

<<Πτυχιακή εργασία της φοιτήτριας Σκευούλη Θεοδώρας>>



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



Πτυχιακή εργασία

<< Ανάπτυξη Λογισμικού Διαχείρισης Εργασιών Φοιτητών >>



Της φοιτήτριας

Σκευούλη Θεοδώρα

Αρ.Μητρώου:04/2547

Επιβλέπων καθηγητής:

Απόστολος Αμπατζόγλου

Θεσσαλονίκη 2010

Ευχαριστίες

Η εργασία αυτή ξεκίνησε και τελείωσε μέσα στο ακαδημαϊκό έτος 2009-2010 σύμφωνα με το υπάρχον κανονισμό του Αλεξάνδρειου τμήματος Πληροφορικής Θεσσαλονίκης.

Ολοκληρώνοντας λοιπόν αυτή την πτυχιακή εργασία θα ήθελα να ευχαριστήσω όλους όσους μου στάθηκαν κατά την διάρκεια των σπουδών μου. Ιδιαίτερα θα ήθελα να ευχαριστήσω τον κ. Απόστολο Αμπατζόγλου, επιβλέποντα της εργασίας μου, για την συνεχή υποστήριξη που μου παρείχε. Επίσης θα ήθελα να τον ευχαριστήσω για την διάθεσή του να με βοηθήσει και να μου λύσει οποιαδήποτε απορία οποιαδήποτε στιγμή το χρειαζόμουν.

Ένα μεγάλο ευχαριστώ στους φίλους μου, οι οποίοι μου συμπαραστάθηκαν όλο αυτό τον καιρό. Τέλος, θέλω να ευχαριστήσω θερμά την οικογένεια μου, γονείς και αδέρφια για την ηθική και οικονομική συμπαράσταση όχι μόνο κατά την διάρκεια της εκπόνησης της πτυχιακής μου εργασίας αλλά και καθ' όλη τη διάρκεια των σπουδών μου.

Πρόλογος

Η συγκεκριμένη εργασία αναπτύχθηκε στα πλαίσια πτυχιακής εργασίας με τίτλο Δυναμικές Ιστοσελίδες : “Ανάπτυξη Λογισμικού Διαχείρισης Εργασιών Φοιτητών” κατά το ακαδημαϊκό έτος 2009-2010 από την φοιτήτρια του τμήματος ΤΕΙ Πληροφορικής Θεοδώρας Σκευούλη, υπό την εποπτεία του καθηγητή κ. Απόστολου Αμπατζόγλου.

Στην εποχή μας το διαδίκτυο είναι το νέο εργαλείο πόρων για όλους τους ανθρώπους. Έχει αλλάξει τον τρόπο που ζούμε στην κοινωνία καθώς όλο και περισσότεροι άνθρωποι συνδέονται σε αυτό. Εξετάζοντας τους τρόπους που έχει αλλάξει την κοινωνία, μπορούμε να αντιληφθούμε ότι έχει επηρεάσει ακόμα και τον τρόπο που σκέφτονται και αισθάνονται κάποιοι άνθρωποι. Αρχικά, παρέχει στους ανθρώπους ένα νέο τρόπο επικοινωνίας μέσω e mail και κάμερας. Επιπλέον, είναι ένα είδος σύγχρονης βιβλιοθήκης, όπου οτιδήποτε μπορεί να φανταστεί κανείς μπορεί να το βρει. Τέλος εκατομμύρια θέσεις εργασίας δημιουργούνται μέσω Διαδικτύου ενώ πλέον γίνονται και αγορές με την χρήση του.

Από όλα τα παραπάνω καταλαβαίνουμε ότι το Διαδίκτυο επηρεάζει θετικά και αρνητικά στην ζωή μας. Μερικά από τα θετικά είναι ότι είναι ένας ευχάριστος χώρος ο οποίος παρέχει δυνατότητες διασκέδασης, παρέχει πλούσια πληροφόρηση και αποτελεί αναπόσπαστο κομμάτι της εκπαιδευτικής κοινότητας, είναι «βολικό» και «εύκολα» προσβάσιμο, παρέχει άμεση και οικονομική επικοινωνία και παρέχει ένα μεγάλο εύρος δραστηριοτήτων. Παρόλα αυτά η έντονη χρήση του και γενικά ο κακός χειρισμός, οδηγούν σε αρνητικές συνέπειες. Έτσι, μπορούν να υπάρξουν οικονομικοί και σωματικοί κίνδυνοι, εκμετάλλευση προσωπικών δεδομένων, αναξιόπιστη πληροφόρηση, έκθεση σε ακατάλληλο περιεχόμενο και τέλος μπορεί να υπάρξει εθισμός στο διαδίκτυο που σημαίνει παραμέληση κοινωνικών δραστηριοτήτων ή ακόμα και στάδια κατάθλιψης.

Περίληψη

Η εργασία αυτή στοχεύει στον σχεδιασμό και την ανάπτυξη ενός δυναμικού διαδικτυακού τόπου, ο οποίος θα διαχειρίζεται τις εργασίες φοιτητών για το μάθημα Μεθοδολογίες Προγραμματισμού II. Χρήστες του συστήματος θα είναι οι φοιτητές και καθηγητές(οι οποίοι θα εκτελούν και ρόλο διαχειριστών συστήματος).

Η εφαρμογή θα παρέχει την δυνατότητα στους φοιτητές δημιουργώντας e-mail να μπορούν να στέλνουν ασκήσεις, να βλέπουν το βαθμό τους καθώς και να κατεβάζουν χρήσιμο υλικό. Επιπλέον, οι καθηγητές θα μπορούν να κατεβάζουν τις ασκήσεις των φοιτητών, να τις βαθμολογούν και να ανεβάζουν υλικό για το μάθημα.

