

Επέκταση Portal για τη διαχείριση  
και αποστολή μαζικών email.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

**Τόπος:**.....

**Ημερομηνία:** .....

**ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ**

.....

.....

.....

## Περίληψη

Αντικείμενο αυτής της πτυχιακής είναι η επέκταση υπάρχουσας πλατφόρμας για την διαχείριση και αποστολή μαζικών μηνυμάτων σε ομάδες – στόχους (target groups). Η κύρια ομάδα - στόχος της εν λόγω πλατφόρμας είναι το ανθρώπινο δυναμικό του ΤΕΙ Ηπείρου. Στο κείμενο που ακολουθεί αναλύεται ο τρόπος υλοποίησης και λειτουργίας του συστήματος και των επεκτάσεων που υλοποιήθηκαν καθώς και τα εργαλεία και οι τεχνολογίες που χρησιμοποιήθηκαν.

Σκοπός της επέκτασης αυτής είναι η δυνατότητα να αποστέλλονται μαζικά μηνύματα σε ομάδες χρηστών τα στοιχεία των οποίων βρίσκονται στην κεντρική βάση του ΤΕΙ Ηπείρου. Η αποστολή των μηνυμάτων γίνεται από τους χρήστες της πλατφόρμας οι οποίοι ιεραρχούνται σε κατηγορίες ανάλογα με τη θέση που κατέχουν στο ΤΕΙ Ηπείρου. Κάθε χρήστης μπορεί να στείλει μηνύματα σε ομάδες στόχους για τους οποίους είναι υπεύθυνος. Το σύστημα διατηρεί ιστορικό αποστολών για κάθε χρήστη ξεχωριστά.

Η υλοποίηση έγινε σε περιβάλλον windows με χρήση Microsoft Visual Studio και του MS SQL DBMS.

## **Ευχαριστίες**

Θα ήθελα να ευχαριστήσω θερμά τον καθηγητή μου κ.Βαρτζιώτη Φώτιο για την πολύτιμη βοήθεια του και καθοδήγηση του κατά την διάρκεια υλοποίησης της πτυχιακής μου εργασίας.

Θα ήθελα επίσης να απευθύνω τις ευχαριστίες μου στους γονείς μου και στον μέλλοντα σύζυγο μου, οι οποίοι με στήριξαν με διάφορους τρόπους ,φροντίζοντας για την καλύτερη δυνατή μόρφωση μου.

ΣΦΙΝΟΓΕΝΝΗ ΜΑΡΙΑΝΝΑ

10/09/2015

## Περιεχόμενα

Περίληψη .....	ii
Ευχαριστίες .....	iii
1. Εισαγωγή .....	1
1.1 Η Εφαρμογή.....	1
1.2 Απαιτήσεις συστήματος.....	1
2. Περιγραφή τεχνολογιών.....	3
2.1 ASP .NET .....	3
2.2 C#.....	5
2.3 WebForms.....	7
2.4 HTML .....	9
2.5 SQLServer.....	10
2.6 SQLServerManagementStudio .....	11
2.7 Microsoftvisualstudio .....	11
2.8 Membershipautomation .....	12
2.9 Υπάρχοντα συστήματα μαζικής αποστολής email .....	12
3. Περιγραφή εφαρμογής.....	14
3.1 Γενική περιγραφή.....	14
3.2 Βάση Δεδομένων .....	17
3.2.1 Πίνακας log_table .....	21
3.2.2 Πίνακας Classes .....	21
3.2.3 Πίνακας classusernames .....	22
3.2.4 Πίνακας departments .....	22
3.3 Δομή εφαρμογής.....	22
3.3.1 UML.....	22
3.4 Δομή καταλόγου .....	27
3.4.1 Φάκελος root.....	28
3.4.2 Φάκελος App_Data.....	28
3.4.3 Φάκελος CSS .....	28
3.4.4 Φάκελος Scritps .....	29
3.4.5 Φάκελος professor .....	29

3.4.6 Φάκελος CEO .....	29
3.4.7 Φάκελος secretariat .....	29
3.5 Τυποποίηση εφαρμογής .....	29
3.5.1 Login.aspx .....	29
3.5.2 Σελίδα seoForm.aspx .....	33
3.5.3 Σελίδα professorForm.aspx .....	39
3.5.4 Σελίδα secretaryat.aspx .....	45
3.5.5 Σελίδα email_history .....	51
3.5.6 web.config .....	52
4. Οδηγίες Χρήσης .....	54
4.1 CEO .....	55
4.2 PROFESSOR .....	63
4.3 SECRETARY .....	66
5. Συμπεράσματα -προτάσεις .....	68
6. Βελτιώσεις .....	69
ΒΙΒΛΙΟΓΡΑΦΙΑ .....	71
ΠΑΡΑΡΤΗΜΑ Α .....	72
ΠΑΡΑΡΤΗΜΑ Β .....	73



## 1. Εισαγωγή

Σκοπός της πτυχιακής είναι η επέκταση ενός portal ώστε να είναι δυνατή η αποστολή μαζικών μηνυμάτων προς ομάδες στόχους, τα στοιχεία των οποίων βρίσκονται στην κεντρική βάση του ΤΕΙ Ηπείρου.

Η κατασκευή της επέκτασης έγινε σε γλώσσα προγραμματισμού C# - ίδια με αυτή του portal - και η ανάπτυξη έγινε σε περιβάλλον Microsoft Visual Studio με χρήση του .NET framework. Ως βάση δεδομένων για την αποθήκευση των δεδομένων της εφαρμογής χρησιμοποιήθηκε η MS Sql.

Οι βασικές εργασίες για την ολοκλήρωση της πτυχιακής ήταν:

- Μελέτη της λειτουργίας του συστήματος
- Αναγνώριση των βασικών απαιτήσεων του συστήματος
- Αναγνώριση των βασικών οντοτήτων και των σχέσεών τους
- Παραμετροποίηση της βάσης δεδομένων
- Υλοποίηση των εργασιών αποστολής μηνυμάτων

Παραδοτέα της πτυχιακή είναι

- Η βάση δεδομένων
- Ο κώδικας της εφαρμογής

### 1.1 Η Εφαρμογή

Η επέκταση που υλοποιήθηκε έχει ως σκοπό την προσθήκη δυνατότητας αποστολής μαζικών email σε ομάδες στόχους που είναι εγγραμμένοι στην κεντρική βάση του ΤΕΙ Ηπείρου, από τους χρήστες του συστήματος. Κάθε χρήστης έχει συγκεκριμένα δικαιώματα και μπορεί να στείλει μηνύματα μόνο σε ομάδες για του οποίους είναι εξουσιοδοτημένος να διαχειρίζεται. Η πιστοποίηση γίνεται μέσω φόρμας login όπου ο χρήστης θα πρέπει να εισάγει το όνομα χρήστη τον κωδικό πρόσβασης και το email του.

Η οθόνη εργασίας μετά την πιστοποίηση των στοιχείων περιλαμβάνει μια φόρμα όπου συντάσσει το email καθώς και μια σειρά από επιλογές μέσω των οποίων ορίζει τους παραλήπτες.

Η εφαρμογή διατηρεί ιστορικό αποστολών για κάθε αποστολή μηνύματος.

### 1.2 Απαιτήσεις συστήματος

Οι απαιτούμενες τεχνικές προδιαγραφές της πλατφόρμας είναι οι παρακάτω



- Λειτουργικό σύστημα windows vista ή μεταγενέστερο 32 ή 64 Bit ή windows server 2008
- IIS version 6 +
- Βάση δεδομένων MSSQL 2008 η μεταγενέστερη.
- Οποιοδήποτε επεξεργαστής INTEL ή AMD βάση προδιαγραφών λειτουργικού συστήματος.
- Ελάχιστη Μνήμη 2GBRAM.
- 30 MB χώρος στον σκληρός δίσκος.

#### Προγράμματα που χρησιμοποιήθηκαν

- Microsoft visual studio 2013
- Microsoft SQL Server Management studio 2014
- Microsoft SQL Server 2014

#### Γλώσσες προγραμματισμού

- C#
- Java script
- ASP.Net Framework
- SQL

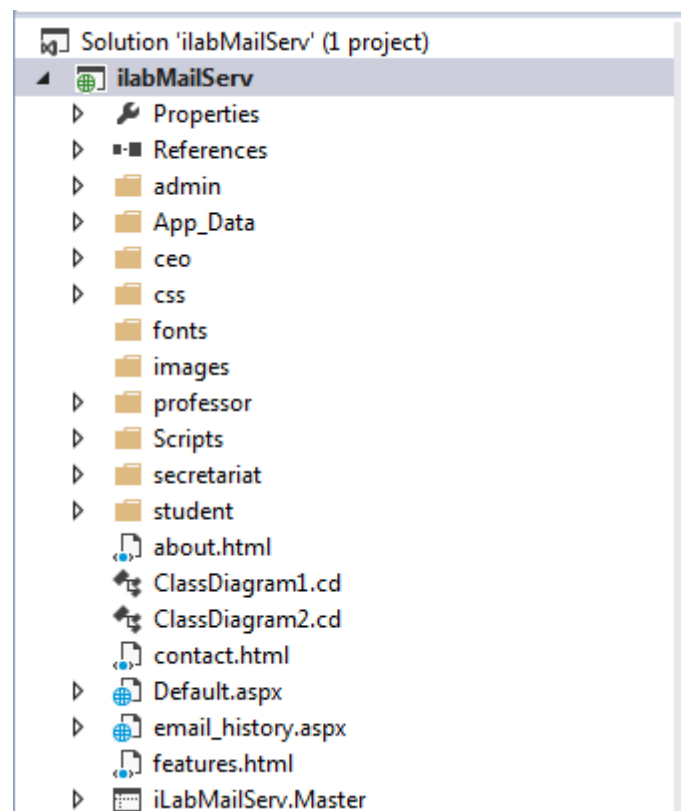
## 2. Περιγραφή τεχνολογιών

Στο κεφάλαιο αυτό γίνεται περιγραφή των τεχνολογιών που χρησιμοποιήθηκαν. Αναλύονται τα χαρακτηριστικά τους και ο τρόπος λειτουργίας τους. Στο τέλος του κεφαλαίου αυτού παρουσιάζονται και ορισμένα παρόμοια συστήματα που κυκλοφορούν στο εμπόριο.

### 2.1 ASP .NET

Η ASP.NET<sup>[1]</sup> είναι μια πλατφόρμα ανάπτυξης Web εφαρμογών – δυναμικών ιστοσελίδων. Βάση της ASP.NET είναι το .NET framework το οποίο αναπτύχθηκε από την Microsoft με σκοπό να προσφέρει στους προγραμματιστές τα εργαλεία που χρειάζονται για την ανάπτυξη εφαρμογών για το Λ/Σ windows.

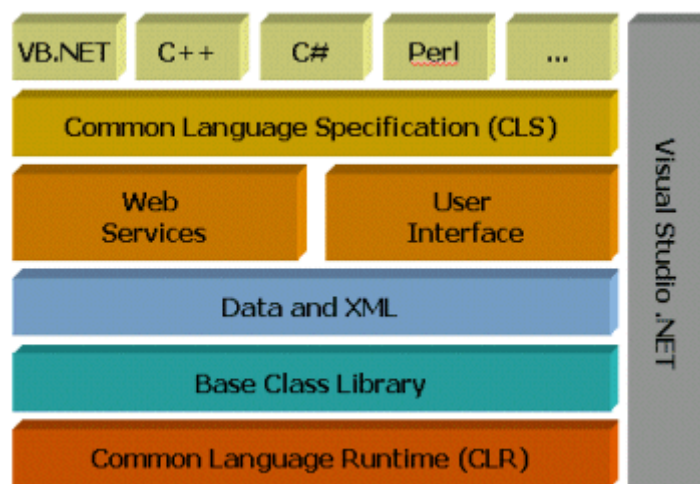
Οι web εφαρμογές που αναπτύσσονται σε ASP.Net φιλοξενούνται σε server με λειτουργικό windows server στον οποίο τρέχει ο IIS.



Εικ. 2.1: Η δομή ενός έργου σε asp

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

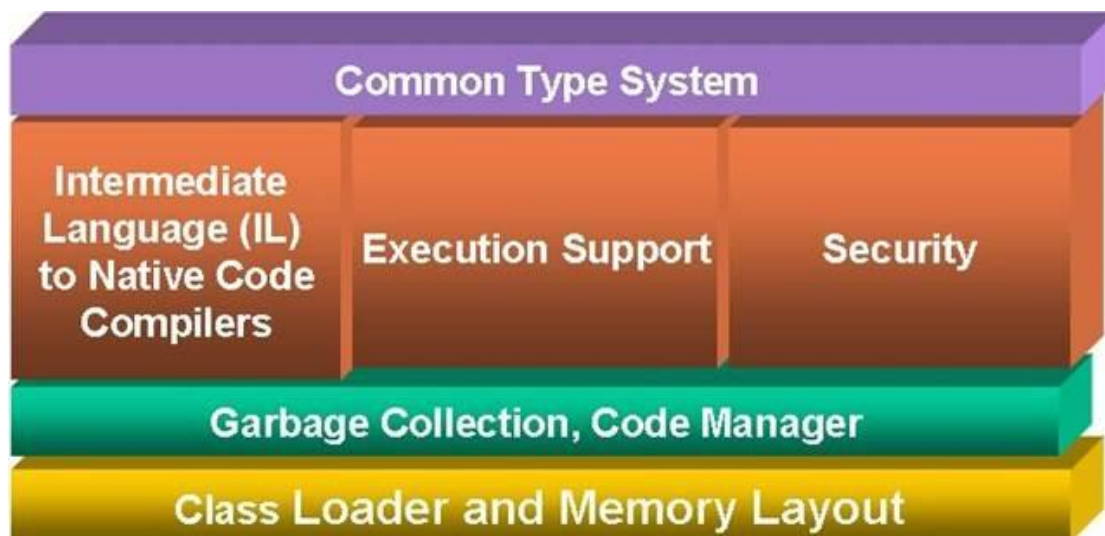
- i. Χρησιμοποιεί το .NET framework, ένα πακέτο που προσφέρει έτοιμη λειτουργικότητα στον προγραμματιστή για την διευκόλυνσή του με βασικές διαδικασίες.
- ii. Επιτρέπει την συγγραφή κώδικα με μια από περισσότερες από 40 γλώσσες προγραμματισμού μεταξύ των οποίων και οι πιο διαδεδομένες C# και Visual Basic.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



Εικ. 2.2: Η δομή του .NET FRAMEWORK<sup>[2]</sup>

Απαραίτητη προϋπόθεση, για την εκτέλεση των εφαρμογών σε .NET είναι το σύστημα το οποίο φιλοξενεί ένα πρόγραμμα ή μια ιστοσελίδα, να έχει εγκατεστημένο το .NET framework runtime.

Ο πυρήνα του .NET είναι το CLR (Common Language Runtime). Το CLR είναι το περιβάλλον μέσα στο οποίο τρέχουν οι εφαρμογές σε .NET. Το CLR είναι αυτό που διαχειρίζεται τις εφαρμογές.



Εικ. 2.3: Η δομή του CLR<sup>[3]</sup>

Το .NET, όπως και η JAVA ,χρησιμοποιεί την λεγόμενη Just in time μεταγλώττιση. Ο κώδικας, πρώτα μεταγλωττίζεται σε MSIL (Microsoft Intermediate Language) και στη συνέχεια αποθηκεύεται σε ένα εκτελέσιμο (.exe) αρχείο. Κατά την εκτέλεση του προγράμματος, το CLR διαβάζει τον MSIL κώδικα του αρχείου και την ίδια στιγμή (Just In Time) το μεταγλωττίζει σε κώδικα έτοιμο προς εκτέλεση.

Επίσης, το CLR παρέχει αυτοματοποιημένη διαχείριση της μνήμης, προς αποφυγήν προβλημάτων τύπου memory leak, αν ο προγραμματιστής μιας εφαρμογής δεν διαχειρίζεται σωστά τη μνήμη που δεσμεύεται από το πρόγραμμά του.

## 2.2 C#

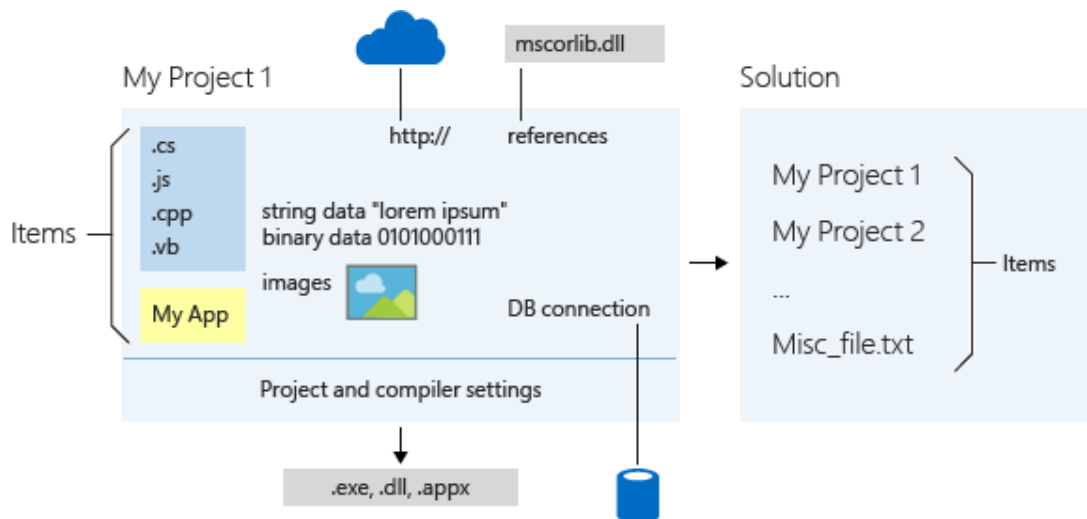
HC#<sup>[8]</sup> είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού η οποία δημιουργήθηκε από την Microsoft. Είναι γλώσσα ειδικά σχεδιασμένη για .NET Framework. Κύριο χαρακτηριστικό της γλώσσας είναι ότι δεν παράγει απευθείας κώδικα μηχανής όπως η C++ αλλά ένα ενδιάμεσο MSIL κώδικα. Το συντακτικό της γλώσσας ομοιάζει με την C++ και την Java, καθιστώντας την εκμάθησή της σχετικά εύκολη υπόθεση.

Εν αντιθέσει με άλλες γλώσσες προγραμματισμού όπως η Visual Basic που προσαρμόστηκαν ώστε να υποστηρίζουν το .NET, η C# έχει σχεδιαστεί εξ αρχής με βάση το .NET πράγμα που της δίνει τη δυνατότητα να το εκμεταλλεύεται καλύτερα.

Ο μεταγλωττιστής της C# παράγει τον ενδιάμεσο κώδικα MSIL κάτι ανάλογο με το object code στη java. Κατά την εκτέλεση, το CLR αναλαμβάνει να μεταγλωττίσει την εφαρμογή σε κώδικα για Λ/Σ windows.

Όταν δημιουργούμε μια εφαρμογή στο Visual Studio, δημιουργείτε ένα έργο. Ένα έργο περιέχει όλα τα αρχεία του πηγαίου κώδικα, εικονίδια, εικόνες, αρχεία δεδομένων καθώς και οτιδήποτε άλλο μπορεί να απαιτείτε από την εφαρμογή. Ένα έργο περιλαμβάνει επίσης τις ρυθμίσεις του μεταγλωττιστή, ρυθμίσεις που μπορεί να χρειάζονται από διάφορες υπηρεσίες ή στοιχεία (components) με τα οποία επικοινωνεί η εφαρμογή.

Τα δεδομένα αυτά αποθηκεύονται σε δομή XML σε αρχεία με κατάληξη ( .csproj ) το οποίο χρησιμοποιείται από τον Solution Explorer για να εμφανίζει τα περιεχόμενα και τις ρυθμίσεις του έργου.



Εικ. 2.4: Διάγραμμα σχέσης μεταξύ έργου (project) και solution<sup>[4]</sup>

Ένα έργο περιέχεται σε ένα solution. Ένα solution αποτελεί ένα πλαίσιο, μια δομή οργάνωσης η οποία μπορεί να περιέχει περισσότερα από ένα έργα. Στο solution περιέχονται επίσης οι ρυθμίσεις του visual studio, οι ρυθμίσεις δόμησης (build) του έργου καθώς και διάφορα άλλα αρχεία τα οποία δεν σχετίζονται με το/ τα έργα.

Ένα solution αποτελείται από δύο αρχεία. Το ένα αρχείο είναι απλό αρχείο κειμένου με κατάληξη .sln και περιέχει τις ρυθμίσεις του visual studio και ένα με κατάληξη .suo στο οποίο αποθηκεύονται οι ρυθμίσεις, κάθε χρήστη που έχει εργαστεί στο έργο.

Με την δημιουργία του έργου δημιουργείται και το namespace το οποίο ορίζει τον χώρο μέσα στον οποίο δημιουργούνται οι μεταβλητές στο πρόγραμμα. Ο ρόλος του είναι να επιτρέπει το διαχωρισμών των μεταβλητών που ανήκουν σε διαφορετικές βιβλιοθήκες αλλά έχουν το ίδιο όνομα. Όλος ο κώδικας αποτελείται από κλάσεις και μόνο και δεν επιτρέπεται η δημιουργία κώδικα εκτός κλάσης όπως για παράδειγμα στην C++.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.Security;

namespace ilabMailServ
{
    public partial class login : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            LoginStatus LoginStatus1 = (LoginStatus)this.Master.
            LoginStatus1.Visible = false;

