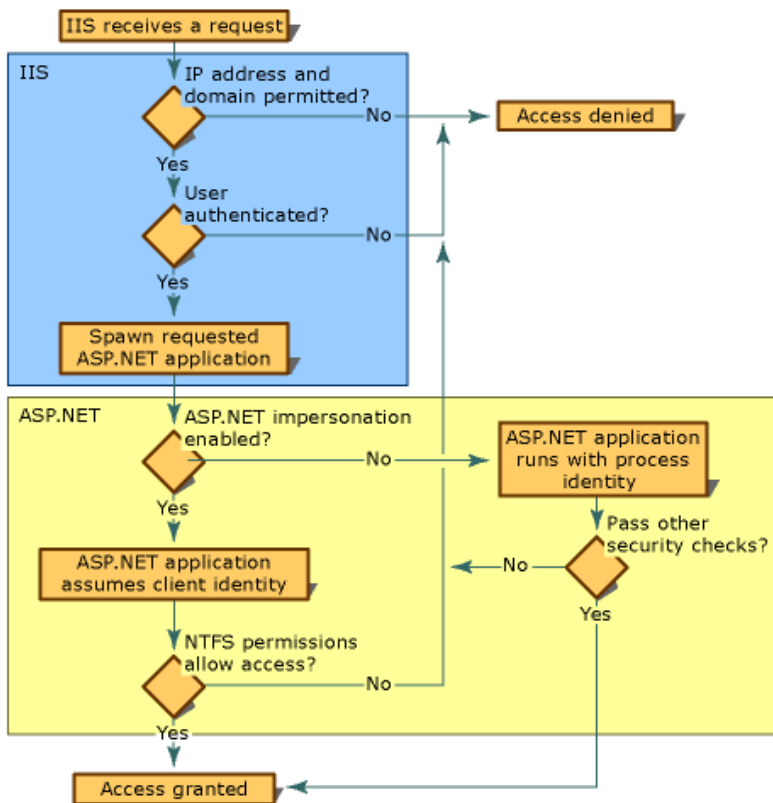
Με έλεγχο ταυτότητας των Windows, το IIS εκτελεί έλεγχο ταυτότητας, και με έλεγχο ταυτότητας διακριτικού προωθείται η ASP. Η διαδικασία εργασίας δικτύου. Το πλεονέκτημα της χρήσης του έλεγχος ταυτότητας Windows είναι ότι απαιτεί ελάχιστη κωδικοποίηση. Αν θέλουμε να χρησιμοποιήσουμε αυτόν τον έλεγχο ταυτότητας, τα Windows απομιμούνται το λογαριασμό χρήστη των Windows που ελέγχει την ταυτότητα των υπηρεσιών IIS πριν παραδώσετε την αίτηση για ASP.NET.

#### ❖ Passport έλεγχος ταυτότητας

Ο έλεγχος ταυτότητας Passport είναι μια υπηρεσία κεντρικού ελέγχου ταυτότητας που η Microsoft παρέχει και προσφέρει ένα μεμονωμένο αρχείο καταγραφής σε βασικών υπηρεσιών προφίλ για τοποθεσίες του μέλους. Συνήθως χρησιμοποιείται έλεγχος ταυτότητας Passport όταν χρειάζεται μόνο αρχείο καταγραφής με δυνατότητα σε πολλούς τομείς.

#### ❖ Προεπιλογή έλεγχος ταυτότητας

Ο προεπιλεγμένος έλεγχος ταυτότητας χρησιμοποιείται όταν δεν θέλουμε ασφάλεια στην Web εφαρμογή. Ανάμεσα σε όλες τις υπηρεσίες παροχής ελέγχου ταυτότητας παρέχει προεπιλεγμένο έλεγχο ταυτότητας με μέγιστη απόδοση για την εφαρμογή μας. Την υπηρεσία παροχής ελέγχου ταυτότητας την χρησιμοποιούμε όταν διαθέτουμε δικό σας λειτουργικό μονάδας ασφαλείας.



Εικόνα 20: Διάγραμμα Ροής με προεπιλογή έλεγχος ταυτότητας  
[http://msdn.microsoft.com/en-us/library/xa68twcb\(v=vs.71\).aspx](http://msdn.microsoft.com/en-us/library/xa68twcb(v=vs.71).aspx)

#### 4.10 **web.config**

Το web.config είναι .xml αρχείο το οποίο περιέχει τις απαραίτητες ρυθμίσεις της εφαρμογής μας όπως:

- connectionStrings (με αυτά συνδεόμαστε σε βάσεις δεδομένων)
- authentication (πως κάνεις authenticate στην εφαρμογή)
- membership, profile, roleManager (είναι τεχνολογίες της Microsoft που μας δίνουν τη δυνατότητα να έχουμε χρήστες και ρόλους στην εφαρμογή μας)

Παρακάτω βλέπουμε αναλυτικά την web.config σελίδα μας η οποία περιλαμβάνει όλες τις βασικές ρυθμίσεις για την εφαρμογή μας όπως το connection string και ρυθμίσεις που χρειάζονται για να κάνουμε σύνδεση με την βάση δεδομένων μας όπως το source, τον κατάλογο και διάφορες ρυθμίσεις ασφάλειας.

```

<configuration>
  <connectionStrings>
    <add name="GraduateConnectionString" connectionString="Data Source=(localdb)\v11.0;Initial
Catalog=GraduateWork; User Id=graduateuser;password=graduateuser123"
    providerName="System.Data.SqlClient" />
    <add name="GraduateWorkConnectionString" connectionString="Data Source=(localdb)\v11.0;Initial
Catalog=GraduateWork;Integrated Security=True"
    providerName="System.Data.SqlClient" />
  
```

```

    <add name="GraduateWorkConnectionString1" connectionString="Data Source=(localdb)\v11.0;Initial
Catalog=GraduateWork;User ID=graduateuser;Password=graduateuser123"
    providerName="System.Data.SqlClient" />
</connectionStrings>
<system.web>
    <compilation debug="true" strict="false" explicit="true" targetFramework="4.0"
optimizeCompilations="true">
    <assemblies>
        <add assembly="System.Data.Linq, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089"/>
        <add assembly="mscorlib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089"/>
    </assemblies>
    </compilation>
    <authentication mode="Forms">
        <forms loginUrl="~/Account/account.aspx" timeout="2880" defaultUrl="~/index.aspx"/>
    </authentication>
    <membership>
        <providers>
            <clear/>
            <add name="AspNetSqlMembershipProvider" type="System.Web.Security.SqlMembershipProvider"
connectionStringName="GraduateConnectionString" enablePasswordRetrieval="true"
enablePasswordReset="true" requiresQuestionAndAnswer="false" passwordFormat="Encrypted"
requiresUniqueEmail="false" maxInvalidPasswordAttempts="5" minRequiredPasswordLength="6"
minRequiredNonalphanumericCharacters="0" passwordAttemptWindow="10" applicationName="/"/>
        </providers>
    </membership>
    <profile>
        <providers>
            <clear/>
            <add name="AspNetSqlProfileProvider" type="System.Web.Profile.SqlProfileProvider"
connectionStringName="GraduateConnectionString" applicationName="/"/>
        </providers>
    </profile>
    <roleManager enabled="true">
        <providers>
            <clear/>
            <add name="AspNetSqlRoleProvider" type="System.Web.Security.SqlRoleProvider"
connectionStringName="GraduateConnectionString" applicationName="/"/>
            <add name="AspNetWindowsTokenRoleProvider" type="System.Web.Security.WindowsTokenRoleProvider"
applicationName="/"/>
        </providers>
    </roleManager>
    <globalization culture="el-gr" uiCulture="el-gr"/>
    <machineKey
validationKey="EE4E855D6559B27ABBB3B666D12BBF8A7FF93FE78AAF7FFB45C50B2995099F97013F95F6D4C0FA560D93063
78DE44C3B0EE8A1B0738B3949EF53E5232C60457D"
decryptionKey="07925EC85781C53ED0DA0705D2C7D0D26AB937C7DF5CB2E6EA70430A4725243F" validation="SHA1"
decryption="AES"/>
</system.web>
<system.webServer>
    <modules runAllManagedModulesForAllRequests="true"/>
    <defaultDocument>
        <files>
            <add value="index.aspx"/>
        </files>
    </defaultDocument>
</system.webServer>
</configuration>

```

## 4.11 Cookies

Τα cookies είναι μικρά αρχεία κειμένου τα οποία αποθηκεύονται στον υπολογιστή μας κατά την πλοήγησή μας στο διαδίκτυο. Συνήθως περιγράφουν τα στοιχεία μας όπως όνομα χρήστη (user name) και συνθηματικό πρόσβασης (password) με σκοπό κατά την επίσκεψή μας στον ίδιο ιστότοπο αργότερα, ο τελευταίος να μας "θυμάται" και να κάνει login χωρίς να χρηστεί να εισάγουμε εμείς καθόλου στοιχεία. Τα cookies μπορεί να προέρχονται από τον ιστότοπο τον οποίο έχουμε επισκεφθεί ή από κάποιον άλλον. Συνήθως είναι άκακα, έχει όμως αποδειχθεί ότι μπορούν να στείλουν πληροφορίες για τη συμπεριφορά μας στο διαδίκτυο. Υπάρχουν προγράμματα που καθαρίζουν τα κακόβουλα cookies, ενώ αν ο χρήστης επιθυμεί να τα διαγράψει δίνεται αυτή η δυνατότητα μέσα από το φυλλομετρητή ιστοσελίδων.

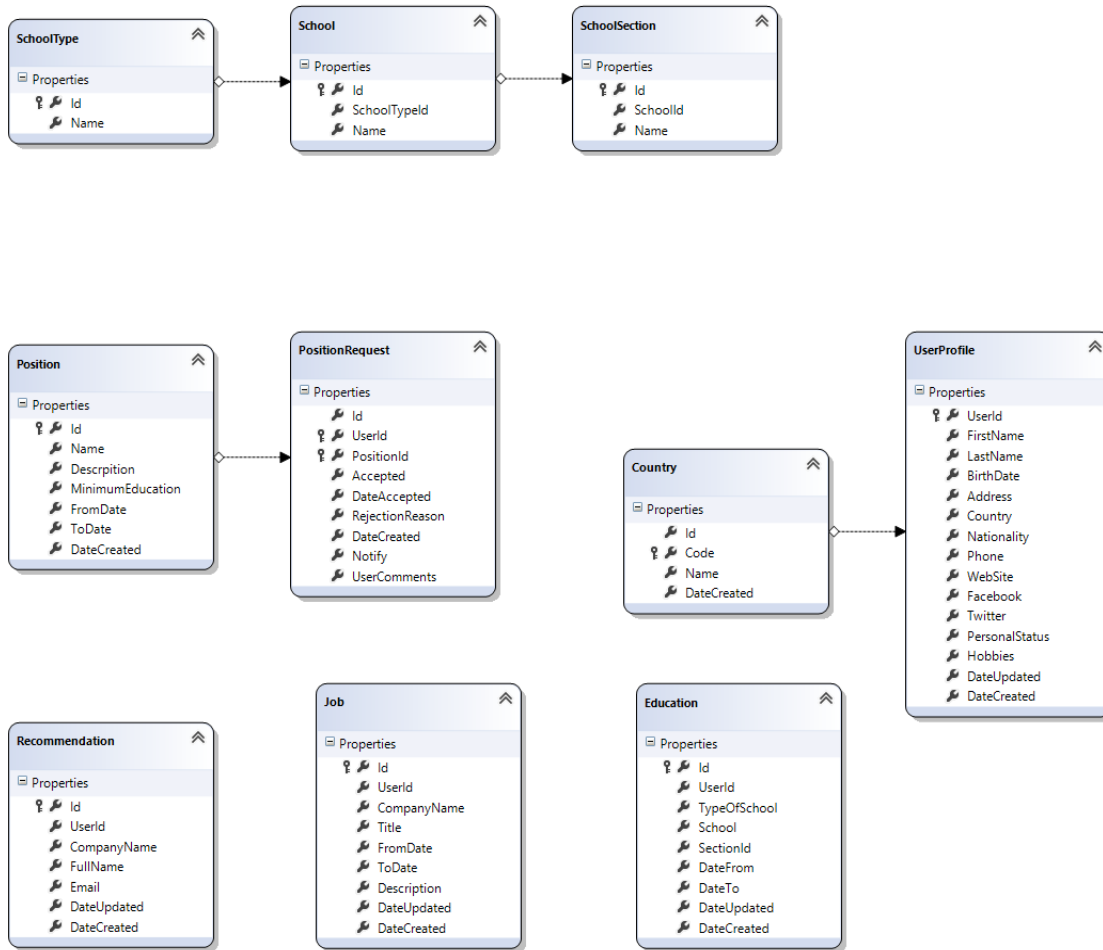
## 4.12 Σύνδεση με βάση δεδομένων

Για να πραγματοποιήσουμε τη σύνδεση με την βάση δεδομένων χρησιμοποιούμε την τεχνολογία LINQ του SQL. Η συγκεκριμένη τεχνολογία μας δίνει τη δυνατότητα να αναπαραστήσουμε τους πίνακες της βάσης δεδομένων σε αντικείμενα (βλέπουμε εικόνα παρακάτω) και να σώσουμε την πληροφορία στη βάση δεδομένων καλώντας κάθε φορά δύο γραμμές κώδικα, πχ:

```
Using db = New GraduateDataContext()  
Dim rq = New PositionRequest()  
rq.PositionId = Convert.ToInt32(Request("id"))  
rq.UserId = New Guid(Membership.GetUser().ProviderUserKey.ToString())  
rq.UserComments = Me.Comments.Text  
rq.DateCreated = DateTime.Now
```

```
db.PositionRequests.InsertOnSubmit(rq)  
db.SubmitChanges()
```

```
Response.Redirect(Request.RawUrl)  
End Using
```



Εικόνα 21: Αναπαράσταση των πινάκων της βάσης δεδομένων σε αντικείμενα στον κώδικα (dbml)

Αν δούμε για παράδειγμα στον παραπάνω διάγραμμα γίνεται εύκολα αντιληπτό ότι το αντικείμενο **PositionRequest** αντιστοιχεί στον πίνακα `dbo.PositionRequest`. Η γραμμή `db.SubmitChanges` δίνει την εντολή να πραγματοποιηθούν όλες οι αλλαγές στην βάση δεδομένων μας. Αυτό είναι κομμάτι του **.NET Framework**. Δημιουργήσαμε ένα `data context` με όνομα `GraduateDataContext` που έχει μέσα τους πίνακες σαν αντικείμενα και με τη βοήθεια της τεχνολογίας **.Net Framework** διαβάζουμε/ενημερώνουμε/διαγράφουμε πληροφορία χωρίς να χρειαστεί να φτιάξουμε ρουτίνες σε επίπεδο `database`.

# 5. Παρουσίαση της Εφαρμογής

Η εφαρμογή **ΠΥΛΗ ΗΛΕΚΤΡΟΝΙΚΩΝ ΑΙΤΗΣΕΩΝ** χωρίζεται σε δύο επίπεδα

- ❖ Στο επίπεδο απλού χρήστη
- ❖ Στο επίπεδο χρήστη διαχειριστή της εφαρμογής

Παρακάτω παρουσιάζονται όλες οι δυνατότητες που παρέχει η εφαρμογή μας τόσο στο διαχειριστή *administrator* όσο και στους χρήστες *users* σε επίπεδο διεπαφής .

## 5.1 Η εφαρμογή σε επίπεδο διαχειριστή

Ο διαχειριστής ορίζεται βάση του username με το οποίο έχει εγγραφεί σε επίπεδο προγραμματισμού.

Ο διαχειριστής της εφαρμογής έχει στη διάθεσή τους όλες τις βασικές λειτουργίες απλού χρήστη αλλά και όλα τα απαραίτητα εργαλεία προκειμένου να είναι σε θέση να διαδραματίσει τους ρόλους που του αναλογούν στην εφαρμογή οι οποίοι είναι :

- ❖ Η δημιουργία θέσεων στο σύστημα
- ❖ Η αξιολόγηση αιτήσεων χρηστών
- ❖ Η έγκριση – απόρριψη αιτήσεων χρηστών

Από την αρχική σελίδα της εφαρμογής ο διαχειριστής, θα πρέπει να συνδεθεί επιλέγοντας «σύνδεση»



Εικόνα 22: Αρχική σελίδα εφαρμογής

Στην σελίδα σύνδεσης στο σύστημα θα πρέπει να καταχωρήσει το username (*E-mail*) και password (*κωδικός πρόσβασης*) που έχουν οριστεί για τον admin user και να επιλέξει «Σύνδεση»

Στην παρουσίασή μας σας Admin User στο σύστημα έχει οριστεί με τα παρακάτω στοιχεία σε επίπεδο βάσης δεδομένων.

E-mail : [aagelis@gmail.com](mailto:aagelis@gmail.com) κωδικός πρόσβασης: 123test!@#

Πατώντας σύνδεση στην αρχική σελίδα της εφαρμογής ο διαχειριστής πρέπει να εισάγει τα στοιχεία του σαν υπάρχον χρήστης όπως φαίνεται παρακάτω



ΝΕΟΣ ΧΡΗΣΤΗΣ ;

Συμπληρώστε τη φόρμα που ακολουθεί για να δημιουργήσετε νέο λογαριασμό.

Ο κωδικός πρόσβασης πρέπει να είναι τουλάχιστον 6 χαρακτήρες.

Account Information

E-mail:

Κωδικός πρόσβασης:

Επιβεβαίωση κωδικού πρόσβασης:

ΣΥΝΔΕΣΗ

Παρακαλώ εισάγετε το e-mail και τον κωδικό πρόσβασης σας.

Στοιχεία Λογαριασμού

E-mail:

Κωδικός πρόσβασης:

Να με θυμάσαι

Εικόνα 23: Σελίδα Εγγραφής / Εισόδου

Μετά την επιτυχημένη σύνδεση στην επικεφαλίδα της εφαρμογής εμφανίζονται τα παρακάτω :

Εικόνα 24: Αναφορά χρήστη συνδεδεμένου στο σύστημα

Η επιπλέον επιλογές του διαχειριστή της εφαρμογής σε σχέση με τον απλό χρήστη της εφαρμογής συνοψίζονται στην επιλογή που παρουσιάζεται δίπλα στο username στην επικεφαλίδα της εφαρμογής «διαχείριση»

Εικόνα 25: Επιπλέον Επιλογή Διαχειριστή Εφαρμογής

Μέσα από τη συγκεκριμένη σελίδα ο διαχειριστής της εφαρμογής ,μπορεί να πραγματοποιήσει τις παρακάτω ενέργειες:

1. Δημιουργία Νέας Θέσης
2. Επεξεργασία Θέσης (αλλαγή στοιχείων απαιτήσεων, αλλαγή περίοδο αιτήσεων, άνοιγμα παλαιότερων θέσεων για υποβολή επιπλέον/νέων αιτήσεων κοκ).
3. Έλεγχο/ Αξιολόγηση (prolifes' χρηστών που έχουν υποβάλει αίτηση για κάποια συγκεκριμένη θέση .
4. Έγκριση Απόρριψη αίτησης χρήστη για συγκεκριμένη θέση με ή χωρίς αιτιολόγηση.
5. Εξαγωγή δεδομένων από τα profiles' χρηστών που έχουν υποβάλει αίτηση για μια συγκεκριμένη θέση, σε αρχείο μορφής csv, με δυνατότητα επιλογής συγκεκριμένων πεδίων που ενδιαφέρουν την παρούσα χρονική στιγμή τον διαχειριστή.

### 5.1.1 Δημιουργία Νέας Θέσης

Όπως προαναφέραμε ο διαχειριστής της εφαρμογής έχει την αρμοδιότητα δημιουργίας νέων θέσεων προς υποβολή αιτήσεων από τους χρήστες της εφαρμογής .

Αφού ο διαχειριστής συνδεθεί στο σύστημα και επιλέξει την επιλογή «διαχείριση» μπορείς να πραγματοποιήσει την δημιουργία νέας θέσης στο σύστημα επιλέγοντας «δημιουργία νέας θέσης»

ΕΠΕΞΕΡΓΑΣΙΑ ΘΕΣΕΩΝ

επιλογή θέσης  ή [δημιουργία νέας θέσης](#)

Όνομασία:

Περιγραφή:

Ελάχιστο επίπεδο εκπαίδευσης:

Ημερομηνία έναρξης:

Ημερομηνία λήξης:

Εικόνα 26: Δημιουργία Θέσης από Διαχειριστή

Ο διαχειριστής καταχωρεί :

- ❖ Ονομασία Θέσης
- ❖ Περιγραφή Θέσης

Ο διαχειριστής επίσης πρέπει να επιλέξει:

- ❖ Ελάχιστο Επίπεδο Εκπαίδευσης (λειτουργία αυτόματης απόρριψη αίτησης χρήστη)
- ❖ Ημερομηνία Έναρξης Υποβολής Αιτήσεων
- ❖ Ημερομηνία Λήξης Υποβολής Αιτήσεων

ΕΠΕΞΕΡΓΑΣΙΑ ΘΕΣΕΩΝ

επιλογή θέσης  ή [δημιουργία νέας θέσης](#)

Όνομασία:

Περιγραφή:

Ελάχιστο επίπεδο εκπαίδευσης:

Ημερομηνία έναρξης:

Ημερομηνία λήξης:

Εικόνα 27: Συμπλήρωση Στοιχείων για δημιουργία νέας θέσης

Αφού συμπληρωθούν όλα τα απαιτούμενα στοιχεία και επιλεγεί το «αποθήκευση» η νέα θέση δημιουργείται στο σύστημα .