Επίσης στην εργασία χρησιμοποιήθηκε μια παραλλαγή της μεθοδολογίας RUP (Rational Unified Process) για την παρουσίαση της διαδικασίας ανάλυσης και σχεδίασης σε διαδικτυακό προγραμματισμό. Η RUP είναι επαναληπτική διαδικασία και χρησιμοποιείται για να μειώσει τους κινδύνους που σχετίζονται με την ανάπτυξη λογισμικού. Τα βασικά στοιχεία που χαρακτηρίζουν μια μεθοδολογία RUP είναι το εξελικτικό μοντέλο με ανάδραση, καθοδηγείται από περιπτώσεις χρήσης, είναι αρχιτεκτονικό - κεντρικό μοντέλο και χρησιμοποιεί την uml σαν γλώσσα μοντελοποίησης.[1] Για την υλοποίηση της εφαρμογής όμως χρησιμοποιήθηκαν διαγράμματα που δεν εμπλέκουν αντικείμενα, όπως είναι τα διαγράμματα περιπτώσεων χρήσης, τα διαγράμματα ακολουθίας συστήματος και τα διαγράμματα δραστηριότητας τα οποία εξετάζονται παρακάτω.

Για να δημιουργηθεί η εφαρμογή χρησιμοποιήθηκε ένας συνδυασμός εργαλείων και γλωσσών προγραμματισμού, όπως η PHP, η MySQL, το XAMPP και το Visual Paradigm. Από αυτά τα πρώτα τρία διανέμονται δωρεάν στο διαδίκτυο, ενώ το εργαλείο Visual Paradigm διανέμεται μέσω φοιτητικής άδεια χρήσης του ΤΕΙΘ. Όλα τα εργαλεία και οι γλώσσες προγραμματισμού που χρησιμοποιήθηκαν αναλύονται στα επόμενα κεφάλαια.

Abstract

This work aims to design and develop a dynamic website, which will manage the work of students for lesson planning methodology II .Users of the system will be the students and teachers (who will perform and a management system).

The application will provide the opportunity for students creating e-mail can send exercises to see their degree and to download useful material. In addition, teachers can download the exercises for students to graduate and upload material for the course.

Also at work used a variant of the methodology RUP (Rational Unified Process) to present the process analysis and design in web programming. The RUP is an iterative process and is used to reduce the risks associated with software development. The key elements of a methodology RUP is an evolutionary model with feedback, driven by use cases, it is architecture - a central model and uses the modeling language uml like.[1] To implement the application but used diagrams that do not involve objects, such as use case diagrams, system sequence diagrams and activity diagrams which are discussed below.

To create the application used a combination of tools and programming languages such as PHP, to MySQL, the XAMPP and Visual Paradigm. From these first three are distributed free on the internet and the tool Visual Paradigm distributed through student license TEITH. All tools and programming languages used are analyzed in subsequent chapters.

ΠΕΡΙΕΧΟΜΕΝΑ

Ευχαριστίες	Σελ.2
Πρόλογος	Σελ.3
Περίληψη	Σελ.4
Abstract	Σελ.5
Περιεχόμενα	Σελ.6
Ευρετήριο Σχημάτων και Πινάκων	Σελ.10
ΚΕΦΑΛΑΙΟ 1	Σελ.12
ΕΙΣΑΓΩΓΗ	Σελ.12
1.1 Τι είναι δυναμικές σελίδες	Σελ.13
1.2 Τι είναι στατικές σελίδες	Σελ.13
1.3 Δυναμική ή στατική σελίδα	Σελ.14
1.4 Τι είναι html	Σελ.15
1.5 Τι είναι css	Σελ.15
1.6 Τι είναι php	Σελ.15
1.7 Τι είναι MySql	Σελ.15
1.8 Τι είναι Apache	Σελ.16
ΚΕΦΑΛΑΙΟ 2	Σελ.17
ΜΕΘΟΔΟΛΟΓΙΕΣ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ ΔΙΑΔΙΚΤΥΟΥ	Σελ.17
2.1 Λίγα λόγια για την RUP	Σελ.17
2.2 Η χρήση της RUP για ανάπτυξη web εφαρμογών	Σελ.18

2.3 Ο ρόλος της RUP σε web εφαρμογές	Σελ.21
2.3.1 Ενσωμάτωση Δημιουργικού σχεδιασμού με την ανάπτυξη παραδοσιακού λογισμικού	Σελ21
ΚΕΦΑΛΑΙΟ 3	Σελ28
ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΗ	Σελ28
3.1 Διάγραμμα Περίπτωσης Χρήσης	Σελ28
3.2 Δημιουργώντας περιπτώσεις χρήσης	Σελ32
3.3 Διάγραμμα Ακολουθίας Συστήματος	Σελ49
3.4 Διάγραμμα Δραστηριοτήτων	Σελ60
ΚΕΦΑΛΑΙΟ 4	Σελ72
ΥΛΟΠΟΙΗΣΗ	Σελ72
4.1 Ιστορική αναδρομή στο διαδικτυακό προγραμματισμό	Σελ72
4.1.1 Ιστορικά της html	Σελ72
4.1.2 Ιστορικά του css	Σελ73
4.1.3 Ιστορικά της php	Σελ.74
4.1.4 Ιστορικά της My-Sql	Σελ.75
4.2 Προγραμματισμός σε html	Σελ.75
4.3 Μορφοποίηση σε css	Σελ.79
4.3.1 Επιλογέας κλάσης (class selector)	Σελ.80
4.3.2 Επιλογέας Ταυτότητας (ID Selector)	Σελ.80
4.3.3 Ομαδοποίηση επιλογέων(Grouping Selectors)	Σελ.81

4.4 Υλοποίηση σε php_____	Σελ.81
4.4.1 Εντολή echo_____	Σελ.81
4.4.2 Μεταβλητές και σταθερές_____	Σελ.82
4.4.3 Τύποι μεταβλητών_____	Σελ.82
4.4.4 Τελεστές Σύγκρισης_____	Σελ.82
4.4.5 Εντολές if_____	Σελ.83
4.4.6 Εντολές elseif_____	Σελ.83
4.4.7 Βρόχοι while και for_____	Σελ.83
4.4.8 Πίνακες_____	Σελ.83
4.4.9 Συναρτήσεις στην Php_____	Σελ.84
4.4.10 Συνάρτηση trim()_____	Σελ.84
4.4.11 Συνάρτηση Strlen()_____	Σελ.84
4.4.12 Εντολές require() και include()_____	Σελ.84
4.4.13 Μεταβλητές \$_get και \$_post_____	Σελ.85
4.4.14.Μεταβλητή \$_session_____	Σελ.85
4.4.15 Σύνδεση με την Βάση Δεδομένων_____	Σελ.86
4.4.16 Συναρτήσεις της εφαρμογής_____	Σελ.87
4.5 Βάσεις Δεδομένων_____	Σελ.91
4.5.1 Σχεσιακό Μοντέλο_____	Σελ.91
4.5.2 Τεκμηρίωση Πινάκων_____	Σελ.92
4.5.3 Περιγραφή πεδίων_____	Σελ.94
ΚΕΦΑΛΑΙΟ 5_____	Σελ.96