            String ξ;
        }
    }
}

```

Εικ. 2.5: Η δομή κώδικα μιας κλάσης σε C#

Στην εικόνα 2.5 βλέπουμε το namespace της εφαρμογής. Εξορισμού, το visual studio ονομάζει το namespace της εφαρμογής χρησιμοποιώντας το όνομα της εφαρμογής.

### 2.3 Web Forms

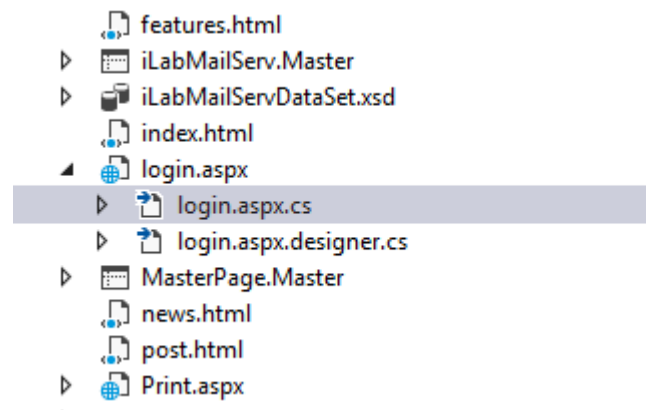
Τα web forms αποτελούν μια νέα μέθοδο στον τρόπο δημιουργίας ιστοσελίδων. Αναγκαία αλλά όχι απαραίτητη προϋπόθεση είναι η ύπαρξη του αντίστοιχου διακομιστή (IIS) της πλατφόρμας .NET. Τα αρχεία αυτής της μορφής έχουν κατάληξη «.aspx».

Εναλλακτικά, μέσω του project MONO<sup>[9]</sup> οι προγραμματιστές μπορούν να δημιουργήσουν cross platform web εφαρμογές χρησιμοποιώντας την τεχνολογία .NET και σε άλλα Λ/Σ όπως Linux, OSX, BDS χωρίς να είναι απαραίτητη η ύπαρξη Λ/Σ windows. Έτσι, ιστοσελίδες γραμμένες σε APS.NET μέσω του MONO μπορούν να τρέξουν σε apache Server με εγκατάσταση του module mod\_mono.

Τα web forms χρησιμοποιούνται κατά κύριο λόγο στην κατασκευή δυναμικών ιστοσελίδων. Ο κώδικας του προγραμματιστή εκτελείται στο web Server ο οποίος στη συνέχεια παράγει HTML σελίδες δυναμικά. Ανάλογα με τη λογική της εφαρμογής μια συγκεκριμένη σελίδα μπορεί να εμφανίζεται διαφορετικά 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 στατικών HTML σελίδων.

Μια web form αποτελείται από δύο μέρη. Το ένα μέρος αφορά στην απεικόνιση του περιεχομένου της σελίδας και είναι σε μορφή HTML. Το δεύτερο μέρος αφορά σε

αρχείο με τον κώδικα της σελίδας στη γλώσσα που έχει επιλέξει ο χρήστης κατά την δημιουργία του έργου π.χ. C#.



Εικ 2.6: Αρχείο webform

Στην παραπάνω εικόνα (2.6) βλέπουμε τη δομή της σελίδας Login της εφαρμογής. Η σελίδα αυτή αποτελείται από το σκιασμένο αρχείο login.aspx το οποίο περιέχει τον html κώδικα και το αρχείο login.aspx.cs το οποίο περιέχει τον κώδικα σε C#. Τέλος το αρχείο Login.aspx.designer.cs δημιουργείται από το visual studio το οποίο περιέχει πληροφορίες σχετικά με το layout της φόρμας μέσα στην οποία μπορεί κάποιος από την εργαλειοθήκη του visual studio να τοποθετήσει συστατικά (components) όπως για παράδειγμα textboxes, combo boxes κτλ. Σε παλαιότερες εκδόσεις του visual studio οι πληροφορίες αυτές υπήρχαν μέσα στο .cs αρχείο. Στις νεότερες εκδόσεις αυτές τις πληροφορίες τις περιλαμβάνει σε ξεχωριστό αρχείο στο οποίο δίνει την κατάληξη designer.cs.

Στο web Server η εκτέλεση μίας σελίδας γίνεται σε στάδια. Σε κάθε στάδιο εκτελούνται διάφορα γεγονότα. Τα βασικότερα στάδια της εκτέλεσης μιας σελίδας είναι τα παρακάτω.

**Page framework initialization:** Σε αυτό το στάδιο όλα τα controls που έχουν δηλωθεί στο αρχείο designer δημιουργούνται. Αν η σελίδα δεν έχει αιτηθεί για πρώτη φορά (έχει γίνει post back) τότε η ASP.NET ανακτά την πληροφορία από το view state και την εφαρμόζει στα controls. Επίσης σε αυτό το στάδιο εκδηλώνεται το Page.Init γεγονός.

**User code initialization:** Σε αυτό το στάδιο δημιουργείται το γεγονός Page.Load. Με αυτό το γεγονός μπορεί να γίνει αρχικοποίηση της σελίδας. Αυτό το γεγονός εκτελείται πάντα είτε υπάρχει postback είτε όχι.

**Validation (Επικύρωση):** Το ASP.NET περιέχει controls ελέγχου τα οποία αυτόματα ελέγχουν τις εισόδους του χρήστη σε σχέση με κάποιους κανόνες και

εμφανίζει μηνύματα λάθους. Αυτά τα γεγονότα εκτελούνται αφού η σελίδα έχει δημιουργηθεί και πριν από την εμφάνιση οποιουδήποτε άλλου συμβάντος (event).

**Event handling:** Σε αυτό το στάδιο η σελίδα έχει δημιουργηθεί πλήρως είναι επικυρωμένη και όλα τα γεγονότα που εκδηλώθηκαν από το τελευταίο postback εκτελούνται.

**Automatic data binding:** Σε αυτό το στάδιο εκτελούνται τα γεγονότα που αφορούν στα δεδομένα της σελίδας.

**Cleanup:** Αυτό το στάδιο είναι το τελευταίο στην εκτέλεση μιας Web Form. Η HTML σελίδα έχει εμφανιστεί στο browser του χρήστη. Το γεγονός Page.Unload εκτελείται. Σε αυτή τη στιγμή τα αντικείμενα που σχετίζονται με την Web Form υπάρχουν αλλά εφόσον έχει δημιουργηθεί η HTML δεν μπορεί να αλλάξει. Τέλος εκτελείται το γεγονός Page.Disposed. Σε αυτό το στάδιο θα πρέπει να γίνει η απελευθέρωση αντικειμένων που δεν χρειάζονται για να μπορέσει η διεργασία garbage collector του .NET framework να τα διαγράψει από την μνήμη.

## 2.4HTML

Η HTML είναι η πιο διαδεδομένη γλώσσα που χρησιμοποιείται για την περιγραφή ιστοσελίδων. Δεν είναι γλώσσα προγραμματισμού αφού δεν χρησιμοποιείται για να καθορίσει κανείς τη συμπεριφορά προγραμμάτων, αλλά μέσω αυτής καθορίζεται το παρουσιαστικό μιας ιστοσελίδας.

Η προδιαγραφή της HTML ορίζει ένα σύνολο από στοιχεία (elements), τα οποία αναπαριστώνται με tags, δηλαδή με λέξεις-κλειδιά που περικλείονται μεταξύ των χαρακτήρων «<» και «>» ή «</» και «>». Τα html tags συνήθως πάνε σε ζευγάρια με το πρώτο κομμάτι να ονομάζεται opening tag και το δεύτερο μέρος closing tag. Ανάλογα με το εκάστοτε στοιχείο μεταξύ του opening και closing tag ενδέχεται να μην υπάρχει τίποτα, να υπάρχει κείμενο ή ακόμα και άλλα στοιχεία. Κάθε tag υποστηρίζει ένα σύνολο από ιδιότητες (attributes). Η τιμή κάθε attribute πρέπει να περικλείεται πάντα μεταξύ των χαρακτήρων “ ”.

Θέτοντας μια τιμή σε ένα attribute καθορίζεται ένα συγκεκριμένο χαρακτηριστικό (π.χ. μέγεθος κ.τ.λ.) του στοιχείου για το οποίο είναι υπεύθυνο το tag.

Κάθε tag αποτελεί μια οδηγία στον browser για την εμφάνιση ενός συγκεκριμένου οπτικού αντικειμένου στην οθόνη ή μια οδηγία η οποία επηρεάσει τον τρόπο με τον οποίο ο browser θα εμφανίσει ένα στοιχείο.

Τα βασικά και απαραίτητα στοιχεία μιας σελίδας είναι τρία:

Το <html></html>, που χρησιμοποιείται για να ορίσει μια ιστοσελίδα.



Το <head></head>, που χρησιμοποιείται για να ορίσει τον τίτλο μια ιστοσελίδας καθώς και άλλες πληροφορίες που δεν είναι ορατές στο χρήστη.

Το <body></body>, που χρησιμοποιείται για να ορίσει το κυρίως μέρος μιας σελίδας που τελικά θα παρουσιάζεται στο χρήστη.