Επιστρέφοντας στην αρχική σελίδα της εφαρμογής διαπιστώνουμε ότι η πλέον η θέση έχει δημιουργηθεί και είναι άμεσα ανοιχτή για αιτήσεις χρηστών (εφόσον είμαστε εντός ημερομηνιών που έχουν οριστεί από το διαχειριστή)

ΚΑΛΩΣΗΡΘΑΤΕ ΣΤΗΝ ΠΥΛΗ ΗΛΕΚΤΡΟΝΙΚΩΝ ΑΙΤΗΣΕΩΝ

Θέσεις για υποβολή αιτήσεων

μόνο ανοιχτές  όλες

1. [Τεχνικός και Μηχανικός Η/Υ - Δικτύων - Τηλεπικοινωνιών \(κωδικός: #12001\)](#)

Η ΕΤΑΙΡΙΑ Μηχανογράφησης Business Software, με έδρα την Θεσσαλονίκη ζητά Τεχνικό Υπολογιστών και Δικτύων, για το τεχνικό τμήμα της εταιρίας. Απαραίτητη προϋπηρεσία 2 ετών και δίπλωμα οδήγησης.

περίοδος αιτήσεων από 2/8/2014 έως 30/9/2014 - (Τεχνολογική εκπαίδευση και άνω) - ανοιχτό

[ πρώτη ] [ προηγούμενη ] 1 [ επόμενη ] [ τελευταία ]

Εικόνα 28: Απεικόνιση ανοιχτής θέσης στην αρχική σελίδα της εφαρμογής

Ακριβώς κάτω από το link υποβολής (τίτλος θέσης) υπάρχει η περιγραφή της θέσης, η περίοδος για την οποία το σύστημα θα είναι ανοιχτό για υποβολή αιτήσεων για τη συγκεκριμένη θέση καθώς και το κατώτατο επίπεδο εκπαίδευσης που είναι απαραίτητο προκειμένου η αίτηση του χρήστη να γίνει αποδεκτή από το σύστημα.

### 5.1.2 Επεξεργασία Θέσης

Ο διαχειριστής έχει την δυνατότητα ανά πάση στιγμή να προβεί σε αλλαγές στην οποιαδήποτε θέση έχει δημιουργήσει στο σύστημα.

Επιλέγοντας τη συγκεκριμένη θέση από το drop down menu θέσεων στη σελίδα διαχείρισης , μπορεί να μεταβάλει όλα τα δεδομένα της θέσης :

- Ονομασία :
- Περιγραφή:
- Ελάχιστο επίπεδο εκπαίδευσης : (ισχύει για αιτήσεις που θα υποβληθούν μετά την αλλαγή)
- Ημερομηνία έναρξης : (ισχύει για αιτήσεις που θα υποβληθούν μετά την αλλαγή)
- Ημερομηνία λήξης :

ΕΠΕΞΕΡΓΑΣΙΑ ΘΕΣΕΩΝ

επιλογή θέσης: [#12001 - Τεχνικός και Μηχανικός Η/Υ - Δικτύων - Τηλεπικοινωνιών](#) ή [δημιουργία νέας θέσης](#)

Κωδικός: [#12001 - Τεχνικός και Μηχανικός Η/Υ - Δικτύων - Τηλεπικοινωνιών](#)  
[#13001 - Διαχειριστής Δικτύου στην Megasoft A.E.](#)

Ονομασία:

Περιγραφή:

Ελάχιστο επίπεδο εκπαίδευσης:

Ημερομηνία έναρξης:

Ημερομηνία λήξης:

Ο διαχειριστής μεταβάλλει τα στοιχεία που επιθυμεί και πατώντας αποθήκευση οι αλλαγές πραγματοποιούνται στο σύστημα.

### 5.1.3 Έλεγχος/Αξιολόγηση αιτήσεων χρηστών

Στο κάτω μέρος της οθόνης διαχείρισης δημιουργείται λίστα χρηστών που έχουν υποβάλει αίτηση για την συγκεκριμένη θέση .

Αναφέρεται επίσης η ημερομηνία και η ώρα υποβολής της αίτησης καθώς και η κατάσταση της αίτησης

#### ΑΙΤΗΣΕΙΣ ΓΙΑ ΤΗ ΘΕΣΗ

[εξαγωγή δεδομένων σε excel \(εμφάνιση\)](#)

Όνοματεπώνυμο	Ημέρ/νία υποβολής	Κατάσταση	Σχόλια	Ενέργεια
<a href="#">Ηλίας Καδής</a>	2/8/2014 2:49:58 μμ	σε εξέλιξη	-	<a href="#">έγκριση</a>   <a href="#">απόρριψη</a>
<a href="#">Νικόλαος Ευαγγέλου</a>	2/8/2014 3:02:57 μμ	σε εξέλιξη	-	<a href="#">έγκριση</a>   <a href="#">απόρριψη</a>

Εικόνα 30: Έλεγχος/Αξιολόγηση αιτήσεων χρηστών από διαχειριστή

Ο διαχειριστής έχει την δυνατότητα να μεταφερθεί στο profile(CV) του εκάστοτε χρήστη προκειμένου να εξετάσει το βιογραφικό του χρήστη και να αποφανθεί σχετικά με την έγκριση ή την απόρριψη της αίτησης του χρήστη .

aagelis@gmail.com | [διαχείριση](#) | [προφίλ](#) | [αιτήσεις](#) | [αποσύνδεση](#)

## ΠΥΛΗ ΗΛΕΚΤΡΟΝΙΚΩΝ ΑΙΤΗΣΕΩΝ

Αρχική

#### ΠΡΟΒΟΛΗ ΠΡΟΦΙΛ

Όνομα	Ηλίας		
Επώνυμο	Καδής		
E-mail	user1@teiep.gr		
Τηλέφωνο	2108614015		
Ημερ. γέννησης	07/07/1980		
Διεύθυνση	Χρ. Μιχαήλ 13		
Εκπαίδευση	<b>σχολή</b>	<b>περίοδος</b>	-
	ΤΕΙ ΙΟΝΙΩΝ ΝΗΣΩΝ / Εφαρμογών Πληροφορικής στη Διοίκηση & στην Οικονομία (Λευκάδα)		από Σεπ 1999 έως Ιουν 2005
Χώρα	Ελλάδα ▼		
Εθνικότητα	Ελληνική		
Οικογενειακή κατάσταση	<input checked="" type="radio"/> Άγαμος <input type="radio"/> Παντρεμένος		
Εργασιακή εμπειρία	<b>εταιρεία</b>	<b>τίτλος</b>	<b>περίοδος</b>
	PLAISIO	Τεχνικών Ηλεκτρονικών Υπολογιστών	από Μαρ 2006 έως Ιουν 2012
σύνολικό διάστημα εμπειρίας: 6,3 χρόνος/τα			
Website			
Facebook			
Twitter			

Εικόνα 31: Προβολή προφίλ/βιογραφικού χρήστη από διαχειριστή

#### 5.1.4 Έγκριση / Απόρριψη αίτησης χρήστη για συγκεκριμένη θέση

Από την σελίδα διαχείρισης ο διαχειριστής έχει την δυνατότητα, μετά από την αξιολόγηση των profiles' των χρηστών, να προχωρήσει σε έγκριση ή απόρριψη αιτήσεων χρηστών .

##### ΑΙΤΗΣΕΙΣ ΓΙΑ ΤΗ ΘΕΣΗ

[εξαγωγή δεδομένων σε excel \(εμφάνιση\)](#)

Όνοματεπώνυμο	Ημέρ/νία υποβολής	Κατάσταση	Σχόλια	Ενέργεια
<a href="#">Ηλίας Καδής</a>	2/8/2014 2:49:58 μμ	σε εξέλιξη	-	<a href="#">έγκριση</a>   <a href="#">απόρριψη</a>
<a href="#">Νικόλαος Ευαγγέλου</a>	2/8/2014 3:02:57 μμ	σε εξέλιξη	-	<a href="#">έγκριση</a>   <a href="#">απόρριψη</a>

Εικόνα 32: Σελίδα έγκρισης/απόρριψης αίτησης από διαχειριστή

Πατώντας έγκριση ή απόρριψη ο διαχειριστής καλείται να σχολιάσει τον λόγο έγκρισης/ απόρριψης όπως για παράδειγμα στην παρακάτω εικόνα:

αιτιολογία απόρριψης

ολογία: Η αίτησή σας δεν έχει επιλεγεί. Σας ευχαριστούμε πολύ

OK Άκυρο

Εικόνα 33: Παράθυρο σχολίων απόρριψης αίτησης

#### 5.1.5 Εξαγωγή δεδομένων χρηστών σε .xls

Ο διαχειριστής έχει επίσης την δυνατότητα να προχωρήσει σε εξαγωγή δεδομένων για όλους τους χρήστες που έχουν υποβάλει αίτηση για μια συγκεκριμένη θέση, σε αρχείο μορφής .xls, με δυνατότητα επιλογής συγκεκριμένων πεδίων που τον ενδιαφέρουν ανά πάσα χρονική στιγμή, προκειμένου να μπορεί να έχει συγκεντρωτικά όλους τους υποψηφίους και τα χαρακτηριστικά αυτών τα οποία θα ήθελε να συγκρίνει.

##### ΑΙΤΗΣΕΙΣ ΓΙΑ ΤΗ ΘΕΣΗ

[εξαγωγή σε excel \(απόκρυψη\)](#)

- |   |  |  |   |
|---|--|--|---|
| <input checked="" type="checkbox"/> Όνοματεπώνυμο       | <input checked="" type="checkbox"/> E-mail         | <input checked="" type="checkbox"/> Διεύθυνση  | <input type="checkbox"/> Εθνικότητα             |
| <input checked="" type="checkbox"/> Ημερομηνία υποβολής | <input checked="" type="checkbox"/> Τηλέφωνο       | <input checked="" type="checkbox"/> Εκπαίδευση | <input type="checkbox"/> Οικογενειακή κατάσταση |
| <input checked="" type="checkbox"/> Κατάσταση           | <input checked="" type="checkbox"/> Ημερ. Γέννησης | <input type="checkbox"/> Χώρα                  | <input type="checkbox"/> Εργασιακή εμπειρία     |
| <input checked="" type="checkbox"/> Σχόλια              |  |  |   |

[εξαγωγή](#)

Όνοματεπώνυμο	Ημέρ/νία υποβολής	Κατάσταση	Σχόλια	Ενέργεια
<a href="#">Ηλίας Καδής</a>	2/8/2014 2:49:58 μμ	σε εξέλιξη	-	<a href="#">έγκριση</a>   <a href="#">απόρριψη</a>
<a href="#">Νικόλαος Ευαγγέλου</a>	2/8/2014 3:02:57 μμ	σε εξέλιξη	-	<a href="#">έγκριση</a>   <a href="#">απόρριψη</a>

Εικόνα 34: Επιλογή στοιχείων προς εξαγωγή σε xls

Αφού επιλέξει τα πεδία που επιθυμεί να εξάγει για τους υποψηφίους, πατώντας «εξαγωγή» ,ο διαχειριστής έχει στην διάθεσή του για αποθήκευση ή εκτύπωση, ένα αρχείο όπως το παρακάτω το οποίο μπορεί να επεξεργαστεί προκειμένου να προβεί σε συγκριτική αξιολόγηση των υποβληθέντων αιτήσεων.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Όνοματεπώνυμο	Τηλέφωνο	Ημερ. Γέννησης	Εργασιακή εμπειρία									
2	Ηλίας Καδής	2108614015	07/07/1980	PLAISIO / Τεχνικών Ηλεκτρονικών Υπολογιστών από Μαρ 2006 έως Ιουν 2012									
3	Νικόλαος Ευαγγέλου	2310234560	02/07/1984	ALTEC ABEE / Προγραμματιστής από Ιουν 2007 έως και σήμερα, CNT Computers / Προγραμματιστής από Μαρ :									
4													
5													
6													
7													

Εικόνα 35: Εξαγωγή στοιχείων χρηστών

## 5.2 Η εφαρμογή σε επίπεδο χρήστη

Η εφαρμογή έχει σχεδιαστεί προκειμένου κάθε χρήστης να έχει την δυνατότητα να διατηρεί το profile του το οποίο μπορεί να επεξεργαστεί προκειμένου να έχει την δυνατότητα να υποβάλει αιτήσεις χωρίς να είναι υποχρεωμένος να συμπληρώνει συνεχώς φόρμες.

Όταν ο χρήστης επισκεφτεί την αρχική σελίδα της εφαρμογής έχει τρεις δυνατότητες :

1. Δημιουργία λογαριασμού στο σύστημα (νέος χρήστης)
2. Σύνδεση στο σύστημα (υπάρχον χρήστης)
3. Άμεση υποβολή αίτησης και ταυτόχρονη υποβολή αίτησης για συγκεκριμένη θέση (νέος και υπάρχον χρήστης)
4. Επεξεργασία Profile (βιογραφικού)
5. Έλεγχος κατάστασης αιτήσεων

### 5.2.1 Δημιουργία λογαριασμού στο σύστημα (νέος χρήστης)

Από την αρχική σελίδα ο χρήστης επιλέγει σύνδεση όπως φαίνεται στην παρακάτω εικόνα.

**ΠΥΛΗ ΗΛΕΚΤΡΟΝΙΚΩΝ ΑΙΤΗΣΕΩΝ** [ σύνδεση ]

Αρχική

ΚΑΛΩΣΗΡΘΑΤΕ ΣΤΗΝ ΠΥΛΗ ΗΛΕΚΤΡΟΝΙΚΩΝ ΑΙΤΗΣΕΩΝ

Θέσεις για υποβολή αιτήσεων

μόνο ανοιχτές  όλες

1. [Διαχειριστής Δικτύου στην Megasoft A.E. \(κωδικός: #13001\)](#)  
Το άτομο που αναζητούμε θα εργαστεί στο Υποκατάστημα μας στην Θεσσαλονίκη και θα αναλάβει ως αρμοδιότητα την υποστήριξη δικτύου των συνεργατών και των πελατών μας στην Βόρεια Ελλάδα όσον αφορά τις υπηρεσίες δικτύων δηλαδή εγκατάσταση, διάρθρωση και λειτουργία βασικών δικτυακών υπηρεσιών.  
περίοδος αιτήσεων από 2/8/2014 έως 31/10/2014 - (Δευτεροβάθμια εκπαίδευση και άνω) - **ανοιχτό**
2. [Τεχνικός και Μηχανικός Η/Υ - Δικτύων - Τηλεπικοινωνιών \(κωδικός: #12001\)](#)  
Η ΕΤΑΙΡΙΑ Μηχανογράφησης Business Software, με έδρα την Θεσσαλονίκη ζητά Τεχνικό Υπολογιστών και Δικτύων, για το τεχνικό τμήμα της εταιρίας. Απαραίτητη προϋπηρεσία 2 ετών και δίπλωμα οδήγησης.  
περίοδος αιτήσεων από 2/8/2014 έως 30/9/2014 - (Τεχνολογική εκπαίδευση και άνω) - **ανοιχτό**

[ πρώτη ] [ προηγούμενη ] 1 [ επόμενη ] [ τελευταία ]

Εικόνα 36: Σελίδα Προβολής Θέσεων

Μεταφέρεται στην σελίδα εισόδου και καταχωρεί τα στοιχεία με τα οποία θέλει να εγγραφεί (e-mail και κωδικό εισόδου) στα ανάλογα πεδία που αφορούν εγγραφές νέων χρηστών

#### ΝΕΟΣ ΧΡΗΣΤΗΣ ;

Συμπληρώστε τη φόρμα που ακολουθεί για να δημιουργήσετε νέο λογαριασμό.

Ο κωδικός πρόσβασης πρέπει να είναι τουλάχιστον 6 χαρακτήρες.

Account Information

E-mail:

Κωδικός πρόσβασης:

Επιβεβαίωση κωδικού πρόσβασης:

#### ΣΥΝΔΕΣΗ

Παρακαλώ εισάγετε το e-mail και τον κωδικό πρόσβασης σας.

Στοιχεία Λογαριασμού

E-mail:

Κωδικός πρόσβασης:

Να με θυμάσαι

Εικόνα 37: Σελίδα Εγγραφής Εισόδου

Αφού πατήσει εγγραφή μετακινείται στην σελίδα που πρέπει να καταχωρήσει τα στοιχεία του profile του. Υπάρχουν πεδία που είναι υποχρεωτικά για την ολοκλήρωση της δημιουργίας του προφίλ.

#### ΕΠΕΞΕΡΓΑΣΙΑ ΠΡΟΦΙΛ

- συμπληρώστε το όνομα σας
- συμπληρώστε το επώνυμο σας
- συμπληρώστε το τηλέφωνο σας
- συμπληρώστε την ημερομηνία γέννησης σας
- συμπληρώστε την διεύθυνση σας
- συμπληρώστε την διεύθυνση σας
- συμπληρώστε την εθνικότητα σας

Όνομα  \*

Επώνυμο  \*

E-mail user3@teiep.gr

Τηλέφωνο  \*

Ημερ. γέννησης  \*

Διεύθυνση  \*

Εκπαίδευση δεν έχει καταχωρηθεί εκπαίδευση. [ [προσθήκη](#) ]

Χώρα  \*

Εθνικότητα  \*

Οικογενειακή κατάσταση  Άγαμος  Παντρεμένος

Εργασιακή εμπειρία δεν έχει καταχωρηθεί εργασιακή εμπειρία. [ [προσθήκη](#) ]

Website

Facebook

Εικόνα 38: Σελίδα Δημιουργία Προφίλ Χρήστη (υποχρεωτικά πεδία)

Ο χρήστης μπορεί να επιλέξει την σειρά με την οποία θα εισάγει τα στοιχεία που επιθυμεί πχ. Μπορεί να πατήσει πρώτα προσθήκη εκπαίδευσης ή εργασιακής εμπειρίας και μετά να επανέλθει στην σελίδα του προφίλ του, όμως προκειμένου το σύστημα να δεχθεί και να ολοκληρώσει επιτυχημένα την δημιουργία της σελίδας του προφίλ του θα πρέπει να έχει συμπληρώσει όλα τα ορισμένα ως υποχρεωτικά πεδία πριν πατήσει αποθήκευση .

Σε διαφορετική περίπτωση το σύστημα συνεχίζει κατά την υποβολή (αποθήκευση) να ενημερώνει για το ποια πεδία είναι απαραίτητα για να ολοκληρωθεί η εγγραφή

#### ΕΠΕΞΕΡΓΑΣΙΑ ΠΡΟΦΙΛ

- συμπληρώστε το τηλέφωνο σας
- συμπληρώστε την ημερομηνία γέννησης σας
- συμπληρώστε την διεύθυνση σας
- συμπληρώστε την διεύθυνση σας
- συμπληρώστε την εθνικότητα σας

Όνομα	<input type="text" value="Μανώλης"/>						
Επώνυμο	<input type="text" value="Εγγατίου"/>						
E-mail	<input type="text" value="user3@teiep.gr"/>						
Τηλέφωνο	<input type="text"/> *						
Ημερ. γέννησης	<input type="text"/> *						
Διεύθυνση	<input type="text"/>						
Εκπαίδευση	<table border="1"><thead><tr><th>σχολή</th><th>περίοδος</th><th>-</th></tr></thead><tbody><tr><td>ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ / Μηχανικών Ορυκτών Πόρων [ <a href="#">προσθήκη</a> ]</td><td>από Σεπ 2004 έως Αυγ 2010</td><td><a href="#">επεξεργασία</a> <a href="#">διαγραφή</a></td></tr></tbody></table>	σχολή	περίοδος	-	ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ / Μηχανικών Ορυκτών Πόρων [ <a href="#">προσθήκη</a> ]	από Σεπ 2004 έως Αυγ 2010	<a href="#">επεξεργασία</a> <a href="#">διαγραφή</a>
σχολή	περίοδος	-					
ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ / Μηχανικών Ορυκτών Πόρων [ <a href="#">προσθήκη</a> ]	από Σεπ 2004 έως Αυγ 2010	<a href="#">επεξεργασία</a> <a href="#">διαγραφή</a>					
Χώρα	<input type="text" value="-"/> *						
Εθνικότητα	<input type="text"/> *						
Οικογενειακή κατάσταση	<input checked="" type="radio"/> Άγαμος <input type="radio"/> Παντρεμένος						

Εικόνα 39: Σελίδα Δημιουργία Προφίλ Χρήστη (υποχρεωτικά πεδία)

Αφού ο χρήστης συμπληρώσει τουλάχιστον όλα τα υποχρεωτικά πεδία κατά την αποθήκευση λαμβάνει το μήνυμα ότι το προφίλ του ενημερώθηκε επιτυχώς.