SCREENSHOTS ΕΦΑΡΜΟΓΗΣ	Σελ.96
5.1 Βασική οθόνη	Σελ.96
5.2 Δημιουργία Χρήστη	Σελ.97
5.3 Υπενθύμιση κωδικού	Σελ.98
5.4 Δυνατότητες Φοιτητή	Σελ.98
5.4.1 Εμφάνιση Βαθμών	Σελ.99
5.4.2 Αποστολή Άσκησης	Σελ.100
5.4.3 Αποστολή διορθωμένης άσκησης	Σελ.101
5.4.4 Κατέβασμα υλικού	Σελ.102
5.5 Δυνατότητες Καθηγητή	Σελ.102
5.5.1 Κατέβασμα ασκήσεων	Σελ.103
5.5.2 Ανέβασμα Βαθμών	Σελ.104
5.5.3 Ανέβασμα υλικού	Σελ.105
ΒΙΒΛΙΟΓΡΑΦΙΑ	Σελ.106

ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ ΚΑΙ ΠΙΝΑΚΩΝ

ΠΙΝΑΚΕΣ

Πίνακας I	Σελ.31
Πίνακας II	Σελ.33
Πίνακας III	Σελ.34
Πίνακας IV	Σελ.36
Πίνακας V	Σελ.38
Πίνακας VI	Σελ.40
Πίνακας VII	Σελ.41
Πίνακας VIII	Σελ.43
Πίνακας IX	Σελ.44
Πίνακας X	Σελ.46
Πίνακας XI	Σελ.47
Πίνακας XII	Σελ.48

ΣΧΗΜΑΤΑ

Σχήμα I	Σελ.31
Σχήμα II	Σελ.51
Σχήμα III	Σελ.52
Σχήμα IV	Σελ.53
Σχήμα V	Σελ.54
Σχήμα VI	Σελ.55
Σχήμα VII	Σελ.56

Σχήμα VIII	Σελ.57
Σχήμα ΙΧ	Σελ.58
Σχήμα Χ	Σελ.59
Σχήμα ΧΙ	Σελ.60
Σχήμα ΧΙΙ	Σελ.62
Σχήμα ΧΙΙΙ	Σελ.63
Σχήμα ΧΙV	Σελ.64
Σχήμα ΧV	Σελ.65
Σχήμα ΧVΙ	Σελ.66
Σχήμα ΧVΙΙ	Σελ.67
Σχήμα ΧVΙΙΙ	Σελ.68
Σχήμα ΧΙΧ	Σελ.69
Σχήμα ΧΧ	Σελ.70
Σχήμα ΧΧΙ	Σελ.71
Σχήμα ΧΧΙΙ	Σελ.92
Σχήμα ΧΧΙΙΙ	Σελ.93
Σχήμα ΧΧΙV	Σελ.93
Σχήμα ΧΧV	Σελ.94

ΕΙΣΑΓΩΓΗ

Σε αυτήν την πτυχιακή εργασία αναλύουμε, σχεδιάζουμε και υλοποιούμε μία εφαρμογή δικτυακού κοινωνικού ιστοχώρου με δυνατότητα υποστήριξης δυναμικής ανάρτησης (upload) αρχείων, εξετάζοντας τις απαραίτητες εκείνες λειτουργίες που χαρακτηρίζουν μία τέτοια εφαρμογή. Θα πρέπει να τονίσουμε ότι ο όρος “δυναμικός” είναι πολυδιάστατος. Τις περισσότερες φορές όμως αναφέρεται στον αυτόματο τρόπο με τον οποίο δημιουργούνται οι σελίδες ενός website. Ο δυναμικός μηχανισμός δημιουργεί τις σελίδες αυτές διαβάζοντας πληροφορίες που προέρχονται από μια βάση δεδομένων, εισάγοντας τις πληροφορίες (data) σε ένα template και στέλνοντας την σελίδα πίσω στον web browser. Ως template στα ελληνικά θα μπορούσαμε να δώσουμε τον όρο πρότυπο εμφάνισης. Δηλαδή τον τρόπο εμφάνισης μιας σελίδας, και το συγκεκριμένο στήσιμό της ανεξάρτητα με το περιεχόμενό της.

Το Διαδικτυακό Περιβάλλον που δημιουργήθηκε έχει σκοπό να βοηθήσει καθηγητές και φοιτητές στη γρήγορη και αποτελεσματική διαχείριση ενός μεγάλου όγκου ασκήσεων που αναφέρονται στο μάθημα " Μεθοδολογίες Προγραμματισμού II" του τμήματος Πληροφορικής, στο οποίο εξετάζονται οι φοιτητές κατά την διάρκεια του εξαμήνου. Είναι στην ουσία μια web εφαρμογή στην οποία μπορούν φοιτητές να διαχειρίζονται διάφορες λειτουργίες όπως να υποβάλουν τις ασκήσεις τους αλλά και να αναζητούν χρήσιμο υλικό για το μάθημα, τα οποία συμβάλλουν στην εξοικονόμηση χρόνου και στην αρτιότερη οργάνωση της δύσκολης προσπάθειάς τους.