```
<html>
<head>
<title>Τίτλος Σελίδας</title>
</head>

<body>
HTML ΕΝΤΟΛΕΣ
</body>
</html>
```

Βασική δομή ιστοσελίδας HTML

Τα αρχεία HTML πρέπει πάντα να έχουν κατάληξη «.html» ή «.htm».

## 2.5 SQL Server

Ο SQL Server είναι μια σχεσιακή βάση δεδομένων, η οποία αναπτύσσεται από τη Microsoft. Οι κύριες γλώσσες που χρησιμοποιούνται είναι η T-SQL και η ANSI SQL. Ο SQL Server βγήκε για πρώτη φορά στην αγορά το 1989 σε συνεργασία με την Sybase. Η κύρια μονάδα αποθήκευσης στοιχείων είναι μια βάση δεδομένων, η οποία αποτελείται από μια συλλογή πινάκων και κώδικα.

Η κεντρική βάση δεδομένων του SQL υποστηρίζει διαφορετικούς τύπους, συμπεριλαμβανομένων των ακεραίων αριθμών, αριθμών κινητής υποδιαστολής, δεκαδικών, αλφαριθμητικών, Varchar (σειρές χαρακτήρων μεταβλητού μήκους), δυαδικών αριθμών (για τα μη δομημένα δεδομένα), κειμένων (για κείμενα).

Επιτρέπει επίσης καθορισμένους από το χρήστη σύνθετους τύπους δεδομένων (UDTs), δηλαδή τύπους που βασίζονται στους βασικούς τύπους αλλά μπορούν να τροποποιηθούν. Τα στοιχεία στη βάση δεδομένων αποθηκεύονται σε ένα (ή περισσότερα) αρχεία με επέκταση .mdf.

Τα δευτεροβάθμια στοιχεία αποθηκεύονται στο αρχείο με επέκταση .ndf.

Το αρχείο καταγραφής το οποίο περιέχει όλες τις πρόσφατες αλλαγές στη βάση δεδομένων αποθηκεύεται σε αρχείο με επέκταση .ldf.

Η ερώτηση (query) είναι ο κύριος τρόπος για την ανάκτηση στοιχείων από μια βάση δεδομένων. Η ερώτηση εκφράζεται χρησιμοποιώντας μια παραλλαγή της αποκαλούμενου SQL T-SQL, είναι μια διάλεκτος SQL που αναπτύχθηκε από την Microsoft και Sybase.

Ο SQL Server επιτρέπει επίσης αποθηκευμένες ερωτήσεις (stored procedures). Οι αποθηκευμένες ερωτήσεις είναι ερωτήσεις με παραμέτρους T-SQL, οι οποίες αποθηκεύονται στον SQL Server.

## **2.6 SQL Server Management Studio**

Το SQL Server management studio είναι ένα δωρεάν εργαλείο που περιλαμβάνεται μέσα στο Microsoft SQL Server 2005 και αργότερα το οποίο χρησιμοποιείται για να τροποποιείς, να διαχειρίζεσαι και οτιδήποτε άλλο χρειαστεί να κάνεις σε μια βάση η στα component της βάσης μέσα στον SQL Server.

Το εργαλείο αυτό περιλαμβάνει και script editors και graphical tools τα όποια και τα δύο δουλεύουν με αντικείμενα του Server.

Υπάρχουν πολλές διαφορετικές εκδόσεις διαθέσιμες από το site της Microsoft.

Ο SQL Server Management Studio είναι το βασικό εργαλείο που χρησιμοποιήθηκε για την διαχείριση της βάσης δεδομένων της πτυχιακής.

## **2.7 Microsoft visual studio**

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

Το Visual Studio εισάγει μια νέα διαδικασία ανάπτυξης προγραμμάτων, τον παραστατικό προγραμματισμό, που αλλάζει τον τρόπο εγγραφής και εκτέλεσης των προγραμμάτων, οδηγώντας σε αύξηση της παραγωγικότητας.

Παρέχει προχωρημένα εργαλεία για τη διόρθωση λαθών, για την τεκμηρίωση και εγγραφή κώδικα, για την ανάπτυξη διεπαφών χρήστη, για τη σχεδίαση κλάσεων καθώς και για τη σχεδίαση του σχήματος μίας βάσης δεδομένων (database schema).

Επίσης υποστηρίζει πολλά plug - ins που προσφέρουν προηγμένες λειτουργίες όπως οι 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.

## 2.8 Membership automation

Το membership είναι ένας ενσωματωμένος τρόπος ελέγχου επικύρωσης και πιστοποίησης χρηστών. Σκοπός του είναι να βοηθήσει στη διαχείριση του ελέγχου ταυτότητας χρήστη σε ιστοσελίδες.

Υποστηρίζει τα παρακάτω χαρακτηριστικά

- Δημιουργία νέων χρηστών.
- Την αποθήκευση των πληροφοριών των μελών (ονόματα χρήστη, κωδικούς πρόσβασης, καθώς και τα συνοδευτικά τους στοιχεία) στον Microsoft SQL Server, Active Directory, ή εναλλακτικό χώρο αποθήκευσης δεδομένων.
- Πιστοποίηση των χρηστών που επισκέπτονται την ιστοσελίδα είτε προγραμματιστικά είτε με τη χρήση του ASP.NET login control
- Διαχείριση κωδικών πρόσβασης η οποία περιλαμβάνει τη δημιουργία, την αλλαγή και την επαναφορά τους. Παρέχεται επίσης αυτοματοποιημένο σύστημα επαναφοράς.

## 2.9 Υπάρχοντα συστήματα μαζικής αποστολής email

Παρόμοια συστήματα μπορεί να βρει κανείς τόσο σε ανοιχτού κώδικα λύσεις (open source) όσο και σε συστήματα επί πληρωμή. Τα περισσότερα εστιάζουν στην αποστολή newsletter και διαφημιστικών μηνυμάτων σε λίστες email διευθύνσεων, και όχι σε συγκεκριμένους χρήστες ενός οργανισμού, παρόλα αυτά όμως μπορούν να χρησιμοποιηθούν και για σκοπούς που εξυπηρετούν έναν οργανισμό. Τα περισσότερα από αυτά είναι γραμμένα σε php και χρησιμοποιούν MySQL και apacheserver, μπορούν όμως να χρησιμοποιηθούν και σε συστήματα με Λ/Σ windows με εγκατάσταση των κατάλληλων μηχανισμών.

Παρακάτω εξετάζονται μερικά από αυτά

**Php list:** είναι ανοιχτού κώδικα διαχείρισης μηνυμάτων. Παρέχει πλήρες περιβάλλον διαχείρισης, δυνατότητα δημιουργίας μηνυμάτων είτε μορφή html είτε σε απλού κειμένου, οργάνωση των παραληπτών σε λίστες και τέλος δυνατότητα εισαγωγής διευθύνσεων από εξωτερικά αρχεία. Επιπλέον, ως ανοικτό λογισμικό, μπορεί να παραμετροποιηθεί σύμφωνα με τις εκάστοτε ανάγκες.

**List messenger:** Το σύστημα αυτό περιέχει και έκδοση επί πληρωμή αλλά διαθέτει και δωρεάν με περιορισμούς. Διαθέτει περιβάλλον διαχείρισης των χρηστών αλλά και των μηνυμάτων, παρέχει ρυθμίσεις στον τρόπο αποστολής μηνυμάτων, διαθέτει έτοιμα templates για τα μηνύματα, παρέχει μηχανισμούς ασφάλειας, και διάφορα χρήσιμα εργαλεία στους χρήστες όπως ορθογραφικό έλεγχο, αυτόματη απάντηση, αυτόματη διαγραφή χρηστών, templates με χρήση μεταβλητών, διάφορες ρυθμίσεις αποστολής, ομάδες χρηστών κτλ.

**Simple Mailing List:** άλλο ένα δωρεάν σύστημα διαχείρισης μηνυμάτων περιορισμένης λειτουργικότητας το οποίο καλύπτει όλες τις βασικές λειτουργίες αποστολής μηνυμάτων. Παρέχει ένα βασικό περιβάλλον διαχείρισης χρηστών, δυνατότητα αποστολής email σε μορφή html ή απλού κειμένου, περιορισμό στην κίνηση των αποστολών, αρχειοθέτηση των μηνυμάτων, έλεγχο της ουράς μηνυμάτων καθώς και έτοιμα templates.

**Mail system.NET:** το συγκεκριμένο δεν είναι σύστημα διαχείρισης μηνυμάτων αλλά ένα σύνολο εργαλείων που υποστηρίζουν λειτουργίες από αποστολής μηνυμάτων ηλεκτρονικού ταχυδρομείου. Στην ουσία είναι εργαλείο για να φτιάξεις κανείς ένα σύστημα διαχείρισης μηνυμάτων. Είναι γραμμένο σε .NET και διατίθεται μόνο επί πληρωμή.

### 3. Περιγραφή εφαρμογής

#### 3.1 Γενική περιγραφή

Υπάρχουν τρεις ενεργοί χρήστες. Αυτοί είναι οι

- ceo
- professor
- secretary

Κάθε χρήστης του συστήματος έχει την δική του οθόνη και τις δικές του επιλογές ως προς την αποστολή των μηνυμάτων.

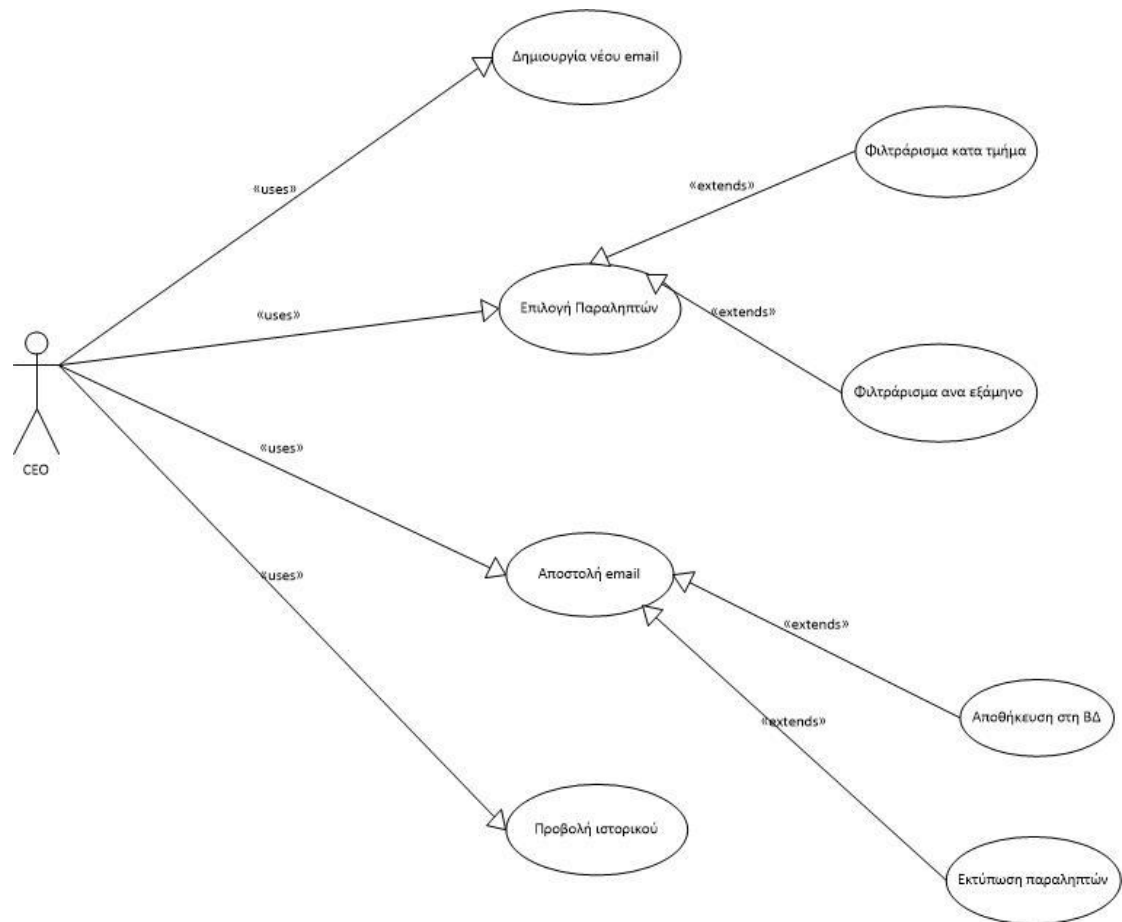
Ο χρήστης ceo έχει δικαίωμα να στέλνει μηνύματα είτε σε όλες, είτε σε επιλεγμένες ομάδες. Οι ομάδες διακρίνονται σε τμήματα και εξάμηνα. πχ. «Τμήμα Πληροφορικής» «Α' εξάμηνο».

Ο χρήστης professor στέλνει σε όλες τις ομάδες που αφορούν το μάθημά του. Οι ομάδες διακρίνονται μεταξύ τους ως προς το μάθημα που παρακολουθούν την ακαδημαϊκή περίοδο που διανύουν – εξάμηνο – την περίοδο –χειμερινή, εαρινή- και το έτος φοίτησης.

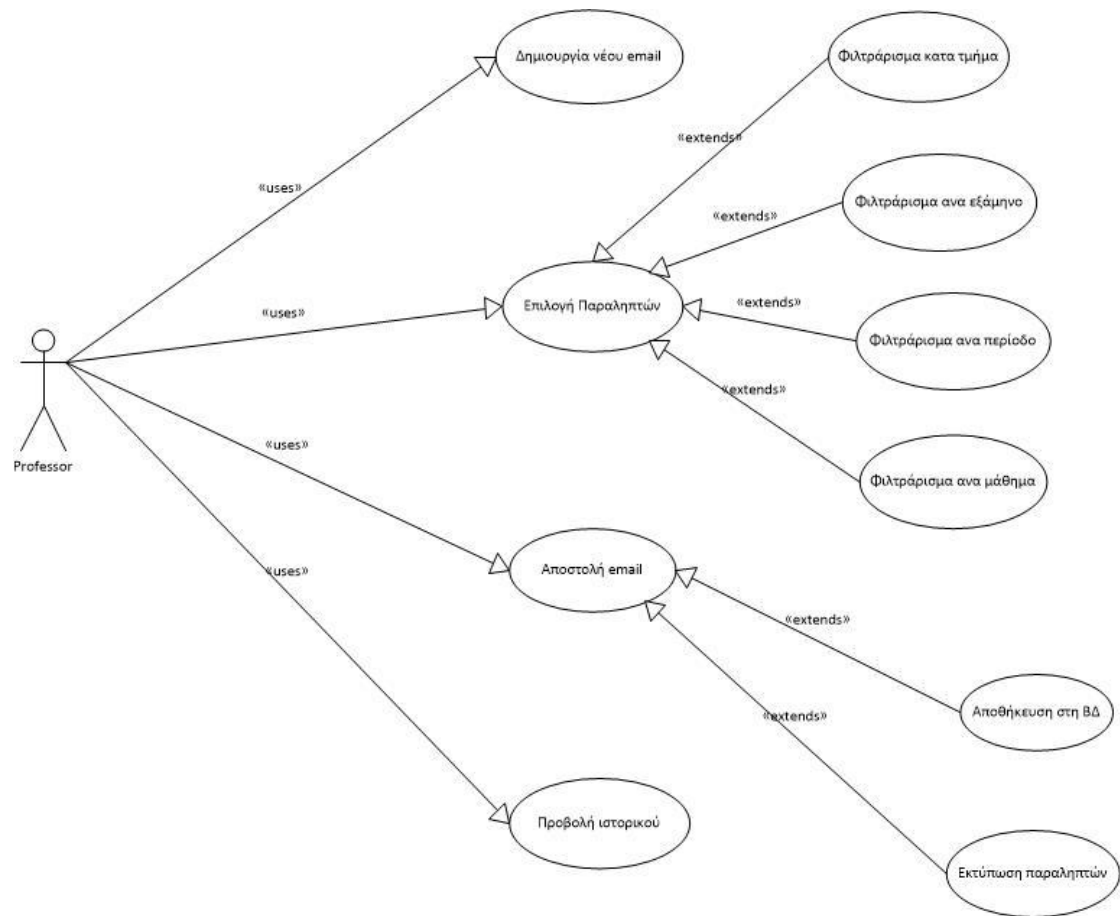
Ο χρήστης secretary μπορεί να στείλει μηνύματα σε όλες τις ομάδες της σχολής. Για τον χρήστη secretary οι ομάδες διακρίνονται σε μαθήματα η εξάμηνα με επιπλέον υποδιαίρεση σε έτος φοίτησης και μάθημα.

Για τον έλεγχο της εισόδου χρησιμοποιείται το ASP.NET membership , αυτοματισμός που παρέχεται από την Microsoft για τον έλεγχο των πρόσβασης των χρηστών.

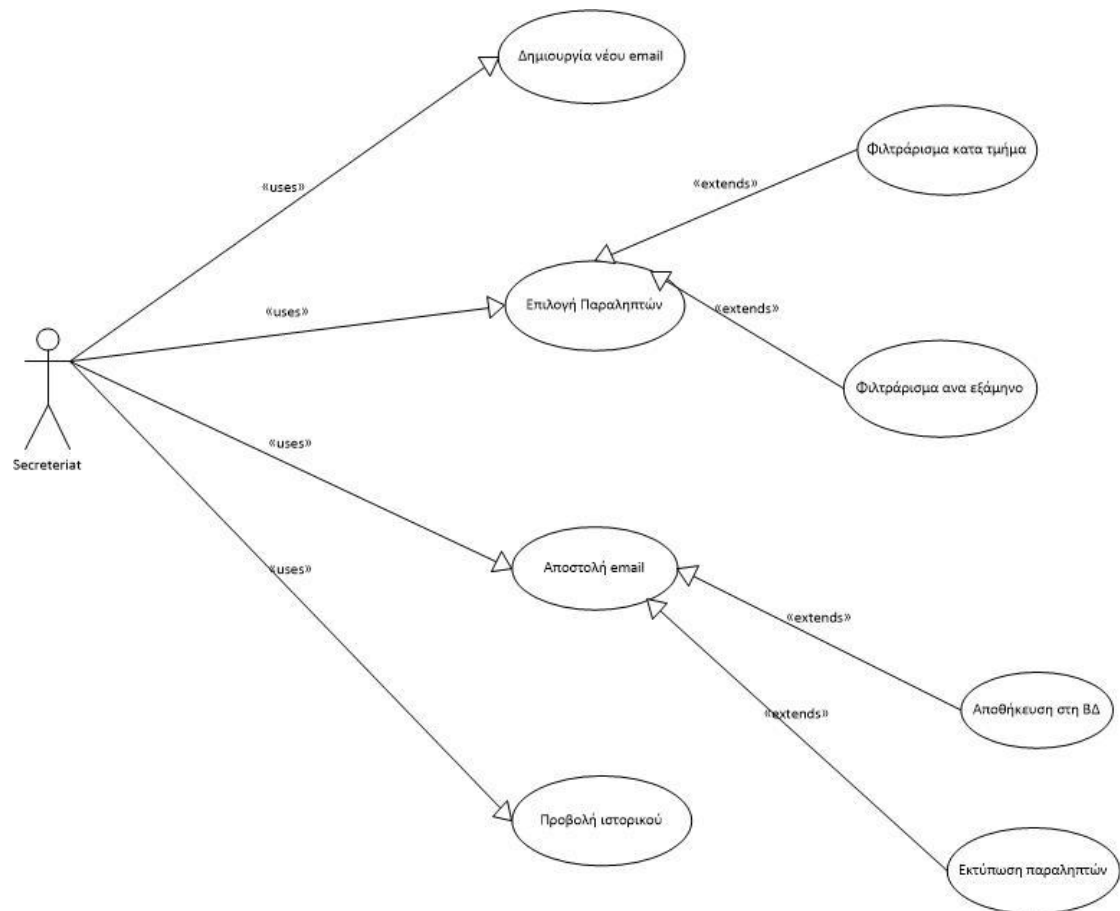
Τέλος, παραθέτονται τα διαγράμματα που απεικονίζουν τους χρήστες και τις ενέργειες που μπορούν να κάνουν.



Εικ. 3.1: Διάγραμμα χρήστη με ρόλο CEO



Εικ 3.2: Διάγραμμα χρήστη με ρόλο Professor

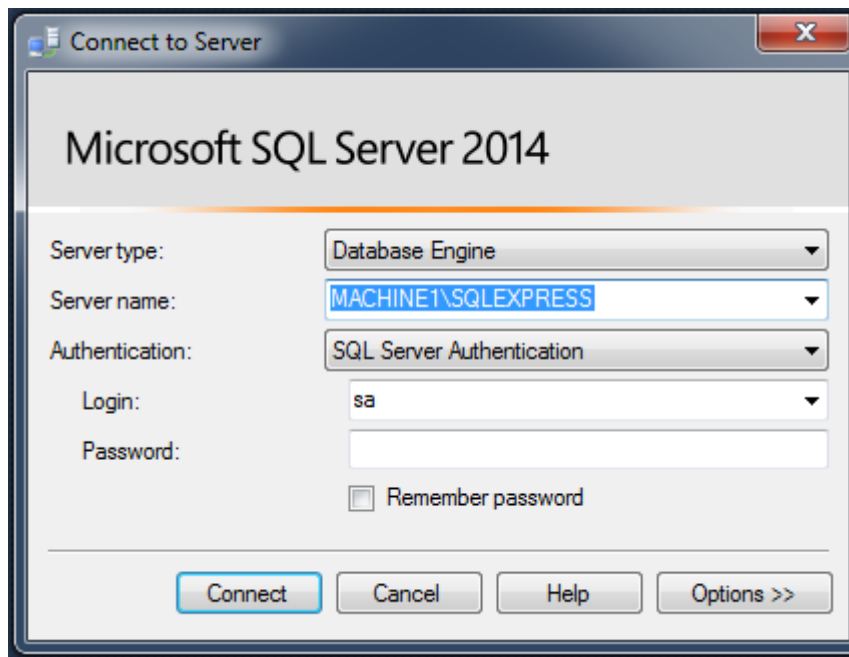


Εικ. 3.3: Διάγραμμα χρήστη με ρόλο secretariat

### 3.2 Βάση Δεδομένων

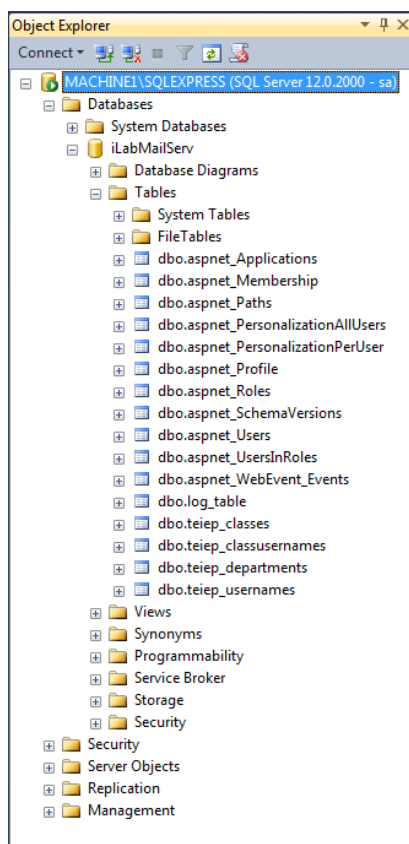
Για την διαχείριση της βάσης δεδομένων χρησιμοποιήθηκε το πρόγραμμα Microsoft SQL Server management studio το οποίο παρέχει ένα γραφικό περιβάλλον για την δημιουργία και διαχείριση της βάσης δεδομένων.





Εικ. 3.4: Η οθόνη login του MS SQL Management studio

Βασικό αντικείμενο του Microsoft SQL Server management studio είναι ο object Explorer, με χρήση του οποίου μπορούμε να επεξεργαστούμε οποιοδήποτε στοιχείο της βάσης δεδομένων.



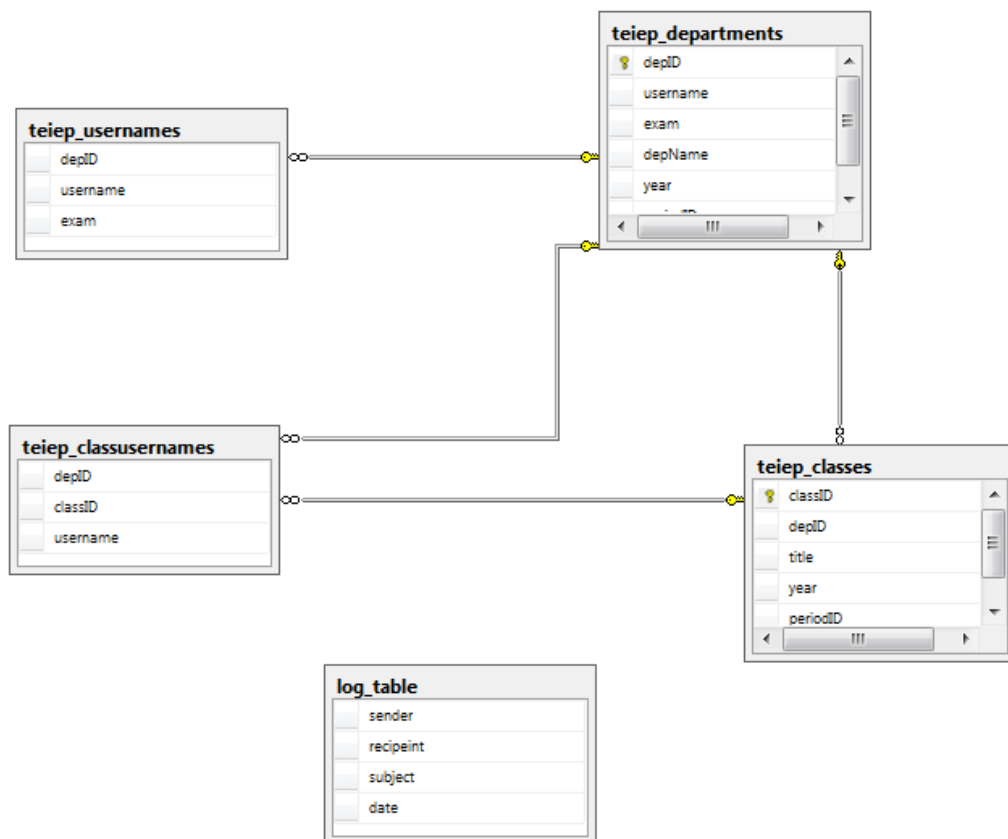
Εικ. 3.5: Ο object explorer

Η βάση δεδομένων της εφαρμογής αποτελείται από τους παρακάτω πίνακες.

ΟΝΟΜΑ ΠΙΝΑΚΑ	ΠΕΡΙΓΡΑΦΗ
log_table	Πίνακας ιστορικού αποστολών
teiep_classes	Πίνακας μαθημάτων
teiep_classusernames	Πίνακας αντιστοίχισης μαθημάτων και χρηστών
teiep_departments	Τμήματα οργανισμού
teiep_usernames	Πίνακας με τους χρήστες του συστήματος.

Πίν. 3.1: Οι πίνακες της εφαρμογής

Οι σχέσεις μεταξύ των πινάκων απεικονίζονται στην εικόνα 3.6



Εικ 3.6: Οι σχέσεις μεταξύ των πινάκων της εφαρμογής

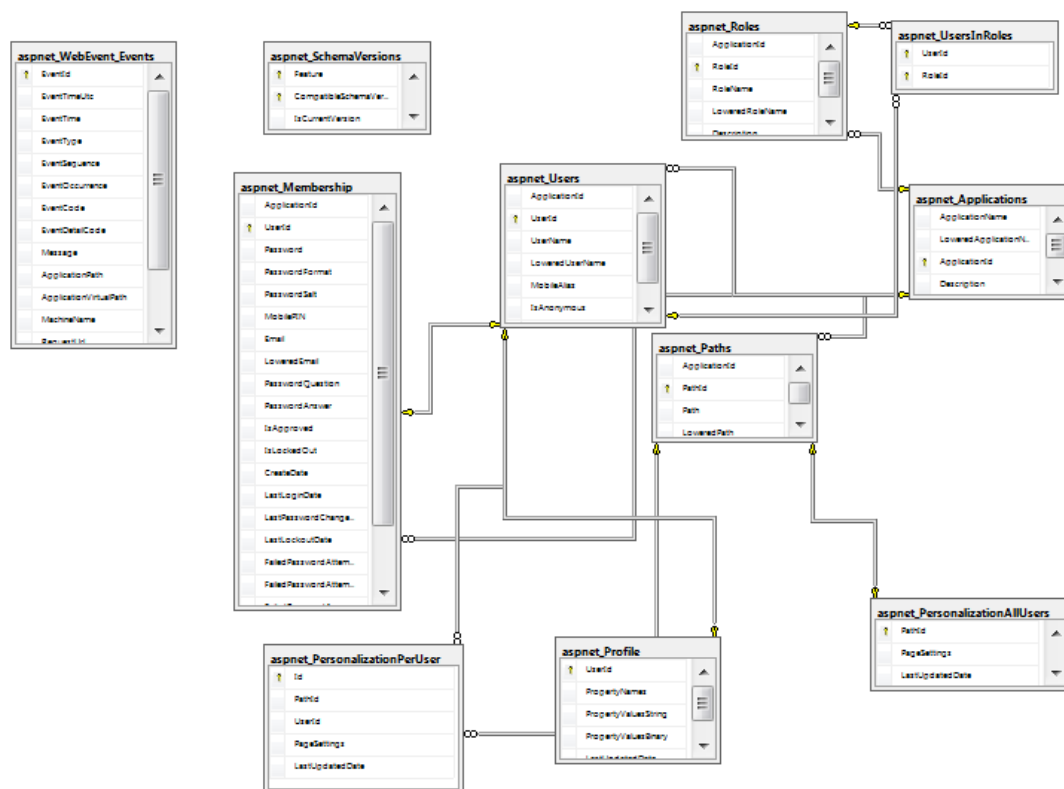
Εκτός των παραπάνω πινάκων η εφαρμογή χρησιμοποιεί το membership automation, το οποίο αποτελείται από τους παρακάτω πίνακες

ΟΝΟΜΑΣΙΑ ΠΙΝΑΚΑ	ΠΕΡΙΓΡΑΦΗ
aspnet_Applications	Περιέχει το id της εφαρμογής
aspnet_Paths	Περιέχει στοιχεία των μελών και πληροφορίες που αφορούν την ανάκτηση του κωδικού πρόσβασης. Στον πίνακα αυτόν καταγράφονται και οι τελευταίες ημερομηνίες διαφόρων δραστηριοτήτων όπως είσοδο έξοδο κτλ.
aspnet_Paths	Χρησιμοποιείται από το membership
aspnet_PersonalizationAllUsers	

aspnet_PersonalizationPerUser	
aspnet_Profile	
aspnet_Roles	Περιέχει τους διαθέσιμους ρόλους για τους εγγεγραμμένους χρήστες
aspnet_SchemaVersions	Πίνακας συστήματος
aspnet_Users	Εγγεγραμμένοι χρήστες
aspnet_UsersInRoles	Αντιστοίχιση χρηστών και ρόλων
aspnet_WebEvent_Events	Πίνακας συστήματος

Πίν. 3.2: Οι πίνακες του membership

Η σχέση μεταξύ των πινάκων απεικονίζεται στην επόμενη εικόνα



Εικ 3.7: Οι σχέσεις των πινάκων του membership

Στη συνέχεια παρουσιάζονται αναλυτικά οι πίνακες που χρησιμοποιεί η εφαρμογή.

### 3.2.1 Πίνακας log\_table

Column Name	Data Type	Allow Nulls
sender	varchar(50)	<input type="checkbox"/>
recipient	text	<input type="checkbox"/>
subject	text	<input type="checkbox"/>
date	varchar(50)	<input type="checkbox"/>

Εικ 3.8: Πίνακας log\_table

Στον πίνακα log\_table αποθηκεύεται το ιστορικό των αποστολών από τους χρήστες της εφαρμογής. Το πεδίο sender περιέχει το όνομα του αποστολέα (ceo, professor, secretariat), το πεδίο recipient περιέχει τα emails των παραληπτών, χωρισμένα με (,), το πεδίο subject περιέχει το κείμενο του θέματος του email και το πεδίο date περιέχει την ημερομηνία αποστολής του μηνύματος.

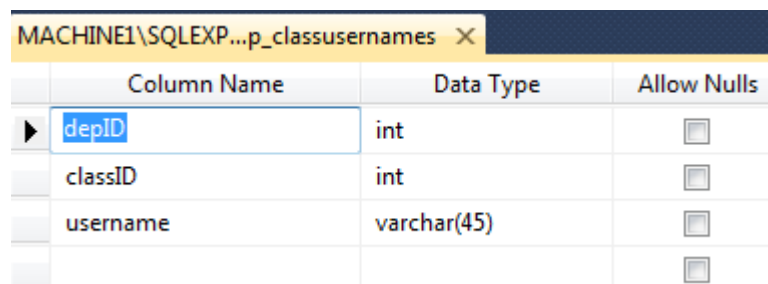
### 3.2.2 Πίνακας Classes

Column Name	Data Type	Allow Nulls
classID	int	<input type="checkbox"/>
depID	int	<input type="checkbox"/>
title	varchar(150)	<input type="checkbox"/>
year	int	<input checked="" type="checkbox"/>
periodID	int	<input checked="" type="checkbox"/>

Εικ 3.9: Πίνακας classes

Ο πίνακας classes περιέχει τα μαθήματα της σχολής. Το πεδίο class ID περιέχει τον κωδικό του μαθήματος, το πεδίο depID περιέχει τον κωδικό του τμήματος που ανήκει – διδάσκεται το μάθημα, το πεδίο title περιέχει τον τίτλο του μαθήματος, το πεδίο year περιέχει το έτος και το πεδίο period ID περιέχει το ID της περιόδου.

### 3.2.3 Πίνακας class usernames

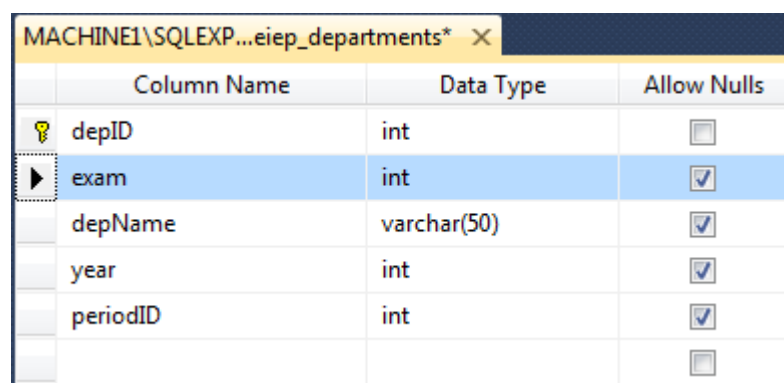


Column Name	Data Type	Allow Nulls
depID	int	<input type="checkbox"/>
classID	int	<input type="checkbox"/>
username	varchar(45)	<input type="checkbox"/>
		<input type="checkbox"/>

Εικ 3.10: Πίνακας class usernames

Ο πίνακας class usernames περιέχει τα email των σπουδαστών. Στον πίνακα αυτόν γίνεται η αντιστοίχσή τους σε τμήμα και μάθημα. Το πεδίο depID περιέχει τον κωδικό του τμήματος, το πεδίο class ID περιέχει τον κωδικό του μαθήματος και το πεδίο username περιέχει την διεύθυνση email του φοιτητή.

### 3.2.4 Πίνακας departments



Column Name	Data Type	Allow Nulls
depID	int	<input type="checkbox"/>
exam	int	<input checked="" type="checkbox"/>
depName	varchar(50)	<input checked="" type="checkbox"/>
year	int	<input checked="" type="checkbox"/>
periodID	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Εικ 3.11: Πίνακας departments

Ο πίνακας departments περιέχει τα τμήματα της σχολής. Το πεδίο depID περιέχει το ID του τμήματος, το πεδίο exam αφορά το εξάμηνο σπουδών, το πεδίο depName αφορά το όνομα του τμήματος, το πεδίο year αφορά το έτος και το πεδίο periodID περιέχει τον κωδικό της περιόδου.

## 3.3 Δομή εφαρμογής

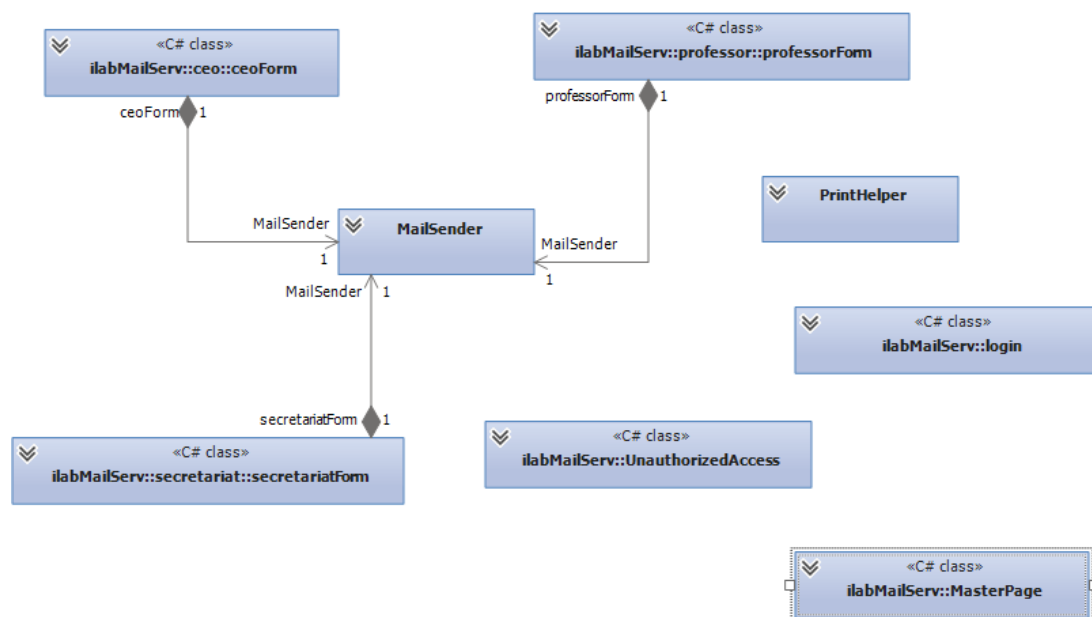
Η εφαρμογή, αποτελείται από ένα σύνολο αρχείων και φακέλων τα οποία περιέχουν τον κώδικα και τις ρυθμίσεις της εφαρμογής που αφορούν την αλληλεπίδρασή της με τη βάση δεδομένων αλλά και την συμπεριφορά της μέσα στον IIS.

### 3.3.1 UML

Οι βασικές κλάσεις της εφαρμογής είναι οι παρακάτω

- class Print
- class email history
- class secretariat Form
- class professor Form
- class ceo Form
- class Mail Sender
- class Print Helper

Το UML διαγράμμάτων κλάσεων που προκύπτει απεικονίζεται στην εικόνα 3.12



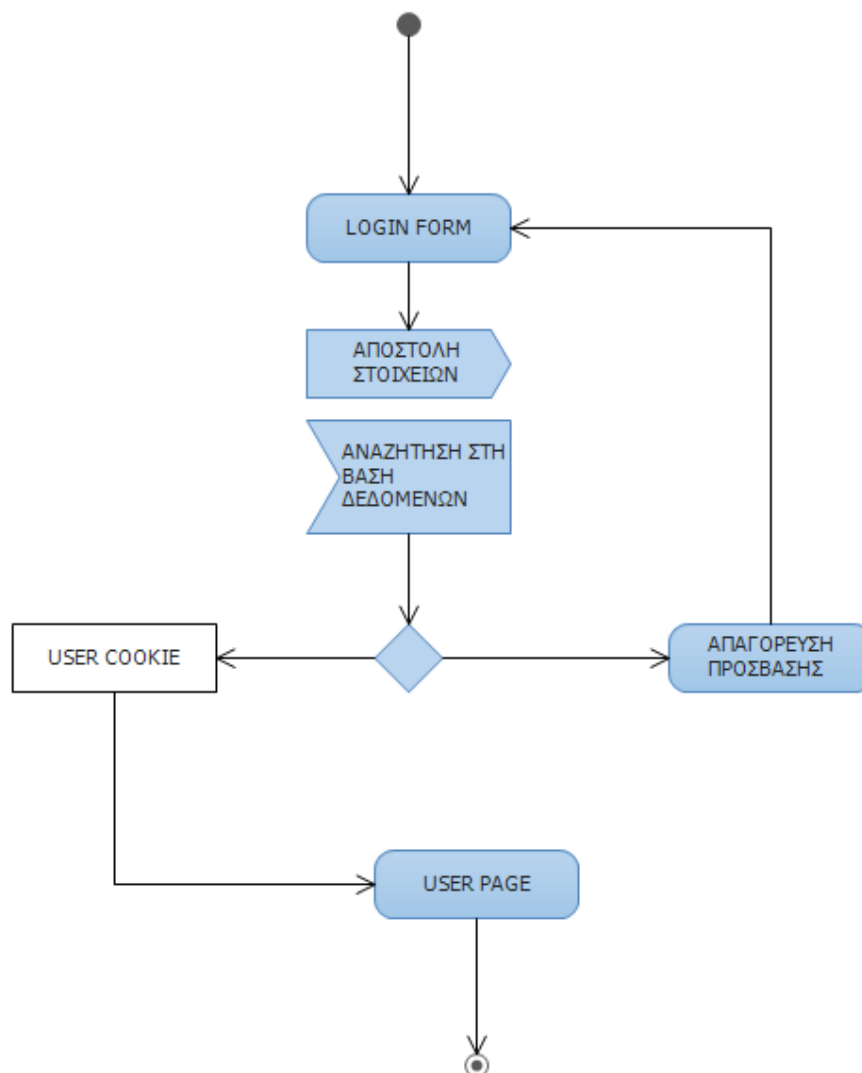
Εικ: 3.12: Διάγραμμα UML

Στο UML διάγραμμα φαίνονται οι σχέσεις μεταξύ των αντικειμένων

Όπως βλέπουμε και στο διάγραμμα υπάρχει μια σχέση aggregation μεταξύ των κλάσεων ceoForm, professor Form, secretariat με την κλάση Mail Sender

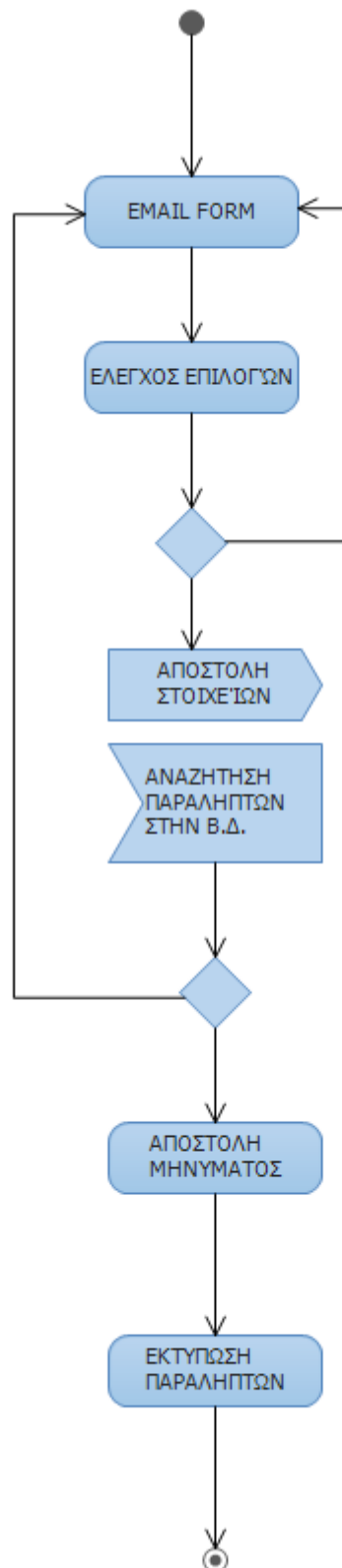
Στη συνέχεια παρουσιάζονται τα activity διαγράμματα για την διαδικασία εισόδου του χρήστη και την διαδικασία αποστολής μηνυμάτων.

Στην εικόνα 3.13 απεικονίζεται το διάγραμμα για την διαδικασία login.



Εικ. 3.13: Διάγραμμα activity για την είσοδο του χρήστη στο σύστημα

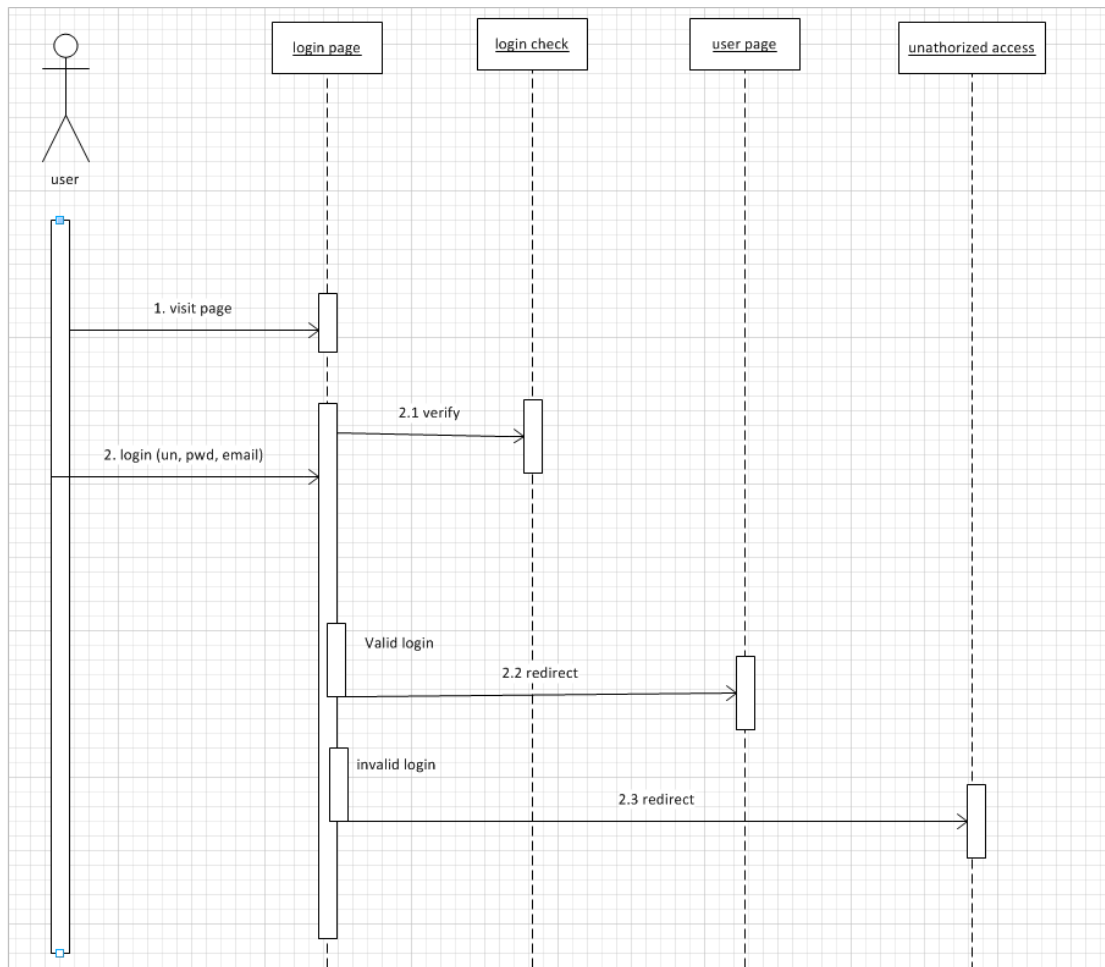
Το επόμενο διάγραμμα, εικόνα 3.14, αφορά την διαδικασία αποστολής μηνύματος.



Εικ 3.14: Διάγραμμα activity αποστολής email

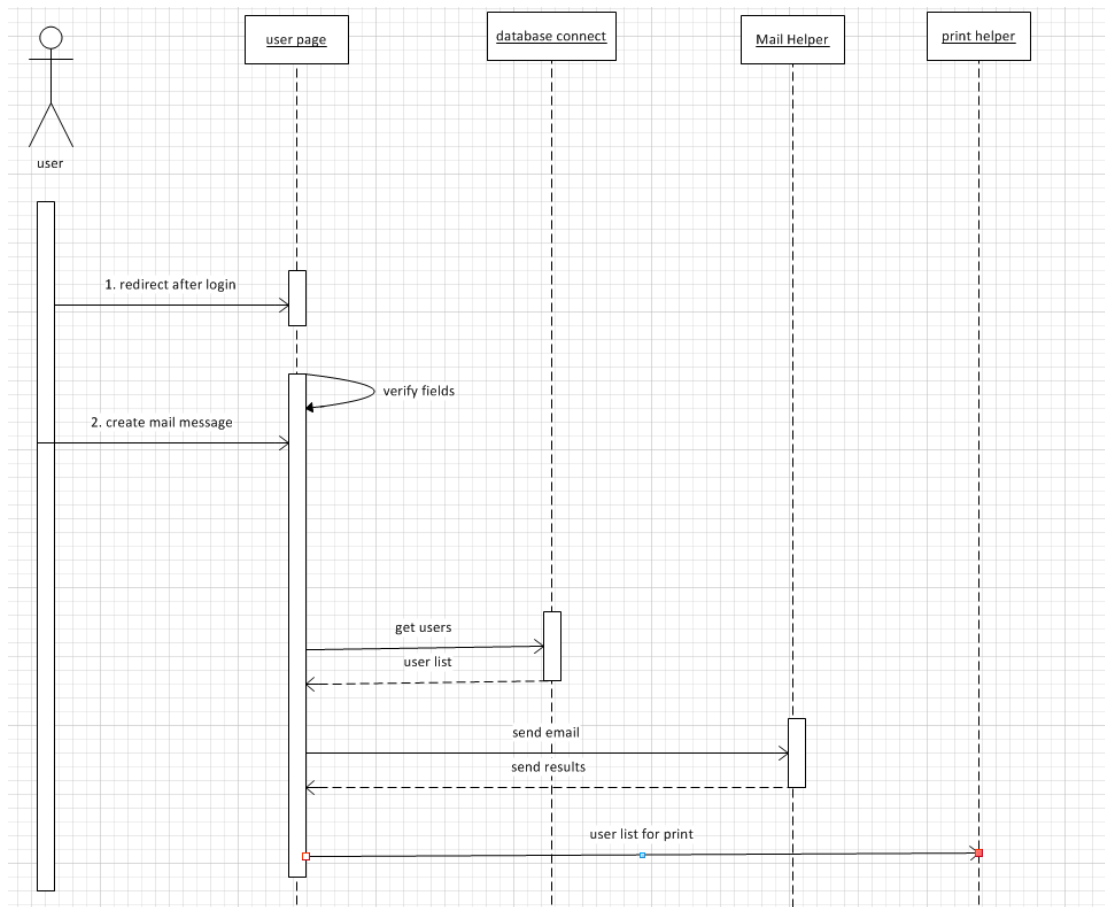
Ακολουθούν τα διαγράμματα sequence για τις ίδιες λειτουργίες. Στην εικόνα 3.15 απεικονίζεται το διάγραμμα της διαδικασίας πιστοποίησης του χρήστη.





**Εικ. 3.15:** Διάγραμμα sequence για την διαδικασία πιστοποίησης του χρήστη

Ακολουθεί το διάγραμμα αποστολής μηνυμάτων



Εικ. 3.16: Διάγραμμα αποστολής μηνυμάτων

### 3.4 Δομή καταλόγου

Η εφαρμογή, οργανώνεται σε φακέλους οι οποίοι περιέχουν τα web forms, τα αρχεία ρυθμίσεων, κώδικα μεμονωμένων κλάσεων, φωτογραφίες, και διάφορα άλλα αρχεία που αφορούν την εφαρμογή.

Κάθε webform ορίζεται από δύο αρχεία. Το ένα με κατάληξη .aspx περιέχει τον HTML κώδικα της σελίδας και των στοιχείων ελέγχου (controls) της εφαρμογής. Το δεύτερο αρχείο με κατάληξη .cs περιέχει τον κώδικα της σελίδας σεC#

Η δομή του καταλόγου της εφαρμογής είναι η παρακάτω

- root
- App data
- Ceo
- Css
- Fonts

- Images
- Professor
- Scripts
- Secretariat

### 3.4.1 Φάκελος root

Ο ριζικός κατάλογος της εφαρμογής περιέχει τα παρακάτω αρχεία

Όνομα αρχείου	Περιγραφή
login.aspx	Είναι η σελίδα μέσω της οποίας συνδέεται ο χρήστης ώστε να χρησιμοποιήσει την εφαρμογή.
Default.aspx	Η αρχική σελίδα της εφαρμογής
email_history.spx	Η σελίδα που προβάλλει το ιστορικό αποστολών ενός συνδεδεμένου χρήστη
Masterpage.master	Η σελίδα από την οποία κληρονομούν οι υπόλοιπες σελίδες της εφαρμογής
web.config	XML αρχείο που περιέχει τις παραμέτρους της εφαρμογής
UnauthorizedAccess.aspx	Η σελίδα αυτή εμφανίζεται στους μη εξουσιοδοτημένους χρήστες
Print.aspx	Η σελίδα αυτή εμφανίζει τα αποτελέσματα της αποστολής προς εκτύπωση

Πίν. 3.4: Τα αρχεία στο ριζικό φάκελο του έργου

Αρχεία που δεν περιέχονται στον παραπάνω πίνακα δεν χρησιμοποιούνται.

### 3.4.2 Φάκελος App\_Data

Ο φάκελος αυτός περιέχει τις κλάσεις του συστήματος που δεν αφορούν σελίδες

Όνομα αρχείου	Περιγραφή
MailSender.cs	Περιέχει τον κώδικα της κλάσης για την αποστολή των μηνυμάτων
PrintHelper.cs	Η κλάση αυτή καλείται κατά την εμφάνιση του παραθύρου των αποτελεσμάτων αποστολής και δημιουργεί το πλαίσιο ελέγχου για τον εκτυπωτή

### 3.4.3 Φάκελος CSS

Περιέχει το αρχείο style.css το οποίο περιέχει τα css της σελίδας

### 3.4.4 Φάκελος Scripts

Περιέχει τα αρχεία JavaScript που χρησιμοποιεί η εφαρμογή

### 3.4.5 Φάκελος professor

Η σελίδα του χρήστη με ρόλο professor

Όνομα αρχείου	Περιγραφή
professor Form.aspx	Η σελίδα του χρήστη με ρόλο PROFESSOR
web.config	Το αρχείο ρυθμίσεων για τον φάκελο professor

### 3.4.6 Φάκελος CEO

Η σελίδα του χρήστη με δικαιώματα CEO

Όνομα αρχείου	Περιγραφή
ceoForm.aspx	Η σελίδα του χρήστη με ρόλο CEO
web.config	Το αρχείο ρυθμίσεων για τον φάκελο ceo

### 3.4.7 Φάκελος secretariat

Όνομα αρχείου	Περιγραφή
secretariatForm.aspx	Η σελίδα του χρήστη με ρόλο SECRETARY
web.config	Το αρχείο ρυθμίσεων για τον φάκελο secretariat

## 3.5 Τυποποίηση εφαρμογής

Η εμφάνιση των σελίδων (layout) της εφαρμογής καθορίζεται από την master page.master. Οι master pages είναι templates μέσω των οποίων ορίζεται ένα σταθερό περιεχόμενο και δηλώνουν μια περιοχή της σελίδας όπου ο προγραμματιστής μπορεί να εισάγει το δυναμικό περιεχόμενο, όπως αυτό διαμορφώνεται κατά την εκτέλεση της εφαρμογής. Το πλεονέκτημα που παρέχει η Master page είναι πως αν γίνει μια αλλαγή στην master page αυτή θα έχει εφαρμογή σε όλες τις σελίδες.

Η master page περιέχει το μενού το λογότυπο και το υποσέλιδο της εφαρμογής. Στον html κώδικα της ίδιας σελίδας περιέχονται όλες οι αναφορές σε css και js αρχεία που χρησιμοποιεί η εφαρμογή.

### 3.5.1 Login.aspx

Η σελίδα login.aspx είναι η πρώτη σελίδα που φορτώνει όταν κάποιος εισάγει το url της εφαρμογής σε ένα πρόγραμμα περιήγησης ιστοσελίδων. Η σελίδα αυτή περιέχει το στοιχείο ελέγχου login view και το στοιχείο ελέγχου login.

Το στοιχείο **login view** χωρίζεται σε δύο μέρη. Το **<Anonymous Template>** και το **<LoggedInTemplate>**. Το πρώτο περιέχει το περιεχόμενο που βλέπει ένας μη πιστοποιημένος χρήστης και το δεύτερο περιέχει το περιεχόμενο που βλέπει ο χρήστης όταν εισέρχεται στο σύστημα.

Το στοιχείο **login** περιέχει δύο βασικές παραμέτρους. Την **on authenticate** και την **on login error**