Αρχική									
ΕΠΕΞΕΡΓΑΣΙΑ ΠΡΟΦΙΛ									
Όνομα	<input type="text" value="Μανώλης"/>								
Επώνυμο	<input type="text" value="Εγγατίου"/>								
E-mail	<input type="text" value="user3@teiep.gr"/>								
Τηλέφωνο	<input type="text" value="2108845324"/>								
Ημερ. γέννησης	<input type="text" value="13/01/1985"/>								
Διεύθυνση	<input type="text" value="Μαυρογιάννη 14"/>								
Εκπαίδευση	<table border="1"><thead><tr><th>σχολή</th><th>περίοδος</th><th>-</th></tr></thead><tbody><tr><td>ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ / Μηχανικών Ορυκτών Πόρων [ <a href="#">προσθήκη</a> ]</td><td>από Σεπ 2004 έως Αυγ 2010</td><td><a href="#">επεξεργασία</a> <a href="#">διαγραφή</a></td></tr></tbody></table>	σχολή	περίοδος	-	ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ / Μηχανικών Ορυκτών Πόρων [ <a href="#">προσθήκη</a> ]	από Σεπ 2004 έως Αυγ 2010	<a href="#">επεξεργασία</a> <a href="#">διαγραφή</a>		
σχολή	περίοδος	-							
ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ / Μηχανικών Ορυκτών Πόρων [ <a href="#">προσθήκη</a> ]	από Σεπ 2004 έως Αυγ 2010	<a href="#">επεξεργασία</a> <a href="#">διαγραφή</a>							
Χώρα	<input type="text" value="Ελλάδα"/>								
Εθνικότητα	<input type="text" value="Ελληνική"/>								
Οικογενειακή κατάσταση	<input checked="" type="radio"/> Άγαμος <input type="radio"/> Παντρεμένος								
Εργασιακή εμπειρία	<table border="1"><thead><tr><th>εταιρεία</th><th>τίτλος</th><th>περίοδος</th><th>-</th></tr></thead><tbody><tr><td>INKMAN</td><td>Administrative Support</td><td>από Αυγ 2010 έως και σήμερα</td><td><a href="#">επεξεργασία</a> <a href="#">διαγραφή</a></td></tr></tbody></table> <p>σύνολικό διάστημα εμπειρίας: 3,9 χρόνος/ια [ <a href="#">προσθήκη</a> ]</p>	εταιρεία	τίτλος	περίοδος	-	INKMAN	Administrative Support	από Αυγ 2010 έως και σήμερα	<a href="#">επεξεργασία</a> <a href="#">διαγραφή</a>
εταιρεία	τίτλος	περίοδος	-						
INKMAN	Administrative Support	από Αυγ 2010 έως και σήμερα	<a href="#">επεξεργασία</a> <a href="#">διαγραφή</a>						
Website	<input type="text"/>								
Facebook	<input type="text"/>								
Twitter	<input type="text"/>								
Δραστηριότητες	<input type="text"/>								
Συστάσεις	<input #"="" type="text" value="δεν έχουν καταχωρηθεί συστάσεις [ &lt;a href="/> προσθήκη ]"/>								
το προφίλ ενημερώθηκε επιτυχώς									
<input type="button" value="αποθήκευση"/>									

Εικόνα 40: Σελίδα Δημιουργία Προφίλ Χρήστη αποθήκευση στοιχείων

Πλέον ο νέος χρήστης είναι σε θέση να υποβάλει αιτήσεις μέσω της εφαρμογής .



Από την αρχική σελίδα και καθώς είναι σε σύνδεση στην εφαρμογή επιλέγει την θέση για την οποία επιθυμεί να υποβάλει αίτηση

#### ΠΡΟΒΟΛΗ ΘΕΣΗΣ

Κωδικός / ονομασία θέσης: #13001 - Διαχειριστής Δικτύου στην Megasoft A.E.

Περίοδος αιτήσεων: από 2/8/2014 έως 31/10/2014

Περιγραφή: Το άτομο που αναζητούμε θα εργαστεί στο Υποκατάστημα μας στην Θεσσαλονίκη και θα αναλάβει ως αρμοδιότητα την υποστήριξη δικτύου των συνεργατών και των πελατών μας στην Βόρεια Ελλάδα όσον αφορά τις υπηρεσίες δικτύων δηλαδή εγκατάσταση, διάρθρωση και λειτουργία βασικών δικτυακών υπηρεσιών.

Σχόλια:

υποβολή αίτησης

Εικόνα 41: Σελίδα υποβολής αίτησης από χρήστη

Στην προβολή θέσης ο χρήστης μπορεί να προσθέσει τα σχόλια που επιθυμεί και να πατήσει υποβολή αίτησης.

#### ΠΡΟΒΟΛΗ ΘΕΣΗΣ

Κωδικός / ονομασία θέσης: #12001 - Τεχνικός και Μηχανικός Η/Υ - Δικτύων - Τηλεπικοινωνιών

Περίοδος αιτήσεων: από 2/8/2014 έως 30/9/2014

Περιγραφή: Η ΕΤΑΙΡΙΑ Μηχανογράφησης Business Software, με έδρα την Θεσσαλονίκη ζητά Τεχνικό Υπολογιστών και Δικτύων, για το τεχνικό τμήμα της εταιρίας. Απαραίτητη προϋπηρεσία 2 ετών και δίπλωμα οδήγησης.

Ημερ. αίτησης Εγκρίθηκε Λόγος απόρριψης  
2/8/2014 σε εξέλιξη -

Εικόνα 42: Ολοκλήρωση υποβολής αίτησης από χρήστη

Πλέον ο χρήστης έχει την δυνατότητα να παρακολουθεί την πορεία της αίτησης στο από την σελίδα «αιτήσεις»

**ΠΥΛΗ ΗΛΕΚΤΡΟΝΙΚΩΝ ΑΙΤΗΣΕΩΝ** user3@teiep.gr | [προφίλ](#) [αιτήσεις](#) [αποσύνδεση](#)

Αρχική

Εικόνα 43: Μεταφορά χρήστη στη σελίδα ελέγχου κατάστασης αιτήσεων

Όπως φαίνεται παρακάτω.

#### ΙΣΤΟΡΙΚΟ ΤΩΝ ΑΙΤΗΣΕΩΝ ΜΟΥ

Θέση	Ημερ. αίτησης	Κατάσταση	Σχόλια
Τεχνικός και Μηχανικός Η/Υ - Δικτύων - Τηλεπικοινωνιών Η ΕΤΑΙΡΙΑ Μηχανογράφησης Business Software, με έδρα την Θεσσαλονίκη ζητά Τεχνικό Υπολογιστών και Δικτύων, για το τεχνικό τμήμα της εταιρίας. Απαραίτητη προϋπηρεσία 2 ετών και δίπλωμα οδήγησης.	2/8/2014	σε εξέλιξη	-

Εικόνα 44: Παρακολούθηση κατάστασης αίτησης από χρήστη

### 5.2.2 Σύνδεση στο σύστημα (υπάρχον χρήστης)

Οι χρήστες που ήδη διαθέτουν λογαριασμό στο σύστημα μπορούν να συνδεθούν με τα στοιχεία τους.

The image displays two side-by-side screenshots of a web application interface. The left screenshot is titled 'NEOS ΧΡΗΣΤΗΣ;' and contains the following text: 'Συμπληρώστε τη φόρμα που ακολουθεί για να δημιουργήσετε νέο λογαριασμό.' and 'Ο κωδικός πρόσβασης πρέπει να είναι τουλάχιστον 6 χαρακτήρες.' Below this is a form box labeled 'Account Information' with three input fields: 'E-mail:', 'Κωδικός πρόσβασης:', and 'Επιβεβαίωση κωδικού πρόσβασης:'. A button labeled 'εγγραφή' is positioned below the form. The right screenshot is titled 'ΣΥΝΔΕΣΗ' and contains the text: 'Παρακαλώ εισάγετε το e-mail και τον κωδικό πρόσβασης σας.' Below this is a form box labeled 'Στοιχεία Λογαριασμού' with two input fields: 'E-mail:' and 'Κωδικός πρόσβασης:'. There is also a checkbox labeled 'Να με θυμάσαι' and a button labeled 'σύνδεση' below the form.

Εικόνα 45: Σελίδα εισόδου υπάρχον χρήστη στην εφαρμογή

### 5.2.3 Άμεση υποβολή αίτησης και ταυτόχρονη υποβολή για συγκεκριμένη θέση (νέος και υπάρχον χρήστης)

Όταν κάποιος χρήστης νέος η υπάρχον επιλέξει εκτός σύνδεσης κάποια διαθέσιμη για υποβολή αίτησης το σύστημα του ζητάει να συνδεθεί για να υποβληθεί η αίτηση

Αμέσως μετά την σύνδεση το σύστημα μεταφέρει τον χρήστη στην σελίδα του προφίλ του

- Αν είναι νέος χρήστης θα πρέπει να ακολουθήσει την διαδικασία συμπλήρωσης των στοιχείων του προφίλ του.
- Αν είναι νέο χρήστης μπορεί να επεξεργαστεί κάποια στοιχεία του προφίλ του πριν την υποβολή της αίτησης.

και στις δύο όπως περιπτώσεις , λόγω του ότι ο λόγος σύνδεσης στο σύστημα έχει καθοριστεί πριν την σύνδεση σε αυτό(επιλογή συγκεκριμένης θέσης για υποβολή αίτησης), στο κάτω μέρος της οθόνης του προφίλ υπάρχουν πλέον δύο επιλογές

ΕΠΕΞΕΡΓΑΣΙΑ ΠΡΟΦΙΛ

Όνομα   
 Επώνυμο   
 E-mail   
 Τηλέφωνο   
 Ημερ. γέννησης   
 Διεύθυνση

Εκπαίδευση

σχολή	περίοδος	-
ΠΑΝΕΠΙΣΤΗΜΙΟ ΣΤΕΡΕΑΣ ΕΛΛΑΔΑΣ / Πληροφορικής από Σεπ 2000 με Εφαρμογές στην Βιοϊατρική (Λαμία)	έως Ιουλ 2006	<a href="#">επεξεργασία</a> <a href="#">διαγραφή</a>
<a href="#">[ προσθήκη ]</a>		

Χώρα   
 Εθνικότητα   
 Οικογενειακή κατάσταση  Άγαμος  Παντρεμένος

Εργασιακή εμπειρία

εταιρεία	τίτλος	περίοδος	-
ALTEC ABEE	Προγραμματιστής	από Ιουν 2007 έως και σήμερα	<a href="#">επεξεργασία</a> <a href="#">διαγραφή</a>
CNT Computes	Προγραμματιστής	από Μαρ 2006 έως Μαϊ 2007	<a href="#">επεξεργασία</a> <a href="#">διαγραφή</a>
σύνολικό διάστημα εμπειρίας: 8,3 χρόνια/ια			
<a href="#">[ προσθήκη ]</a>			

Website   
 Facebook   
 Twitter   
 Δραστηριότητες

Συστάσεις

εταιρεία	ονοματεπώνυμο	e-mail	-
ALTEC ABEE ΠΑΝΑΓΙΩΤΟΥ	m.panagiotou@altec.gr	<a href="#">επεξεργασία</a> <a href="#">διαγραφή</a>	
<a href="#">[ προσθήκη ]</a>			

Εικόνα 46: Άμεση υποβολή αίτησης από την σελίδα του προφίλ

Μετά την δημιουργία του λογαριασμού του, ο χρήστης έχει την δυνατότητα να πραγματοποιήσει τις παρακάτω ενέργειες στο σύστημα:

#### 5.2.4 Επεξεργασία Profile(βιογραφικού)

Ο χρήστης της εφαρμογής έχει την δυνατότητα ανά πάσα στιγμή αφού συνδεθεί στο λογαριασμό του και μεταφερθεί στην σελίδα του προφίλ του.



Εικόνα 47: Μεταφορά συνδεδεμένου χρήστη στην σελίδα προφίλ

Ο χρήστης της εφαρμογής έχει την δυνατότητα να τροποποιήσει όλα τα στοιχεία του εκτός του email και του κωδικού πρόσβασης στην εφαρμογή

Παρακάτω μερικές από τις επιλογές του χρήστη κατά την επεξεργασία του προφίλ

- Προσθήκη/Επεξεργασία ή Διαγραφή Εκπαίδευσης

	σχολή	περίοδος	-
Εκπαίδευση	ΤΕΙ ΙΟΝΙΩΝ ΝΗΣΩΝ / Εφαρμογών Πληροφορικής στη Διοίκηση & στην Οικονομία (Λευκάδα) [ προσθήκη ]	από Σεπ 1999 έως Ιουν 2005	επεξεργασία διαγραφή

Εικόνα 48: Επεξεργασία/Διαγραφή εγγραφής εκπαίδευσης


Ο χρήστης έχει την δυνατότητα να επεξεργαστεί κάποια εγγραφή εκπαίδευσης που έχει πραγματοποιήσει,


## ΠΡΟΣΘΗΚΗ / ΕΠΕΞΕΡΓΑΣΙΑ ΕΚΠΑΙΔΕΥΣΗΣ

Τύπος σχολής: Τεχνολογική εκπαίδευση ▼

Σχολή: ΤΕΙ ΙΟΝΙΩΝ ΝΗΣΩΝ ▼

Τμήμα: Εφαρμογών Πληροφορικής στη Διοίκηση & στην Οικονομία (Λευκάδα) ▼

Από: 15/9/1999 

Έως: 15/6/2005 

αποθήκευση και επιστροφή στο προφίλ    ακύρωση & επιστροφή στο προφίλ

Εικόνα 49: Προσθήκη εγγραφής εκπαίδευσης

ή να προσθέσει μια νέα εγγραφή εκπαίδευσης. Κατά την καταχώρηση εγγραφής εκπαίδευσης ο χρήστης υποχρεούται να επιλέξει το από συγκεκριμένη λίστα το επίπεδο εκπαίδευσης για τη συγκεκριμένη εγγραφή.

## ΠΡΟΣΘΗΚΗ / ΕΠΕΞΕΡΓΑΣΙΑ ΕΚΠΑΙΔΕΥΣΗΣ

Τύπος σχολής: Δευτεροβάθμια εκπαίδευση ▼

Σχόλια

Από:

Έως:

- επιλέξτε -  
Υποχρεωτική εκπαίδευση  
Δευτεροβάθμια εκπαίδευση  
Τεχνολογική εκπαίδευση  
Πανεπιστημιακή εκπαίδευση

αποθήκευση και επιστροφή στο προφίλ    ακύρωση & επιστροφή στο προφίλ

Εικόνα 50: Επιλογή Τύπου Σχολής

Σε Καταχώρηση Υποχρεωτικής ή Δευτεροβάθμιας εκπαίδευσης ο χρήστης καλείτε υποχρεωτικά να καταχωρήσει κάποια σχόλια *πχ Σχολείο – Ίδρυμα.*

## ΠΡΟΣΘΗΚΗ / ΕΠΕΞΕΡΓΑΣΙΑ ΕΚΠΑΙΔΕΥΣΗΣ

### • συμπληρώστε τα σχόλια

Τύπος σχολής:

Σχόλια:

Από:

Έως:

Εικόνα 51: Υποχρεωτικά πεδία σε επιλογή σχολής Υποχρεωτικής εκπαίδευσης

## ΠΡΟΣΘΗΚΗ / ΕΠΕΞΕΡΓΑΣΙΑ ΕΚΠΑΙΔΕΥΣΗΣ

### • συμπληρώστε τα σχόλια

Τύπος σχολής:

Σχόλια:

Από:

Έως:

Εικόνα 52: Υποχρεωτικά πεδία σε επιλογή σχολής Δευτεροβάθμιας εκπαίδευσης

Στην περίπτωση επιλογής Τεχνολογικής ή Πανεπιστημιακής εκπαίδευσης ο χρήστης καλείται να επιλέξει την σχολή – ίδρυμα από συγκεκριμένες λίστες σχολών όπως φαίνεται παρακάτω.

**ΠΥΛΗ ΗΛΕΚΤΡΟΝΙΚΩΝ ΑΙΤΗΣΕΩΝ** user4@teiep.gr | [προφίλ](#) | [αιτήσεις](#) | [αποσύνδεση](#)

Αρχική

---

### ΠΡΟΣΘΗΚΗ / ΕΠΕΞΕΡΓΑΣΙΑ ΕΚΠΑΙΔΕΥΣΗΣ

Τύπος σχολής:

Σχολή:

Τμήμα:

Από:

Έως:

- Οικονομικής Επιστήμης
- Οικονομικής Επιστήμης
- Διεθνών & Ευρωπαϊκών Οικονομικών Σπουδών
- Οργάνωσης & Διοίκησης Επιχειρήσεων
- Στατιστικής
- Πληροφορικής
- Μάρκετινγκ & Επικοινωνίας
- Λογιστικής & Χρηματοοικονομικής
- Διοικητικής Επιστήμης & Τεχνολογίας

Εικόνα 53: Επιλογή τύπου σχολής σε επιλογή Πανεπιστημιακής Εκπαίδευσης

### 5.2.5 Έλεγχος κατάστασης αιτήσεων

Κάθε χρήστης έχει την δυνατότητα να ελέγχει την κατάσταση των αιτήσεων του

The screenshot shows the user interface for 'user3@teiep.gr'. The page title is 'ΠΥΛΗ ΗΛΕΚΤΡΟΝΙΚΩΝ ΑΙΤΗΣΕΩΝ'. The user is logged in, with links for 'προφίλ', 'αιτήσεις', and 'αποσύνδεση'. A 'Αρχική' button is visible. The main heading is 'ΙΣΤΟΡΙΚΟ ΤΩΝ ΑΙΤΗΣΕΩΝ ΜΟΥ'. Below this is a table with columns: 'Θέση', 'Ημερ. αίτησης', 'Κατάσταση', and 'Σχόλια'. The first row shows a request for 'Τεχνικός και Μηχανικός Η/Υ - Δικτύων - Τηλεπικοινωνιών' with a status of 'σε εξέλιξη'.

Θέση	Ημερ. αίτησης	Κατάσταση	Σχόλια
Τεχνικός και Μηχανικός Η/Υ - Δικτύων - Τηλεπικοινωνιών Η ΕΤΑΙΡΙΑ Μηχανογράφησης Bussiness Software, με έδρα την Θεσσαλονίκη ζητά Τεχνικό Υπολογιστών και 2/8/2014 σε εξέλιξη - Δικτύων, για το τεχνικό τμήμα της εταιρίας. Απαραίτητη προϋπηρεσία 2 ετών και δίπλωμα οδήγησης.	2/8/2014	σε εξέλιξη	-

Εικόνα 54: Έλεγχος κατάστασης αίτησης από χρήστη

Κάθε φορά που υπάρχει τροποποίηση στην κατάσταση κάποιας αίτησής χρήση το σύστημα του παρέχει σχετική ενημέρωση με ειδοποίηση (*notifications*).

The screenshot shows the user interface for 'user3@teiep.gr'. The page title is 'ΠΥΛΗ ΗΛΕΚΤΡΟΝΙΚΩΝ ΑΙΤΗΣΕΩΝ'. The user is logged in, with links for 'προφίλ', 'αιτήσεις', and 'αποσύνδεση'. A 'Αρχική' button is visible. The main heading is 'ΙΣΤΟΡΙΚΟ ΤΩΝ ΑΙΤΗΣΕΩΝ ΜΟΥ'. Below this is a table with columns: 'Θέση', 'Ημερ. αίτησης', 'Κατάσταση', and 'Σχόλια'. The first row shows a request for 'Διαχειριστής Δικτύου στην Megasoft A.E.' with a status of 'σε εξέλιξη'.

Θέση	Ημερ. αίτησης	Κατάσταση	Σχόλια
Διαχειριστής Δικτύου στην Megasoft A.E. Το άτομο που αναζητούμε θα εργαστεί στο Υποκατάστημα μας στην Θεσσαλονίκη και θα αναλάβει ως αρμοδιότητα την υποστήριξη δικτύου των συνεργατών και των πελατών μας στην Βόρεια Ελλάδα όσον αφορά τις υπηρεσίες δικτύων δηλαδή εγκατάσταση, διάρθρωση και λειτουργία βασικών δικτυακών υπηρεσιών. Τεχνικός και Μηχανικός Η/Υ - Δικτύων - Τηλεπικοινωνιών Η ΕΤΑΙΡΙΑ Μηχανογράφησης Bussiness Software, με έδρα την Θεσσαλονίκη ζητά Τεχνικό Υπολογιστών και Δικτύων, για το τεχνικό τμήμα της εταιρίας. Απαραίτητη προϋπηρεσία 2 ετών και δίπλωμα οδήγησης.	2/8/2014	σε εξέλιξη	-

Εικόνα 55: Ενημέρωση χρήστη για αλλαγή κατάστασης σε αίτηση

Πατώντας αιτήσεις χρήστης έχει την δυνατότητα να δει αναλυτικά όλες τις αιτήσεις που έχει υποβάλει τα σχόλια καθώς και την κατάσταση της κάθε αίτησής του.