Η εργασία αυτή χωρίζεται σε πέντε κεφάλαια. Στο πρώτο κεφάλαιο παρουσιάζονται τα βασικά εργαλεία που χρησιμοποιήθηκαν στην εφαρμογή. Ουσιαστικά γίνεται μια εισαγωγή στις δυναμικές και στατικές ιστοσελίδες και τη χρησιμοποιήθηκε για την δημιουργία αυτής της δυναμικής ιστοσελίδας. Στο δεύτερο κεφάλαιο γίνεται η επεξήγηση της RUP μεθοδολογίας καθώς και η χρήση της σε web εφαρμογές. Το τρίτο κεφάλαιο αναφέρεται στην ανάλυση και σχεδίαση. Περιγράφει δηλαδή τα διαγράμματα που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής αλλά και την εξήγηση τους. Το τέταρτο κεφάλαιο παρουσιάζει την υλοποίηση του κώδικα την εξήγησή του καθώς και κάποιες

βασικές έννοιες. Τέλος το πέμπτο κεφάλαιο είναι τα Screenshots της εφαρμογής, δηλαδή εικόνες και ο τρόπος με τον οποίο μπορούν να χρησιμοποιηθούν.

1.1 Τι είναι Δυναμικές Ιστοσελίδες

Οι δυναμικές ιστοσελίδες, σε αντίθεση με τις στατικές ιστοσελίδες, δεν είναι απλά HTML έγγραφα, αλλά συμπεριλαμβάνουν προγραμματισμό (σε μία γλώσσα προγραμματισμού κατάλληλη για το διαδίκτυο, όπως π.χ. είναι η php) και ουσιαστικά είναι web εφαρμογές.

Οι δυναμικές ιστοσελίδες, μπορεί στην εμφάνιση, σε πολλές περιπτώσεις, να μην έχουν μεγάλη διαφορά με τις στατικές, όμως οι δυνατότητές είναι πολύ περισσότερες, από πολλές πλευρές, καθώς στην περίπτωση αυτή ουσιαστικά πρόκειται για μία εφαρμογή (πρόγραμμα), και όχι ένα απλό ηλεκτρονικό έγγραφο.

Συνήθως, οι δυναμικές ιστοσελίδες, χρησιμοποιούν κάποια βάση δεδομένων (database), όπου αποθηκεύουν πληροφορίες και απ' όπου αντλούν το περιεχόμενό τους, ανάλογα με το τι ζητάει ο χρήστης/επισκέπτης σε κάθε του "κλικ". Η χρήση των βάσεων δεδομένων, είναι αυτή που επιτρέπει την εύκολη προσθαφαίρεση περιεχομένου στις δυναμικές ιστοσελίδες, καθώς δεν απαιτείται να επεξεργάζεται κανείς κάθε φορά την ίδια την ιστοσελίδα, αλλά απλά να διαχειρίζεται έμμεσα το περιεχόμενο στην βάση δεδομένων και οι υπόλοιπες διαδικασίες γίνονται αυτοματοποιημένα από τον "μηχανισμό" της ιστοσελίδας.[11]

1.2 Τι είναι Στατικές Ιστοσελίδες

Οι στατικές ιστοσελίδες είναι ουσιαστικά απλά ηλεκτρονικά "έγγραφα". Μπορούν να περιέχουν κείμενα, φωτογραφίες, συνδέσμους (links), αρχεία για download, κινούμενα σχέδια (animated graphics). Είναι κατάλληλες κυρίως για την δημιουργία "μόνιμων/ στατικών παρουσιάσεων", όπου δεν υπάρχει συχνά η ανάγκη να τροποποιείται το περιεχόμενό τους. Δηλαδή, θα μπορούσε να χρησιμοποιηθεί μια στατική ιστοσελίδα για την γενική παρουσίαση μιας επιχείρησης και των προϊόντων ή υπηρεσιών της.

Ο χαρακτηρισμός "στατική", δεν σημαίνει ότι τίποτα στην σελίδα δεν κουνιέται, αντιθέτως μπορεί να έχει κινούμενα γραφικά και άλλα μπιχλιμπίδια. Σημαίνει ότι

το περιεχόμενο της κάθε σελίδας είναι σταθερό και συγκεκριμένο. Πρόκειται δηλαδή για σταθερά ηλεκτρονικά έγγραφα.

Το κύριο μειονέκτημά τους είναι ότι δεν είναι πρακτική/βολική η χρήση τους όταν χρειάζεται το περιεχόμενο της ιστοσελίδας να αλλάζει συχνά, π.χ. τουλάχιστον μία φορά την εβδομάδα να προστίθεται κάποιο άρθρο, ή να αλλάζουν τα προϊόντα κτλ. Αυτό συμβαίνει γιατί για να τροποποιηθεί μια στατική ιστοσελίδα, πρέπει να γίνουν επεμβάσεις απευθείας πάνω στην ιστοσελίδα, με κάποιο σχετικό πρόγραμμα (html editor είτε απλό είτε WYSIWYG), μια διαδικασία δηλαδή όπως το να σβήνεις και να γράφεις σε ένα τετράδιο, που όμως χρειάζεται μια σχετική εμπειρία.[11]

1.3 Δυναμική ή στατική ιστοσελίδα;

Γενικά, αυτό που προσφέρουν οι δυναμικές ιστοσελίδες, είναι μεγαλύτερη αλληλεπίδραση του χρήστη με την σελίδα π.χ. να προσθέτει τα σχόλια του στην σελίδα, αλλά και πολλές ευκολίες στον διαχειριστή του περιεχομένου της ιστοσελίδας π.χ. τον ιδιοκτήτη της σελίδας.

Πάντως το γεγονός ότι μια δυναμική ιστοσελίδα προσφέρει περισσότερες δυνατότητες, δεν σημαίνει ότι αυτές είναι απαραίτητες σε όλους, δηλαδή σε αρκετές περιπτώσεις, μία στατική ιστοσελίδα μπορεί να καλύπτει πλήρως τις ανάγκες μιας συνοπτικής παρουσίασης.

Από πλευράς κόστους, η στατική ιστοσελίδα είθισται να είναι η φτηνή επιλογή, καθώς είναι πιο απλή η κατασκευή της ιστοσελίδας, ενώ οι δυναμικές ιστοσελίδες λόγω της πολυπλοκότητας τους κοστίζουν ακριβότερα και αυτό είναι λογικό τουλάχιστον όταν γίνονται κατά παραγγελία.