```
<asp: Login ID="myLogin" runat="server" RememberMeSet="True" TitleText=""
UserNameLabelText="Username:" onauthenticate="myLogin_Authenticate"
onloginerror="myLogin_LoginError">
```

Όπως βλέπουμε και στον κώδικα παραπάνω, η παράμετρος **onauthenticate** περιέχει την τιμή **myLogin\_Authenticate**. Η τιμή αυτή είναι το όνομα της μεθόδου που βρίσκεται στην κλάση **login** στο αρχείο **login.aspx.cs** το οποίο περιέχει τον κώδικα σε **C#** για την σελίδα αυτή.

```
protected void myLogin_Authenticate(object sender, AuthenticateEventArgs e)
{
    // Get the email address entered
    Login myLogin = (Login)LoginView1.FindControl("myLogin");
    Textbox Email Textbox = myLogin.FindControl("Email") as Textbox;
    string email = EmailTextBox.Text.Trim();
    // ΠΡΟΣΘΗΚΗ TRYCATCH ΓΙΑ ΤΟΝ ΕΛΕΓΧΟ ΤΗΣ ΣΥΝΔΕΣΗΣ ΜΕ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ
    try
    {
        // Verify that the username/password pair is valid
        if (Membership.ValidateUser(myLogin.UserName, myLogin.Password))
        {
            // Username/password are valid, check email
            MembershipUser usrInfo = Membership.GetUser(myLogin.UserName);

            if (usrInfo != null && string.Compare(usrInfo.Email, email,
true) == 0)
            {
                //sessions
                //HttpContext.Current.Session["roleOfTheUser"] =
Roles.GetRolesForUser(myLogin.UserName) [0];
                //HttpContext.Current.Session["nameOfTheUser"] =
myLogin.UserName;

                // δημιουργία cookie
                HttpCookie cookie = Request.Cookies["basicElements"];
                if (cookie != null)
                {
                    Response.Cookies["basicElements"].Expires =
DateTime.Now.AddDays (-1);
                }

                cookie = new HttpCookie("basicElements");
                cookie["roleOfTheUser"] =
Roles.GetRolesForUser(myLogin.UserName) [0];
                cookie["nameOfTheUser"] = myLogin.UserName;
                cookie.Expires = DateTime.Now.AddYears(1);
                Response.Cookies.Add(cookie);

                // Email matches, the credentials are valid
                e.Authenticated = true;
            }
        }
    }
}
```

```

        else
        {
            // Email address is invalid...
            e.Authenticated = false;
        }
    }
    else
    {
        // Username/password are not valid...
        e.Authenticated = false;
    }
}

catch ( Exception ex )
{
    //myLogin.Failure Text = "ΑΠΕΤΥΧΕ Η ΣΥΝΔΕΣΗ ΜΕ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ.";
    //ex.Message;
}
}

```

Ο παραπάνω κώδικας υλοποιεί την κλάση **myLogin\_Authenticate**.

Οι δύο αυτές παράμετροι, του αντικειμένου Login, αποτελούν τα συμβάντα της κλάσης login τα οποία εγείρονται όταν ο χρήστης πατήσει το πλήκτρο αποστολή της φόρμας εισόδου.

Αν τα στοιχεία είναι σωστά

**if (Membership.ValidateUser(myLogin.UserName, myLogin.Password))**

τότε ελέγχεται αν και το email είναι σωστό

**if (usrInfo != null && string.Compare(usrInfo.Email, email, true) == 0).**

Αν και το email είναι σωστό τότε η παράμετρο authenticated γίνεται true και δημιουργείτε ένα cookie με τα στοιχεία του χρήστη, διαφορετικά γίνεται false και εγείρεται το συμβάν **onloginerror** και εκτελείται η μέθοδο **myLogin\_LoginError** της οποίας ο κώδικας παραθέτετε παρακάτω

```

protectedvoid myLogin_LoginError(object sender, EventArgs e)
{
    // Determine why the user could not login...
    Login myLogin = (Login)LoginView1.FindControl("myLogin");
    myLogin.FailureText = "Your login attempt was not successful. Please try
again.";

    // ΠΡΟΣΘΗΚΗ TRY CATCH ΓΙΑ ΤΟΝ ΕΛΕΓΧΟ ΤΗΣ ΣΥΝΔΕΣΗΣ ΜΕ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ
    try
    {
        // Does there exist a User account for this user?
        MembershipUser usrInfo = Membership.GetUser(myLogin.UserName);
        if (usrInfo != null)
        {
            // Is this user locked out?
            if (usrInfo.IsLockedOut)
            {
                myLogin.FailureText = "Your account has been locked out
because of too many invalid login attempts. Please contact the administrator to have
your account unlocked.";
            }
        }
    }
}

```

```

    }
elseif (!usrInfo.IsApproved)
    {
        myLogin.FailureText = "Your account has not yet been approved.
You cannot login until an administrator has approved your account.";
    }
}
}
catch (Exception ex)
    {
        myLogin.FailureText = "ΑΠΕΤΥΧΗΣΥΝΔΕΣΗΜΕΤΗΒΑΣΗΔΕΛΟΜΕΝΩΝ.";
//ex.Message;
    }
}
}

```

Η μέθοδο αυτή εμφανίζει και το ανάλογο μήνυμα στην οθόνη μέσω του πεδίου myLogin.

Τέλος, αν ο χρήστης πιστοποιηθεί από το σύστημα, για να ανακατευθυνθεί στη σωστή σελίδα με βάση τον ρόλο του, στο αρχείο login.aspx μέσα στην περιοχή της ετικέτας <LoggedInTemplate>υπάρχει ο παρακάτω κώδικας

```

HttpCookie cookie = Request.Cookies["basicElements"];
    if (cookie != null)
    {
        usrRole = cookie["roleOfTheUser"];
    }

    switch (usrRole)
    {
        case "professor":
            Response.Redirect("professor/professorForm.aspx", true);
            break;

        case "ceo":
            Response.Redirect("professor/ceoForm.aspx", true);
            break;

        case "secretariat":
            Response.Redirect("secretariat/secretariatForm.aspx", true);
            break;

        case "student":
            Response.Redirect("student/studentForm.aspx", true);
            break;

        default:
            Response.Redirect("default.aspx", true);
    }
break;
}

```

Ο κώδικας αυτός, με βάση την ιδιότητα του χρήστη, όπως αυτή αποθηκεύτηκε στο cookie, ανακατευθύνει τον χρήστη στη σωστή σελίδα.

### 3.5.2 Σελίδα ceoForm.aspx

Η σελίδα αυτή αφορά τους χρήστη σε ρόλο ceo. Κληρονομεί το template της από την master page όπως φαίνεται και από τον κώδικα παρακάτω

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.Master"
AutoEventWireup="true" CodeBehind="ceoForm.aspx.cs"
Inherits="ilabMailServ.ceo.ceoForm" %>
```

Η σελίδα αυτή περιέχει όπως και η login.aspx το στοιχείο **loginview** στο οποίο περιέχεται η ετικέτα **<LoggedInTemplate>** η οποία περιέχει τα πεδία που απαιτούνται για την αποστολή μηνυμάτων. Στιγμιότυπο της σελίδας απεικονίζεται στην εικόνα 3.17

The screenshot shows the user interface for composing an email. At the top, there is a navigation menu with links for 'Αρχική', 'Διοίκηση', 'Γραμματεία', and 'Καθηγητής'. Below the menu is a 'Logout' button. The main content area is titled 'e - mail από Διοίκηση' and is divided into three steps:

- 1ο Βήμα:** 'Θέμα' (Subject) with a text input field.
- 2ο Βήμα:** 'Περιεχόμενο' (Content) with a large text area.
- 3ο Βήμα:** 'Υπογραφή' (Signature) with a text input field.

On the right side, there is a 'Παραλήπτες' (Recipients) section with a dropdown menu for 'Επιλογή με βάση:' (selected 'Τμήμα') and another dropdown for 'Τμήμα' (selected 'Επιλέξτε Τμήμα...'). Below this is a large 'ΑΠΟΣΤΟΛΗ' (Send) button and a smaller 'Ιστορικό' (History) button. At the bottom of the recipients section, there is a link for '...Περιοχή μηνυμάτων'.

At the bottom of the page, there is a footer with the text: '© 2014 Fotios Vartzolis. All Rights Reserved.'

Εικ 3.17: Οθόνη του χρήστη με ρόλο CEO

Μια σημαντική μέθοδο μιας .aspx σελίδας είναι η Page\_Load.



```
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        DDLhide();
    }
}
```

Το συμβάν **IsPostBack** μας επιτρέπει να ελέγξουμε αν η σελίδα φορτώνει για πρώτη φορά π.χ. μετά από κλικ σε ένα σύνδεσμο ή φορτώνει μετά από αποστολή στοιχείων του χρήστη π.χ. submit τα δεδομένα μιας φόρμας. Στον παραπάνω κώδικα, η μέθοδος **DDL Hide()**, η οποία κρύβει τα πεδία φιλτραρίσματος, εκτελείται μόνο όταν ο χρήστης φορτώσει τη σελίδα, χωρίς να στείλει δεδομένα σε αυτή.

Οι επιπλέον επιλογές φιλτραρίσματος εμφανίζονται μόνο όταν ο χρήστης επιλέξει μία επιλογή από το select box του πεδίου με τίτλο «Επιλογή με βάση» Ανάλογα με την επιλογή που κάνει εμφανίζονται και τα επιπλέον πεδία. Την εμφάνιση των πεδίων αυτών την καθορίζει η μέθοδος

**protectedvoidDDLChoose\_SelectedIndexChanged(objectsender, EventArgs).**

Με βάση τη θέση της επιλογής –πρώτο δεύτερο κτλ – επιλέγεται ποια πεδία θα εμφανιστούν. Αυτό πετυχαίνεται με τις παρακάτω γραμμές κώδικα

```
switch (Int32.Parse(DDLSelect.Selected.Value))
{
    case 1:
        DDLDept.Visible = true;
        DDLDept.Enabled = true;
        theDept.Visible = true;
        LoadDepts();
        break;
    case 2:
        DDLDept.Visible = true;
        DDLDept.Enabled = true;
        theDept.Visible = true;
        LoadDepts();
        break;
    case 3:
        DDLPriority.Visible = true;
        DDLPriority.Enabled = true;
        thePriority.Visible = true;
        break;
    default:
        Console.WriteLine("Default case");
        break;
}
```

Αντίστοιχη μέθοδος υπάρχει για το πεδίο των τμημάτων

```
protected void DDLDept_SelectedIndexChanged(object sender, EventArgs e)
```

για το πεδίο των εξαμήνων

```
protected void DDLSemester_SelectedIndexChanged(object sender, EventArgs e)
```

και για το πεδίο της προτεραιότητας

```
protected void DDLPriority_SelectedIndexChanged(object sender, EventArgs e)
{
    Button email Submit = (Button)LoginView2.FindControl("email Submit");
    email Submit.Enabled = true;
}
```

Η επιλογή αυτή εμφανίζεται τελευταία και με την επιλογή προτεραιότητας ενεργοποιείται το πλήκτρο **ΑΠΟΣΤΟΛΗ**.