The screenshot shows the user interface for 'user2@teiep.gr'. The page title is 'ΠΥΛΗ ΗΛΕΚΤΡΟΝΙΚΩΝ ΑΙΤΗΣΕΩΝ'. The user is logged in, with links for 'προφίλ', 'αιτήσεις', and 'αποσύνδεση'. A 'Αρχική' button is visible. The main heading is 'ΙΣΤΟΡΙΚΟ ΤΩΝ ΑΙΤΗΣΕΩΝ ΜΟΥ'. Below this is a table with columns: 'Θέση', 'Ημερ. αίτησης', 'Κατάσταση', and 'Σχόλια'. The first row shows a request for 'Διαχειριστής Δικτύου στην Megasoft A.E.' with a status of 'σε εξέλιξη'. The second row shows a request for 'Τεχνικός και Μηχανικός Η/Υ - Δικτύων - Τηλεπικοινωνιών' with a status of 'απορρίφθηκε'.

Θέση	Ημερ. αίτησης	Κατάσταση	Σχόλια
Διαχειριστής Δικτύου στην Megasoft A.E. Το άτομο που αναζητούμε θα εργαστεί στο Υποκατάστημα μας στην Θεσσαλονίκη και θα αναλάβει ως αρμοδιότητα την υποστήριξη δικτύου των συνεργατών και των πελατών μας στην Βόρεια Ελλάδα όσον αφορά τις υπηρεσίες δικτύων δηλαδή εγκατάσταση, διάρθρωση και λειτουργία βασικών δικτυακών υπηρεσιών. Τεχνικός και Μηχανικός Η/Υ - Δικτύων - Τηλεπικοινωνιών Η ΕΤΑΙΡΙΑ Μηχανογράφησης Bussiness Software, με έδρα την Θεσσαλονίκη ζητά Τεχνικό Υπολογιστών και Δικτύων, για το τεχνικό τμήμα της εταιρίας. Απαραίτητη προϋπηρεσία 2 ετών και δίπλωμα οδήγησης.	2/8/2014	σε εξέλιξη	-
Τεχνικός και Μηχανικός Η/Υ - Δικτύων - Τηλεπικοινωνιών Η ΕΤΑΙΡΙΑ Μηχανογράφησης Bussiness Software, με έδρα την Θεσσαλονίκη ζητά Τεχνικό Υπολογιστών και Δικτύων, για το τεχνικό τμήμα της εταιρίας. Απαραίτητη προϋπηρεσία 2 ετών και δίπλωμα οδήγησης.	2/8/2014	απορρίφθηκε	Η αίτησή σας δεν έχει επιλεγεί. Σας ευχαριστούμε πολύ

Εικόνα 56: Σελίδα έλεγχου κατάστασης αιτήσεων χρήστη

## 6. Παράρτημα

Στο παράρτημα την παρουσίασης θα δούμε αναλυτικά με επεξηγήσεις των κώδικα της εφαρμογής.

### 6.1 Συνοπτικός πίνακας σελίδων της εφαρμογής

Παρακάτω παρουσιάζεται ένας συνοπτικός πίνακας των σελίδων και των υπό-σελίδων που απαρτίζουν την εφαρμογή καθώς και μια σύντομη περιγραφή της λειτουργίας κάθε σελίδας και υπό-σελίδας.

#### Όνομασία σελίδας

#### Περιγραφή

**index.aspx**

Σε αυτή τη σελίδα εμφανίζει λίστα με τις αιτήσεις οι οποίες είναι διαθέσιμες καθώς και η επιλογή "όλες" μέσω της οποίας να δεις τις αιτήσεις που δεν είναι πλέον διαθέσιμες .

**view-position.aspx**

Είναι η σελίδα προβολή της θέσης. Στη σελίδα αυτή εμφανίζονται οι εξής πληροφορίες :

1. Στοιχεία θέσης
2. Προηγούμενες / τρέχουσες αιτήσεις χρήστη για τη θέση
3. Δυνατότητα να αιτηθώ για τη θέση συμπληρώνοντας τη φόρμα

**account.aspx**

Στη σελίδα αυτή μπορεί να εγγραφεί νέος χρήστης ή να συνδεθεί υπάρχον χρήστης . Είναι η login Page της εφαρμογής

**profile.aspx**

Η σελίδα διαχείρισης του προφίλ του χρήστη. Μέσα από αυτή τη σελίδα ο χρήστης μπορεί να ορίσει τις βασικές πληροφορίες του προφίλ του

*profile.aspx (sub page)* **edit-school.aspx**

Προσθήκη / επεξεργασία επιπέδου μόρφωσης στο προφίλ

*profile.aspx (sub page)* **edit-jobs.aspx**

Προσθήκη / επεξεργασία εργασιακής εμπειρίας στο προφίλ

*profile.aspx (sub page)* **edit-recommendations.aspx**

Προσθήκη / επεξεργασία συστάσεων στο προφίλ

**applications.aspx**

Η σελίδα εμφάνισης κατάστασης των αιτήσεων μου. Στη σελίδα αυτή ο χρήστης βλέπει την κατάσταση για τις αιτήσεις που έχει υποβάλλει

**modify-position.aspx**

Είναι η σελίδα διαχείρισης αιτήσεων. Παρέχει τις παρακάτω Δυνατότητες :

1. Δημιουργία νέας θέσης
2. Διαγραφή θέσης
3. Επεξεργασία θέσης
4. Προβολή αιτήσεων για μία θέση
5. Εξαγωγή αιτήσεων σε μορφή excel για μία θέση

## 6.2 index.aspx

Παρακάτω βλέπουμε τον κώδικα της αρχικής σελίδας της εφαρμογής

```
<%@ Page Title="Home Page" Language="VB" MasterPageFile="~/Site.Master" AutoEventWireup="false"
    CodeFile="index.aspx.vb" Inherits="_Default" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
    <h2>
        ΚΑΛΩΣΗΡΘΑΤΕ ΣΤΗΝ ΠΥΛΗ ΗΛΕΚΤΡΟΝΙΚΩΝ ΑΙΤΗΣΕΩΝ
    </h2>
    <p>
        Θέσεις για υποβολή αιτήσεων
        <div class="hsep5"></div>
        <asp:RadioButtonList ID="ShowOnly" runat="server" RepeatColumns="2"
            AutoPostBack="true">
            <asp:ListItem Value="0" Selected="True">μόνο ανοιχτές</asp:ListItem>
            <asp:ListItem Value="1">όλες</asp:ListItem>
        </asp:RadioButtonList>
    </p>
    <asp:ObjectDataSource ID="Positions" runat="server" EnablePaging="True"
        MaximumRowsParameterName="pageSize"
        SelectCountMethod="GetActiveItemCount" SelectMethod="GetActivePagedData"
        StartRowIndexParameterName="rowIndex"
        TypeName="Position" SortParameterName="sortExpression" />

    <asp:ListView ID="Results" runat="server" DataSourceID="Positions"
        ItemPlaceholderID="Item">
        <LayoutTemplate>
            <table>
                <asp:Placeholder ID="Item" runat="server" />
            </table>
            <div class="hsep10"></div>
            <asp:DataPager PageSize="10" runat="server" QueryStringField="index">
                <Fields>
                    <asp:NextPreviousPagerField ShowPreviousPageButton="true"
                        ShowNextPageButton="false" ShowLastPageButton="false"
                        FirstPageText="[ πρώτη ]" PreviousPageText="[ προηγούμενη ]" />
                    <asp:NumericPagerField />
                    <asp:NextPreviousPagerField ShowPreviousPageButton="false"
                        ShowNextPageButton="true" ShowLastPageButton="true"
                        NextPageText="[ επόμενη ]" LastPageText="[ τελευταία ]" />
                </Fields>
            </asp:DataPager>
        </LayoutTemplate>
        <ItemTemplate>
            <tr>
                <td width="20" valign="top"><asp:Literal ID="Index" runat="server"
                    /></td>
                <td>
                    <a href="view-position.aspx?id=<#Eval("Id")
                        %>"><#Eval("Name").ToString() %> (κωδικός: #<#Eval("Id").ToString()%)</a>
                    <div class="hsep2"></div>
                    <div style="color: #000000;">
                        <asp:Literal ID="FullText" runat="server" />
                    </div>
                    <div class="hsep5"></div>
                    <asp:Literal id="Period" runat="server" />
                </td>
            </tr>
        </ItemTemplate>
    </asp:ListView>
</asp:Content>
```



```

                </td>
            </tr>
        </ItemTemplate>
        <EmptyDataTemplate>
        Δεν υπάρχουν ανοιχτές θέσεις
        </EmptyDataTemplate>
    </asp:ListView>
</asp:Content>

```

Γυρνάμε στην index.aspx.vb για να δούμε πως εκμεταλλευόμαστε τα περιεχόμενα της index.aspx

```

Imports System.Data.Linq
Imports System.Data.Linq.Mapping

Partial Class _Default
    Inherits System.Web.UI.Page

    ''' <summary>
    ''' Τρέχουσα ώρα
    ''' </summary>
    ''' <remarks></remarks>
    Dim now As DateTime = DateTime.Now

    ''' <summary>
    ''' </summary>
    ''' <remarks></remarks>
    Protected Overrides Sub OnLoad(ByVal e As System.EventArgs)
        MyBase.OnLoad(e)

        If Not Page.IsPostBack Then Me.CreatePageDataView()
    End Sub

    ''' <summary>
    ''' Εμφάνιση λίστας με τις θέσεις ανάλογα με την επιλογή του χρήστη
    (ανοιχτές ή όλες)
    ''' </summary>
    ''' <remarks></remarks>
    Private Sub CreatePageDataView()
        Dim view = Request("view")
        If view = "1" Then
            Me.ShowOnly.SelectedIndex = 1
        Else
            Me.ShowOnly.SelectedIndex = 0
        End If

        If Me.ShowOnly.SelectedIndex = 1 Then
            Me.Positions.SelectMethod = "GetPagedData"
            Me.Positions.SelectCountMethod = "GetItemCount"
        End If
    End Sub

    ''' <summary>
    ''' Ρουτίνα που καλείται όταν ο χρήστης επιλέξει να δει όλες τις θέσεις ή
    τις ανοιχτές μόνο. Κάνει re-direct η σελίδα στον εαυτό της με άλλη παράμετρο στο
    url
    ''' </summary>
    ''' <param name="sender"></param>
    ''' <param name="e"></param>
    ''' <remarks></remarks>
    Protected Sub ShowOnly_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles ShowOnly.SelectedIndexChanged
        Response.Redirect(String.Format("{0}?view={1}", Request.CurrentExecutionFilePath,
Me.ShowOnly.SelectedIndex), True)
    End Sub

```

```

''' <summary>
''' Ρουτίνα που καλείται όταν δημιουργείται το item που εμφανίζει τη θέση
για να μορφοποιήσουμε το item όπως επιθυμούμε (ενδεικτικά σχόλια σε γραμμές
παρακάτω)
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>
''' <remarks></remarks>
Protected Sub Results_ItemCreated(ByVal sender As Object, ByVal e As
System.Web.UI.WebControls.ListViewItemEventArgs) Handles Results.ItemCreated
    Dim item = DirectCast(e.Item.DataItem, Position)
    If Not item Is Nothing Then
        CType(e.Item.FindControl("Index"), Literal).Text = e.Item.DataItemIndex + 1

        Dim period As Literal = DirectCast(e.Item.FindControl("Period"), Literal)
        Dim open As Boolean = (now >= item.FromDate AndAlso now <= item.ToDate)
        period.Text = String.Format("περίοδος αιτήσεων από {0} έως {1} -",
item.FromDate.ToString("d"), item.ToDate.ToString("d")) 'Κείμενο που δείχνει την περίοδο
αιτήσεων σε ανοιχτή θέση
        If item.MinimumEducation.HasValue Then
            Using Context = New GraduateDataContext()
                Dim type = Context.SchoolTypes.First(Function(q) q.Id =
item.MinimumEducation.Value)
                period.Text &= " (" & type.Name & " και άνω) - "
            End Using
        End If

        period.Text &= String.Format("<strong>{0}</strong>", If(open, "ανοιχτό",
"κλειστό")) 'Κείμενο που δείχνει αν η θέση είναι ανοιχτή ή όχι για υποβολή
αιτήσεων

        Dim text As Literal = DirectCast(e.Item.FindControl("FullText"), Literal)
        text.Text = item.Description
    End If
End Sub

End Class

```

## 6.3 view-position.aspx

Ο χρήστης έρχεται στη συγκεκριμένη σελίδα όταν επιλέξει μια συγκεκριμένη θέση στην index.aspx από εδώ μπορεί να συγκεντρωτικά τα στοιχεία για την προσφερόμενη θέση

```

<%@ Page Title="" Language="VB" MasterPageFile="~/Site.master" AutoEventWireup="false"
CodeFile="view-position.aspx.vb" Inherits="view_position" %>

<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" Runat="Server">
    <h2>ΠΡΟΒΟΛΗ ΘΕΣΗΣ</h2>
    <div class="hsep10"></div>
    <hr />
    <div class="hsep5"></div>
    <asp:ValidationSummary ID="LoginUserValidationSummary" runat="server"
CssClass="failureNotification"
        ValidationGroup="Position"/>
    <table>
        <tr>
            <td>Κωδικός / ονομασία θέσης:</td>
            <td><strong><asp:Literal ID="Name" runat="server" /></strong></td>
        </tr>
        <tr>
            <td>Περίοδος αιτήσεων:</td>
            <td><asp:Literal ID="Period" runat="server" /></td>
        </tr>
    </table>

```

```

        <tr>
            <td>Περιγραφή:</td>
            <td><asp:Literal ID="Description" runat="server" /></td>
        </tr>
</table>

<asp:PlaceHolder ID="NotLoggedIn" runat="server" Visible="false">
<div class="hsep10"></div>
Για να κάνετε αίτηση για τη θέση πρέπει να συνδεθείτε. <a
href="account/account.aspx?returnUrl=<%=server.urlencode(Request.RawUrl &
If(Request.RawUrl.Contains("?"), "&", "?") & "submit=yes") %>">Πατήστε εδώ για να συνδεθείτε</a>
</asp:PlaceHolder>

<asp:PlaceHolder ID="History" runat="server" Visible="false" ViewStateMode="Disabled">
<div class="hsep10"></div>
<asp:ListView ID="HistoryView" runat="server" ItemPlaceholderID="Item">
<LayoutTemplate>
    <table>
        <tr class="row-h">
            <th>Ημερ. αίτησης</th>
            <th>Σχόλια αίτησης</th>
            <th>Κατάσταση</th>
            <th>Λόγος απόρριψης</th>
        </tr>
        <asp:PlaceHolder ID="Item" runat="server" />
    </table>
</LayoutTemplate>
<ItemTemplate>
    <tr id="Row" runat="server">
        <td></td>
        <td></td>
        <td></td>
        <td></td>
    </tr>
</ItemTemplate>
</asp:ListView>
</asp:PlaceHolder>

<asp:PlaceHolder ID="NoEducation" runat="server" Visible="false">
<div class="hsep10"></div>
για να αιτηθίτε για τη συγκεκριμένη θέση πρέπει το επίπεδο εκπαίδευσης σας να είναι
τουλάχιστον <strong><asp:Literal ID="Minimum" runat="server" /></strong>
</asp:PlaceHolder>

<asp:PlaceHolder ID="Closed" runat="server" Visible="false">
<div class="hsep10"></div>
δεν δεχόμαστε αιτήσεις για την συγκεκριμένη θέση πλέον.
</asp:PlaceHolder>

<asp:PlaceHolder ID="AlreadySubmitted" runat="server" Visible="false">
    <div class="hsep10"></div>
<strong>έχετε ήδη μία αίτηση για τη συγκεκριμένη θέση σε εξέλιξη.</strong>
</asp:PlaceHolder>

<asp:PlaceHolder ID="PlaceHolder1" runat="server" Visible="false">
<div class="hsep10"></div>
<table>
    <tr>
        <td></td>
    </tr>
</table>
</asp:PlaceHolder>

<asp:PlaceHolder ID="CreateApplication" runat="server" Visible="false">
<div class="hsep10"></div>
<table>
    <tr valign="top">
        <td>Σχόλια:</td>
        <td></td>
    </tr>
</table>

```

```

        <asp:TextBox ID="Comments" runat="server" TextMode="MultiLine"
Columns="50" Rows="5"></asp:TextBox>
        <asp:RequiredFieldValidator ID="CommentsRequired" runat="server"
ControlToValidate="Comments"
Text="*"
ValidationGroup="Position"
Display="None"
ErrorMessage="συμπληρώστε τα σχόλια" />
    </td>
</tr>
<tr>
<td>&nbsp;</td>
<td>
        <asp:Button ID="Send" runat="server" Text="υποβολή αίτησης"
ValidationGroup="Position" />
    </td>
</tr>
</table>
</asp:Placeholder>

</asp:Content>

```

Παρακάτω θα εξετάσουμε τον κώδικα της view-position.aspx.vb μέσω της οποίας ενεργούμε στην view-position.aspx.

```

Partial Class view_position
    Inherits System.Web.UI.Page

    ''' <summary>
    ''' Προβολή μιας συγκεκριμένης θέσης, αν δεν υπάρχει θέση για προβολή τότε
    επιστροφή στην αρχική σελίδα
    ''' </summary>
    ''' <param name="sender"></param>
    ''' <param name="e"></param>
    ''' <remarks></remarks>
    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load
        Dim value As String = Request("id")
        Dim id As Integer
        If Not String.IsNullOrEmpty(value) AndAlso Int32.TryParse(value, id) Then
            Me.ShowPosition()
        Else
            Response.Redirect("~/index.aspx", True)
        End If
    End Sub

    ''' <summary>
    ''' Εμφάνιση της θέσης
    ''' Αν ο χρήστης είναι logged in στο σύστημα τότε εμφανίζει την κατάσταση
    του σχετικά με τη θέση
    ''' 1. Προηγούμενες αιτήσεις
    ''' 2. Κουμπί για να αιτηθεί αν του επιτρέπεται βάση επιπέδου μορφωτικού
    επιπέδου του έχει δηλωθεί στο profile '''
    ''' αν δεν είναι logged in τον παραπέμπει στο να συνδεθεί ή να εγγραφεί
    και να γυρίσει για να υποβάλει την αίτησή του '''
    </summary>
    ''' <remarks></remarks>
    Private Sub ShowPosition()
        Dim id As Integer = Convert.ToInt32(Request("id"))
        Using db = New GraduateDataContext()
            Dim ps = db.Positions.SingleOrDefault(Function(item) item.Id = id)
            Me.Name.Text = String.Format("#{0} - {1}", ps.Id, ps.Name)
            Me.Period.Text = String.Format("από {0} έως {1}", ps.FromDate.ToString("d"),
ps.ToDate.ToString("d"))
            Me.Description.Text = ps.Description
        End Using
    End Sub

```

```

Dim now = DateTime.Now
Dim open As Boolean = (now >= ps.FromDate AndAlso now <= ps.ToDate)

Me.NotLoggedIn.Visible = False
Me.Closed.Visible = False
Me.CreateApplication.Visible = False

If Not My.User.IsAuthenticated Then
    Me.NotLoggedIn.Visible = True
Else
    If Not open Then
        Me.Closed.Visible = True
    Else
        Dim userId As Guid = New
Guid(Membership.GetUser().ProviderUserKey.ToString())

        Dim canSubmit As Boolean = True
        If ps.MinimumEducation.HasValue Then
            Dim education = (From item In db.Educations Where
item.UserId = userId AndAlso item.TypeOfSchool.Value >= ps.MinimumEducation).ToList()
            If education.Count > 0 Then
                canSubmit = True
            Else
                canSubmit = False
            End If
        End If

        If Not canSubmit Then
            Me.NoEducation.Visible = True
            Me.CreateApplication.Visible = False

            Dim type = db.SchoolTypes.First(Function(q) q.Id =
ps.MinimumEducation.Value)

            Me.Minimum.Text = type.Name
        Else
            'Dim already =
db.PositionRequests.Where(Function(item) item.PositionId = ps.Id andalso item.UserId = userId
andalso Not item.Accepted.HasValue

            Dim hv = (From position In db.PositionRequests
Where position.PositionId = ps.Id AndAlso
position.UserId = userId Order By position.DateCreated Descending).ToList()

            Dim approved = hv.AsQueryable().Count(Function(item)
item.Accepted.HasValue AndAlso item.Accepted.Value)
            Dim pending = hv.AsQueryable().Count(Function(item)
Not item.Accepted.HasValue)

            If hv.Count > 0 Then
                Me.History.Visible = True
                Me.HistoryView.DataSource = hv
                Me.HistoryView.DataBind()
            End If

            If pending = 0 Then
                Me.CreateApplication.Visible = True

                If Not Page.IsPostBack AndAlso
Request("submit") = "yes" Then
                    Me.Send_Click(Me.Send, EventArgs.Empty)
                End If
            Else
                Me.AlreadySubmitted.Visible = True
            End If
        End If
    End If
End If

End If