Υπάρχει βέβαια και η περίπτωση υλοποίησης μιας δυναμικής ιστοσελίδας με την χρήση κάποιας open source εφαρμογής η οποία διατίθεται δωρεάν μέσω του internet και σε αυτήν την περίπτωση η κατασκευή της ιστοσελίδας μπορεί να έχει μηδενικό κόστος (αν γίνει self-service) ή να υπάρξει κάποια χρέωση (η τελική τιμή μιας ιστοσελίδας είναι υποκειμενική υπόθεση) αν η εγκατάσταση και η τυχόν παραμετροποίηση της δωρεάν εφαρμογής ανατεθεί σε κάποια εταιρία.[11]

Για την δημιουργία της εφαρμογής χρησιμοποιήθηκαν τα παρακάτω εργαλεία για τα οποία θα δούμε μερικά εισαγωγικά στοιχεία.

1.4 Τι είναι html

Η HTML (Hyper Text Markup Language) είναι μια περιγραφική γλώσσα, δηλαδή ένας ειδικός τρόπος γραφής κειμένου και κλήσης άλλων αρχείων ή εφαρμογών βασισμένος σε οδηγίες (tags). Ο Web client αναγνωρίζει αυτόν τον ειδικό τρόπο γραφής και εκτελεί τις εντολές που περιέχονται σε αυτόν. Web Client ονομάζουμε το λογισμικό που τρέχει ένας Η/Υ και του επιτρέπει να "διαβάσει" Web σελίδες (από το σκληρό του δίσκο ή από το Internet αν είναι συνδεδεμένος με αυτό).[2]

1.5 Τι είναι Css

Η CSS (*Cascading Style Sheets*) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων στυλ που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Πιο πρακτικά χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστοτόπου.[8]

1.6 Τι είναι Php

Η PHP είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML.[8]

1.7 Τι είναι MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακής βάσης ανοικτού κώδικα όπως λέγεται (relational database management system - RDBMS) που χρησιμοποιεί την Structured Query Language (SQL), την πιο γνωστή γλώσσα για την προσθήκη, την πρόσβαση και την επεξεργασία δεδομένων σε μία Βάση Δεδομένων.[8]

1.8 Τι είναι Apache

Ο Apache HTTP γνωστός και απλά σαν Apache είναι ένας εξυπηρετητής του παγκόσμιου ιστού (web). Όταν επισκέπτεστε έναν ιστότοπο ο πλοηγός σας επικοινωνεί με έναν διακομιστή HTTP. Ο Apache είναι ένας από τους δημοφιλέστερους, εν μέρει γιατί λειτουργεί σε διάφορες πλατφόρμες σαν τα Windows, το Linux το Unix και το MAC Os X. Ο Apache HTTP υποστηρίζει επίσης αρκετές διάσημες εφαρμογές και γλώσσες προγραμματισμού όπως MySql, PHP, Perl, Python κ.λπ.

Ο ρόλος του apache είναι να αναμένει αιτήσεις από διάφορα προγράμματα- χρήστες(clients) όπως είναι ένας φυλλομετρητής (browser) ενός χρήστη και στην συνέχεια να εξυπηρετεί αυτές τις υπηρεσίες <<σερβίροντας>> τις σελίδες που ζητούν είτε απευθείας είτε μέσω μιας ηλεκτρονικής διεύθυνσης (url), είτε μέσω ενός συνδέσμου(link). Ο τρόπος με τον οποίο ο Apache εξυπηρετεί αυτές τις αιτήσεις, είναι σύμφωνος με τα πρότυπα του HTTP(Hypertext Transfer Protocol).

Ο Apache αναπτύσσεται από την <<Κοινότητα Ανοιχτού Λογισμικού>> και η εποπτεία, υποστήριξη και διάθεση του προγράμματος γίνεται από το Apache Software Foundation. Το πρόγραμμα είναι ανοιχτού κώδικα (open source), κάτι που σημαίνει ότι σύμφωνα με την άδεια χρήσης του, διατίθεται δωρεάν και μπορούν να γίνουν ελεύθερα από το χρήστη προσθήκες και τροποποιήσεις στον κώδικα του.[10]

Κεφάλαιο 2

ΜΕΘΟΔΟΛΟΓΙΕΣ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ ΔΙΑΔΙΚΤΥΟΥ

2.1 Λίγα λόγια για την RUP

Η RUP είναι μια μέθοδος διαχείρισης, σχεδιασμού και ανάπτυξης έργων λογισμικού μεγάλης κλίμακας. Χρησιμοποιεί επαναληπτική προσέγγιση για την κατάρτιση σχεδίων που οργανώνονται από την άποψη της ροής εργασίας και των φάσεων, το καθένα αποτελούμενο από μία ή περισσότερες επαναλήψεις. Κάθε κύκλος του έργου ξεκινά με ένα σχέδιο σχετικά με το τι θα πρέπει να υλοποιηθεί και τελειώνει με την ολοκλήρωση μιας αξιολόγησης του κατά πόσο θα έχουν επιτευχθεί οι στόχοι. Η RUP είναι σαν ένας ηλεκτρονικός μέντορας που παρέχει κατευθυντήριες γραμμές, πρότυπα, καθώς και παραδείγματα για όλες τις πτυχές και τα στάδια ανάπτυξης προγράμματος. Η μεθοδολογία RUP χρησιμοποιείται για αντικειμενοστραφή συστήματα.

Το μοντέλο RUP έχει τέσσερις φάσεις στο χρόνο και είναι 1)η έναρξη όπου γίνεται ορισμός και του έργου και της έκτασής του, 2)η επεξεργασία όπου αναφέρεται στην κατάστρωση μεθόδου υλοποίησης του έργου, στην μοντελοποίηση χαρακτηριστικών του έργου και στον ορισμό της αρχιτεκτονικής του συστήματος. 3) η κατασκευή, δηλαδή στην υλοποίηση του έργου και τέλος 4)η μετάβαση δηλαδή στην ανάπτυξη του συστήματος στο περιβάλλον χρήσης του.