Το συμβάν του πλήκτρου αποστολή εκτελεί η μέθοδο

```
protected void emailSubmit_Click(object sender, EventArgs e)
```

Στο συμβάν αυτό, ελέγχεται η τιμή **DDLSelect.SelectedValue** του πρώτου πεδίου. Ανάλογα με την τιμή της επιλέγονται και τα κατάλληλα δεδομένα βάση των επιλογών από τα άλλα πεδία.

```
try
{
    switch (Int32.Parse(DDLSelect.SelectedValue))
    {
        case 1:
            selectedDepts = getddlselections(DDLDept);
            foreach (var DeptItem in selectedDepts)
            {
                DatasetBinding(out ds, SelectUserNamesByDepID(DeptItem));
                if (ds != null)
                {
                    for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
                    {
                        selectedUsers.Add(ds.Tables[0].Rows[i][0].ToString());
                    }
                }
            }
        }
    }
}
```

```

        Console.WriteLine("Case Τμήμα");

        break;

    case 2:

        selectedDepts = getddlselections(DDLDept);

        selectedSemesters = getddlselections(DDLSemester);

        foreach (var DeptItem in selectedDepts)

        {

            foreach (var SemesterItem in selectedSemesters)

            {

                DatasetBinding(out ds,
SelectUserNamesByDepIDAndExam(DeptItem, Int32.Parse(SemesterItem)));

                if (ds != null)

                {

                    for (int i = 0; i < ds.Tables[0].Rows.Count; i++)

                    {

selectedUsers.Add(ds.Tables[0].Rows[i][0].ToString());

                    }

                }

            }

        }

        Console.WriteLine("Case DeptUserNames");

        break;

    case 3:

        DataSet Deptds = new DataSet();

        DataSet Userds = new DataSet();

        DatasetBinding(out Deptds, Select Tiep Departments());

        if (Deptds != null)

        {

            for (int i = 0; i < Deptds.Tables[0].Rows.Count; i++)

            {

                DatasetBinding(out Userds,
SelectUserNamesByDepID(ds.Tables[0].Rows[i][0].ToString()));

                if (Userds != null)

                {

```

```

        for (int j = 0; j < Userds.Tables[0].Rows.Count;
j++)
        {
            selectedUsers.Add(Userds.Tables[0].Rows[j][0].ToString());
        }
    }
}

Console.WriteLine("Case All");

break;

default:
    Console.WriteLine("Default case");

    break;
}
}

catch (SqlException ex)
{
    error.Text = ex.Message;
}
}

```

Για κάθε μια από τις παραπάνω περιπτώσεις υπάρχει αντίστοιχη συνάρτηση που συλλέγει τα δεδομένα που απαιτούνται.

Έτσι, για την πρώτη επιλογή η μέθοδο **Select User Names By DepID (DeptItem)** συλλέγει τους παραλήπτες με βάση το τμήμα που επιλέχθηκε, η μέθοδος **Select UserNames By DepID(DeptItem)** συλλέγει τους χρήστες με βάση το τμήμα και το εξάμηνο και για την περίπτωση 3 καλείτε πάλι η συνάρτηση **Select User Names By DepID (SelectUserNamesByDepID(DeptItem))** με την διαφορά πως σε αυτή την περίπτωση συλλέγει όλους τους χρήστες από όλα τα τμήματα.

Στη συνέχεια γίνεται έλεγχος στα πεδία του μηνύματος

```

if (header.Text == "")
{
    error.Text = "Εισάγετε θέμα";
}
else if (content.Text == "")
{
    error.Text = "Εισάγετε περιεχόμενο";
}
else if (signature.Text == "")
{
    error.Text = "Εισάγετε υπογραφή";
}
}

```

```

else
{
    /*
    * αποστολη μηνυματος
    */
    foreach (var user in selectedUsers)
    {
        //SendMail(user, header.Text, content.Text, signature.Text,
        Int32.Parse(DDLPriority.SelectedValue));

        x = MailSender.SendEmail("teleinfomlab@gmail.com", "password", user,
        header.Text, content.Text, System.Web.Mail.MailFormat.Html, "");
        if (!x)
        {
            error.Text = "Σφάλμα κατά την αποστολή";
        }
        else
        {
            recipeint += user + ", ";
        }
        mailcounter++;

        /*
        * καθυστέρηση για κάθε 100 email
        */
        if (mailcounter == 100)
        {
            mailcounter = 0;
            System.Threading.Thread.Sleep(milliseconds);
        }

    }
    msgSent = true;
    Login Login1 = (Login)LoginView2.FindControl("Login1");

    string ConnString =
    ConfigurationManager.ConnectionStrings["LocaliLabMailServConnectionString"].Connection
    String;
    SqlConnection conn = new SqlConnection(ConnString);

    logQuery = "INSERT INTO LOG_TABLE VALUES ('" + "ceo" + "', '" + recipeint + "',
    '" + header.Text + "', '" + DateTime.Now.ToString("d/M/yyyy") + "')";

    SqlCommand mycomm = new SqlCommand(logQuery, conn);

    try
    {
        conn.Open();
        mycomm.ExecuteNonQuery();
        conn.Close();
    }
    catch (SqlException SqlExc)
    {
        error.Text = "Σφάλμα στην επικοινωνία. Προσπαθήστε αργότερα";
    }
}

```

και αν δεν βρεθεί σφάλμα εκτελείτε ο κώδικας στο else όπου αποστέλλεται το μήνυμα στους παραλήπτες.

Η αποστολή του μηνύματος εκτελείται μέσω της κλήσης του αντικειμένου **MailSender**

```

x = MailSender.SendEmail("teleinfomlab@gmail.com", "password", user, header.Text, content.Text,
System.Web.Mail.MailFormat.Html, "");

```

Η μεταβλητή **recipeint** αποθηκεύει τους παραλήπτες του μηνύματος και το περιεχόμενό της καταχωρείτε στον πίνακα `log_table`

```
logQuery = "INSERT INTO LOG_TABLE VALUES ('" + ceo + "', '" + recipeint + "', '" + header.Text + "', '" + DateTime.Now.ToString("d/M/yyyy") + "')";
```

Τέλος, από τον παρακάτω κώδικα καλείται η σελίδα `print.aspx` η οποία προβάλλει τα email των παραληπτών.

```
selusers.DataSource = selectedUsers;

        selusers.DataBind();

        Session["ctrl"] = selusers;

        ClientScript.RegisterStartupScript(this.GetType(), "onclick", "<script language=javascript>window.open('../Print.aspx','PrintMe','height=300px,width=300px,scrollbars=1');</script>");
```

Η σελίδα ανοίγει σε νέο παράθυρο και κατά το φόρτωμα της δημιουργείται το αντικείμενο **PrintHelper** το οποίο εμφανίζει το παράθυρο διαλόγου του εκτυπωτή ώστε ο χρήστης να επιλέξει τον εκτυπωτή που θέλει για να εκτυπώσει τη λίστα με τα emails των παραληπτών του μηνύματος.

Για την προβολή του ιστορικού, εκτελείτε το συμβάν `click` στο πλήκτρο ιστορικό

```
protected void Button2_Click(object sender, EventArgs e)

    {

        ClientScript.RegisterStartupScript(this.GetType(), "onclick", "<script language=JavaScript>window.Open('../email_history.aspx','Ιστορικό Αποστολών','height=500px,width=800px,scrollbars=1');</script>");

    }
```

όπου εκτελείται κώδικας JavaScript που ανοίγει νέο παράθυρο στο οποίο φορτώνει η σελίδα `email_history.aspx`.

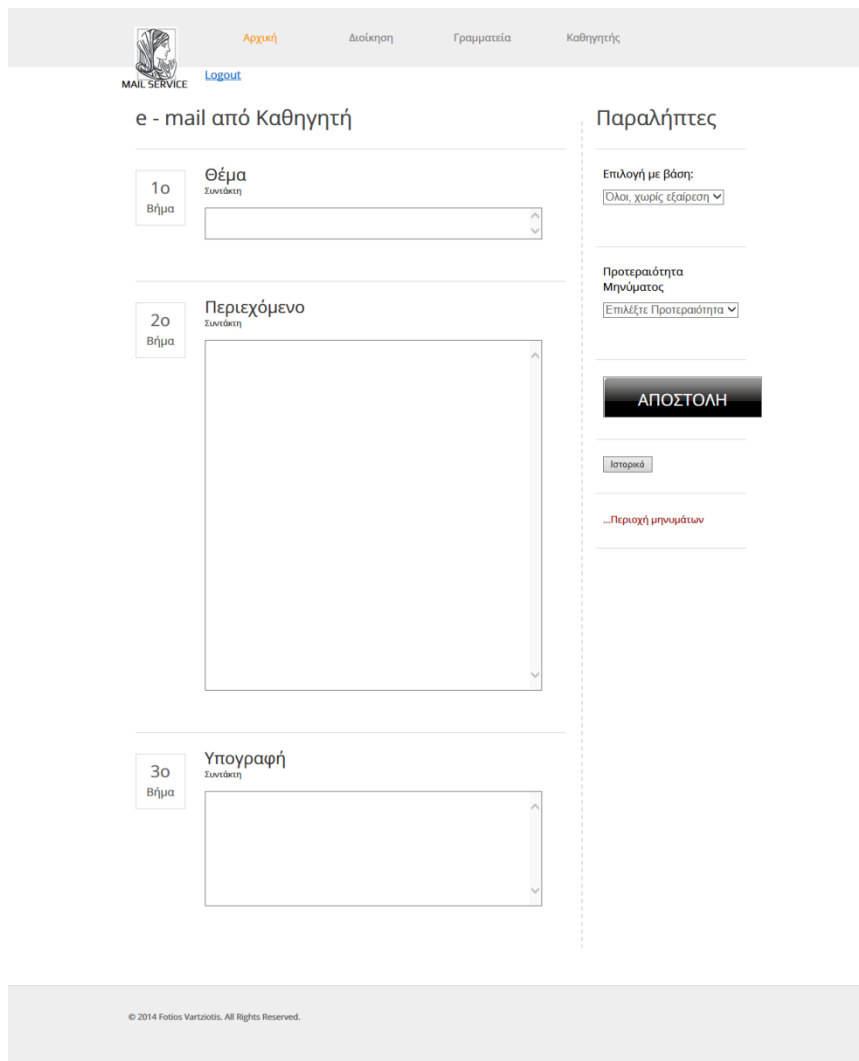
### 3.5.3 Σελίδα professor Form.aspx

Η σελίδα αυτή αφορά τους χρήστες σε ρόλο professor. Η σελίδα αυτή κληρονομεί το template της από την master page όπως φαίνεται και από τον κώδικα παρακάτω

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.Master"
AutoEventWireup="true" CodeBehind="professorForm.aspx.cs"
Inherits="ilabMailServ.professor.professorForm" %>
```

Το στοιχείο **loginview** χρησιμοποιείται και σε αυτή τη σελίδα, και η ετικέτα **<LoggedInTemplate>** περιέχει τα πεδία που απαιτούνται για την αποστολή μηνυμάτων.

Και σε αυτή τη σελίδα η μέθοδο Page\_Load εκτελεί την ίδια λειτουργία με αυτή της σελίδας ceoForm.aspx.



Εικ. 3.18: Η οθόνη του χρήστη με ρόλο professor

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        DDHide();
    }
}
```

Οι επιπλέον επιλογές φιλτραρίσματος εμφανίζονται μόνο όταν ο χρήστης επιλέξει μία επιλογή από το select box του πεδίου με τίτλο «Επιλογή με βάση». Ανάλογα με την επιλογή που κάνει εμφανίζονται και τα επιπλέον πεδία. Την εμφάνιση των πεδίων αυτών την καθορίζει η μέθοδο

```
protected void DDChoose_SelectedIndexChanged(object sender, EventArgs e) .
```

Με βάση τη θέση του πεδίου –πρώτο δεύτερο κτλ – επιλέγεται ποια πεδία θα εμφανιστούν. Αυτό πετυχαίνεται με τις παρακάτω γραμμές κώδικα. Στον ρόλο αυτό, όπως φαίνεται και στον παρακάτω κώδικα, τα φίλτρα των δεδομένων αποτελούν τα πεδία **DDLYear**, και **DDLSemester** ενώ η Τρίτη επιλογή περιλαμβάνει όλους τους χρήστες.

```
switch (Int32.Parse(DDLSelect.SelectedValue))
{
    case 1:
        DDLYear.Visible = true;
        DDLYear.Enabled = true;
        theYear.Visible = true;
        LoadYears(DDLYear);
        break;
    case 2:
        DDLSemester.Visible = true;
        DDLSemester.Enabled = true;
        theSemester.Visible = true;
        LoadSemesters(DDLSemester);
        break;
    case 3:
        DDLPriority.Visible = true;
        DDLPriority.Enabled = true;
        thePriority.Visible = true;
        break;
    default:
        Console.WriteLine("Default case");
        break;
}
```

Το συμβάν SelectedIndexChanged εκτελείται στα πεδία έτος

```
protected void DDLYear_SelectedIndexChanged(object sender, EventArgs e)
```

περίοδο

```
protected void DDLPeriod_SelectedIndexChanged(object sender, EventArgs e)
```

Μαθήματα

```
protected void DDLCourse_SelectedIndexChanged(object sender, EventArgs e)
```

και για το πεδίο της προτεραιότητας

```
protected void DDLPriority_SelectedIndexChanged(object sender, EventArgs e)
{
    Button emailSubmit = (Button)LoginView2.FindControl("emailSubmit");
    emailSubmit.Enabled = true;
}
```

Η επιλογή αυτή, εμφανίζεται τελευταία. Στον κώδικα της ενεργοποιείται το πλήκτρο ΑΠΟΣΤΟΛΗ.



Το συμβάν του πλήκτρου αποστολή εκτελεί η μέθοδο

```
protected void emailSubmit_Click(object sender, EventArgs e)
```

Στο συμβάν αυτό, ελέγχεται η τιμή **DDLSelect.SelectedValue** του πρώτου πεδίου. Ανάλογα με την τιμή της επιλέγονται και τα κατάλληλα δεδομένα βάση των επιλογών από τα άλλα πεδία.

```
switch (Int32.Parse(DDLSelect.SelectedValue))
{
    case 1:
        selectedClass = getddlselections(DDLCourse);
        foreach (var ClassItem in selectedClass)
        {
            /*
             * ΓΙΑΤΙ ΕΠΙΛΕΓΕΙ ΤΟ ΠΕΔΙΟ comment για depID;;;;;
             * ΟΡΙΣΑ ΣΤΑΘΕΡΗ ΤΙΜΗ ΓΙΑ ΤΟ DEPID
             * */
            //DatasetBinding(out ds,
            SelectClassUserNames(Membership.GetUser(User.Identity.Name.ToString()).Comment,
            ClassItem));
            DatasetBinding(out ds, SelectClassUserNames("1", ClassItem));
            if (ds != null)
            {
                for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
                {
                    selectedUsers.Add(ds.Tables[0].Rows[i][0].ToString());
                }
            }
            Console.WriteLine("Case Μάθημα");
            break;
        case 2:
            selectedSemesters = getddlselections(DDLSemester);
            foreach (var SemesterItem in selectedSemesters)
            {
                /*
                 * ΓΙΑΤΙ ΕΠΙΛΕΓΕΙ ΤΟ ΠΕΔΙΟ comment για depID;;;;;
                 * ΟΡΙΣΑ ΣΤΑΘΕΡΗ ΤΙΜΗ ΓΙΑ ΤΟ DEPID
                 * */
                //DatasetBinding(out ds,
                SelectUserNamesByDepIDAndExam(Membership.GetUser(User.Identity.Name.ToString()).Comment,
                Int32.Parse(SemesterItem)));
                DatasetBinding(out ds, SelectUserNamesByDepIDAndExam("1",
                Int32.Parse(SemesterItem)));
                if (ds != null)
                {
                    for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
                    {
                        selectedUsers.Add(ds.Tables[0].Rows[i][0].ToString());
                    }
                }
            }
            Console.WriteLine("Case DeptUserNames");
            break;
        case 3:
            /*
             * ΓΙΑΤΙ ΕΠΙΛΕΓΕΙ ΤΟ ΠΕΔΙΟ comment για depID;;;;;
             * ΟΡΙΣΑ ΣΤΑΘΕΡΗ ΤΙΜΗ ΓΙΑ ΤΟ DEPID
             * */
            //DatasetBinding(out ds,
            SelectUserNamesByDepID(Membership.GetUser(User.Identity.Name.ToString()).Comment));
            DatasetBinding(out ds, SelectUserNamesByDepID("1"));

```

```

        if (ds != null)
        {
            for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
            {
                selectedUsers.Add(ds.Tables[0].Rows[i][0].ToString());
            }
        }
        Console.WriteLine("Case All");
        break;
    default:
        Console.WriteLine("Default case");
        break;
}

```

Για κάθε μια από τις παραπάνω περιπτώσεις υπάρχει αντίστοιχη συνάρτηση που συλλέγει τα δεδομένα που απαιτούνται.

Έτσι, για την πρώτη επιλογή η μέθοδο

```

SelectClassUserNames (Membership.GetUser (User.Identity.Name.ToString()) .Comment,
ClassItem)

```

Συλλέγει τους παραλήπτες με βάση το τμήμα που επιλέχθηκε , η μέθοδο

```

SelectUserNamesByDepIDAndExam (Membership.GetUser (User.Identity.Name.ToString()) .Comment,
Int32.Parse (SemesterItem) )

```

συλλέγει τους χρήστες με βάση το τμήμα και το εξάμηνο και για την περίπτωση 3 καλείτε πάλι η συνάρτηση

```

SelectUserNamesByDepIDAndExam (Membership.GetUser (User.Identity.Name.ToString()) .Comment,
Int32.Parse (SemesterItem) )

```

με την διαφορά πως σε αυτή την περίπτωση συλλέγει όλους τους χρήστες από όλα τα τμήματα του χρήστη.

Στη συνέχεια γίνεται έλεγχος στα πεδία του μηνύματος

```

if (header.Text == "")
{
    error.Text = "Εισάγετε θέμα";
}
else if (content.Text == "")
{
    error.Text = "Εισάγετε περιεχόμενο";
}
else if (signature.Text == "")
{
    error.Text = "Εισάγετε υπογραφή";
}
else
{
    /*foreach (var user in selectedUsers)
    {
        SendMail(user, header.Text, content.Text, signature.Text,
Int32.Parse(DDLPriority.SelectedValue));
    }*/

    /*
    * αποστολη μηνυματος
    */
    foreach (var user in selectedUsers)
    {

```

```

//SendMail(user, header.Text, content.Text, signature.Text,
Int32.Parse(DDLPriority.SelectedValue));

x = MailSender.SendEmail("teleinfomlab@gmail.com", "password", user,
header.Text, content.Text, System.Web.Mail.MailFormat.Html, "");
if (!x)
{
    error.Text = "Σφάλμα κατά την αποστολή";
}
else
{
    receipt += user + ", ";
}
mailcounter++;

/*
 * καθυστέρηση για κάθε 100 email
 */
if (mailcounter == 100)
{
    mailcounter = 0;
    System.Threading.Thread.Sleep(milliseconds);
}

}

Login Login1 = (Login)LoginView2.FindControl("Login1");

string ConnString =
ConfigurationManager.ConnectionStrings["LocaliLabMailServConnectionString"].Connection
String;
SqlConnection conn = new SqlConnection(ConnString);

logQuery = "INSERT INTO LOG_TABLE VALUES ('" + "professor" + "', '" + receipt
+ "', '" + header.Text + "', '" + DateTime.Now.ToString("d/M/yyyy") + "')";

SqlCommand mycomm = new SqlCommand(logQuery, conn);

try
{
    conn.Open();
    mycomm.ExecuteNonQuery();
    conn.Close();
}
catch (SqlException SqlExc)
{
    error.Text = "Σφάλμα στην επικοινωνία. Προσπαθήστε αργότερα";
}

error.Text = "Το μήνυμά σας έχει σταλεί";
}

```

και αν δεν βρεθεί σφάλμα εκτελείτε ο κώδικας στο else όπου αποστέλλεται το μήνυμα στους παραλήπτες.

Η αποστολή του μηνύματος γίνεται μέσω της κλήσης του αντικειμένου MailSender

```
x = MailSender.SendEmail("teleinfomlab@gmail.com", "password", user, header.Text,
content.Text, System.Web.Mail.MailFormat.Html, "");
```

Η μεταβλητή receipt αποθηκεύει τους παραλήπτες του μηνύματος και το περιεχόμενό της καταχωρείτε στον πίνακα log\_table

```
logQuery = "INSERT INTO LOG_TABLE VALUES ('" + "professor" + "', '" + receipt + "',
'" + header.Text + "', '" + DateTime.Now.ToString("d/M/yyyy") + "')";
```

Τέλος, από τον παρακάτω κώδικα καλείται η σελίδα `print.aspx` η οποία προβάλλει τα email των παραληπτών.

```
selusers.DataSource = selectedUsers;
selusers.DataBind();
Session["ctrl"] = selusers;
ClientScript.RegisterStartupScript(this.GetType(), "onclick", "<script
language=javascript>window.open('../Print.aspx','PrintMe','height=300px,width=300px,sc
rollbars=1');</script>");
```

Η σελίδα ανοίγει σε νέο παράθυρο και κατά το φόρτωμα της δημιουργείτε το αντικείμενο `PrintHelper` το οποίο εμφανίζει το παράθυρο διαλόγου του εκτυπωτή ώστε ο χρήστης να επιλέξει τον εκτυπωτή που θέλει για να εκτυπώσει τη λίστα με τα emails των παραληπτών του μηνύματος.

Για την προβολή του ιστορικού, εκτελείτε το συμβάν `click` στο πλήκτρο ιστορικό

```
protectedvoid Button2_Click(object sender, EventArgs e)
{
    ClientScript.RegisterStartupScript(this.GetType(), "onclick", "<script
language=JavaScript>window.                    Open('../email_history.aspx','Ιστορικό
Αποστολών','height=500px, width=800px, scrollbars=1');</script>");
}
```

όπου εκτελείται κώδικας JavaScript που ανοίγει νέο παράθυρο στο οποίο φορτώνει η σελίδα `email_history.aspx`.