```

## 6.4 account.aspx

Στην account.aspx γίνεται η διαπίστευση του χρήστη με το σύστημα. Όπως έχουμε προαναφέρει και στο κεφάλαιο 4, σε αυτή την σελίδα πραγματοποιούνται και όλοι οι απαραίτητοι έλεγχοι ορθότητας στοιχείων εγγραφής εισόδου. Παρακάτω ο κώδικάς της σελίδας.

```
<%@ Page Title="" Language="VB" MasterPageFile="~/Site.master" AutoEventWireup="false"
CodeFile="Account.aspx.vb" Inherits="Account_Account" %>

<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" Runat="Server">

    <table>
        <tr valign="top">
            <td id="register"> <asp:CreateUserWizard ID="RegisterUser" runat="server"
EnableViewState="false"
                RequireEmail="false"
                InvalidPasswordErrorMessage="Ο κωδικός πρόσβασης δεν είναι αποδεκτός.
Πρέπει να είναι τουλάχιστον {0} χαρακτήρες"
                DuplicateUserNameErrorMessage="Ο κωδικός χρήστη υπάρχει ήδη. Επιλέξτε
ένα άλλο email για εγγραφή ή κάντε είσοδο σαν υπάρχον χρήστης">
                <LayoutTemplate>
                    <asp:Placeholder ID="wizardStepPlaceholder" runat="server"></asp:Placeholder>
                    <asp:Placeholder ID="navigationPlaceholder" runat="server"></asp:Placeholder>
                </LayoutTemplate>
                <WizardSteps>
                    <asp:CreateUserWizardStep ID="RegisterUserWizardStep" runat="server">

                        <ContentTemplate>
                            <h2>
                                ΝΕΟΣ ΧΡΗΣΤΗΣ ;
                            </h2>
                            <p>
                                Συμπληρώστε τη φόρμα που ακολουθεί για να δημιουργήσετε νέο λογαριασμό.
                            </p>
                            <p>
                                Ο κωδικός πρόσβασης πρέπει να είναι τουλάχιστον <%=
Membership.MinRequiredPasswordLength %> χαρακτήρες.
                            </p>
                            <span class="failureNotification">
                                <asp:Literal ID="ErrorMessage" runat="server"></asp:Literal>
                            </span>
                            <asp:ValidationSummary ID="RegisterUserValidationSummary" runat="server"
CssClass="failureNotification"
                                ValidationGroup="RegisterUserValidationGroup"/>
                            <div class="accountInfo">
                                <fieldset class="register">
                                    <legend>Account Information</legend>
                                    <p>
                                        <asp:Label ID="UserNameLabel" runat="server"
AssociatedControlID="UserName">E-mail:</asp:Label>
                                        <asp:TextBox ID="UserName" runat="server"
CssClass="textEntry"></asp:TextBox>
                                        <asp:RequiredFieldValidator ID="UserNameRequired" runat="server"
ControlToValidate="UserName"
                                            CssClass="failureNotification" ErrorMessage="to e-mail είναι
υποχρεωτικό" ToolTip="to e-mail είναι υποχρεωτικό."
ValidationGroup="RegisterUserValidationGroup">*</asp:RequiredFieldValidator>
                                        <asp:RegularExpressionValidator
ID="EmailRegexValidator" runat="server"
                                            ValidationGroup="RegisterUserValidationGroup"

```



```
  |
```



```

        </p>
    </div>
</LayoutTemplate>
</asp:Login>
</td>

</tr>
</table>

<script type="text/javascript">
    $('#register').keydown(function(event)
    {
        if(event.keyCode == 13 && event.target.tagName == 'INPUT' &&
(event.target.type == 'text' || event.target.type == 'password'))
        {
            $('#MainContent_RegisterUser_CreateUserStepContainer_CreateUserButton').focus();
        }
    });

    $('#login').keydown(function(event)
    {
        if(event.keyCode == 13 && event.target.tagName == 'INPUT' &&
(event.target.type == 'text' || event.target.type == 'password'))
        {
            $('#MainContent_LoginUser_LoginButton').focus();
        }
    });
</script>
</asp:Content>

```

Παρακάτω θα δούμε συνοπτικά την σελίδα account.aspx.vb μέσω της οποίας χρησιμοποιείται η account.aspx και γίνονται τα ανάλογα redirections

```

partial Class Account_Account
    Inherits System.Web.UI.Page

    ''' <summary>
    ''' Ορίζουμε το URL στο οποίο θα επιστρέψει ο χρήστης μόλις εγγραφεί AN
    εγγραφεί
    ''' </summary>
    ''' <param name="sender"></param>
    ''' <param name="e"></param>
    ''' <remarks></remarks>
    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load
        RegisterUser.ContinueDestinationPageUrl = "~/profile.aspx?returnUrl=" &
Server.UrlEncode(Request.QueryString("ReturnUrl"))
    End Sub

    ''' <summary>
    ''' Αν γίνει επιτυχώς η εγγραφή αποθηκεύουμε ένα cookie με το username του
    χρήστη και στέλνουμε τον χρήστη στο url που ορίσαμε στο Page_Load
    ''' </summary>
    ''' <param name="sender"></param>
    ''' <param name="e"></param>
    ''' <remarks></remarks>
    Protected Sub RegisterUser_CreatedUser(ByVal sender As Object, ByVal e As EventArgs)
Handles RegisterUser.CreatedUser
        FormsAuthentication.SetAuthCookie(RegisterUser.UserName, False)

        Dim continueUrl As String = RegisterUser.ContinueDestinationPageUrl
        If String.IsNullOrEmpty(continueUrl) Then
            continueUrl = "~/

```

```

        End If

        Response.Redirect(continueUrl)
    End Sub

    ''' <summary>
    ''' Αν είναι επιτυχημένο το login τότε στέλνουμε τον χρήστη στο url που
ορίσαμε στο page_load αν υπάρχει, αλλιώς επιστροφή στην αρχική σελίδα
    ''' </summary>
    ''' <param name="sender"></param>
    ''' <param name="e"></param>
    ''' <remarks></remarks>
    Protected Sub LoginUser_LoggedIn(sender As Object, e As EventArgs) Handles
LoginUser.LoggedIn
        Dim url As String = RegisterUser.ContinueDestinationPageUrl
        If String.IsNullOrEmpty(url) Then
            Response.Redirect("~/index.aspx", True)
        Else
            Response.Redirect(url, True)
        End If
    End Sub
End Class

```

## 6.5 profile.aspx

Η profile.aspx είναι η σελίδα που παρουσιάζει τα απαραίτητα πεδία προκειμένου ο χρήστης της εφαρμογής να αποθηκεύσει/αλλάξει/δεί τα προσωπικά του στοιχεία/βιογραφικό του.

```

<%@ Page Title="" Language="VB" MasterPageFile="~/Site.master" AutoEventWireup="false"
CodeFile="profile.aspx.vb" Inherits="Account_profile" %>

<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" Runat="Server">
    <h2 id="Section" runat="server">
        επεξεργασία προφιλ
    </h2>
    <div class="hsep10"></div>
    <asp:ValidationSummary ID="LoginUserValidationSummary" runat="server"
CssClass="failureNotification"
        ValidationGroup="Profile"/>
    <table>
        <tr>
            <td class="label">Όνομα</td>
            <td class="value">
                <asp:TextBox ID="FirstName" runat="server" ValidationGroup="Profile"
Width="200px" onchange="set_changed();" />
                <asp:RequiredFieldValidator ID="FirstNameRequired" runat="server"
                    ControlToValidate="FirstName"
                    ValidationGroup="Profile"
                    Text="*"
                    ErrorMessage="συμπληρώστε το όνομα σας" />
            </td>
        </tr>
        <tr>
            <td class="label">Επώνυμο</td>
            <td class="value">
                <asp:TextBox ID="LastName" runat="server" ValidationGroup="Profile"
Width="200px" onchange="set_changed();" />
                <asp:RequiredFieldValidator ID="LastNameRequired" runat="server"
                    ControlToValidate="LastName"
                    ValidationGroup="Profile"
                    Text="*"
                    ErrorMessage="συμπληρώστε το επώνυμο σας" />
            </td>
        </tr>
    </table>

```

```

<tr>
  <td class="label">E-mail</td>
  <td class="value">
    <asp:Label ID="Email" runat="server" />
  </td>
</tr>
<tr>
  <td class="label">Τηλέφωνο</td>
  <td class="value">
    <asp:TextBox ID="Phone" runat="server" ValidationGroup="Profile"
onchange="set_changed();" />
    <asp:RequiredFieldValidator ID="PhoneFieldRequired" runat="server"
      ControlToValidate="Phone"
      ValidationGroup="Profile"
      Text="*"
      ErrorMessage="συμπληρώστε το τηλέφωνο σας" />
    <asp:RegularExpressionValidator ID="PhoneRegexValidator"
runat="server"
      ValidationGroup="Profile"
      ControlToValidate="Phone"
      ValidationExpression="^((\+|00)?)[0-9]{10,25}$"
      ErrorMessage="συμπληρώστε σωστά το τηλέφωνο"
      Display="Dynamic"
      SetFocusOnError="true" />
  </td>
</tr>
<tr>
  <td class="label">Ημερ. γέννησης</td>
  <td class="value">
    <asp:Textbox ID="BirthDate" runat="server" ValidationGroup="Profile"
onchange="set_changed();" />
    <asp:RequiredFieldValidator ID="BirthDateRequiredValidator"
runat="server"
      ControlToValidate="BirthDate"
      ValidationGroup="Profile"
      Text="*"
      ErrorMessage="συμπληρώστε την ημερομηνία γέννησης σας" />
  </td>
</tr>
<tr>
  <td class="label">Διεύθυνση</td>
  <td class="value">
    <asp:Textbox ID="Address" runat="server"
      ValidationGroup="Profile"
      TextMode="Multiline"
onchange="set_changed();"
      Rows="2"
      Columns="50" />
    <asp:RequiredFieldValidator ID="AddressRequired" runat="server"
      ControlToValidate="Address"
      ValidationGroup="Profile"
      Text="*"
      ErrorMessage="συμπληρώστε την διεύθυνση σας" />
  </td>
</tr>
<tr>
  <td>Εκπαίδευση</td>
  <td>
    <asp:ListView ID="StudiesList" runat="server"
ItemPlaceholderID="Item">
      <LayoutTemplate>
        <table width="500">
          <tr class="row-h">
            <th>σχολή</th>
            <th>περίοδος</th>
            <th>-</th>
          </tr>
          <asp:PlaceHolder ID="Item" runat="server" />
        </table>
      </LayoutTemplate>
    </td>
</tr>

```

```

        </LayoutTemplate>
        <ItemTemplate>
            <tr id="Row" runat="server">
                <td></td>
                <td></td>
                <td>
                    <%--<a href="edit-
school.aspx?id=<#Eval("Id").ToString() %">επεξεργασία</a>--%>
                    <asp:LinkButton runat="server"
CommandName="edit" CommandArgument='<#Eval("Id")%>' Text="επεξεργασία" />
                    &nbsp;
                    <%--<a href="edit-
school.aspx?id=<#Eval("Id").ToString() %">&action=delete" onclick="return confirm('είσαι σίγουρος
/η ;')">διαγραφή</a>--%>
                    <asp:LinkButton runat="server"
CommandName="delete" CommandArgument='<#Eval("Id")%>' OnClientClick="return confirm('είσαι
σίγουρος /η ;')" Text="διαγραφή" />
                </td>
            </tr>
        </ItemTemplate>
        <EmptyDataTemplate><i>δεν έχει καταχωρηθεί εκπαίδευση.</i></EmptyDataTemplate>
    </asp:ListView>
    <asp:PlaceHolder ID="AddStudies" runat="server">
        <%--[ <a href="edit-school.aspx">προσθήκη</a> ]--%>
        [ <asp:LinkButton ID="EditSchool" runat="server" CommandArgument="edit-
school.aspx">προσθήκη</asp:LinkButton> ]
    </asp:PlaceHolder>
</td>
</tr>
<tr>
<td colspan="2">&nbsp;</td>
</tr>
<tr>
<td class="label">Χώρα</td>
<td class="value">
    <asp:DropDownList ID="Country" runat="server"
onchange="set_changed();" />
    <asp:RequiredFieldValidator ID="RequiredFieldValidator1"
runat="server"
ControlToValidate="Address"
ValidationGroup="Profile"
Text="*"
ErrorMessage="συμπληρώστε την διεύθυνση σας" />
</td>
</tr>
<tr>
<td class="label">Εθνικότητα</td>
<td class="value">
    <asp:Textbox ID="Nationality" runat="server"
onchange="set_changed();"
ValidationGroup="Profile" />
    <asp:RequiredFieldValidator ID="NationalityRequired" runat="server"
ControlToValidate="Nationality"
ValidationGroup="Profile"
Text="*"
ErrorMessage="συμπληρώστε την εθνικότητα σας" />
</td>
</tr>
<tr>
<td class="label">Οικογενειακή κατάσταση</td>
<td class="value">
    <asp:RadioButtonList ID="MaritalStatus" runat="server"
RepeatColumns="2">
        <asp:ListItem Value="0" Selected="True"
onclick="set_changed();">Άγαμος</asp:ListItem>
        <asp:ListItem Value="1"
onclick="set_changed();">Παντρεμένος</asp:ListItem>
    </asp:RadioButtonList>
</td>
</tr>

```



```

</tr>
<tr>
  <td class="label">Twitter</td>
  <td class="value">
    <asp:Textbox ID="Twitter" runat="server" onchange="set_changed();"
      ValidationGroup="Profile" />
  </td>
</tr>

<tr>
  <td class="label">Δραστηριότητες</td>
  <td class="value">
    <asp:Textbox ID="Hobbies" runat="server"
      ValidationGroup="Profile"
      TextMode="MultiLine"
      onchange="set_changed();"
      Rows="3"
      Columns="50" />
  </td>
</tr>

<tr>
  <td colspan="2">&nbsp;</td>
</tr>

<tr>
  <td class="label">Συστάσεις</td>
  <td class="value">
    <asp:ListView ID="RecommendationsList" runat="server"
      ItemPlaceholderID="Item">
      <LayoutTemplate>
        <table>
          <tr class="row-h">
            <th>εταιρεία</th>
            <th>ονοματεπώνυμο</th>
            <th>e-mail</th>
            <th>-</th>
          </tr>
          <asp:PlaceHolder ID="Item" runat="server" />
        </table>
      </LayoutTemplate>
      <ItemTemplate>
        <tr id="Row" runat="server">
          <td></td>
          <td></td>
          <td></td>
          <td>
            <%--<a href="edit-
recommendation.aspx?id=<#Eval("Id").ToString() %>">επεξεργασία</a>--%>
            <asp:LinkButton runat="server"
              CommandName="edit" CommandArgument='<#Eval("Id")%>' Text="επεξεργασία" />
            &nbsp;  
            <%--<a href="edit-
recommendation.aspx?id=<#Eval("Id").ToString() %>&action=delete" onclick="return confirm('είσαι
σίγουρος /η ;')">διαγραφή</a>--%>
            <asp:LinkButton runat="server"
              CommandName="delete" CommandArgument='<#Eval("Id")%>' OnClientClick="return confirm('είσαι
σίγουρος /η ;')" Text="διαγραφή" />
          </td>
        </tr>
      </ItemTemplate>
      <EmptyDataTemplate><i>δεν έχουν καταχωρηθεί συστάσεις.</i></EmptyDataTemplate>
    </asp:ListView>
    <asp:PlaceHolder ID="AddRecommendation" runat="server">
      [ <asp:LinkButton ID="EditRecommendation" runat="server"
        CommandArgument="edit-recommendation.aspx" Text="προσθήκη" /> ]
    </asp:PlaceHolder>
  </td>
</tr>

```

```

        <tr>
            <td colspan="2">&nbsp;</td>
        </tr>

<tr id="Update1" runat="server">
    <td colspan="2" align="left">
        <span style="color: green;">το προφιλ ενημερώθηκε επιτυχώς</span>
    </td>
</tr>
<tr id="Update2" runat="server">
    <td colspan="2">&nbsp;</td>
</tr>

        <tr>
            <td>&nbsp;</td>
            <td>
                <asp:Button ID="Save" runat="server"
                    ValidationGroup="Profile"
                    Text="αποθήκευση" />
                &nbsp;<asp:Button ID="SubmitApplication" runat="server"
                    ValidationGroup="Profile" Visible="false"
                    Text="υποβολή αίτησης" />
                <asp:Button ID="ReturnBack" runat="server"
                    CausesValidation="false"
                    Visible="false"
                    Text="επιστροφή" />
            </td>
        </tr>
</table>

<script>

    var changed = false;

    $.datepicker.setDefaults($.datepicker.regional['el']);
    $('input[id$=BirthDate]').datepicker();

    function set_changed()
    {
        changed = true;
    }

</script>

</asp:Content>

```

Η συμπλήρωση των στοιχείων στην φόρμα της profile.aspx πραγματοποιείται με την βοήθεια της profile.aspx.vb. Παρακάτω ο κώδικας με σχόλια στα βασικότερα σημεία.

```

Partial Class Account_profile
    Inherits System.Web.UI.Page

    Dim admin As Boolean = Me.IsAdmin()
    Dim other As Boolean

    ''' <summary>
    ''' Αν δεν έχει γίνει υποβολή της σελίδας τότε καλείται η ρουτίνα LoadPage
    ''' Γενικώς το Page.IsPostBack σημαίνει ότι πατήθηκε κάποιο κουμπί και
    έχει γίνει submit η σελίδα προς τον server. Ανάλογα εκτελούμε actions ή όχι '''
    </summary>
    ''' <remarks></remarks>
    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load
        If Not Page.IsPostBack Then Me.LoadData()

```

```

        Me.SubmitApplication.Visible = Not String.IsNullOrEmpty(Request("returnUrl"))
    End Sub

    ''' <summary>
    ''' Εμφάνιση του profile του χρήστη.
    ''' Αν ο χρήστης είναι διαχειριστής και υπάρχει στο url η παράμετρος e-
    mail τότε φορτώνει το profile του χρήστη που ζήτησε να δει ο διαχειριστής χωρίς
    τη δυνατότητα επεξεργασίας
    ''' περισσότερα εσωτερικά '''
</summary>
    ''' <remarks></remarks>
    Private Sub LoadData()
        Dim userName As String = Me.GetUserName()

        Me.other = (Not String.IsNullOrEmpty(userName) AndAlso admin)
        Me.Email.Text = My.User.Name

        Using db = New GraduateDataContext()
            Γέμισα λίστας χωρών
            Me.Country.DataSource = db.Countries
            Me.Country.DataTextField = "Name"
            Me.Country.DataValueField = "Id"
            Me.Country.DataBind()
            Me.Country.Items.Insert(0, New ListItem("-", "0"))

            If Not other Then userName = My.User.Name
            Dim user = Membership.GetUser(userName)
            Dim userId As Guid = New Guid(user.ProviderUserKey.ToString())

            Me.Email.Text = user.UserName

            Φόρτωση λίστας μόρφωσης
            Me.StudiesList.DataSource = From item In db.Educations
            Where item.UserId = userId
            Order By item.DateTo Descending
            Me.StudiesList.DataBind()

            Φόρτωση λίστας εργασιακής εμπειρίας
            Dim jobs = (From item In db.Jobs
            Where item.UserId = userId
            Order By If(item.ToDate.HasValue, item.ToDate.Value, DateTime.Now)
            Descending).ToList()
            Me.JobsList.DataSource = jobs
            Me.JobsList.DataBind()

            Φόρτωση λίστας συστάσεων
            Me.RecommendationsList.DataSource = db.Recommendations.Where(Function(item)
            item.UserId = userId).OrderByDescending(Function(item) item.DateUpdated)
            Me.RecommendationsList.DataBind()

            Βασικά πεδία προφίλ
            Dim profile = db.UserProfiles.FirstOrDefault(Function(u) u.UserId = userId)
            If Not profile Is Nothing Then
                Me.FirstName.Text = profile.FirstName
                Me.LastName.Text = profile.LastName
                Me.Phone.Text = profile.Phone
                Me.BirthDate.Text = profile.BirthDate.ToString("dd/MM/yyyy")
                Me.Address.Text = profile.Address
                Me.Country.SelectedValue = profile.Country
                Me.Nationality.Text = profile.Nationality
                Me.MaritalStatus.SelectedIndex = If(profile.PersonalStatus.Value = 0,
0, 1)

                Me.Website.Text = profile.WebSite
                Me.Facebook.Text = profile.Facebook
                Me.Twitter.Text = profile.Twitter
                Me.Hobbies.Text = profile.Hobbies