Στην εφαρμογή έγινε μια παραλλαγή της μεθοδολογίας RUP, αφού οι γλώσσες που χρησιμοποιήθηκαν στην εφαρμογή δεν είναι αντικειμενοστραφής. Κατά τις επαναλήψεις της ανάπτυξης χρησιμοποιήσαμε τα εξής διαγράμματα: Διάγραμμα περιπτώσεων χρήσης, Διάγραμμα ακολουθίας συστήματος και Διάγραμμα δραστηριοτήτων. Παρακάτω περιγράφονται τα στάδια που ακολουθεί η μεθοδολογία RUP για την ανάπτυξη τέτοιων εφαρμογών.

2.2 Η χρήση της RUP για την ανάπτυξη web εφαρμογών

Η RUP (Rational Unified Process), είναι μια μεθοδολογία Ανάπτυξης Λογισμικού που είναι ένα από τα ισχύοντα Standards Διεθνώς για WEB Εφαρμογές Λογισμικού με προοπτικές επέκτασης. Τα στάδια που ακολουθεί για την ανάπτυξη μιας τέτοιας εφαρμογής είναι:

1. Προσδιορισμός του Μηχανογραφικού Συστήματος

Η ανάπτυξη ενός Μηχανογραφικού Συστήματος μπορεί να χωριστεί σε μερικές βασικές ενότητες οι οποίες εκτελούνται γραμμικά (waterfall lifecycle) η επαναλαμβανόμενα (iterative lifecycle) στον χρόνο ανάπτυξης του.

2. Object oriented (Αντικειμενοστραφής προσέγγιση)

Η "Object oriented" μεθοδολογική προσέγγιση απαντά στα παραπάνω με: Αντικειμενοστραφή Ανάλυση, Αντικειμενοστραφή Σχεδίαση, Αντικειμενοστραφή Προγραμματισμό.

Ανάλυση

Η ανάλυση προσπαθεί να καταλάβει, να εξηγήσει και να παρουσιάσει την φύση ενός συστήματος. Η ανάλυση δεν ασχολείται με τις λύσεις αλλά με ερωτήσεις που προσδιορίζουν το «τι κάνει» και «το περιβάλλον» ενός συστήματος. Πιο συγκεκριμένα οι φάσεις της ανάλυσης είναι:

- Προσδιορισμός των απαιτήσεων του συστήματος.

Ο προσδιορισμός των απαιτήσεων του συστήματος μπορεί να γίνει με την εφαρμογή των Περιπτώσεων Χρήσης (User Cases) που αποδεικνύονται πιο αποτελεσματικά αλλά απαιτούν μεγαλύτερη εκπαίδευση των χρηστών του συστήματος.

- Μοντελοποίηση του περιβάλλοντος της εφαρμογής

Αυτό συνίσταται στον προσδιορισμό των αντικειμένων που αποτελούν τον κορμό μιας εφαρμογής και των αλληλεπιδράσεων μεταξύ αυτών των αντικειμένων. Η μοντελοποίηση γίνεται με την χρήση των διάφορων τύπων των διαγραμμάτων της UML όπως User Case diagrams, Class diagrams, Collaboration diagrams, Sequence diagrams, State chart diagrams.

- Προσδιορισμός των χαρακτηριστικών και των περιορισμών του περιβάλλοντος που θα βοηθήσουν αποτελεσματικά στην σωστή σχεδίαση που θα ακολουθήσει την ανάλυση
- Αναλυτική καταγραφή της υπάρχουσας κατάστασης

Σχεδίαση

Η σχεδίαση και η υλοποίηση είναι έννοιες συμπληρωματικές. Η σχεδίαση απαιτεί πειραματισμό (κατά συνέπεια επί μέρους υλοποιήσεις) και η υλοποίηση σταθερές αρχές (αρχιτεκτονική συστήματος) που προκύπτει από την σχεδίαση. Για αυτό τον λόγο η σχεδίαση χωρίζεται κυρίως σε δύο στάδια:

- Λογική σχεδίαση (Logical step)
- Φυσική σχεδίαση (Physical step)

Προγραμματισμός

Ο προγραμματισμός είναι ο τελευταίος κρίκος της αλυσίδας στην αντικειμενοστραφή σχεδίαση. Η χρήση αντικειμενοστραφών γλωσσών προγραμματισμού διευκολύνει κατά μεγάλο βαθμό την μετάβαση από την φάση της σχεδίασης στην φάση της υλοποίησης της εφαρμογής.

3.Μεθοδολογία Ανάπτυξης Λογισμικού

Η μεθοδολογία αποτελείται από:

- Τα στοιχεία που αποτελούν τα βασικά δομικά υλικά ενός μοντέλου
- Μια σημειολογία που στόχο έχει να απεικονίσει γραφικά τα στοιχεία του μοντέλου
- Μια διαδικασία η οποία να περιγράφει αναλυτικά τα βήματα για την υλοποίηση του συστήματος
- Τεχνογνωσία κατακτημένη από την εφαρμογή της μεθόδου

4. Αρχιτεκτονική Software

Μέσα στα πλαίσια της αντικειμενοστραφούς προσέγγισης η Αρχιτεκτονική συνίσταται από τα πρωτογενή στοιχεία (αντικείμενα και κλάσεις) που θα χρησιμοποιηθούν, τον τρόπο που θα ομαδοποιηθούν ώστε να δημιουργήσουν μεγαλύτερα και πιο λειτουργικά δομικά στοιχεία στον σκελετό της αρχιτεκτονικής (patterns) και την αιτιολόγηση της επιλογής αυτών των patterns.

Η αρχιτεκτονική δίνει μία σφαιρική άποψη για την εφαρμογή στην οποία εφαρμόζεται περιγράφοντας τις στρατηγικές επιλογές οι οποίες προσδιορίζουν την ποιότητα, την αξιοπιστία, την προσαρμοστικότητα και την εγγύηση καλής λειτουργίας της εφαρμογής καθώς επίσης πρέπει να αφήνει περιθώρια για παρεμβάσεις για να ικανοποιήσει ανάγκες που θα προκύψουν στην φάση της ανάπτυξης. Η σωστή αρχιτεκτονική χαρακτηρίζεται από απλότητα, ευκολία κατανόησης, περιορισμό των γενικεύσεων και σαφή διαχωρισμό ανάμεσα στις προδιαγραφές και στην υλοποίηση κάθε επιπέδου.