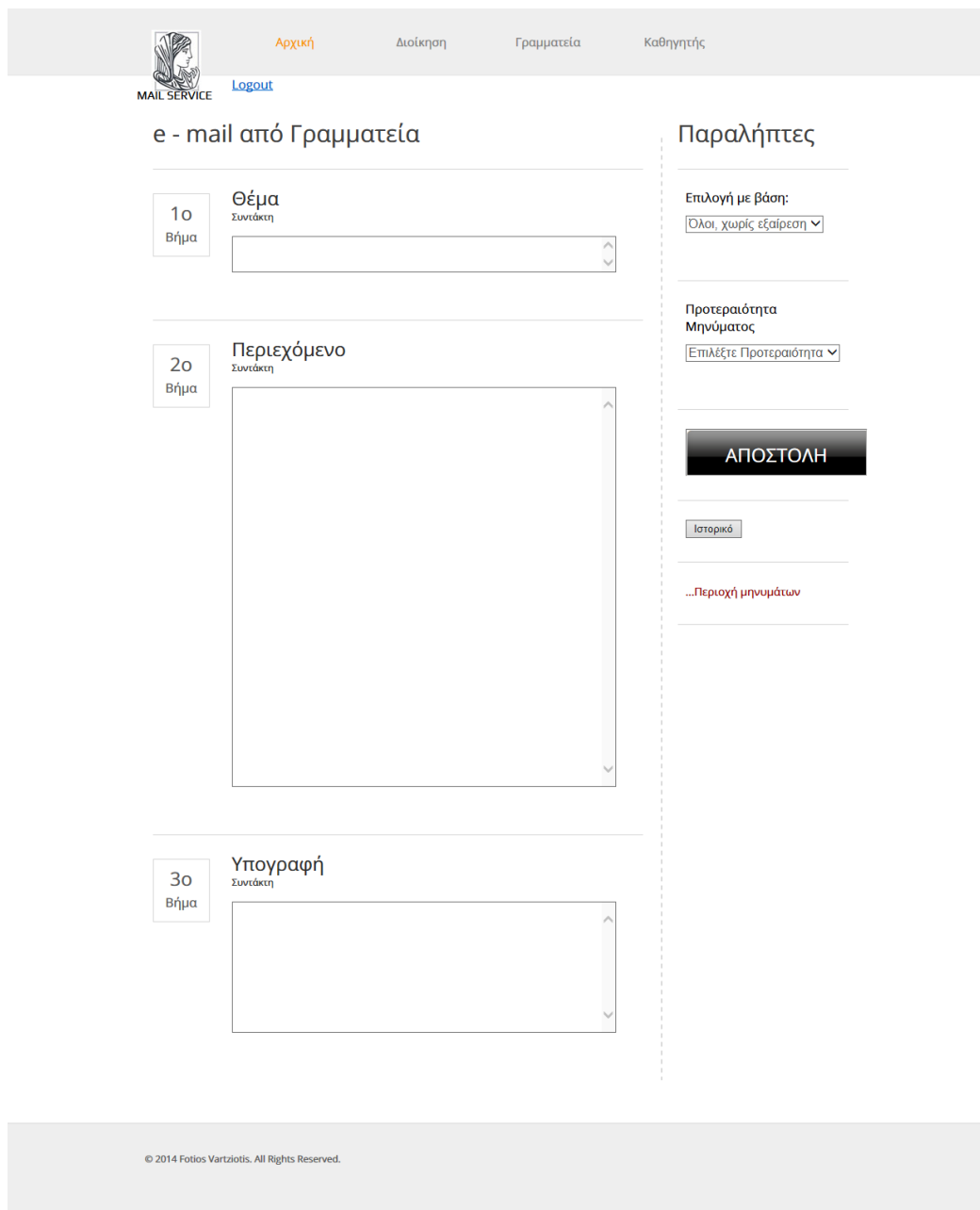
### 3.5.4 Σελίδα `secretariat.aspx`

Η σελίδα αυτή αφορά τους χρήστες σε ρόλο `secretariat`. Η σελίδα αυτή κληρονομεί το `template` της από την `master page` όπως φαίνεται και από τον κώδικα παρακάτω

```
<%@PageTitle=""Language="C#"MasterPageFile="~/MasterPage.Master"AutoEventWireup="true"
CodeBehind="secretariatForm.aspx.cs"Inherits="ilabMailServ.secretariat.secretariatForm
"%>
```

Το στοιχείο `loginview` χρησιμοποιείται και σε αυτή τη σελίδα, και η ετικέτα `<LoggedInTemplate>` περιέχει τα πεδία που απαιτούνται για την αποστολή μηνυμάτων.

Και σε αυτή τη σελίδα η μέθοδος `Page_Load` εκτελεί την ίδια λειτουργία με αυτή της σελίδας `ceoForm.aspx`.



MAIL SERVICE [Logout](#)

Αρχική Διοίκηση Γραμματεία Καθηγητής

## e - mail από Γραμματεία

**1ο Βήμα** **Θέμα**  
Συντάκτη

**2ο Βήμα** **Περιεχόμενο**  
Συντάκτη

**3ο Βήμα** **Υπογραφή**  
Συντάκτη

**Παραλήπτες**

Επιλογή με βάση:  
Όλοι, χωρίς εξαίρεση

Προτεραιότητα Μηνύματος  
Επιλέξτε Προτεραιότητα

**ΑΠΟΣΤΟΛΗ**

Ιστορικό

[...Περιοχή μηνυμάτων](#)

© 2014 Fotios Vartziotis. All Rights Reserved.

Εικ. 3.19: Οθόνη του χρήστη με ρόλο secretariat

```
protectedvoid Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        DDLhide();
    }
}
```

Οι επιπλέον επιλογές φιλτραρίσματος εμφανίζονται μόνο όταν ο χρήστης επιλέξει μία επιλογή από το select box του πεδίου με τίτλο «Επιλογή με βάση» Ανάλογα με την επιλογή που κάνει εμφανίζονται και τα επιπλέον πεδία. Την εμφάνιση των πεδίων αυτών την καθορίζει η μέθοδος

```
protected void DDLChoose_SelectedIndexChanged(object sender, EventArgs e) .
```

Με βάση τη θέση του πεδίου –πρώτο δεύτερο κτλ – επιλέγεται ποια πεδία θα εμφανιστούν. Αυτό πετυχαίνεται με τις παρακάτω γραμμές κώδικα. Στον ρόλο αυτό, όπως φαίνεται και στον παρακάτω κώδικα, τα φίλτρα των δεδομένων αποτελούν τα πεδία DDLYear, και DDLSemester ενώ η Τρίτη επιλογή περιλαμβάνει όλους τους χρήστες.

```
switch (Int32.Parse(DDLSelect.Selected.Value))
{
case 1:
    DDLYear.Visible = true;
    DDLYear.Enabled = true;
    theYear.Visible = true;
    LoadYears(DDLYear);
break;
case 2:
    DDLSemester.Visible = true;
    DDLSemester.Enabled = true;
    theSemester.Visible = true;
    LoadSemesters(DDLSemester);
break;
case 3:
    DDLPriority.Visible = true;
    DDLPriority.Enabled = true;
    thePriority.Visible = true;
break;
default:
    Console.WriteLine("Default case");
break;
}
```

Το συμβάν SelectedIndexChanged εκτελείται στα πεδία έτος

```
protected void DDLYear_SelectedIndexChanged(object sender, EventArgs e)
```

περίοδο

```
protected void DDLPeriod_SelectedIndexChanged(object sender, EventArgs e)
```

Μαθήματα

```
protected void DDLCourse_SelectedIndexChanged(object sender, EventArgs e)
```

και για το πεδίο της προτεραιότητα

```
protected void DDLPriority_SelectedIndexChanged(object sender, EventArgs e)
{
    Button emailSubmit = (Button)LoginView2.FindControl("emailSubmit");
    emailSubmit.Enabled = true;
}
```

Η επιλογή αυτή, εμφανίζεται τελευταία. Στον κώδικα της ενεργοποιείται το πλήκτρο ΑΠΟΣΤΟΛΗ.

Το συμβάν του πλήκτρου αποστολή εκτελεί η μέθοδο

```
protected void emailSubmit_Click(object sender, EventArgs e)
```

Στο συμβάν αυτό, ελέγχεται η τιμή DDLSelect.SelectedValue του πρώτου πεδίου. Ανάλογα με την τιμή της επιλέγονται και τα κατάλληλα δεδομένα βάση των επιλογών από τα άλλα πεδία.

```
switch (Int32.Parse(DDLSelect.SelectedValue))
{
    case 1:
        selectedClass = getddlselections(DDLCourse);
        foreach (var ClassItem in selectedClass)
        {
            /*
                * ΓΙΑΤΙ ΕΠΙΛΕΞΕΙ ΤΟ ΠΕΔΙΟ comment για depID;;;;;
                * ΟΠΙΣΑ ΣΤΑΘΕΡΗ ΤΙΜΗ ΓΙΑ ΤΟ DEPID
            */
            DatasetBinding(out ds,
                SelectClassUserNames(Membership.GetUser(User.Identity.Name.ToString()).Comment,
                ClassItem));
            //DatasetBinding(out ds, SelectClassUserNames("1", ClassItem));
            if (ds != null)
            {
                for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
                {
                    selectedUsers.Add(ds.Tables[0].Rows[i][0].ToString());
                }
            }
            Console.WriteLine("Case Μάθημα");
            break;
        case 2:
            selectedSemesters = getddlselections(DDLSemester);
            foreach (var SemesterItem in selectedSemesters)
            {
                /*
                    * ΓΙΑΤΙ ΕΠΙΛΕΞΕΙ ΤΟ ΠΕΔΙΟ comment για depID;;;;;
                    * ΟΠΙΣΑ ΣΤΑΘΕΡΗ ΤΙΜΗ ΓΙΑ ΤΟ DEPID
                */
                DatasetBinding(out ds,
                    SelectUserNamesByDepIDAndExam(Membership.GetUser(User.Identity.Name.ToString()).Comment,
                    Int32.Parse(SemesterItem)));
                //DatasetBinding(out ds,
                    SelectUserNamesByDepIDAndExam("1",
                    Int32.Parse(SemesterItem)));
                if (ds != null)
                {
                    for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
                    {
```

```

                selectedUsers.Add(ds.Tables[0].Rows[i][0].ToString());
            }
        }
    }
    Console.WriteLine("Case DeptUserNames");
    break;
    case 3:
    /*
        * ΓΙΑΤΙ ΕΠΙΛΕΓΕΙΤΟ ΠΕΔΙΟ comment για depID;;;;;
        * ΟΡΙΣΑΣΤΑ ΘΕΡΤΙΜΗ ΓΙΑΤΟ DEPID
    */
    DatasetBinding(out ds,
    SelectUserNamesByDepID(Membership.GetUser(User.Identity.Name.ToString()).Comment));
    //DatasetBinding(out ds, SelectUserNamesByDepID("1"));
    if (ds != null)
    {
        for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
        {
            selectedUsers.Add(ds.Tables[0].Rows[i][0].ToString());
        }
    }
    Console.WriteLine("Case All");
    break;
    default:
    Console.WriteLine("Default case");
    break;
}
}

```

Για κάθε μια από τις παραπάνω περιπτώσεις υπάρχει αντίστοιχη συνάρτηση που συλλέγει τα δεδομένα που απαιτούνται.

Έτσι, για την πρώτη επιλογή η μέθοδο

```
SelectClassUserNames (Membership.GetUser (User . Identity . Name . ToString ()) . Comment
```

Συλλέγει τους παραλήπτες με βάση το τμήμα που επιλέχθηκε , η μέθοδο

```
SelectUserNamesByDepIDAndExam (Membership.GetUser (User . Identity . Name . ToString ()) . Comment, Int32.Parse (SemesterItem))
```

Συλλέγει τους χρήστες με βάση το τμήμα και το εξάμηνο και για την περίπτωση 3 καλείτε πάλι η συνάρτηση

```
SelectUserNamesByDepIDAndExam (Membership.GetUser (User . Identity . Name . ToString ()) . Comment, Int32.Parse (SemesterItem))
```

με την διαφορά πως σε αυτή την περίπτωση συλλέγει όλους τους χρήστες από όλα τα τμήματα του χρήστη.

Στη συνέχεια γίνεται έλεγχος στα πεδία του μηνύματος

```

if (header.Text == "")
{
    error.Text = "Εισάγετε θέμα";
}
elseif (content.Text == "")
{
    error.Text = "Εισάγετε περιεχόμενο";
}
elseif (signature.Text == "")
{
    error.Text = "Εισάγετε υπογραφή";
}
else

```



```

        {
/*
            foreach (var user in selectedUsers)
            {
                SendMail(user, header.Text, content.Text, signature.Text,
Int32.Parse(DDLPriority.SelectedValue));
            }
*/

/*
            * αποστολημηνυματος
            */
foreach (var user in selectedUsers)
            {
//SendMail(user, header.Text, content.Text, signature.Text,
Int32.Parse(DDLPriority.SelectedValue));

                x = MailSender.SendEmail("teleinfomlab@gmail.com",
"password", user, header.Text, content.Text, System.Web.Mail.MailFormat.Html, "");
if (!x)
                {
                    error.Text = "Σφάλμα κατά την αποστολή";
                }
else
                {
                    recipeint += user + ", ";
                }
                mailcounter++;
            }

/*
            * καθυστέρηση για κάθε 100 email
            */
if (mailcounter == 100)
            {
                mailcounter = 0;
                System.Threading.Thread.Sleep(milliseconds);
            }

        }

Login Login1 = (Login)LoginView2.FindControl("Login1");

string ConnString =
ConfigurationManager.ConnectionStrings["LocaliLabMailServConnectionString"].Connection
String;
SqlConnection conn = new SqlConnection(ConnString);

        logQuery = "INSERT INTO LOG_TABLE VALUES ('" + "secretariat" +
"', '" + recipeint + "', '" + header.Text + "', '" + DateTime.Now.ToString("d/M/yyyy")
+ "')";

SqlCommand mycomm = new SqlCommand(logQuery, conn);

try
        {
            conn.Open();
            mycomm.ExecuteNonQuery();
            conn.Close();
        }
catch (SqlException SqlExc)
{
            error.Text = "Σφάλμα στην επικοινωνία. Προσπαθήστε
αργότερα";
        }
    }
}

```

και αν δεν βρεθεί σφάλμα εκτελείτε ο κώδικας στο else όπου αποστέλλεται το μήνυμα στους παραλήπτες.

Η αποστολή του μηνύματος γίνεται μέσω της κλήσης του αντικειμένου MailSender

```
x = MailSender.SendEmail("teleinfomlab@gmail.com", "password", user, header.Text, content.Text, System.Web.Mail.MailFormat.Html, "");
```

Η μεταβλητή receipt αποθηκεύει τους παραλήπτες του μηνύματος και το περιεχόμενό της καταχωρείτε στον πίνακα log\_table

```
logQuery = "INSERT INTO LOG_TABLE VALUES ('" + "secretariat" + "', '" + receipt + "', '" + header.Text + "', '" + DateTime.Now.ToString("d/M/yyyy") + "')";
```

Τέλος, από τον παρακάτω κώδικα καλείται η σελίδα printing.aspx η οποία προβάλλει τα email των παραληπτών.

```
selusers.DataSource = selectedUsers;

selusers.DataBind();

Session["ctrl"] = selusers;

ClientScript.RegisterStartupScript(this.GetType(), "onclick", "<script language=javascript>window.open('../Print.aspx', 'PrintMe', 'height=300px,width=300px,scrollbars=1');</script>");
```

Η σελίδα ανοίγει σε νέο παράθυρο και κατά το φόρτωμα της δημιουργείτε το αντικείμενο [PrintHelper](#) το οποίο εμφανίζει το παράθυρο διαλόγου του εκτυπωτή ώστε ο χρήστης να επιλέξει τον εκτυπωτή που θέλει για να εκτυπώσει τη λίστα με τα emails των παραληπτών του μηνύματος.

Για την προβολή του ιστορικού, εκτελείτε το συμβάν click στο πλήκτρο ιστορικό

```
protected void Button2_Click(object sender, EventArgs e)

{

    ClientScript.RegisterStartupScript(this.GetType(), "onclick", "<script language=JavaScript>window. Open('../email_history.aspx', 'Ιστορικό Αποστολών', 'height=500px,width=800px,scrollbars=1');</script>");

}
```

όπου εκτελείται κώδικας JavaScript που ανοίγει νέο παράθυρο στο οποίο φορτώνει η σελίδα email\_history.aspx.

### 3.5.5 Σελίδα email\_history

Η σελίδα αυτή προβάλλει το ιστορικό του χρήστη. Εμφανίζεται σε ξεχωριστό παράθυρο από αυτό της εφαρμογής. Ο χρήστης μπορεί να δει το ιστορικό των αποστολών του πατώντας στο πλήκτρο «ιστορικό» στη σελίδα αποστολής μηνύματος.

Η μέθοδος Page\_Load περιέχει τον κώδικα της εμφάνισης των δεδομένων. Τα στοιχεία του χρήστη ανακτώνται μέσω του cookie που δημιουργείται κατά την είσοδο του χρήστη.

Τα δεδομένα, εφόσον υπάρχουν, αποθηκεύονται σε ένα dataset αντικείμενο. Το dataset παράγεται από την μέθοδο

```
DataSet Get History(String query)
```

το οποίο διοχετεύεται ως data source στο αντικείμενο data grid της σελίδας, όπως φαίνεται και στον κώδικα παρακάτω.

```
DataSet ds = GetHistory(query);
if (ds.Tables.Count > 0)
{
    HistoryGrid.DataSource = ds;
    HistoryGrid.DataBind();
}
else
{
    errMsg.Text = "ΔΕΝΒΡΕΘΗΚΑΝΔΕΔΟΜΕΝΑ";
}
```

Η παρακάτω εικόνα απεικονίζει το ιστορικό του χρήστη με δικαιώματα professor

**professor**

Θέμα	Παραλήπτες	Ημερομηνία
sdfsdf	user1@tei.gr, user2@tei.gr, user3@tei.gr,	20/2/2015
ssssssssss	user1@tei.gr, user2@tei.gr, user3@tei.gr,	20/2/2015
sdfsdf	user1@tei.gr, user2@tei.gr, user3@tei.gr,	20/2/2015
asdas	user1@tei.gr, user2@tei.gr, user3@tei.gr,	13/6/2015
asdasd	user14@tei.gr,	22/6/2015
sdfsdf	user14@tei.gr,	22/6/2015

Εικ. 3.20:Ιστορικό του χρήστη με ρόλο professor

### 3.5.6 web.config

Το web.config είναι ένα Xml αρχείο το οποίο περιέχει διάφορες ρυθμίσεις για την ιστοσελίδα.

Για παράδειγμα, το παρακάτω δηλώνει στην εφαρμογή το connection string για τη σύνδεση με τη βάση δεδομένων

```
<connection Strings>
```

```
<add name="LocaliLabMailServConnectionString" connectionString="Data
Source=MACHINE1\SQLEXPRESS;Initial Catalog=iLabMailServ;User ID=sa;Password=0000"
providerName="System.Data.SqlClient"/>

</connectionStrings>
```

Το κομμάτι αυτό του web.config υποδηλώνει ότι θέλουμε να έχουμε debugmode ενεργό και η έκδοση του framework είναι η 4.

```
<compilation debug="true" targetFramework="4.0"/>
```

### Και κάποιες ρυθμίσεις για τον smtp

```
<mailSettings>

<smtp deliveryMethod="Network" from="teleinfomlab@gmail.com">

<network defaultCredentials="false" host="smtp.gmail.com"
userName="teleinfomlab@gmail.com" password="foti#bar"/>

</smtp>

</mailSettings>
```

### Οι ρυθμίσεις για το membership

```
<membership>

<providers>

<clear/>

<add name="AspNetSqlMembershipProvider"
type="System.Web.Security.SqlMembershipProvider, System. Web, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
connectionStringName="LocaliLabMailServConnectionString"
enablePasswordRetrieval="false" enablePasswordReset="true"
requiresQuestionAndAnswer="true" applicationName="/" requiresUniqueEmail="false"
passwordFormat="Hashed" maxInvalidPasswordAttempts="5" minRequiredPasswordLength="7"
minRequiredNonalphanumericCharacters="1" passwordAttemptWindow="10"
passwordStrengthRegularExpression=""/>

</providers>

</membership>
```

### Τέλος ο τρόπος πιστοποίησης των χρηστών

```
<authentication mode="Forms"/>
```

## 4. Οδηγίες Χρήσης

Παρακάτω περιγράφονται όλα τα βήματα που πρέπει να ακολουθήσει ένας χρήστης για να στείλει email. Κάθε χρήστης, ανάλογα την ιδιότητα του, έχει ορισμένες επιλογές ως προς το φιλτράρισμα των παραληπτών. Να σημειωθεί εδώ πως η εφαρμογή αυτή αποστέλλει ομαδικά μηνύματα και όχι ατομικά. Ο συντάκτης επιλέγει την ομάδα που θα αποστείλει το μήνυμα χωρίς να εισάγει παραλήπτες χειροκίνητα.

Γράψε απλά...

© 2014 Fotios Vartziotis. All Rights Reserved.

Εικ. 4.1: Η αρχική σελίδα

Στην εικόνα 4.1 η οποία είναι η αρχική σελίδα της εφαρμογής, ο χρήστης, για να μπορέσει να χρησιμοποιήσει την εφαρμογή, θα πρέπει να εισάγει τα στοιχεία του στα αντίστοιχα πεδία της φόρμας. Η πιστοποίηση, εκτός από το όνομα χρήστη και τον κωδικό πρόσβασης περιλαμβάνει και ένα ακόμη πεδίο, το email του.

Το πεδίο remember me next time διατηρεί το session ώστε όταν ξανανοίξει το πρόγραμμα περιήγησης να φορτώσει η σελίδα του.

Κάθε χρήστης έχει την δική του οθόνη. Στους χρήστες ανατίθεται και ένας ρόλος ανάλογα με την ιδιότητα του. Η εφαρμογή διαθέτει τρεις ρόλους χρηστών.

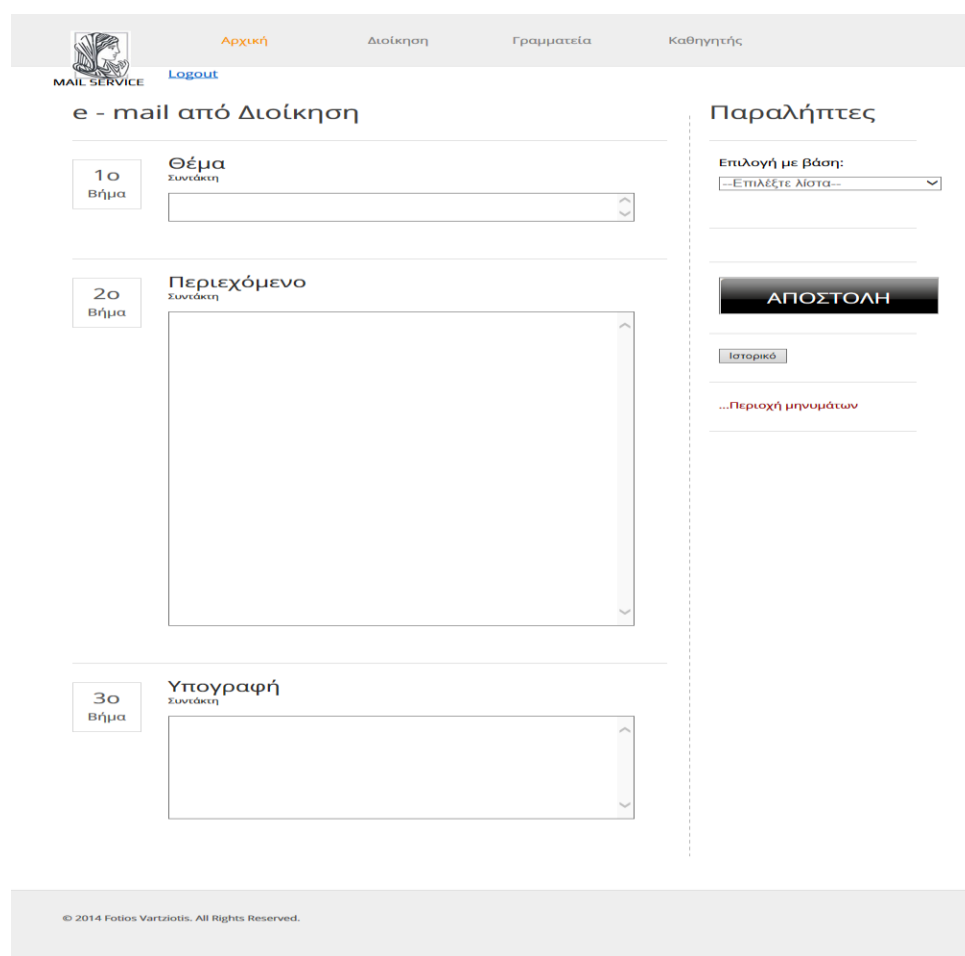
- CEO
- PROFESSOR
- SECRETARY

Υπάρχει και μια κατηγορία χρηστών «σπουδαστές» οι οποίοι είναι οι παραλήπτες των μηνυμάτων που αποστέλλονται από τους χρήστες της εφαρμογής. Η κατηγορία αυτή των χρηστών δεν έχει πρόσβαση στην εφαρμογή.

#### 4.1 CEO

Ο χρήστης με δικαιώματα CEO μπορεί να στείλει μηνύματα σε όλες τις ομάδες του οργανισμού.

Η αρχική σελίδα του χρήστη με δικαιώματα CEO φαίνεται στην εικόνα 4.2



MAIL SERVICE Logout

Αρχική Διοίκηση Γραμματεία Καθηγητής

e - mail από Διοίκηση

1ο Βήμα **Θέμα**  
Συντόκη

2ο Βήμα **Περιεχόμενο**  
Συντόκη

3ο Βήμα **Υπογραφή**  
Συντόκη

Παραλήπτες

Επιλογή με βάση:  
--Επιλέξτε λίστα--

ΑΠΟΣΤΟΛΗ

Ιστορικό

...Περιοχή μηνυμάτων

© 2014 Fotios Vartziotis. All Rights Reserved.

Εικ. 4.2: Η αρχική σελίδα του χρήστη με δικαιώματα CEO

Η σελίδα χωρίζεται σε δύο τμήματα. Στο αριστερό τμήμα, τα πεδία με τίτλους θέμα, περιεχόμενο, υπογραφή αφορούν το μήνυμα που θα αποσταλεί. Η δεξιά στήλη με τίτλο παραλήπτες περιέχει ένα σύνολο πεδίων που αφορούν τους τελικούς παραλήπτες του μηνύματος.