```



```

End If

Me.Update1.Visible = (Request("saved") = "true")
Me.Update2.Visible = (Request("saved") = "true")
End Using

```

Αν βλέπει άλλο προφίλ ο διαχειριστής τότε απενεργοποιούμε την τροποποίηση

```

If other Then
    Me.FirstName.ReadOnly = True
    Me.FirstName.CssClass = "view"

    Me.LastName.ReadOnly = True
    Me.LastName.CssClass = "view"

    Me.Phone.ReadOnly = True
    Me.Phone.CssClass = "view"

    Me.BirthDate.ReadOnly = True
    Me.BirthDate.CssClass = "view"

    Me.Address.ReadOnly = True
    Me.Address.CssClass = "view"

    Me.AddStudies.Visible = False

    Me.Country.Enabled = False
    Me.Country.CssClass = "view"

    Me.Nationality.ReadOnly = True
    Me.Nationality.CssClass = "view"

    Me.MaritalStatus.Enabled = False
    Me.AddJob.Visible = False

    Me.Website.ReadOnly = True
    Me.Website.CssClass = "view"

    Me.Facebook.ReadOnly = True
    Me.Facebook.CssClass = "view"

    Me.Twitter.ReadOnly = True
    Me.Twitter.CssClass = "view"

    Me.Hobbies.ReadOnly = True
    Me.Hobbies.CssClass = "view"

    Me.AddRecommendation.Visible = False
    Me.Save.Visible = False

    If Not String.IsNullOrEmpty("position") Then
        Me.ReturnBack.Visible = True
    End If

    Me.Section.InnerHtml = "ΠΡΟΒΟΛΗ ΠΡΟΦΙΛ"
Else

```

'Αν ο χρήστης έχει έρθει από υποσέλιδα προφίλ (όπως επεξεργασίας εργασιακής εμπειρίας, συστάσεις κλπ) τότε γεμίσουμε τα πεδία με τις τιμές που είχαν πριν πατήσει στον σύνδεσμό για να μεταφερθεί στην υποσελίδα και τα νέα στοιχεία που έχει προσθέσει/επεξεργαστεί/διαγράψει στην υποσελίδα'

```

If Not Session("FirstName") Is Nothing Then Me.FirstName.Text =
Session("FirstName")
If Not Session("LastName") Is Nothing Then Me.LastName.Text =
Session("LastName")
If Not Session("Phone") Is Nothing Then Me.Phone.Text = Session("Phone")

```

```

Session("BirthDate") If Not Session("BirthDate") Is Nothing Then Me.BirthDate.Text =
Session("Address") If Not Session("Address") Is Nothing Then Me.Address.Text =
Session("Address") If Not Session("Country") Is Nothing Then Me.Country.SelectedValue =
Session("Nationality") If Not Session("Nationality") Is Nothing Then Me.Nationality.Text =
Session("Marital") If Not Session("Marital") Is Nothing Then Me.MaritalStatus.SelectedValue =
Session("Website") If Not Session("Website") Is Nothing Then Me.Website.Text =
Session("Facebook") If Not Session("Facebook") Is Nothing Then Me.Website.Text =
Session("Twitter") If Not Session("Twitter") Is Nothing Then Me.Website.Text =
Session("Hobbies") If Not Session("Hobbies") Is Nothing Then Me.Hobbies.Text =
Session("Country") If Not Session("Country") Is Nothing Then Me.Country.SelectedValue =

If Not Session("ReturnUrl") Is Nothing Then
    Dim returnUrl = Session("ReturnUrl")
    If Not String.IsNullOrEmpty(returnUrl) Then
        Session.Remove("ReturnUrl")

        Dim targetUrl = Request.RawUrl &
If(Request.RawUrl.Contains("?"), "&", "?") & "returnUrl=" & Server.UrlEncode(returnUrl)
        Response.Redirect(targetUrl)
    End If
End If

ClearSession()
End If
End Sub

```

## 6.6 edit-school.aspx

Η συγκεκριμένη σελίδα αποτελεί sub-page της profile.aspx και αφορά όπως είναι εύκολα αντιληπτό στην επεξεργασία του μορφωτικού επιπέδου του χρήστη .

```

<%@ Page Title="" Language="VB" MasterPageFile="~/Site.master" AutoEventWireup="false"
CodeFile="edit-school.aspx.vb" Inherits="edit_school" %>

<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" Runat="Server">
<h2>ΠΡΟΣΘΗΚΗ / ΕΠΕΞΕΡΓΑΣΙΑ ΕΚΠΑΙΔΕΥΣΗΣ</h2>
<div class="hsep10"></div>
<hr />
<div class="hsep10"></div>
<asp:ValidationSummary ID="LoginUserValidationSummary" runat="server"
CssClass="failureNotification"
ValidationGroup="School"/>
<table>
<tr>
<td>Τύπος σχολής:</td>
<td>
<asp:DropDownList ID="TypeOfSchool" runat="server"
AutoPostBack="true"></asp:DropDownList>
<asp:RequiredFieldValidator ID="TypeOfSchoolValidator" runat="server"
InitialValue="0" ValidationGroup="School" Text="*"
ControlToValidate="TypeOfSchool"
ErrorMessage="επιλέξτε τον τύπο της σχολής"
CssClass="failureNotification" />

```

```

        </td>
</tr>
<asp:PlaceHolder ID="TypeAB" runat="server" Visible="false">
<tr id="SchoolName" runat="server" visible="false">
    <td>Σχολή:</td>
    <td>
        <asp:DropDownList ID="School" runat="server"
AutoPostBack="true"></asp:DropDownList>
        <asp:RequiredFieldValidator ID="SchoolValidator" runat="server"
            InitialValue="0" ValidationGroup="School" Text="*"
            ControlToValidate="School"
            ErrorMessage="επιλέξτε τη σχολή"
            CssClass="failureNotification" />
    </td>
</tr>
<tr id="SectionName" runat="server" visible="false">
    <td>Τμήμα:</td>
    <td>
        <asp:DropDownList ID="Section" runat="server"></asp:DropDownList>
        <asp:RequiredFieldValidator ID="SectionValidator" runat="server"
            InitialValue="0" ValidationGroup="School" Text="*"
            ControlToValidate="Section"
            ErrorMessage="επιλέξτε το τμήμα"
            CssClass="failureNotification" />
    </td>
</tr>
</asp:PlaceHolder>
<asp:PlaceHolder ID="TypeCD" runat="server" Visible="false">
<tr>
    <td>Σχόλια</td>
    <td>
        <asp:TextBox ID="Comments" runat="server" Width="300px"
            ValidationGroup="School" />
        <asp:RequiredFieldValidator ID="NameFieldRequired" runat="server"
            ControlToValidate="Comments"
            ValidationGroup="School"
            Text="*"
    </td>
</tr>
</asp:PlaceHolder>

```

Ας εξετάσουμε τον κώδικα της edit-school.aspx.vb για να δούμε πως λειτουργεί.

```

Partial Class edit_school
    Inherits System.Web.UI.Page

    ''' <summary>
    ''' Αν δεν έχει γίνει υποβολή της σελίδας τότε καλείται η ρουτίνα LoadPage
    ''' Γενικώς το Page.IsPostBack σημαίνει ότι πατήθηκε κάποιο κουμπί και
    έχει γίνει submit η σελίδα προς τον server. Ανάλογα εκτελούμε actions ή όχι
    ''' </summary>
    ''' <remarks></remarks>
    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load
        If Not Page.IsPostBack Then Me.LoadData()
    End Sub

    ''' <summary>
    ''' Φορτώνει την επιλεγμένη από την account.aspx εγγραφή εκπαίδευσης αν
    έχουμε ζητήσει από το profile να δούμε μια συγκεκριμένη αλλιώς εμφανίζει κενή
    φόρμα για καταχώρηση
    ''' </summary>
    ''' <remarks></remarks>
    Private Sub LoadData()
        Dim now = DateTime.Now
        Using context = New GraduateDataContext()
            Dim selected = Request("id")

```

```

Dim id As Integer
Dim userId = New Guid(Membership.GetUser().ProviderUserKey.ToString())

'fill schoools
With (Me.TypeOfSchool)
    .DataSource = context.SchoolTypes.ToList()
    .DataTextField = "Name"
    .DataValueField = "Id"
    .DataBind()

    .Items.Insert(0, New ListItem("- επιλέξτε -", "0"))
End With

If Not String.IsNullOrEmpty(selected) AndAlso Int32.TryParse(selected, id)
Then
    Dim rc = context.Educations.SingleOrDefault(Function(item) item.Id =
id AndAlso item.UserId = userId)

    Me.Comments.Text = rc.School
    Me.StartDate.Text = rc.DateFrom.ToString("d")
    Me.EndDate.Text = rc.DateTo.ToString("d")

    If rc.TypeOfSchool.HasValue Then
        Me.TypeOfSchool.SelectedValue = rc.TypeOfSchool
    End If

    Me.TypeOfSchool_SelectedIndexChanged(Me.TypeOfSchool,
EventArgs.Empty)

    If rc.TypeOfSchool > 2 Then
        Dim section =
context.SchoolSections.FirstOrDefault(Function(q) q.Id = rc.SectionId.Value)
        Dim school = section.School

        Me.School.SelectedValue = school.Id.ToString()
        Me.School_SelectedIndexChanged(Me.School, EventArgs.Empty)

        Me.Section.SelectedValue = section.Id.ToString()
    End If

    If Request("action") = "delete" Then
        context.Educations.DeleteOnSubmit(rc)
        context.SubmitChanges()

        Response.Redirect("~/profile.aspx", True)
    End If
End If
End Using
End Sub

''' <summary>
''' Αν έχουμε έρθει πατώντας στο profile σε εγγραφή εκπαίδευσης τότε
ενημερώνονται τα στοιχεία της, αλλιώς γίνεται αποθήκευση της νέας και επιστροφή
στο profile
''' </summary>
''' <remarks></remarks>
Protected Sub Save_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Save.Click, SaveAndReturnToProfile.Click
    Page.Validate("School")
    If Page.IsValid Then
        Using db = New GraduateDataContext()
            Dim rc As Education
            Dim selected = Request("id")
            Dim id As Integer
            Dim userId = New
Guid(Membership.GetUser().ProviderUserKey.ToString())

            If Not String.IsNullOrEmpty(selected) AndAlso
Int32.TryParse(selected, ID) Then

```

```

rc = db.Educations.SingleOrDefault(Function(item) item.Id = id
Also item.UserId = userId)
Else
rc = New Education()
rc.DateCreated = DateTime.Now
rc.UserId = userId

db.Educations.InsertOnSubmit(rc)
End If

Dim type = Convert.ToInt32(Me.TypeOfSchool.SelectedValue)
rc.TypeOfSchool = type

If type > 2 Then
rc.SectionId = Convert.ToInt32(Me.Section.SelectedValue)
Else
rc.School = Me.Comments.Text
End If

rc.DateFrom = Convert.ToDateTime(Me.StartDate.Text)
rc.DateTo = Convert.ToDateTime(Me.EndDate.Text)

rc.DateUpdated = DateTime.Now

db.SubmitChanges()

If sender Is SaveAndReturnToProfile Then
Response.Redirect("~/profile.aspx", True)
Else
Me.MessageRow.Cells(1).Style("color") = "green"
Me.MessageRow.Cells(1).InnerHTML = "η αποθήκευση έγινε
επιτυχώς"
End If
End Using
End If
End Sub

''' <summary>
''' Ρουτίνα που εκτελείται όταν αλλάζει το επίπεδο μόρφωσης και αντίστοιχα
φορτώνει την επόμενη λίστα
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>
''' <remarks></remarks>
Protected Sub TypeOfSchool_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
TypeOfSchool.SelectedIndexChanged
Me.SchoolName.Visible = False
Me.SectionName.Visible = False

If Me.TypeOfSchool.SelectedIndex > 0 Then
Me.SchoolName.Visible = True

Dim value = Convert.ToInt32(Me.TypeOfSchool.SelectedValue)
If value > 2 Then
Me.TypeAB.Visible = True
Me.TypeCD.Visible = False

Using db = New GraduateDataContext()
Dim schools = From item In db.Schools Where item.SchoolTypeId
= Me.TypeOfSchool.SelectedValue
With (Me.School)
.DataSource = schools.ToList()
.DataValueField = "Id"
.DataTextField = "Name"
.DataBind()

.Items.Insert(0, New ListItem("- επιλέξτε -"))
End With
End Using

```

```

        Else
            Me.TypeAB.Visible = False
            Me.TypeCD.Visible = True
        End If
    End If
End Sub

''' <summary>
''' Ρουτίνα που φορτώνεται όταν αλλάζει η σχολή και φορτώνει την επόμενη
λίστα
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>
''' <remarks></remarks>
Protected Sub School_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
School.SelectedIndexChanged
    Me.SectionName.Visible = False

    If Me.School.SelectedIndex > 0 Then
        Me.SectionName.Visible = True

        Using db = New GraduateDataContext()
            Dim schools = From item In db.SchoolSections Where item.SchoolId =
Me.School.SelectedValue
                With (Me.Section)
                    .DataSource = schools.ToList()
                    .DataValueField = "Id"
                    .DataTextField = "Name"
                    .DataBind()
                End With
            End Using
        End If
    End Sub

''' <summary>
''' Ακύρωση, καμία αλλαγή και επιστροφή στο προφίλ
''' </summary>
''' <remarks></remarks>
Protected Sub Cancel_Click(sender As Object, e As System.EventArgs) Handles Cancel.Click
    Response.Redirect("~/profile.aspx", True)
End Sub

End Class

```

## 6.7 edit-jobs.aspx

Η συγκεκριμένη σελίδα αποτελεί sub-page της profile.aspx και αφορά στις εγγραφές εργασιακής εμπειρίας του χρήστη. Λειτουργεί ακριβώς όπως και η παραπάνω edit-school.aspx. Παρακάτω ο κώδικας

```

<%@ Page Title="" Language="VB" MasterPageFile="~/Site.master" AutoEventWireup="false"
CodeFile="edit-job.aspx.vb" Inherits="edit_job" %>

<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" Runat="Server">
<h2>ΠΡΟΣΘΗΚΗ / ΕΠΕΞΕΡΓΑΣΙΑ ΕΡΓΑΣΙΑΚΗΣ ΕΜΠΕΙΡΙΑΣ</h2>
<div class="hsep10"></div>
<hr />
<div class="hsep10"></div>
<asp:ValidationSummary ID="LoginUserValidationSummary" runat="server"
CssClass="failureNotification"
ValidationGroup="Job"/>

```

```

<table>
  <tr>
    <td>Επωνυμία εταιρείας</td>
    <td>
      <asp:TextBox Id="CompanyName" runat="server" Width="200px"
ValidationGroup="Job" />
      <asp:RequiredFieldValidator ID="CompanyNameRequired" runat="server"
ControlToValidate="CompanyName"
Text="*"
ValidationGroup="Job"
CssClass="failureNotification"
ErrorMessage="συμπληρώστε την επωνυμία της εταιρείας" />
    </td>
  </tr>
  <tr>
    <td>Θέση</td>
    <td>
      <asp:TextBox Id="JobTitle" runat="server" Width="200px"
ValidationGroup="Job" />
      <asp:RequiredFieldValidator ID="JobTitleFieldRequired" runat="server"
ControlToValidate="JobTitle"
Text="*"
CssClass="failureNotification"
ValidationGroup="Job"
ErrorMessage="συμπληρώστε τον τίτλο" />
    </td>
  </tr>
  <tr>
    <td>Περιγραφή</td>
    <td>
      <asp:TextBox Id="Description" runat="server" TextMode="MultiLine"
Columns="50" Rows="5" ValidationGroup="Job" />
      <asp:RequiredFieldValidator ID="DescriptionRequired" runat="server"
ControlToValidate="Description"
Text="*"
CssClass="failureNotification"
ValidationGroup="Job"
ErrorMessage="συμπληρώστε την περιγραφή" />
    </td>
  </tr>
  <tr>
    <td class="label">Από:</td>
    <td class="value">
      <asp:TextBox ID="StartDate" runat="server"
ValidationGroup="Job"
width="100px" />
      <asp:RequiredFieldValidator ID="StartDateRequired" runat="server"
ValidationGroup="Job"
ControlToValidate="StartDate"
Display="None"
Text="*"
ErrorMessage="συμπληρώστε την ημερομηνία έναρξης" />
      <asp:CompareValidator ID="StartDateDataValidator" runat="server"
ControlToValidate="StartDate"
ValidationGroup="Position"
Operator="DataTypeCheck"
Text="*"
Display="None"
ErrorMessage="συμπληρώστε σωστά την ημερομηνία έναρξης
εργασίας"
Type="Date" />
    </td>
  </tr>
  <tr>
    <td class="label">Έως:</td>
    <td class="value">
      <asp:TextBox ID="EndDate" runat="server"
ValidationGroup="Job"
width="100px" />
    </td>
  </tr>

```

```

        <asp:RequiredFieldValidator ID="EndDateRequired" runat="server"
            ValidationGroup="Job"
            ControlToValidate="EndDate"
            Display="None"
            Enabled="false"
            Text="*"
            ErrorMessage="συμπληρώστε την ημερομηνία λήξης" />
        <asp:CompareValidator ID="EndDateDataValidator" runat="server"
            ControlToValidate="EndDate"
            ValidationGroup="Job"
            Operator="DataTypeCheck"
            Display="None"
            ErrorMessage="συμπληρώστε σωστά την ημερομηνία λήξης εργασίας"
            Type="Date" />
        <asp:CompareValidator ID="StartDateEndDateCompare" runat="server"
            ControlToValidate="StartDate"
            ControlToCompare="EndDate"
            ValidationGroup="Job"
            Operator="LessThanEqual"
            Text="*"
            Display="None"
            ErrorMessage="η ημερομηνία λήξης πρέπει να είναι μεγαλύτερη
της ημερομηνίας έναρξης"
            Type="Date" />
    </td>
</tr>
<tr id="MessageRow" runat="server">
    <td></td>
    <td></td>
</tr>
<tr>
    <td>&nbsp;</td>
    <td>
        <asp:Placeholder runat="server" Visible="false">
        <asp:Button ID="Save" runat="server"
            ValidationGroup="Job"
            Text="αποθήκευση" />
        &nbsp;<br>
        </asp:Placeholder>
        <asp:Button ID="SaveAndReturnToProfile" runat="server"
            ValidationGroup="Job"
            Text="αποθήκευση και επιστροφή στο προφίλ" />
        &nbsp;<br>
        <asp:Button ID="Cancel" runat="server" Text="ακύρωση & επιστροφή στο
προφίλ" />
    </td>
</tr>
</table>
<script language="javascript" type="text/javascript">
    $.datepicker.setDefaults($.datepicker.regional['el']);
    $('input[id$=StartDate],input[id$=EndDate]').datepicker(
    {
        showOn: 'both'
        ,changeMonth: true
        ,changeYear: true
        ,buttonImageOnly: true
        ,buttonImage: 'images/calendar.gif'
    });
</script>
</asp:Content>

```

Παρακάτω θα δούμε τα actions που πραγματοποιεί στη συγκεκριμένη σελίδα η edit-job.aspx.vb.