5. Διαδικασίες Ανάπτυξης

Ένας παραδοσιακός τρόπος αντιμετώπισης στην σχεδίαση συστημάτων απαιτεί τις φάσεις της Ανάλυσης, Σχεδίασης, Προγραμματισμού και Ελέγχου της εφαρμογής. Στην αντικειμενοστραφή προσέγγιση αυτά τα τέσσερα βήματα μπορούν να επαναλαμβάνονται και να εξελίσσουν διαρκώς ένα σύστημα μέχρι να ολοκληρωθεί η λίστα των απαιτήσεων έτσι όπως προκύπτει από την αρχική καταγραφή των User Cases. Σε αυτό το βασικό μοντέλο προστίθενται δύο επιπλέον φάσεις που έχουν να κάνουν αφενός με την σχεδίαση του συγκεκριμένου κύκλου και αφετέρου με τα παραδοτέα κάθε κύκλου.

6. Υλοποίηση Αντικειμενοστραφή Συστήματος

Η υλοποίηση ενός συστήματος έχει δύο βασικές συμπληρωματικές όψεις.

Support View: Αφορά σε στρατηγικές, διαχείριση ανθρώπινων πόρων και οικονομικών.

Technical View: Αφορά στην ανάπτυξη, στον ποιοτικό έλεγχο και στην μεθοδολογία υλοποίησης. [9]

2.3 Ο ρόλος της RUP σε web εφαρμογές

Μια νέα γενιά ανάπτυξης λογισμικού έχει διαμορφωθεί τα τελευταία χρόνια, και μερικές φορές αναφέρεται με τους όρους e-Commerce, e-business. Αυτές οι Web εφαρμογές παρέχουν στους πελάτες και στις επιχειρήσεις, λειτουργίες όπως η on-line επεξεργασία των συναλλαγών, η ενοποίηση της αλυσίδας προμηθειών, το δυναμικό περιεχόμενο, και την εξατομίκευση. Τα συστήματα αυτά έχουν ισχυρή, ευέλικτη και κλιμακούμενη αρχιτεκτονική, με σκοπό να χειρίζονται τις επιδόσεις και το φορτίο που απαιτούνται για την πραγματικότητα των υψηλών όγκων.

Τυπικά αυτές οι αρχιτεκτονικές είναι πολυεπίπεδες, παρέχοντας ενθουσίαση της επιχειρηματικής λογικής και ενσωμάτωση πολλών διαφορετικών συστημάτων «κληρονομιάς».

Η ανάπτυξη Web λύσεων έχει πολλές ομοιότητες με την ανάπτυξη των άλλων εφαρμογών λογισμικού, αλλά και διαφέρει σημαντικά σε πολλούς τομείς. Με πολλούς τρόπους αυτό που δημιουργείται είναι μια μορφή των μέσων μαζικής ενημέρωσης του καταναλωτή, όπως μια κινηματογραφική ταινία ή βίντεο. Όπως κάθε καταναλωτικό προϊόν, η όψη και η αίσθηση της λύσης Web είναι απολύτως κρίσιμη και προϋποθέτει τη συμμετοχή νέων ομάδων ατόμων, συμπεριλαμβανομένης της εμπορίας, δημιουργικών σχεδιαστών, και στελέχη επιχειρήσεων.

2.3.1 Ενσωμάτωση Δημιουργικού σχεδιασμού με την ανάπτυξη παραδοσιακού λογισμικού

Ποτέ πριν δεν είχε δοθεί τόση σημασία στην διασύνδεση χρήστη του λογισμικού εφαρμογών, όπως είναι σήμερα με το web. Τώρα, η οικονομία της αγοράς του web έχει πολύ μεγαλύτερη επιρροή στην επιτυχία ενός λογισμικού με βάση το επιχειρηματικό εγχείρημα. Διότι, στη νέα αγορά των web εφαρμογών, οι πελάτες δεν αγοράζουν το λογισμικό για να το εγκαταστήσουν στις μηχανές τους. Κάνοντας μια εφαρμογή web οπτικά είναι ένα από τα κλειδιά για την επιτυχή λύση στο web, έτσι είναι σημαντικό ότι η δημιουργική διαδικασία σχεδιασμού πρέπει να ενσωματωθεί με τη συνολική διαδικασία ανάπτυξης λογισμικού.

Χρησιμοποιήστε Περιπτώσεις Χρήσης για την ενοποίηση και δημιουργία στην Διαδικασία Τεχνολογίας Λογισμικού.

Οι web λύσεις επεκτείνουν την παραδοσιακή ανάπτυξη λογισμικού από τον συνδυασμό του κόσμου της τέχνης και του δημιουργικού σχεδιασμού με τον κόσμο της τεχνολογίας λογισμικού. Αυτοί οι δύο κόσμοι διαφέρουν όσον αφορά τη διαδικασία, τις δεξιότητες, τον πολιτισμό και τα την κουλτούρα. Οι διαφορές αυτές μπορεί συχνά να αποτελέσουν μείζον εμπόδιο για web έργα όπως η σύγκρουση πολιτισμών. Χρησιμοποιήστε περιπτώσεις που παρέχουν έναν τρόπο για να εκφράσουν, στην κοινή άποψη, μια κοινή κατανόηση για την αναμενόμενη συμπεριφορά της εφαρμογής web.

Χρησιμοποιήστε περιπτώσεις που προβλέπουν την κοινή γλώσσα για τα έργα, μια γλώσσα που ο καθένας που συνδέεται με το πρόγραμμα την χρησιμοποιεί για να περιγράψει αυτό που θα κάνει η λύση στο web και αυτοί οι χρήστες είναι, οι διαχειριστές, οι σχεδιαστές, οι αρχιτέκτονες και οι προγραμματιστές. Επιτρέπουν στον καθένα να «μιλήσει» όσον αφορά την επιχειρηματική λύση που πρέπει να επιλυθεί, και ο καθένας την προσφέρει σύμφωνα με την παρούσα προδιαγραφή.