Το πρώτο πεδίο περιέχει τρεις επιλογές.

Επιλογή με βάση

- το τμήμα
- το τμήμα και εξάμηνο φοιτητών
- όλοι χωρίς εξαίρεση

Η πρώτη αφορά το τμήμα που ανήκει ο φοιτητής η δεύτερη αφορά τμήμα και εξάμηνο και η Τρίτη δεν κάνει καμία διάκριση.

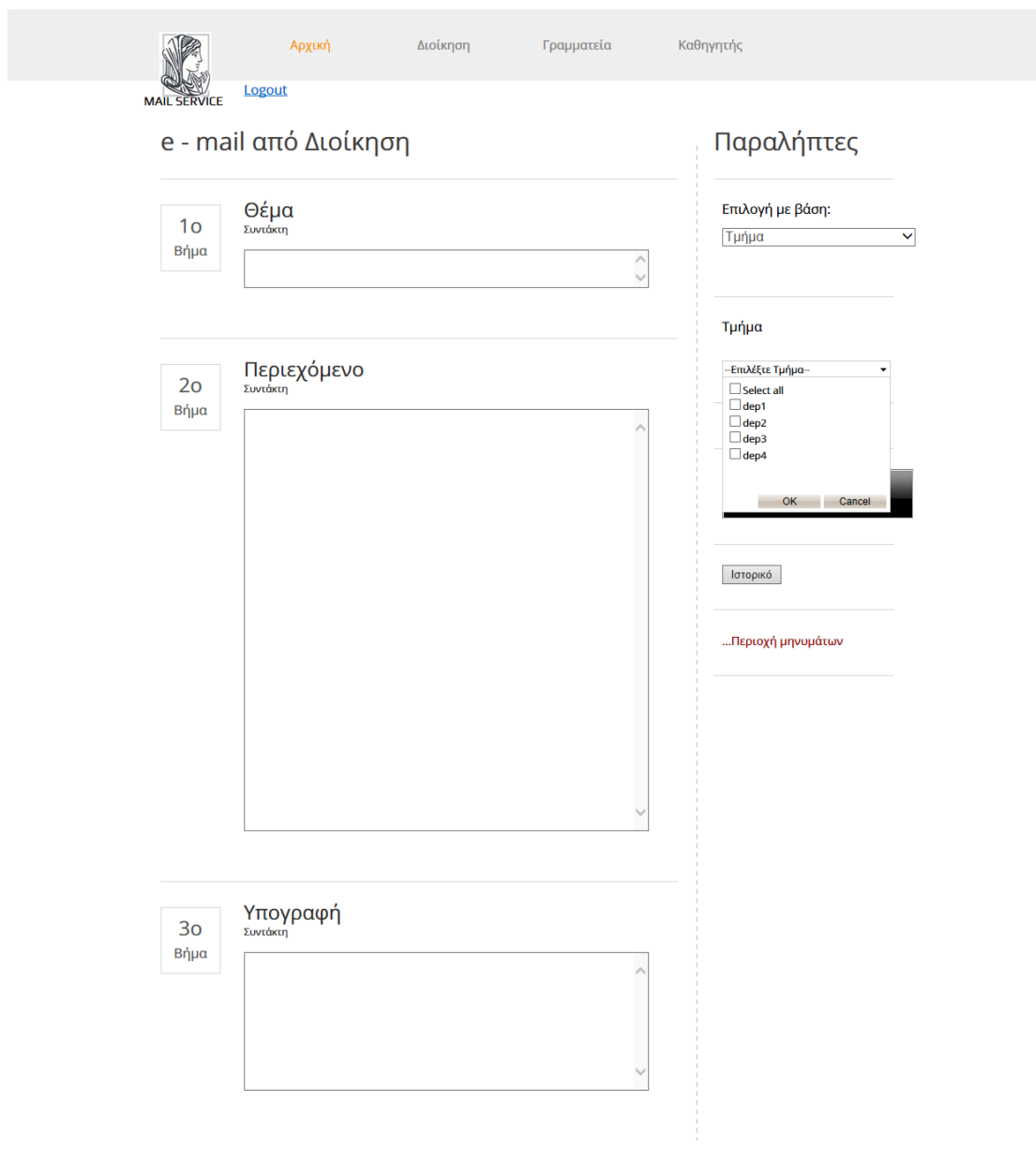
Η επιλογή στο πρώτο πεδίο καθορίζει την εμφάνιση των επόμενων φίλτρων.

Έτσι, αν επιλεγεί το τμήμα, το επόμενο φίλτρο που θα εμφανιστεί θα είναι μόνο το πεδίο τμήμα όπως φαίνεται και στην εικόνα 4.3.

The screenshot shows an email composition interface. At the top, there is a navigation bar with a logo and links for 'Αρχική', 'Διοίκηση', 'Γραμματεία', and 'Καθηγητής'. Below this, the text 'e - mail από Διοίκηση' is displayed. The main area is divided into three sections: '1ο Βήμα' (Step 1) for 'Θέμα' (Subject), '2ο Βήμα' (Step 2) for 'Περιεχόμενο' (Content), and '3ο Βήμα' (Step 3) for 'Υπογραφή' (Signature). On the right side, there is a 'Παραλήπτες' (Recipients) sidebar. It includes a dropdown menu for 'Επιλογή με βάση:' (Select by) with 'Τμήμα' (Department) selected. Below it is another dropdown for 'Τμήμα' (Department) with the value '-Επιλέξτε Τμήμα-' (Select Department). A large 'ΑΠΟΣΤΟΛΗ' (SEND) button is present, along with a smaller 'Ιστορικό' (History) button and a link for '...Περιοχή μηνυμάτων' (Message area).

Εικ. 4.3: Πεδίο τμήμα

Οι διαθέσιμες επιλογές φαίνονται στην εικόνα 4.4



MAIL SERVICE [Logout](#)

Αρχική Διοίκηση Γραμματεία Καθηγητές

### e - mail από Διοίκηση

**1ο Βήμα** **Θέμα**  
Συντάκτη

**2ο Βήμα** **Περιεχόμενο**  
Συντάκτη

**3ο Βήμα** **Υπογραφή**  
Συντάκτη

### Παραλήπτες

Επιλογή με βάση:  
Τμήμα

Τμήμα

-Επιλέξτε Τμήμα-

Select all

dep1

dep2

dep3

dep4

OK Cancel

Ιστορικό

...Περιοχή μηνυμάτων

© 2014 Fotios Vartziotis. All Rights Reserved.

Εικ. 4.4: Οι επιλογές του χρήστη για το τμήμα.

Όπως φαίνεται και στην εικόνα 4.4 ο χρήστης μπορεί να επιλέξει ένα ή περισσότερα τμήματα, ενώ η επιλογή select all τα επιλέγει όλα.

Αφού επιλέξει από τη λίστα – με ok καταχωρούνται οι επιλογές – εμφανίζεται το πεδίο προτεραιότητα, εικόνα 4.5.



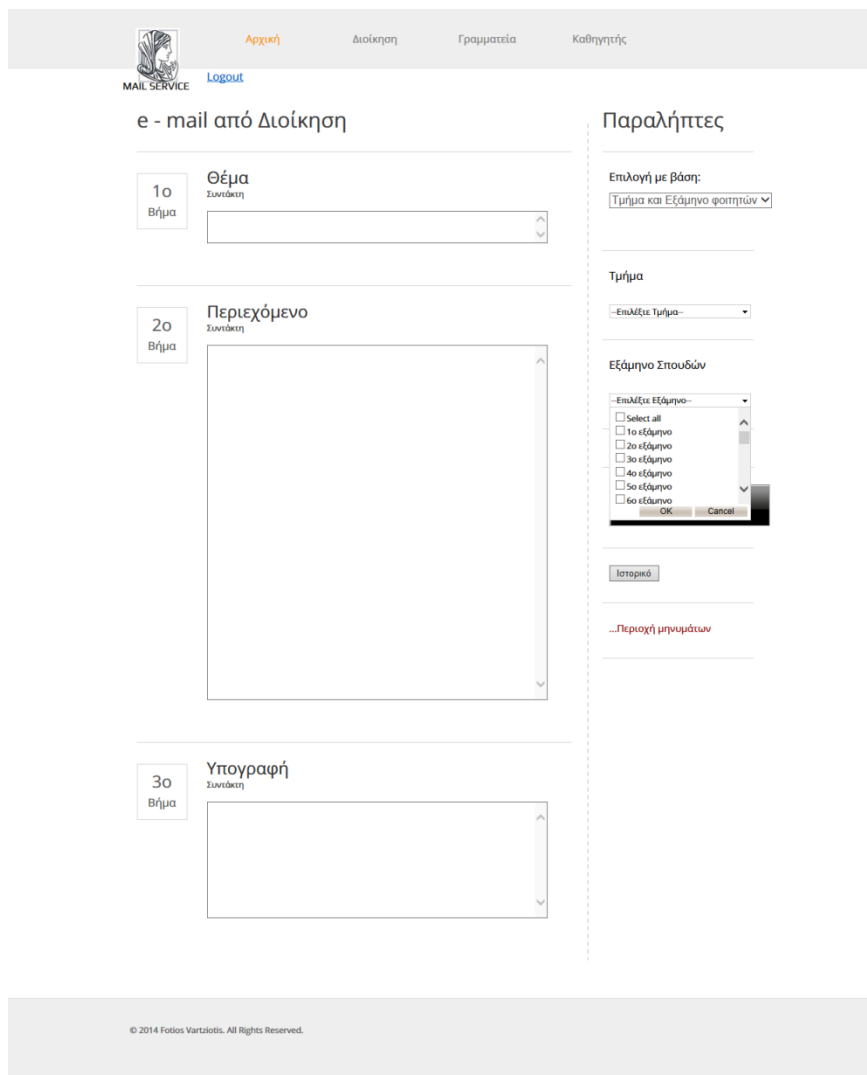
The screenshot shows a web interface for sending an email. The main content area is titled 'e - mail από Διοίκηση' and is divided into three steps: '1ο Βήμα' (Subject), '2ο Βήμα' (Content), and '3ο Βήμα' (Signature). On the right side, there is a 'Παραλήπτες' (Recipients) section with a dropdown menu for 'Επιλογή με βάση:' (Select by) set to 'Τμήμα' (Department). Below this, there is a 'Προτεραιότητα Μηνύματος' (Message Priority) dropdown menu with the option 'Επιλέξτε Προτεραιότητα' (Select Priority) highlighted. A 'ΑΠΟΣΤΟΛΗ' (SEND) button is located below the priority selection. At the bottom of the page, there is a copyright notice: '© 2014 Fotios Vartzolis. All Rights Reserved.'

Εικ. 4.5: Η επιλογή ορισμού προτεραιότητας.

Στο πεδίο αυτό ο χρήστης μπορεί να επιλέξει μεταξύ των παρακάτω τριών επιλογών

- μεγάλη
- κανονική
- μικρή

Σε περίπτωση που ο χρήστης, στην αρχή, επιλέξει την επιλογή «τμήμα και εξάμηνο», τότε μετά την επιλογή του τμήματος (εικόνα 4.4), θα εμφανιστεί το πεδίο που περιέχει τα εξάμηνα (εικόνα 4.6).



MAIL SERVICE Logout

Αρχική Διοίκηση Γραμματεία Καθηγητής

e - mail από Διοίκηση

1ο Βήμα **Θέμα** Συντάκτη

2ο Βήμα **Περιεχόμενο** Συντάκτη

3ο Βήμα **Υπογραφή** Συντάκτη

Παραλήπτες

Επιλογή με βάση:  
Τμήμα και Εξάμηνο φοιτητών

Τμήμα  
Επιλέξτε Τμήμα

Εξάμηνο Σπουδών  
Επιλέξτε Εξάμηνο

Select all  
 1ο εξάμηνο  
 2ο εξάμηνο  
 3ο εξάμηνο  
 4ο εξάμηνο  
 5ο εξάμηνο  
 6ο εξάμηνο

OK Cancel

Ιστορικό

...Περιοχή μηνυμάτων

© 2014 Fotios Vartziotis. All Rights Reserved.

Εικ. 4.6: Επιλογή εξαμήνων

Και εδώ ο χρήστης μπορεί να επιλέξει ένα ή περισσότερα εξάμηνα κατοχυρώνοντας την επιλογή του πατώντας στο OK.

Αφού συμπληρωθούν και τα πεδία του μηνύματος (εικόνα 4.7), ο χρήστης αποστέλλει το μήνυμα πατώντας στο κουμπί αποστολή. Όπως απεικονίζεται και στην εικόνα 4.7 το κουμπί αποστολή έχει αλλάξει χρώμα πράγμα που σημαίνει ότι έχουν γίνει οι κατάλληλες επιλογές και το μήνυμά μας είναι έτοιμο για αποστολή.

MAIL SERVICE [Logout](#)

Αρχική Διοίκηση Γραμματεία Καθηγητής

### e - mail από Διοίκηση

**1ο Βήμα** **Θέμα**  
Συντάκτη

Δοκιμαστικό μήνυμα

**2ο Βήμα** **Περιεχόμενο**  
Συντάκτη

Δοκιμαστικό μήνυμα της εφαρμογής.

**3ο Βήμα** **Υπογραφή**  
Συντάκτη

υπογραφή συντάκτη

### Παραλήπτες

Επιλογή με βάση:  
Τμήμα και Εξάμηνο φοιτητών

Τμήμα  
-Επιλέξτε Τμήμα-

Εξάμηνο Σπουδών  
-Επιλέξτε Εξάμηνο-

Προτεραιότητα Μηνύματος  
Κανονική

**ΑΠΟΣΤΟΛΗ**

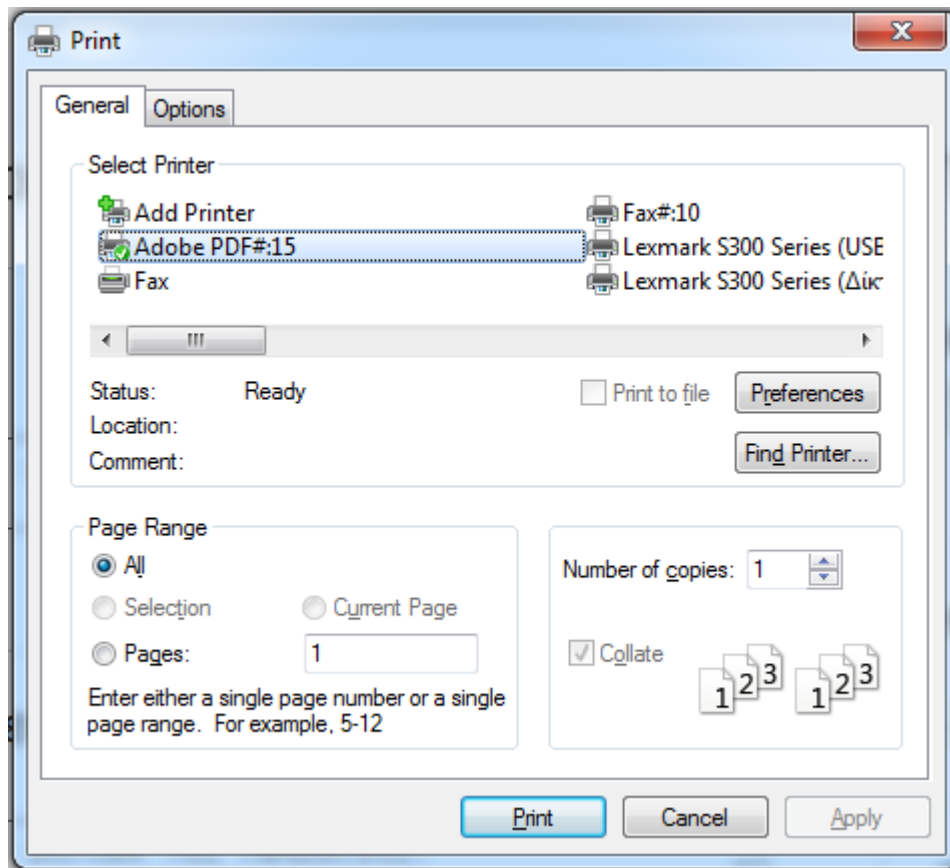
Ιστορικό

[...Περιοχή μηνυμάτων](#)

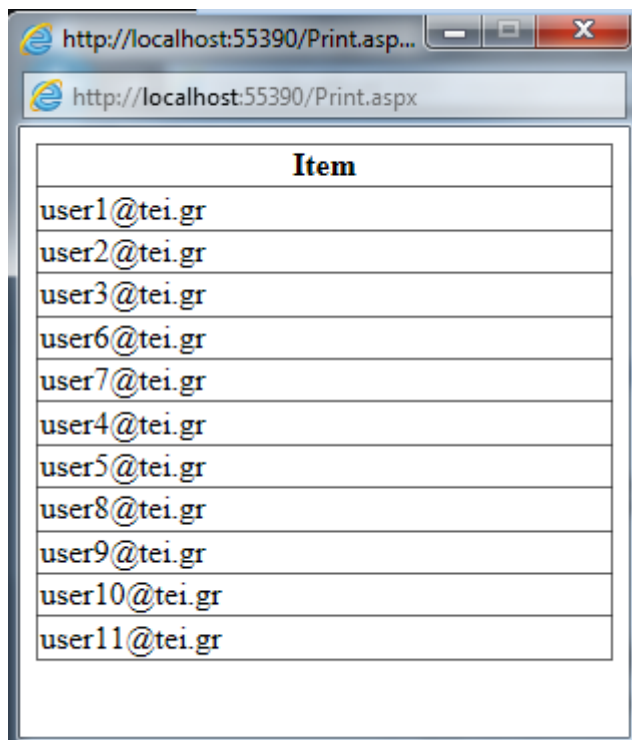
© 2014 Fotios Vartziotis. All Rights Reserved.

Εικ. 4.7: Το μήνυμα είναι έτοιμο για αποστολή.

Μετά την αποστολή, η εφαρμογή μας εμφανίζει τους παραλήπτες (εικόνα 4.9) στους οποίους αποστέλλει το μήνυμα σε ξεχωριστό παράθυρο. Μας εμφανίζει επίσης και παράθυρο διαλόγου (εικόνα 4.8) για εκτύπωση της λίστας των παραληπτών.

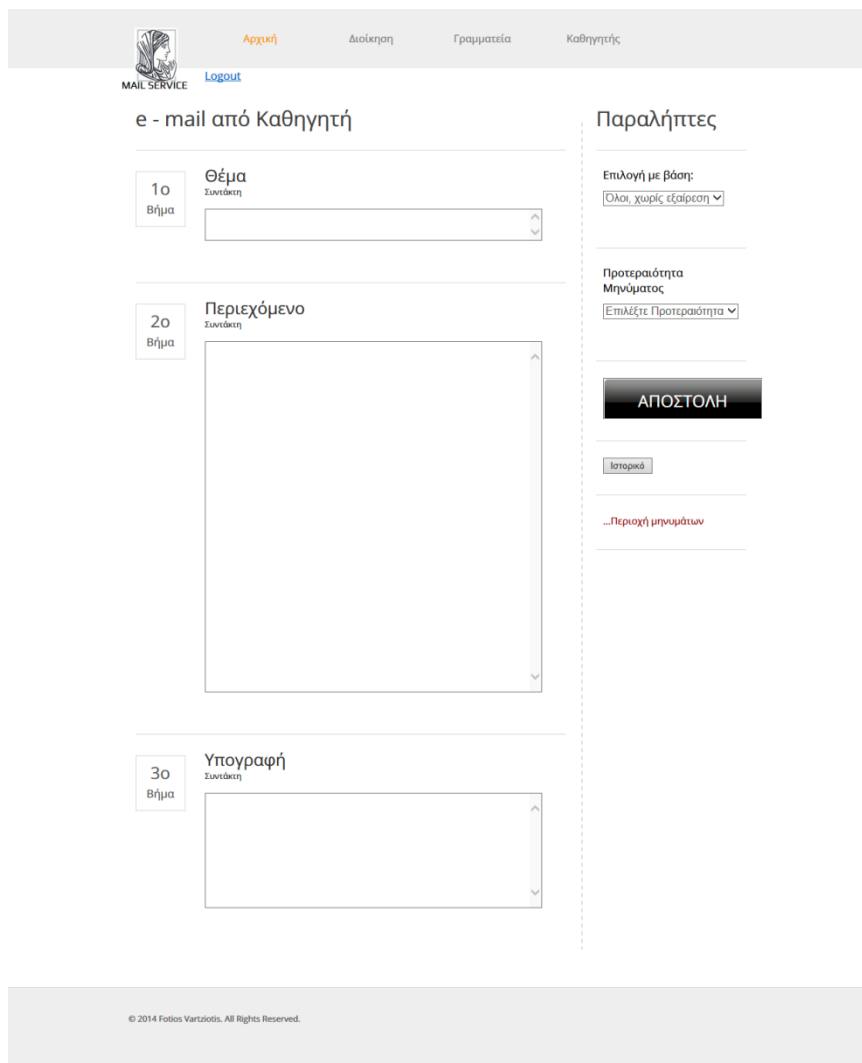


Εικ. 4.8: Παράθυρο διαλόγου για εκτύπωση της λίστας παραληπτών



Εικ. 4.9: Οι λίστα παραληπτών

Στην περίπτωση που ο χρήστης επιλέξει στο πεδίο «Επιλογή με βάση» την επιλογή «όλοι, χωρίς εξαίρεση» (εικόνα 4,10), τότε, το μήνυμα θα το λάβουν όλες οι ομάδες του οργανισμού.



Εικ. 4.10: Αποστολή μηνύματος σε όλους τους χρήστες

Μια επιπρόσθετη επιλογή που έχει ο χρήστης είναι και η προβολή του ιστορικού. Η προβολή γίνεται πατώντας στο πλήκτρο ιστορικό που βρίσκετε κάτω από το πλήκτρο αποστολή.

Το ιστορικό εμφανίζεται σε νέο παράθυρο στην παρακάτω μορφή

Θέμα	Παραλήπτες	Ημερομηνία
Δοκιμαστικό μήνυμα	user1@tei.gr, user2@tei.gr, user3@tei.gr, user6@tei.gr, user7@tei.gr, user4@tei.gr, user5@tei.gr, user8@tei.gr, user9@tei.gr,	11/2/2015

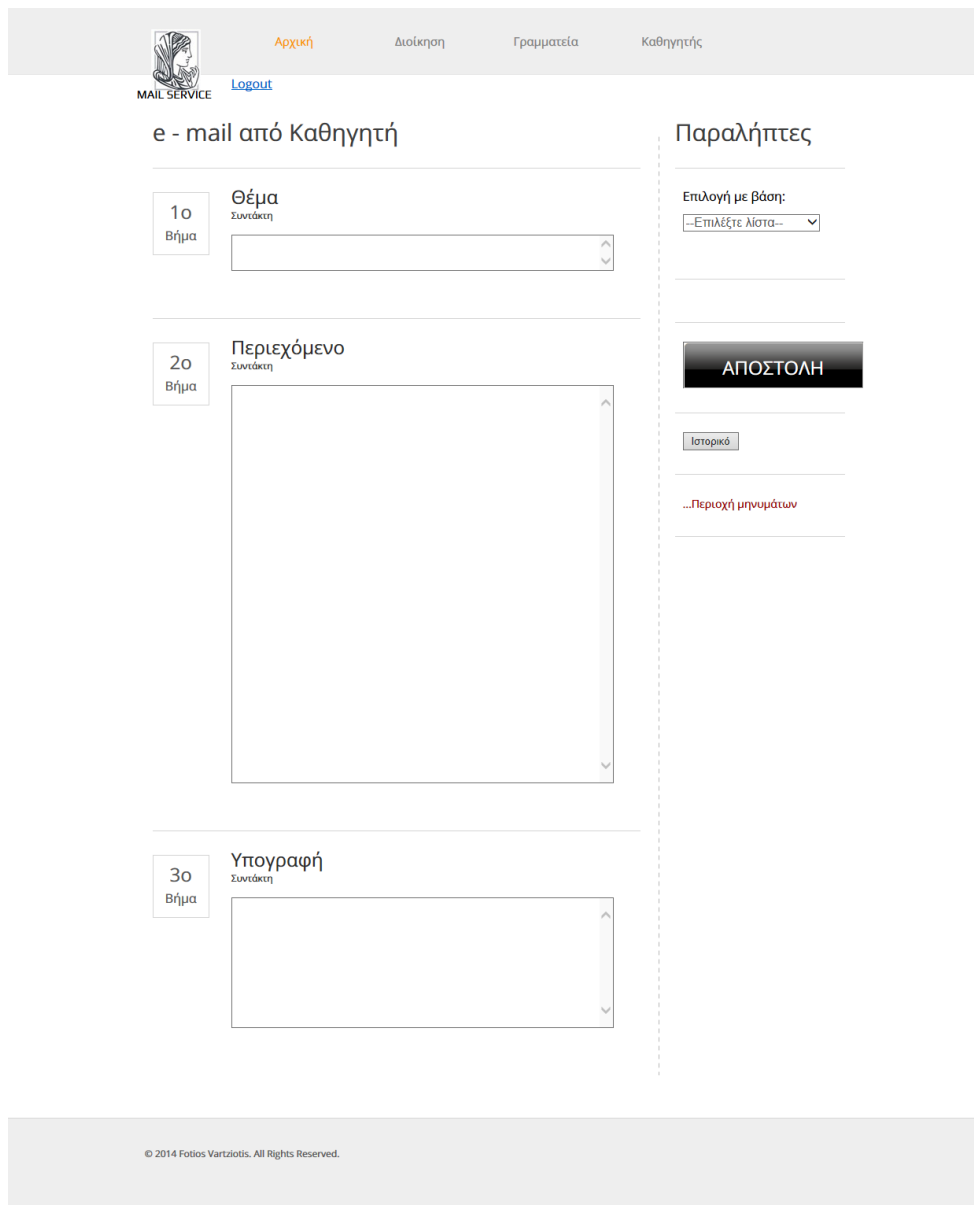
	user10@tei.gr, user11@tei.gr	
--	------------------------------	--

Σχετικά μηνύματα που αφορούν την επιτυχή ή όχι αποστολή εμφανίζονται κάτω από το πλήκτρο ιστορικό.