Partial Class edit\_job



```

Inherits System.Web.UI.Page

''' <summary>
''' Η σελίδα αυτή δείχνει την επιλεγμένη εργασιακή εμπειρία ή καταχωρεί
νέα
'''
''' Αν δεν έχει γίνει υποβολή της σελίδας τότε καλείται η ρουτίνα LoadPage
''' Γενικώς το Page.IsPostBack σημαίνει ότι πατήθηκε κάποιο κουμπί και
έχει γίνει submit η σελίδα προς τον server. Ανάλογα εκτελούμε actions ή όχι
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>
''' <remarks></remarks>
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load
    If Not Page.IsPostBack Then Me.LoadData()
End Sub

''' <summary>
''' Φορτώνει την επιλεγμένη εργασιακή εμπειρία αν έχουμε ζητήσει από το
profile να δούμε μια συγκεκριμένη αλλιώς εμφανίζει κενή φόρμα για καταχώρηση νέας
εγγραφής
''' </summary>
''' <remarks></remarks>
Private Sub LoadData()
    Dim now = DateTime.Now
    Using context = New GraduateDataContext()
        Dim selected = Request("id")
        Dim id As Integer
        Dim userId = New Guid(Membership.GetUser().ProviderUserKey.ToString())

        If Not String.IsNullOrEmpty(selected) AndAlso Int32.TryParse(selected, id)
Then
            Dim rc = context.Jobs.SingleOrDefault(Function(item) item.Id = id
AndAlso item.UserId = userId)
            Me.CompanyName.Text = rc.CompanyName
            Me.JobTitle.Text = rc.Title
            Me.Description.Text = rc.Description
            Me.StartDate.Text = rc.FromDate.ToString("d")
            If rc.ToDate.HasValue Then Me.EndDate.Text = rc.ToDate.Value.ToString("d")

            If Request("action") = "delete" Then
                context.Jobs.DeleteOnSubmit(rc)
                context.SubmitChanges()

                Response.Redirect("~/profile.aspx", True)
            End If
        End If
    End Using
End Sub

''' <summary>
''' Αν έχουμε έρθει πατώντας στο profile σε επιλεγμένη εργασιακή εμπειρία
τότε ενημερώνονται τα στοιχεία της, αλλιώς γίνεται αποθήκευση της νέας και
επιστροφή στο profile
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>
''' <remarks></remarks>
Protected Sub Save_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Save.Click, SaveAndReturnToProfile.Click
    Page.Validate("Recommendation")
    If Page.IsValid Then
        Using db = New GraduateDataContext()
            Dim rc As Job
            Dim selected = Request("id")
            Dim id As Integer

```

```

        Dim userId = New
Guid(Membership.GetUser().ProviderUserKey.ToString())

        If Not String.IsNullOrEmpty(selected) AndAlso
Int32.TryParse(selected, ID) Then
            rc = db.Jobs.SingleOrDefault(Function(item) item.Id = id
AndAlso item.UserId = userId)
        Else
            rc = New Job()
            rc.DateCreated = DateTime.Now
            rc.UserId = userId

            db.Jobs.InsertOnSubmit(rc)
        End If

        rc.CompanyName = Me.CompanyName.Text
        rc.Title = Me.JobTitle.Text
        rc.Description = Me.Description.Text
        rc.FromDate = Convert.ToDateTime(Me.StartDate.Text)
        If Not String.IsNullOrEmpty(Me.EndDate.Text) Then
            rc.ToDate = Convert.ToDateTime(Me.EndDate.Text)
        Else
            rc.ToDate = Nothing
        End If

        rc.DateUpdated = DateTime.Now

        db.SubmitChanges()

        If sender Is SaveAndReturnToProfile Then
            Response.Redirect("~/profile.aspx", True)
        Else
            Me.MessageRow.Cells(1).Style("color") = "green"
            Me.MessageRow.Cells(1).InnerHtml = "η αποθήκευση έγινε
επιτυχώς"
        End If
    End Using
End If
End Sub

''' <summary>
''' Ακύρωση, καμία αλλαγή και επιστροφή στο προφιλ
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>
''' <remarks></remarks>
Protected Sub Cancel_Click(sender As Object, e As System.EventArgs) Handles Cancel.Click
    Response.Redirect("~/profile.aspx", True)
End Sub

End Class

```

## 6.8 edit-recommendations.aspx

Η συγκεκριμένη σελίδα επίσης αποτελεί sub-page της profile.aspx και αφορά σε εγγραφές του χρήστη σχετικά με διαθέσιμες συστάσεις. Λειτουργεί όπως και edit-school.aspx αλλά είναι αρκετά πιο απλή σελίδα.

```
<%@ Page Title="" Language="VB" MasterPageFile="~/Site.master" AutoEventWireup="false"
CodeFile="edit-recommendation.aspx.vb" Inherits="edit_recommendations" %>

<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" Runat="Server">
<h2>ΚΑΤΑΧΩΡΗΣΗ / ΕΠΕΞΕΡΓΑΣΙΑ ΣΥΣΤΑΣΗΣ</h2>
<div class="hsep10"></div>
<hr />
<div class="hsep10"></div>
    <asp:ValidationSummary ID="LoginUserValidationSummary" runat="server"
    CssClass="failureNotification"
        ValidationGroup="Recommendation"/>
<table>
    <tr>
        <td>Επωνυμία εταιρείας:</td>
        <td>
            <asp:TextBox ID="CompanyName" runat="server"
    ValidationGroup="Recommendation" Width="200px"></asp:TextBox>
            <asp:RequiredFieldValidator ID="CompanyNameRequired" runat="server"
                ControlToValidate="CompanyName"
                ValidationGroup="Recommendation"
                CssClass="failureNotification"
                Text="*"
                ErrorMessage="συμπληρώστε την επωνυμία της εταιρείας" />
        </td>
    </tr>
    <tr>
        <td>Όνοματεπώνυμο:</td>
        <td>
            <asp:TextBox ID="FullName" runat="server" ValidationGroup="Recommendation"
    Width="200px"></asp:TextBox>
            <asp:RequiredFieldValidator ID="FullNameRequired" runat="server"
                ControlToValidate="FullName"
                ValidationGroup="Recommendation"
                CssClass="failureNotification"
                Text="*"
                ErrorMessage="συμπληρώστε το ονοματεπώνυμο" />
        </td>
    </tr>
    <tr>
        <td>E-mail:</td>
        <td>
            <asp:TextBox ID="Email" runat="server" ValidationGroup="Recommendation"
    Width="200px"></asp:TextBox>
            <asp:RequiredFieldValidator ID="EmailFieldRequired" runat="server"
                ControlToValidate="Email"
                ValidationGroup="Recommendation"
                CssClass="failureNotification"
                Text="*"
                ErrorMessage="συμπληρώστε το e-mail" />
            <asp:RegularExpressionValidator ID="EmailRegexValidator" runat="server"
                ValidationGroup="Recommendation"
                ControlToValidate="Email"
                CssClass="failureNotification"
                Display="Dynamic"
                ValidationExpression="^([a-zA-Z0-9_\-\ ]+)(\.[a-zA-Z0-9_\-\ ]+)*@(\[[\ ]*\]|([0-9]{1,3}\.){0,3}[0-9]{1,3})\.(\[[\ ]*\]|([0-9]{1,3}\.){0,3}[0-9]{1,3})\.(([0-9]{1,3}\.){0,3}[0-9]{1,3})|([0-9]{1,3}\.){0,3}[0-9]{1,3})" />
        </td>
    </tr>
</table>
```

```

9])?[0-9])|(2[0-4][0-9])|(2[0-5][0-5]))\))|((([a-zA-Z0-9]+(([\-])+([a-zA-Z0-9])+\.)+([a-zA-Z])+(([\-])+([a-zA-Z0-9])+)*)$"
        ErrorMessage="συμπληρώστε σωστά το e-mail"
        ToolTip="συμπληρώστε σωστά το e-mail"
        Text="*"
        SetFocusOnError="true" />
    </td>
</tr>
<tr id="MessageRow" runat="server">
    <td></td>
    <td></td>
</tr>
<tr>
    <td>&nbsp;</td>
    <td>
        <asp:Placeholder runat="server" Visible="false">
        <asp:Button ID="Save" runat="server"
            ValidationGroup="Recommendation"
            Text="αποθήκευση" />
        &nbsp;  
        </asp:Placeholder>
        <asp:Button ID="SaveAndReturnToProfile" runat="server"
            ValidationGroup="Recommendation"
            Text="αποθήκευση και επιστροφή στο προφίλ" />
        &nbsp;  
        <asp:Button ID="Cancel" runat="server" Text="ακύρωση & επιστροφή στο προφίλ" />
    </td>
</tr>
</table>
</asp:Content>

```

Ο χειρισμός της σελίδας αυτής πραγματοποιείται από την edit-recommendations.aspx.vb.

```

Partial Class edit_recommendations
    Inherits System.Web.UI.Page

```

```

    ''' <summary>
    ''' αν δεν έχει γίνει υποβολή της σελίδας τότε καλείται η ρουτίνα LoadPage
    ''' Γενικώς το Page.IsPostBack σημαίνει ότι πατήθηκε κάποιο κουμπί και
    έχει γίνει submit η σελίδα προς τον server. Ανάλογα εκτελούμε actions ή όχι
    ''' </summary>
    ''' <param name="sender"></param>
    ''' <param name="e"></param>
    ''' <remarks></remarks>
    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load
        If Not Page.IsPostBack Then Me.LoadData()
    End Sub

    ''' <summary>
    ''' Φορτώνει την επιλεγμένη σύσταση εργασίας στην περίπτωση που έχουμε
    ζητήσει από το profile να δούμε μια συγκεκριμένη εγγραφή αλλιώς εμφανίζει κενή
    φόρμα για καταχώρηση νέας
    ''' </summary>
    ''' <remarks></remarks>
    Private Sub LoadData()
        Dim now = DateTime.Now
        Using context = New GraduateDataContext()
            Dim selected = Request("id")
            Dim id As Integer
            Dim userId = New Guid(Membership.GetUser().ProviderUserKey.ToString())

```

```

        If Not String.IsNullOrEmpty(selected) AndAlso Int32.TryParse(selected, id)
Then
        Dim rc = context.Recommendations.SingleOrDefault(Function(item)
item.Id = id AndAlso item.UserId = userId)
        Me.CompanyName.Text = rc.CompanyName
        Me.FullName.Text = rc.FullName
        Me.Email.Text = rc.Email

        If Request("action") = "delete" Then
            context.Recommendations.DeleteOnSubmit(rc)
            context.SubmitChanges()

            Response.Redirect("~/profile.aspx", True)
        End If
    End If
End Using
End Sub

''' <summary>
''' Αν έχουμε έρθει πατώντας στο profile σε επιλεγμένη σύσταση εργασίας
τότε ενημερώνονται τα στοιχεία της, αλλιώς γίνεται αποθήκευση της νέας και
επιστροφή στο profile
''' </summary>
''' <remarks></remarks>
Protected Sub Save_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Save.Click, SaveAndReturnToProfile.Click
    Page.Validate("Recommendation")
    If Page.IsValid Then
        Using db = New GraduateDataContext()
            Dim rc As Recommendation
            Dim selected = Request("id")
            Dim id As Integer
            Dim userId = New
Guid(Membership.GetUser().ProviderUserKey.ToString())

            If Not String.IsNullOrEmpty(selected) AndAlso
Int32.TryParse(selected, ID) Then
                rc = db.Recommendations.SingleOrDefault(Function(item) item.Id
= id AndAlso item.UserId = userId)
            Else
                rc = New Recommendation()
                rc.DateCreated = DateTime.Now
                rc.UserId = userId

                db.Recommendations.InsertOnSubmit(rc)
            End If

            rc.CompanyName = Me.CompanyName.Text
            rc.FullName = Me.FullName.Text
            rc.Email = Me.Email.Text
            rc.DateUpdated = DateTime.Now

            db.SubmitChanges()

            If sender Is SaveAndReturnToProfile Then
                Response.Redirect("~/profile.aspx", True)
            Else
                Me.MessageRow.Cells(1).Style("color") = "green"
                Me.MessageRow.Cells(1).InnerHtml = "η αποθήκευση έγινε
επιτυχώς"
            End If
        End Using
    End If
End S
''' <summary>
''' Ακύρωση, καμία αλλαγή και επιστροφή στο προφίλ
''' </summary>
''' <remarks></remarks>
Protected Sub Cancel_Click(sender As Object, e As System.EventArgs) Handles Cancel.Click

```

```

        Response.Redirect("~/profile.aspx", True)
    End Sub
ub
End Class

```

## 6.9 applications.aspx

Η σελίδα εμφάνισης κατάστασης των αιτήσεων μου. Στη σελίδα αυτή ο χρήστης βλέπει την κατάσταση για τις αιτήσεις που έχει υποβάλει.

```

<%@ Page Title="" Language="VB" MasterPageFile="~/Site.master" AutoEventWireup="false"
CodeFile="applications.aspx.vb" Inherits="applications" %>

<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" Runat="Server">
<h2>ΙΣΤΟΡΙΚΟ ΤΩΝ ΑΙΤΗΣΕΩΝ ΜΟΥ</h2>
<div class="hsep10"></div>
<hr />
<div class="hsep5"></div>

<asp:ListView ID="HistoryView" runat="server" ItemPlaceholderID="Item">
    <LayoutTemplate>
        <table width="500">
            <tr class="row-h">
                <th>Θέση</th>
                <th>Ημερ. αίτησης</th>
                <th>Κατάσταση</th>
                <th>Σχόλια</th>
            </tr>
            <asp:PlaceHolder ID="Item" runat="server" />
        </table>
    </LayoutTemplate>
    <ItemTemplate>
        <tr id="Row" runat="server">
            <td></td>
            <td></td>
            <td></td>
            <td></td>
        </tr>
    </ItemTemplate>
    <EmptyDataTemplate>
        δεν έχετε κάνει καμία αίτηση ως τώρα.
    </EmptyDataTemplate>
</asp:ListView>
</asp:Content>

```

Ο χειρισμός της παραπάνω σελίδας πραγματοποιείται από την applications.aspx.vb. Παρακάτω ο κώδικας με σχόλια των ενεργειών.

```

Partial Class applications
    Inherits System.Web.UI.Page

```

```

        ''' <summary>
        ''' Δεν πραγματοποιεί κάποιο action, το Page_Load είναι το entry point σε
        κάθε σελίδα
        ''' </summary>
        ''' <param name="sender"></param>
        ''' <param name="e"></param>
        ''' <remarks></remarks>
        Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load

```

```

        Me.CreateHistoryView()
    End Sub

    ''' <summary>
    ''' Το παρακάτω κομμάτι κώδικά, φέρνει τις αιτήσεις του χρήστη από τη βάση
    δεδομένων και δημιουργεί τη λίστα
    ''' </summary>
    ''' <remarks></remarks>
    Private Sub CreateHistoryView()
        Using db = New GraduateDataContext()
            Dim userId = New Guid(Membership.GetUser().ProviderUserKey.ToString())

            Dim positions = From item In db.PositionRequests
                Where item.UserId = userId Order By item.DateCreated Descending

            Me.HistoryView.DataSource = positions
            Me.HistoryView.DataBind()

            db.ExecuteCommand("UPDATE dbo.PositionRequest SET Notify = 2 WHERE Accepted IS NOT
            NULL AND UserId = {0}", userId)

            db.SubmitChanges()
        End Using
    End Sub

    ''' <summary>
    ''' Ρουτίνα που καλείται όταν δημιουργείται το item της λίστας και
    τροποποιεί το item ώστε να το κάνουμε όπως επιθυμούμε (επιπλέον σχόλια σε
    γραμμές παρακάτω)
    ''' </summary>
    ''' <param name="sender"></param>
    ''' <param name="e"></param>
    ''' <remarks></remarks>
    Protected Sub HistoryView_ItemCreated(ByVal sender As Object, ByVal e As
    System.Web.UI.WebControls.ListViewItemEventArgs) Handles HistoryView.ItemCreated
        Dim item As PositionRequest = DirectCast(e.Item.DataItem, PositionRequest)
        If Not item Is Nothing Then
            Dim row As HtmlTableRow = DirectCast(e.Item.FindControl("row"),
            HtmlTableRow)
            If Not row Is Nothing Then
                Using Context = New GraduateDataContext()
                    Dim position As Position = Context.Positions.First(Function(p) p.Id =
                    item.PositionId)
                    row.Cells(0).InnerHtml = position.Name & "<br />" &
                    position.Descripition 'π.χ: δείχνουμε την ονομασία μαζί με τη θέση
                    row.Cells(1).InnerHtml = item.DateCreated.ToString("d")
                    'ημερομηνία σε μορφή μερα/μηνας/ετος ωρα:λεπτα
                    row.Cells(3).InnerHtml = "-"
                End Using

                If item.Accepted.HasValue Then
                    If item.Accepted.Value Then
                        row.Cells(2).InnerHtml = "εγκρίθηκε"
                        row.Cells(3).InnerHtml = "συγχαρητήρια έχετε επιλεγεί για τη συγκεκριμένη
                        θέση. Επικοινωνήστε μαζί μας το συντομότερο"
                    Else
                        row.Cells(2).InnerHtml = "απορρίφθηκε"
                        row.Cells(3).InnerHtml = item.RejectionReason
                    End If
                Else
                    row.Cells(2).InnerHtml = "σε εξέλιξη"
                End If
            End If
        End If
    End Sub

End Class

```

## 6.10 modify-position.aspx

Τέλος θα δούμε την σελίδα modify-position.aspx που είναι η σελίδα διαχείρισης για τον administrator του συστήματος και παρέχει τις εξής δυνατότητες .

- Δημιουργία νέας θέσης
- Διαγραφή θέσης
- Επεξεργασία θέσης
- Προβολή αιτήσεων για μία θέση
- Εξαγωγή αιτήσεων σε μορφή excel για μία θέση

```
<%@ Page Title="" Language="VB" MasterPageFile="~/Site.master" AutoEventWireup="false"
CodeFile="modify-position.aspx.vb" Inherits="modify_position" %>

<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" Runat="Server">
  <h2>
    ΕΠΕΞΕΡΓΑΣΙΑ ΘΕΣΕΩΝ
  </h2>
  <div class="hsep10"></div>
  επιλογή θέσης <asp:DropDownList Id="CurrentPositions" runat="server" AutoPostBack="true" />
  ή <a href="modify-position.aspx">δημιουργία νέας θέσης</a>
  <div class="hsep5"></div>
  <hr />
  <div class="hsep5"></div>
  <asp:ValidationSummary ID="LoginUserValidationSummary" runat="server"
  CssClass="failureNotification"
  ValidationGroup="Position"/>
  <table>
    <tr id="CodeContainerRow" runat="server" visible="false" viewstatemode="Disabled">
      <td class="label">Κωδικός:</td>
      <td class="value"><asp:Label ID="Code" runat="server" /></td>
    </tr>
    <tr>
      <td class="label">Όνομασία:</td>
      <td class="value">
        <asp:TextBox ID="Name" runat="server" ValidationGroup="Position"
        Width="400px" />
        <asp:RequiredFieldValidator ID="NameFieldRequired" runat="server"
        ValidationGroup="Position"
        ControlToValidate="Name"
        Text="*"
        ErrorMessage="συμπληρώστε την ονομασία" />
      </td>
    </tr>
    <tr>
      <td class="label">Περιγραφή:</td>
      <td class="value">
        <asp:TextBox ID="Description" runat="server"
        TextMode="Multiline"
        Rows="4"
        ValidationGroup="Position"
        Columns="50" />
        <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
        ValidationGroup="Position"
        ControlToValidate="Name"
        Text="*"
        ErrorMessage="συμπληρώστε την περιγραφή" />
      </td>
    </tr>
  </table>
</asp:Content>
```





```

                <asp:Button ID="Save" runat="server"
                    Text="αποθήκευση"
                    ValidationGroup="Position" />
            </td>
        </tr>
    </table>
    <br />

    <asp:PlaceHolder ID="All" runat="server" Visible="false">

    <h2>
        ΑΙΤΗΣΕΙΣ ΓΙΑ ΤΗ ΘΕΣΗ
    </h2>
    <hr />
    <asp:LinkButton ID="ShowColumnPicker" runat="server" Text="εξαγωγή δεδομένων σε excel (εμφάνιση)"
/>
<br /><br />

<asp:PlaceHolder ID="ColumnsContainer" runat="server" Visible="false">

    <asp:CheckBoxList ID="Columns" runat="server" RepeatLayout="Table" RepeatColumns="4">
        <asp:ListItem Value="0" Text="Όνοματεπώνυμο" Selected="true"/>
        <asp:ListItem Value="1" Text="Ημερομηνία υποβολής" Selected="true" />
        <asp:ListItem Value="2" Text="Κατάσταση" Selected="true" />
        <asp:ListItem Value="3" Text="Σχόλια" Selected="true" />
        <asp:ListItem Value="4" Text="E-mail" Selected="true" />
        <asp:ListItem Value="5" Text="Τηλέφωνο" Selected="true" />
        <asp:ListItem Value="6" Text="Ημερ. Γέννησης" Selected="true" />
        <asp:ListItem Value="7" Text="Διεύθυνση" Selected="true" />
        <asp:ListItem Value="8" Text="Εκπαίδευση" Selected="true" />
        <asp:ListItem Value="9" Text="Χώρα" />
        <asp:ListItem Value="10" Text="Εθνικότητα" />
        <asp:ListItem Value="11" Text="Οικογενειακή κατάσταση" />
        <asp:ListItem Value="12" Text="Εργασιακή εμπειρία" />
    </asp:CheckBoxList>
    <br />
    <asp:Button ID="Export" runat="server" Text="εξαγωγή" />
    <br /><br />
</asp:PlaceHolder>

<asp:ListView ID="Applications" runat="server" ItemPlaceholderID="Item">
<LayoutTemplate>

<table width="700">
<tr class="row-h">
    <th>Όνοματεπώνυμο</th>
    <th>Ημέρ/νία υποβολής</th>
    <th>Σχόλια αίτησης</th>
    <th>Κατάσταση</th>
    <th>Λόγος έγκρισης / απόρριψης</th>
    <th>Ενέργεια</th>
</tr>
<asp:PlaceHolder ID="Item" runat="server" />
</table>
</LayoutTemplate>
<ItemTemplate>
<tr id="Row" runat="server">
    <td></td>
    <td></td>
    <td></td>
    <td></td>
    <td>-</td>
    <td align="center">
        <a href="javascript:" onclick="_approve(this);">έγκριση</a> | <a href="javascript:"
onclick="_reject(this);">απόρριψη</a>
        <asp:Button ID="Approve" runat="server" CommandName="Approve" Text="έγκριση"
style="position: absolute; left: -100px; top: -100px;" />
        <asp:Button ID="Reject" runat="server" CommandName="Reject" Text="απόρριψη"
style="position: absolute; left: -100px; top: -100px;" />
    </td>
</tr>

```

```

</tr>
</ItemTemplate>
<EmptyDataTemplate>
δεν υπάρχουν αιτήσεις για τη θέση
</EmptyDataTemplate>
</asp:ListView>

<input type="hidden" id="reject" name="reject" />
<div id="dialog" style="display: none;">
  <table>
    <tr valign="top">
      <td>αιτιολογία:</td>
      <td><textarea id="reason" cols="40" rows="3"></textarea></td>
    </tr>
  </table>
</div>
</asp:PlaceHolder>

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

$.datepicker.setDefaults($.datepicker.regional['el']);
$('#input[id$=StartDate],input[id$=EndDate]').datepicker(
{
  showOn: 'both'
  ,buttonImageOnly: true
  ,buttonImage: 'images/calendar.gif'
});

function _reject(caller)
{
  var reject = $('#reject');
  var reason = $('#reason');

  $('#dialog').dialog(
  {
    title: 'αιτιολογία απόρριψης'
    , modal: true
    , resizable: false
    , width: 420
    , height: 170
    , buttons:
    {
      'OK': function () {

        reject.val(reason.val());
        $(this).dialog('close');

        //$(caller).next('input').click();
        $('#MainContent_Applications_Reject_0').click();
      }

      , 'Ακυρο': function () {
        $(this).dialog('close');
      }
    }
  }
  });
}

function _approve(caller)
{
  var approve = $('#approve');
  var reason = $('#reason');

  $('#dialog').dialog(
  {
    title: 'σχόλια'
    , modal: true
    , resizable: false
    , width: 420
    , height: 170
  }
  );
}

```

```

    , buttons:
    {
        'OK': function () {
            approve.val(reason.val());
            $(this).dialog('close');

            $('#MainContent_Applications_Approve_0').click();
        }
        , 'Ακυρο': function () {
            $(this).dialog('close');
        }
    }
});
}
</script>
</asp:Content>

```

Η διαχείριση της *modify-position.aspx* καλούμε την *modify-position.aspx.vb*. Βλέπουμε παρακάτω τον κώδικα.

```

Partial Class modify_position
    Inherits System.Web.UI.Page

    ''' <summary>
    ''' Αν δεν έχει γίνει υποβολή της σελίδας τότε καλείται η ρουτίνα LoadPage
    ''' Γενικώς το Page.IsPostBack σημαίνει ότι πατήθηκε κάποιο κουμπί και έχει
    ''' γίνει submit η σελίδα προς τον server. Ανάλογα εκτελούμε actions ή όχι
    ''' </summary>
    ''' <remarks></remarks>
    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        If Not Page.IsPostBack Then Me.LoadData()
    End Sub

    ''' <summary>
    ''' Σελίδα διαχείρισης των θέσεων από τον διαχειριστή του συστήματος
    ''' </summary>
    ''' <remarks></remarks>
    Private Sub LoadData()
        Dim now = DateTime.Now
        Using context = New GraduateDataContext()
            Dim list = From item In context.Positions

            Me.CurrentPositions.Items.Insert(0, New ListItem("-", "0"))

            'Δημιουργία περιγραφής για κάθε θέση
            Dim data = list.ToList()
            For Each item In data
                Dim li = New ListItem(String.Format("#{0} - {1}", item.Id, item.Name),
item.Id)

                Me.CurrentPositions.Items.Add(li)
            Next

            'Γέμισα λιστών επιπέδου μόρφωσης
            Dim typeOfSchool = context.SchoolTypes.ToList()
            With (Me.MinimumEducation)
                .DataValueField = "Id"
                .DataTextField = "Name"
                .DataSource = typeOfSchool
                .DataBind()
                .Items.Insert(0, New ListItem("- επιλέξτε -", "0"))
            End With
        End Using
    End Sub

```

**'Αν έχει επιλέξει να δει μια συγκεκριμένη θέση ο διαχειριστής τότε εμφανίζουμε τα στοιχεία της**

```
Dim selected = Request("id")
If Not String.IsNullOrEmpty(selected) Then
    Dim li = Me.CurrentPositions.Items.FindByValue(selected)
    If Not li Is Nothing Then
        Me.CurrentPositions.ClearSelection()
        li.Selected = True

        Dim position = context.Positions.Single(Function(item) item.Id =
Convert.ToInt32(li.Value))

        Me.Name.Text = position.Name
        Me.Description.Text = position.Descripition
        Me.StartDate.Text = position.FromDate.ToString("d")
        Me.EndDate.Text = position.ToDate.ToString("d")
        If position.MinimumEducation.HasValue Then
            Me.MinimumEducation.SelectedValue =
position.MinimumEducation.Value.ToString()
        End If

        Me.CodeContainerRow.Visible = True
        Me.Code.Text = String.Format("#{0}", position.Id)

        Dim admin = Roles.IsUserInRole("Administrators")
        Me.All.Visible = admin
        If admin Then
            Dim apps = context.PositionRequests.Where(Function(item)
item.PositionId = position.Id).ToList()
            Me.Applications.DataSource = apps
            Me.Applications.DataBind()
        End If
    End If
End If
```

**'Δείχνουμε μήνυμα όταν έχει γίνει επιτυχής αποθήκευση των στοιχείων**

```
Dim saved = Request("saved") = "true"
If saved Then
    Me.MessageRow.Visible = True
    Me.MessageRow.Cells(1).InnerHtml = "η αποθήκευση έγινε επιτυχώς"
End If
```

End Using

End Sub

```
''' <summary>
```

**''' Η παρακάτω ρουτίνα καλείται όταν επιλέγει μία θέση από τη λίστα και φορτώνει τα στοιχεία της θέσης για τροποποίηση**

```
''' </summary>
```

```
''' <param name="sender"></param>
```

```
''' <param name="e"></param>
```

```
''' <remarks></remarks>
```

```
Protected Sub CurrentPositions_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles CurrentPositions.SelectedIndexChanged
```

```
    Dim index = Me.CurrentPositions.SelectedItem.Value
```

```
    Response.Redirect(String.Format("{0}?id={1}", Request.CurrentExecutionFilePath, index))
```

End Sub

```
''' <summary>
```

**''' Αποθήκευση της επιλεγμένης θέσης (αν έχει επιλεγεί μία για τροποποίηση) ή αποθήκευση νέας**

```
''' </summary>
```

```
''' <param name="sender"></param>
```

```
''' <param name="e"></param>
```

```
''' <remarks></remarks>
```

```
Protected Sub Save_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Save.Click
    Page.Validate("Position")
```

```

    If Page.IsValid Then
        Dim position As Position
        Dim now = DateTime.Now

        Using context = New GraduateDataContext()
            Dim selected = Request("id")
            If Not String.IsNullOrEmpty(selected) Then
                Dim li = Me.CurrentPositions.Items.FindByValue(selected)
                If Not li Is Nothing Then
                    position = context.Positions.Single(Function(item) item.Id =
Convert.ToInt32(li.Value))

                    Else
                        Throw New ApplicationException("item does not exist")
                    End If

                    position.Name = Me.Name.Text
                    position.Description = Me.Description.Text
                    position.FromDate = DateTime.Parse(Me.StartDate.Text)
                    position.ToDate = DateTime.Parse(Me.EndDate.Text)

                    If Me.MinimumEducation.SelectedIndex > 0 Then
position.MinimumEducation = Convert.ToInt32(Me.MinimumEducation.SelectedValue)
                    Else
                        position = New Position() With {.DateCreated = DateTime.Now}
                        position.Name = Me.Name.Text
                        position.Description = Me.Description.Text
                        position.FromDate = DateTime.Parse(Me.StartDate.Text)
                        position.ToDate = DateTime.Parse(Me.EndDate.Text)

                        If Me.MinimumEducation.SelectedIndex > 0 Then
position.MinimumEducation = Convert.ToInt32(Me.MinimumEducation.SelectedValue)

                                context.Positions.InsertOnSubmit(position)
                            End If

                            context.SubmitChanges()

                            Response.Redirect(String.Format("{0}?id={1}&saved=true",
Request.CurrentExecutionFilePath, position.Id))
                        End Using
                    End If
                End Sub

                ''' <summary>
                ''' Η ρουτίνα αυτή καλείται όταν ο διαχειριστής πατήσει ένα κουμπί δίπλα σε μια
                νέα αίτηση (απόρριψη / έγκριση) και τότε αυτόματα εγκρίνει ή απορρίπτει μια αίτηση
                ''' </summary>
                ''' <param name="sender"></param>
                ''' <param name="e"></param>
                ''' <remarks></remarks>
                Protected Sub Applications_ItemCommand(ByVal sender As Object, ByVal e As
System.Web.UI.WebControls.ListViewCommandEventArgs) Handles Applications.ItemCommand
                    Using context = New GraduateDataContext()
                        Dim position = context.PositionRequests.First(Function(item) item.Id =
Convert.ToInt32(e.CommandArgument))
                        position.Notify = 1
                        If e.CommandName.ToLower() = "approve" Then
                            position.Accepted = True
                            position.DateAccepted = DateTime.Now
                        Else
                            position.RejectionReason = Request("reject")
                            position.Accepted = False
                            position.DateAccepted = DateTime.Now
                        End If
                        context.SubmitChanges()

                        Response.Redirect(Request.RawUrl, True)
                    End Using

```

```

End Sub

''' <summary>
''' Με τη ρουτίνα παρακάτω τροποποιούμε τα items στη λίστα με τις αιτήσεις για
τη θέση
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>
''' <remarks></remarks>
Protected Sub Applications_ItemCreated(ByVal sender As Object, ByVal e As
System.Web.UI.WebControls.ListViewItemEventArgs) Handles Applications.ItemDataBound
    If e.Item.ItemType = ListViewItemType.DataItem Then
        Dim data As PositionRequest = CType(e.Item.DataItem, PositionRequest)
        If Not data Is Nothing Then
            Using context = New GraduateDataContext()
                Dim row As HtmlTableRow = CType(e.Item.FindControl("Row"),
HtmlTableRow)

                Dim user As UserProfile =
context.UserProfiles.FirstOrDefault(Function(u) u.UserId = data.UserId)
                Dim email = Membership.GetUser(data.UserId).UserName
                Dim name As String = email
                If Not user Is Nothing Then
                    name = user.FirstName & " " & user.LastName
                End If

                row.Cells(0).InnerHtml = String.Format("<a
href=""profile.aspx?email={0}&position={2}"">{1}</a>", email, name, Request("id"))

                row.Cells(1).InnerHtml = data.DateCreated.ToString()
                row.Cells(2).InnerHtml = data.UserComments
                row.Cells(3).InnerHtml = "σε εξέλιξη"
                If data.Accepted.HasValue Then
                    row.Cells(3).InnerHtml = If(data.Accepted.Value, "εγκρίθηκε",
"απορρίφθηκε")

                    row.Cells(5).InnerHtml = "-"
                    If Not data.Accepted.Value Then row.Cells(3).InnerHtml =
data.RejectionReason

                Else
                    CType(e.Item.FindControl("Approve"), Button).CommandArgument =
data.Id
                    CType(e.Item.FindControl("Reject"), Button).CommandArgument =
data.Id

                End If
            End Using
        End If
    End If
End Sub

''' <summary>
''' Η ρουτίνα αυτή καλείται όταν πατήσουμε το κουμπί εξαγωγή δεδομένων σε excel
για να επιλέξουμε τις στήλες που επιθυμούμε να εξάγουμε
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>
''' <remarks></remarks>
Protected Sub ShowColumnPicker_Click(sender As Object, e As EventArgs) Handles
ShowColumnPicker.Click
    If Me.ColumnsContainer.Visible Then
        Me.ColumnsContainer.Visible = False
        Me.ShowColumnPicker.Text = "εξαγωγή σε excel (εμφάνιση)"
    Else
        Me.ColumnsContainer.Visible = True
        Me.ShowColumnPicker.Text = "εξαγωγή σε excel (απόκρυψη)"
    End If
End Sub

''' <summary>

```

''' Η ρουτίνα Export\_Click καλείται όταν πατήσουμε εξαγωγή σε excel και δημιουργεί το αρχείο excel παίρνοντας τις στήλες που επιλέξαμε μόνο και στη συνέχεια πετάει prompt για download.

Για την σωστή εξαγωγή χρησιμοποιούμε και ένα εξωτερικό library που λέγεται ExcelXMLExportHelper και βοηθάει στην δημιουργία του excel '''

```
</summary>
''' <param name="sender"></param>
''' <param name="e"></param>
''' <remarks></remarks>
Protected Sub Export_Click(sender As Object, e As EventArgs) Handles Export.Click
    Dim sb As StringBuilder = New StringBuilder()
    Dim headers As StringBuilder = New StringBuilder()
    Dim rs As DataTable = New DataTable("data")

    For Each item As ListItem In Me.Columns.Items
        If item.Selected Then
            If headers.Length > 0 Then headers.Append(",")
            headers.Append(item.Text)

            rs.Columns.Add(item.Text)
        End If
    Next

    Using context = New GraduateDataContext()
        Dim selected = Convert.ToInt32(Request("id"))
        If Not String.IsNullOrEmpty(selected) Then
            Dim apps = context.PositionRequests.Where(Function(item) item.PositionId =
selected).ToList()

            For Each item In apps
                Dim r = rs.NewRow()
                Dim user As UserProfile =
context.UserProfiles.FirstOrDefault(Function(u) u.UserId = item.UserId)
                Dim email = Membership.GetUser(item.UserId).UserName
                Dim row As List(Of String) = New List(Of String)()

                If Me.Columns.Items(0).Selected Then
                    row.Add(user.FirstName & " " & user.LastName)

                    r(Me.Columns.Items(0).Text) = row.Last()
                End If

                If Me.Columns.Items(1).Selected Then
                    row.Add(item.DateCreated.ToString("d"))

                    r(Me.Columns.Items(1).Text) = row.Last()
                End If

                If Me.Columns.Items(2).Selected Then
                    If item.Accepted.HasValue Then
                        row.Add(If(item.Accepted.Value, "εγκρίθηκε",
"απορρίφθηκε"))

                    Else
                        row.Add("-")
                    End If
                    r(Me.Columns.Items(2).Text) = row.Last()
                End If

                If Me.Columns.Items(3).Selected Then
                    row.Add(item.RejectionReason)
                    r(Me.Columns.Items(3).Text) = row.Last()
                End If
            End For
        End Using
    End Sub
```



```

If Me.Columns.Items(4).Selected Then
    row.Add(email)
    r(Me.Columns.Items(4).Text) = row.Last()
End If

If Me.Columns.Items(5).Selected Then
    row.Add(user.Phone)
    r(Me.Columns.Items(5).Text) = row.Last()
End If

If Me.Columns.Items(6).Selected Then
    row.Add(user.BirthDate.ToString("dd/MM/yyyy"))
    r(Me.Columns.Items(6).Text) = row.Last()
End If

If Me.Columns.Items(7).Selected Then
    row.Add(user.Address)
    r(Me.Columns.Items(7).Text) = row.Last()
End If

If Me.Columns.Items(8).Selected Then
    Dim education = (From edu In context.Educations Where
edu.UserId = item.UserId).ToList()
    If education.Count > 0 Then
        Dim d = New StringBuilder()
        For Each edu In education
            If d.Length > 0 Then d.Append(", ")
            If edu.SectionId.HasValue Then
                Dim section =
context.SchoolSections.FirstOrDefault(Function(q) q.Id = edu.SectionId.Value)
                Dim school = section.School
                d.Append(school.Name & " / " &
section.Name)
            Else
                d.Append(edu.School)
            End If
            Dim period As String = " - από " &
edu.DateFrom.ToString("MMM yyyy")
            period = period & " έως " &
edu.DateTo.ToString("MMM yyyy")
            d.Append(period)
        Next
        row.Add(d.ToString())
    Else
        row.Add("-")
    End If
    r(Me.Columns.Items(8).Text) = row.Last()
End If

If Me.Columns.Items(9).Selected Then
    row.Add(user.Country1.Name)
    r(Me.Columns.Items(9).Text) = row.Last()
End If

If Me.Columns.Items(10).Selected Then
    row.Add(user.Nationality)
    r(Me.Columns.Items(10).Text) = row.Last()
End If

```

```

        If Me.Columns.Items(11).Selected Then
            row.Add(user.PersonalStatus)
            r(Me.Columns.Items(11).Text) = row.Last()
        End If

        If Me.Columns.Items(12).Selected Then
            Dim jobs = (From job In context.Jobs Where job.UserId =
item.UserId Order By If(job.ToDate.HasValue, job.ToDate.Value, DateTime.Now) Descending).ToList()
            If jobs.Count > 0 Then
                Dim d = New StringBuilder()
                For Each job In jobs
                    If d.Length > 0 Then d.Append(",")
                    d.Append(job.CompanyName & " / " & job.Title)

                    Dim period As String = "από " &
job.FromDate.ToString("MMM yyyy")

                    If job.ToDate.HasValue Then
                        period = period & " έως " &
job.ToDate.Value.ToString("MMM yyyy")

                    Else
                        period = period & " έως και σήμερα"
                    End If

                    d.Append(period)
                Next

                row.Add(d.ToString())
            End If
            r(Me.Columns.Items(12).Text) = row.Last()
        End If

        If sb.Length > 0 Then sb.Append(ControlChars.CrLf)
        sb.Append(String.Join(", ", row.ToArray()))
        rs.Rows.Add(r)

        'For i As Integer = 0 To 10
        '    Dim t = rs.NewRow()
        '    For Each c As DataColumn In rs.Columns
        '        t(c.ColumnName) = r(c.ColumnName)
        '    Next
        '    rs.Rows.Add(t)
        'Next
    Next
End If
End Using

Dim ds = New DataSet("All")
ds.Tables.Add(rs)
Dim data = ExcelXMlexportHelper.ToFormattedExcel(ds)

Dim tmp = System.IO.Path.ChangeExtension(System.IO.Path.GetTempFileName(), ".xls")
System.IO.File.WriteAllText(tmp, data)

sb.Insert(0, headers.ToString() & ControlChars.CrLf)
With (Response)
    .Clear()
    .ContentType = "application/vnd.ms-excel; charest=windows-1253"
    .AddHeader("Content-Disposition", String.Format("attachment;
filename=export{0}.xls", DateTime.Now.ToString("yyyyMMdHHmm")))
    .ContentEncoding = System.Text.Encoding.GetEncoding(1253)
    .BinaryWrite(System.IO.File.ReadAllBytes(tmp))
End With

Try
    System.IO.File.Delete(tmp)
Catch ex As Exception

```

```
        End Try
    .End()
End With
End Sub
End Class
```

# 7. Βιβλιογραφία

## 7.1 Έντυπη βιβλιογραφία

- Microsoft SQL Server 2008 For Dummies , Chapple Mike
- K. Delaney et al, Microsoft SQL Server 2008 Internals. Microsoft Press, 2009
- Thearon Willis, Bryan Newsome, 2006, Beginning Visual Basic 2005, Wiley Publishing, Indianapolis USA
- Microsoft ASP.NET προγραμματισμός με τη Microsoft Visual Basic .NET βήμα βήμα, Duthie G. Andrew

## 7.2 Σελίδες στο διαδίκτυο

- <http://www.microsoft.com>
- <http://support.microsoft.com>
- <http://msdn2.microsoft.com>
- <http://www.microsoft.com/sql/default.aspx>
- <http://www.microsoft.com/hellas/msdn/products/vsteamsystem.aspx>
- <http://www.getdotnetcode.com>
- <http://www.dotnet2themax.com>
- <http://www.freeprogrammingresources.com>
- <http://www.wrox.com>
- <http://vb-helper.com>
- <http://www.javascriptkit.com/howto/aspnet.shtml>
- <http://en.wikipedia.org/wiki/ASP.NET>
- [http://en.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](http://en.wikipedia.org/wiki/Microsoft_SQL_Server)
- <http://www.techopedia.com/definition/1243/sql-server>
- [http://el.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](http://el.wikipedia.org/wiki/Microsoft_SQL_Server)
- <http://mkasoft.com/downloads/VB.NET%20programming.pdf>