Η έξοδος από το σύστημα γίνεται με την επιλογή logout που υπάρχει δίπλα στο λογότυπο της εφαρμογής.

#### **4.2 PROFESSOR**

Ο ρόλος αυτός ανατίθεται στους καθηγητές του οργανισμού. Μετά το login, εμφανίζεται η παρακάτω οθόνη



MAIL SERVICE [Logout](#)

Αρχική Διοίκηση Γραμματεία Καθηγητής

### e - mail από Καθηγητή

**1ο Βήμα** **Θέμα**  
Συντάκτη

**2ο Βήμα** **Περιεχόμενο**  
Συντάκτη

**3ο Βήμα** **Υπογραφή**  
Συντάκτη

Επιλογή με βάση:  
--Επιλέξτε λίστα--

**ΑΠΟΣΤΟΛΗ**

Ιστορικό

...Περιοχή μηνυμάτων

© 2014 Fotios Vartziotis. All Rights Reserved.

Εικ. 4.11: Η οθόνη του ρόλου PROFESSOR.

Οι διαθέσιμες επιλογές του πρώτου πεδίου τώρα αλλάζουν. Ο ρόλος αυτός έχει τις παρακάτω επιλογές.

- Μάθημα
- Εξάμηνο φοιτητών
- Όλοι, χωρίς εξαίρεση

Αν ο χρήστης επιλέξει την πρώτη επιλογή υπάρχουν διαθέσιμες 3 υπό επιλογές. Έτος, περίοδο, μάθημα (εικόνα 4.12)

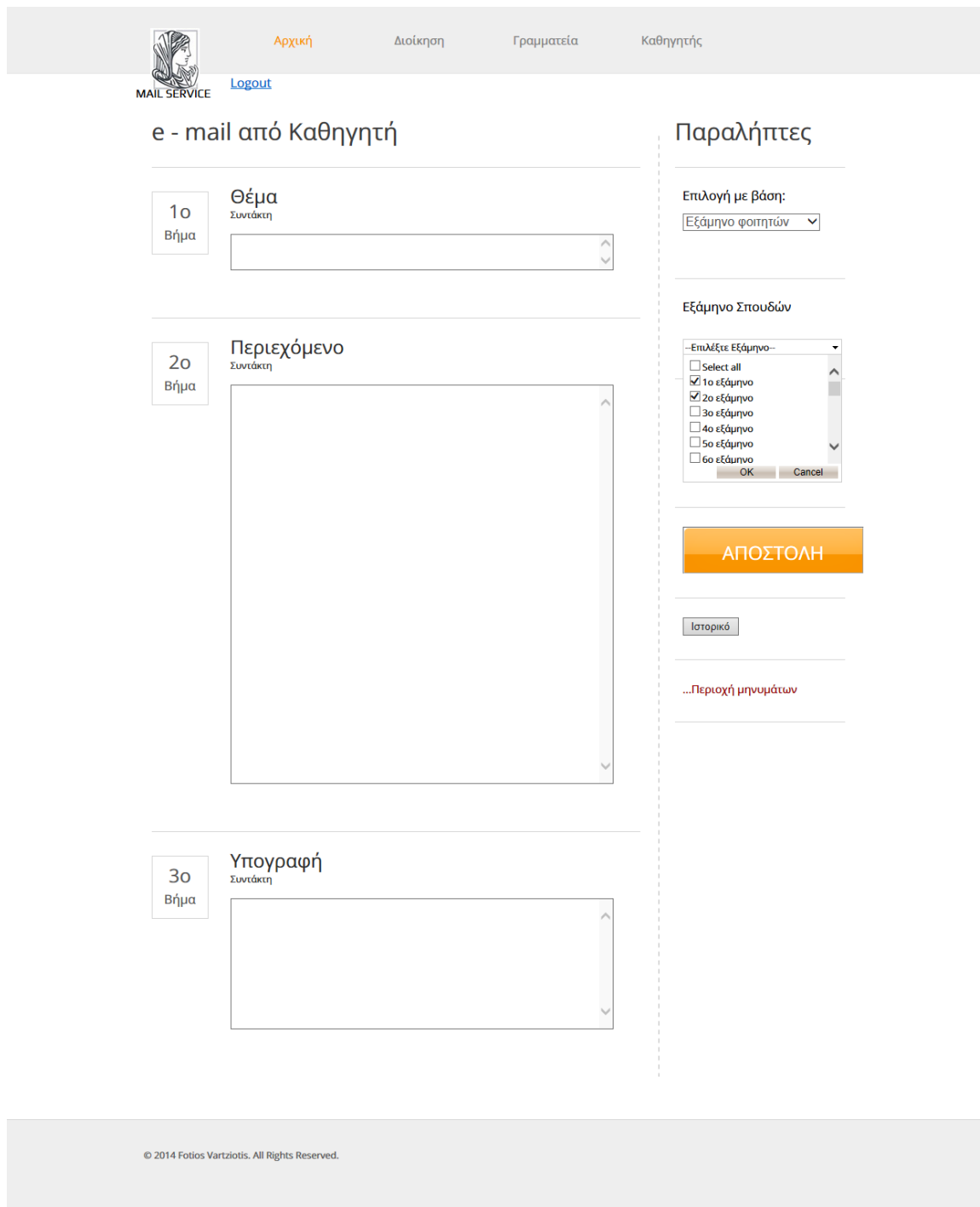
The screenshot shows a web interface for sending an email from a professor. The interface is divided into two main sections: 'e-mail από Καθηγητή' on the left and 'Παραλήπτες' on the right. The left section has three steps: '1ο Βήμα' (Θέμα), '2ο Βήμα' (Περιεχόμενο), and '3ο Βήμα' (Υπογραφή). The right section contains several dropdown menus for filtering recipients: 'Επιλογή με βάση: Μάθημα', 'Έτος: 2014', 'Περίοδος: --Επιλέξτε Περίοδο--', 'Μάθημα: --Επιλέξτε Μάθημα--', and 'Προτεραιότητα Μηνύματος: Επιλέξτε Προτεραιότητα'. There is a 'ΑΠΟΣΤΟΛΗ' button and a '...Περιοχή μηνυμάτων' link.

Εικ. 4.12: Οι διαθέσιμες επιλογές για την επιλογή ΜΑΘΗΜΑ

Το πεδίο έτος περιέχει το έτος φοίτησης, το πεδίο περίοδο περιέχει τις επιλογές χειμερινή και εαρινή και το πεδίο μάθημα περιέχει τα μαθήματα που διδάσκει ο καθηγητής. Για παράδειγμα οι επιλογές 2014, Εαρινή, Μαθηματικά ορίζουν πως το μήνυμα θα το λάβουν οι φοιτητές της εαρινής περιόδου του μαθήματος μαθηματικά του έτους 2014.

Σε περίπτωση που ο χρήστης επιλέξει την πρώτη επιλογή τότε οι διαθέσιμες δευτερεύουσες επιλογές είναι το εξάμηνο (εικόνα 4.13).





MAIL SERVICE [Logout](#)

Αρχική Διοίκηση Γραμματεία Καθηγητής

## e - mail από Καθηγητή

**1ο Βήμα** **Θέμα**  
Συντάκτη

**2ο Βήμα** **Περιεχόμενο**  
Συντάκτη

**3ο Βήμα** **Υπογραφή**  
Συντάκτη

**Παραλήπτες**

Επιλογή με βάση:  
Εξάμηνο φοιτητών

Εξάμηνο Σπουδών

-Επιλέξτε Εξάμηνο-

Select all

1ο εξάμηνο

2ο εξάμηνο

3ο εξάμηνο

4ο εξάμηνο

5ο εξάμηνο

6ο εξάμηνο

OK Cancel

**ΑΠΟΣΤΟΛΗ**

Ιστορικό

...Περιοχή μηνυμάτων

© 2014 Fotios Vartziotis. All Rights Reserved.

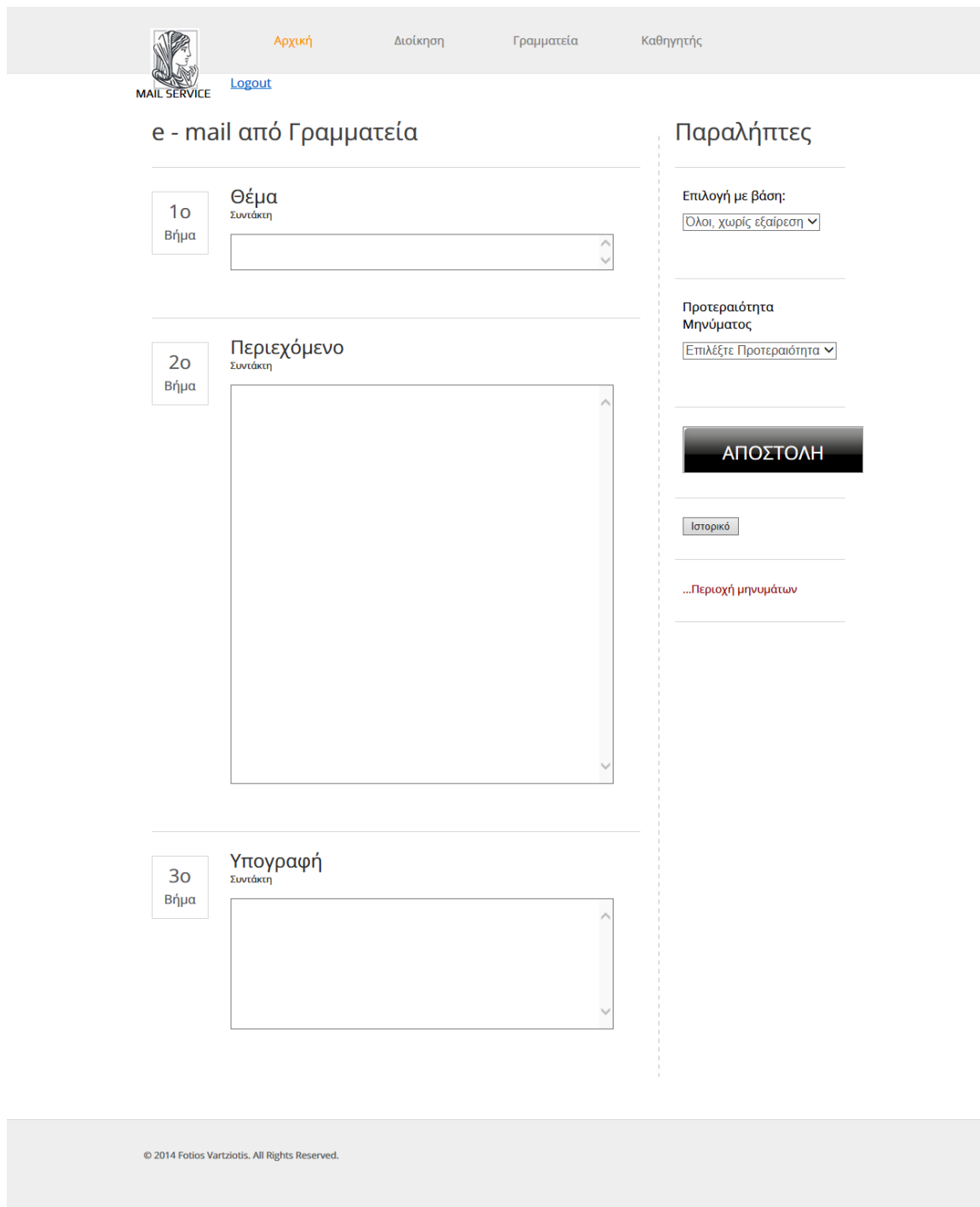
Εικ. 4.13: Επιλογή εξάμηνο

Με την επιλογή «όλοι χωρίς εξαίρεση» το μήνυμα θα αποσταλεί σε όλους τους μαθητές του καθηγητή.

Και εδώ υπάρχει η επιλογή ιστορικό όπου ο χρήστης μπορεί να δει τα μηνύματα που έχει στείλει.

### 4.3 SECRETARY

Η αρχική οθόνη του χρήστη αυτού του ρόλου είναι η παρακάτω



MAIL SERVICE [Logout](#)

Αρχική Διοίκηση Γραμματεία Καθηγητής

### e - mail από Γραμματεία

**1ο Βήμα** **Θέμα**  
Συντάκτη

**2ο Βήμα** **Περιεχόμενο**  
Συντάκτη

**3ο Βήμα** **Υπογραφή**  
Συντάκτη

**Παραλήπτες**

Επιλογή με βάση:  
Όλοι, χωρίς εξαίρεση

Προτεραιότητα Μηνύματος  
Επιλέξτε Προτεραιότητα

**ΑΠΟΣΤΟΛΗ**

Ιστορικό

[...Περιοχή μηνυμάτων](#)

© 2014 Fotios Vartziotis. All Rights Reserved.

Εικ. 4.14: Οθόνη του χρήστη με δικαιώματα SECRETARY

Το περιβάλλον του χρήστη αυτού του ρόλου ομοιάζει με αυτό του καθηγητή. Και εδώ υπάρχουν επιλογές για «μάθημα», «μάθημα ή εξάμηνο», «όλοι χωρίς εξαίρεση»

Η διαφορά από τον ρόλο PROFESSOR είναι ότι η επιλογή «όλοι, χωρίς εξαίρεση» αφορά όλους τους σπουδαστές.

## 5. Συμπεράσματα -προτάσεις

Αντικείμενο της πτυχιακής ήταν η βελτίωση υπάρχοντος συστήματος αποστολής μαζικών μηνυμάτων. Δόθηκε βαρύτητα περισσότερο στις λειτουργίες αποστολής των μηνυμάτων και διαχείρισης σφαλμάτων του συστήματος. Μετατροπές έγιναν σχεδόν σε όλη την έκταση του έργου ενώ προστέθηκε και η δυνατότητα διατήρησης ιστορικού αποστολών για κάθε χρήστη του συστήματος.

Κατά την διάρκεια των εργασιών, ανακαλύφθηκαν αρκετές από τις αρετές του .NET, την ευκολία χρήσης του αλλά και το πλούσιο περιεχόμενο του και σε συνδυασμό με τα κατάλληλα προγραμματιστικά εργαλεία, ο προγραμματιστής έχει στα χέρια του ένα πλήρες προγραμματιστικό περιβάλλον το οποίο του δίνει τη δυνατότητα να φτιάξει εφαρμογές οι οποίες αλληλεπιδρούν με οποιοδήποτε πληροφοριακό σύστημα.

Η εφαρμογή αυτή αποτελεί ένα απλό εργαλείο που εξυπηρετεί τις ανάγκες ενός οργανισμού. Όπως και κάθε άλλο έργο λογισμικού επιδέχεται βελτιώσεις και διορθώσεις. Λόγο της δομής του η παραμετροποίηση και η προσαρμογή του στις εκάστοτε ανάγκες ενός οργανισμού είναι σχετικά εύκολη διαδικασία.

## **6. Βελτιώσεις**

Ως πιθανές βελτιώσεις και επεκτάσεις της εφαρμογής προτείνονται οι παρακάτω

Δημιουργία περιβάλλοντος διαχείρισης χρηστών, μαθημάτων, τμημάτων κτλ.

Αλλαγή στον τρόπο αποστολής των μηνυμάτων ώστε αυτά να καταχωρούνται σε βάση δεδομένων αντί να αποστέλλονται με email, και παράλληλα δημιουργία περιβάλλοντος παραληπτών ώστε να διαχειρίζονται μέσω της εφαρμογής τα μηνύματά τους.

Δυνατότητα επιλογής παραληπτών ένα προς ένα

Δυνατότητα δημιουργίας λογαριασμού από το περιβάλλον της εφαρμογής.



## **ΒΙΒΛΙΟΓΡΑΦΙΑ**

1. ASP.NET Wikipedia. [Ηλεκτρονικό]  
<https://en.wikipedia.org/wiki/ASP.NET>
2. .NET Architecture [Ηλεκτρονικό]  
[http://aspalliance.com/264\\_Introducing\\_Microsoft\\_DotNet.3](http://aspalliance.com/264_Introducing_Microsoft_DotNet.3)
3. .NET Programming [Ηλεκτρονικό]  
<http://ustudy.in/ce/dnt>
4. Visual Studio IDE [Ηλεκτρονικό]  
<https://msdn.microsoft.com/en-us/library/b142f8e7.aspx>
5. MSDN Library. Microsoft Developer Network. [Ηλεκτρονικό] [Παραπομπή: 9 Οκτώβριος 2008.] <http://msdn.microsoft.com/en-us/library/default.aspx>.
6. Language References [Ηλεκτρονικό]  
<http://www.w3schools.com/>
7. MSSQL Server MSDN Microsoft [Ηλεκτρονικό]  
<https://msdn.microsoft.com/en-us/library/bb545450.aspx>
8. c# MSDN Microsoft [Ηλεκτρονικό]  
<https://msdn.microsoft.com/en-us/library/aa287558%28v=vs.71%29.aspx>
9. Project MONO [Ηλεκτρονικό]  
<http://www.mono-project.com/docs/web/aspnet/>

## ΠΑΡΑΡΤΗΜΑ Α

### Περιγραφή βάσης δεδομένων

Οι πίνακες της εφαρμογής μπορούν να χωριστούν νοητά, σε δύο ομάδες. Στους πίνακες της διαδικασίας Login και στους πίνακες των δεδομένων του οργανισμού οι οποίοι περιέχουν τα τμήματα τα εξάμηνα και τα μαθήματα.

Οι πίνακες με πρόθεμα asp net\_ δημιουργήθηκαν από το ASP.NET membership.

ΟΝΟΜΑΣΙΑ ΠΙΝΑΚΑ	ΠΕΡΙΓΡΑΦΗ
aspnet_Applications	Περιέχει το id της εφαρμογής
aspnet_Paths	Περιέχει στοιχεία των μελών και πληροφορίες που αφορούν την ανάκτηση του κωδικού πρόσβασης. Στον πίνακα αυτόν καταγράφονται και οι τελευταίες ημερομηνίες διαφόρων δραστηριοτήτων όπως είσοδο έξοδο κτλ.
aspnet_Paths	Χρησιμοποιείται από το membership
aspnet_PersonalizationAllUsers	
aspnet_PersonalizationPerUser	
aspnet_Profile	
aspnet_Roles	Περιέχει τους διαθέσιμους ρόλους για τους εγγεγραμμένους χρήστες
aspnet_SchemaVersions	Πίνακας συστήματος
aspnet_Users	Εγγεγραμμένοι χρήστες
aspnet_UsersInRoles	Αντιστοίχιση χρηστών και ρόλων
aspnet_WebEvent_Events	Πίνακας συστήματος
log_table	Πίνακας ιστορικού αποστολών
teiep_classes	Πίνακας μαθημάτων
teiep_classusernames	Πίνακας αντιστοίχισης μαθημάτων και χρηστών
teiep_departments	Τμήματα οργανισμού
teiep_usernames	Πίνακας με τους χρήστες του συστήματος.

## ΠΑΡΑΡΤΗΜΑ Β

Εικόνες

<b>ΕΙΚΟΝΑ</b>	<b>ΣΕΛ</b>	<b>ΠΕΡΙΓΡΑΦΗ</b>
Εικόνα 2.1	3	Η δομή του .NET FRAMEWORK
Εικόνα 2.2	4	Η δομή του CRL
Εικόνα 2.3	5	Η δομή ενός έργου σε C#
Εικόνα 2.4	6	Η δομή κώδικα μιας κλάσης σε C#
Εικόνα 2.5	7	Αρχείο web form
Εικόνα 3.1	15	Διάγραμμα χρήστη με ρόλο CEO
Εικόνα 3.2	16	Διάγραμμα χρήστη με ρόλο Professor
Εικόνα 3.3	17	Διάγραμμα χρήστη με ρόλο secretariat
Εικόνα 3.4	18	Η οθόνη login του MS SQL Management studio
Εικόνα 3.5	18	Ο object explorer
Εικόνα 3.6	19	Οι σχέσεις μεταξύ των πινάκων της εφαρμογής
Εικόνα 3.7	20	Οι σχέσεις των πινάκων του membership
Εικόνα 3.8	21	Πίνακας log_table
Εικόνα 3.9	21	Πίνακας classes
Εικόνα 3.10	22	Πίνακας class usernames
Εικόνα 3.11	22	Πίνακας departments
Εικόνα 3.12	23	Διάγραμμα UML
Εικόνα 3.13	24	Διάγραμμα activity για την είσοδο του χρήστη στο σύστημα
Εικόνα 3.14	25	Διάγραμμα activity αποστολής email



Εικόνα 3.15	26	Διάγραμμα sequence για την διαδικασία πιστοποίησης του χρήστη
Εικόνα 3.16	27	Διάγραμμα αποστολής μηνυμάτων
Εικόνα 3.17	28	Οθόνη του χρήστη με ρόλο CEO
Εικόνα 3.18	40	Η οθόνη του χρήστη με ρόλο professor
Εικόνα 3.19	46	Οθόνη του χρήστη με ρόλο secretariat
Εικόνα 3.20	52	Ιστορικό του χρήστη με ρόλο professor
Πίνακας 3.1	14	Οι πίνακες της εφαρμογής
Πίνακας 3.2	15	Οι πίνακες του membership
Πίνακας 3.3	23	Τα αρχεία στο ριζικό φάκελο του έργου
Εικόνα 4.1	54	Αρχική οθόνη της εφαρμογής
Εικόνα 4.2	55	Η αρχική σελίδα του χρήστη με δικαιώματα CEO
Εικόνα 4.3	56	Πεδίο τμήμα
Εικόνα 4.4	57	Οι επιλογές του χρήστη για το τμήμα.
Εικόνα 4.5	58	Η επιλογή ορισμού προτεραιότητας
Εικόνα 4.6	59	Επιλογή εξαμήνων
Εικόνα 4.7	60	Το μήνυμα είναι έτοιμο για αποστολή.
Εικόνα 4.8	61	Παράθυρο διαλόγου για εκτύπωση της λίστας παραληπτών
Εικόνα 4.9	61	Οι λίστα παραληπτών
Εικόνα 4.10	62	Αποστολή μηνύματος σε όλους τους χρήστες
Εικόνα 4.11	63	Η οθόνη του ρόλου PROFESSOR.
Εικόνα 4.12	64	Οι διαθέσιμες επιλογές για την επιλογή ΜΑΘΗΜΑ
Εικόνα 4.13	65	Επιλογή εξάμηνο
Εικόνα 4.14	66	Οθόνη του χρήστη με δικαιώματα

		SECRETARY
--	--	-----------

## Πίνακες

<b>ΠΙΝΑΚΑΣ</b>	<b>ΣΕΛ</b>	<b>ΠΕΡΙΓΡΑΦΗ</b>
Πίνακας 3.1	14	Οι πίνακες της εφαρμογής
Πίνακας 3.2	15	Οι πίνακες του memebership
Πίνακας 3.3	23	Τα αρχεία στο ριζικό φάκελο του έργου