Απαιτήσεις

Μια web εφαρμογή περιλαμβάνει τους διάφορους ενδιαφερόμενους περισσότερο από κάθε άλλη εφαρμογή. Αυτοί οι ενδιαφερόμενοι είναι συνήθως στελέχη των επιχειρήσεων, του μάρκετινγκ, του δημιουργικού σχεδιασμού, της υποστήριξη πελατών, και της ομάδα ανάπτυξης της τεχνολογίας.

Οι επιτυχημένες λύσεις Web αρχίζουν με μια συγκλονιστική δήλωση οράματος. Η δήλωση όραμα πρέπει να αναπτυχθεί προσωπικά από τους ενδιαφερόμενους φορείς της λύσης, εξασφαλίζοντας συμφωνία σχετικά με τους στόχους του έργου εξαρχής. Το όραμα αυτό εξυπηρετεί διάφορους σκοπούς:

- Επιτυγχάνει συμφωνία σχετικά με το τι προβλήματα πρέπει να επιλυθούν.
- Προσδιορίζει τα όρια του συστήματος.
- Περιγράφει τα πιο σημαντικά χαρακτηριστικά του συστήματος.

Μόλις το όραμα αυτό συμφωνηθεί, διευκολύνονται εργαστήρια που πραγματοποιήθηκαν με τους ενδιαφερόμενους για τον περαιτέρω εντοπισμό των χρηστών του συστήματος και των υπηρεσιών του συστήματος που θα πρέπει να παρέχεται στους χρήστες. Οι χρήστες του συστήματος και η τεκμηρίωση της χρήσης γίνεται μέσω της UML (Unified Modeling Language) γλώσσας. Οι χρήστες εκπροσωπούνται από την UML ως ηθοποιοί, ενώ οι υπηρεσίες εκπροσωπούνται χρησιμοποιώντας την uml έννοια των Περιπτώσεων Χρήσης. Με αυτό τον τρόπο τεκμηρίωσης των απαιτήσεων καθίσταται εύκολο για τους τελικούς χρήστες να διατυπώνουν με σαφήνεια τι υπηρεσίες που χρειάζονται, και για την ομάδα ανάπτυξης στο να επικυρώνει ότι οι απαιτήσεις είναι πλήρεις και ακριβείς.

Δημιουργικό Design Brief

Η ανάπτυξη web λύσεων απαιτεί μεγαλύτερη επικέντρωση στον σχεδιασμό διεπαφής- χρήστη. Παράλληλα με τον εντοπισμό παραγόντων και των περιπτώσεων χρήσης, δεν γνωρίζουμε αν οι πρώτες κατευθύνσεις διεπαφής χρήστη είχαν αναπτυχθεί. Οι κατευθυντήριες γραμμές υψηλού επιπέδου αναφέρονται συχνά ως Σύνομη Δημιουργική Σχεδίαση (Creative Design Brief) στην κοινωνία του web. Η Creative Design Brief ορίζει: την διάθεση του χώρου, πως οι χρήστες θα έχουν πρόσβαση στην υπηρεσία, πως θα χρησιμοποιούν τις μηχανές αναζήτησης, αν το site θα χρησιμοποιεί frames, τους περιορισμούς όσο αφορά το χρώμα του site, και αν απαιτείται οδηγός γραφικών.

Χάρτης πλοήγησης

Οι χάρτες πλοήγησης δείχνουν στους χρήστες πως θα έχουν πλοήγηση σε μια web εφαρμογή και αναπαριστούνται σε ένα ιεραρχικό «δέντρο» διάγραμμα. Κάθε επίπεδο του διαγράμματος δείχνει τον αριθμό των κλικ που απαιτούνται για να φτάσει ο χρήστης στην εκάστοτε σελίδα/οθόνη. Η ανάπτυξη του χάρτη πλοήγησης ξεκινά με τον προσδιορισμό μεγάλων παραθύρων ή ιστοσελίδων για κάθε μια από τις περιπτώσεις χρήσης. Σε αυτό το στάδιο δεν γνωρίζουμε ακριβώς τι κάνει η κάθε οθόνη ή πως θα μοιάζει η κάθε σελίδα έτσι δεν μπορούμε να προσδιορίσουμε ακριβώς μια «λογική σελίδα». Οι λογικές σελίδες αναφέρονται στην διεπαφή του χρήστη αλλά επειδή το περιβάλλον εργασίας εξελίσσεται μπορεί να διαιρεθεί ή να συγχωνευθεί. Οι «λογικές σελίδες» εκπροσωπούνται σε ανάλυση με την χρήση της UML. Ενώ για τον σχεδιασμό και

την εφαρμογή χρησιμοποιούνται HTML έγγραφα. Μόλις προσδιοριστούν οι λογικές σελίδες, ο χάρτης πλοήγησης εξετάζει το πώς ένας χρήστης θα περιηγηθεί από μία λογική σελίδα στην άλλη, καθώς και τις μεγάλες δυνατότητες που παρέχονται από τη λογική οθόνη.

Δημιουργικός Σχεδιασμός Comps

“Comps” αποτελούνται από πλακέτες που δείχνουν πως είναι και πως μοιάζουν οι τοποθεσίες. Η Ορθολογική ενοποιημένη διαδικασία παρέχει μια δραστηριότητα, το Πρωτότυπο το User Interface, το οποίο προβλέπει μια γενική προσέγγιση για τη συγκέντρωση ανάδρασης σχετικά με τη διεπαφή χρήστη. Για την ανάπτυξη Ιστού, την επεκτείνει ελαφρά χρησιμοποιώντας την έννοια του δημιουργικού σχεδιασμού comps. Για να δημιουργηθεί το comps, παίρνουμε την πιο σημαντική περίπτωση χρήσης, για να αναπτυχθούν πολλές εναλλακτικές λύσεις (π.χ., τουλάχιστον 10) για την εμφάνιση και αίσθηση. Από αυτό το σύνολο, επιλέγουμε τριών πλέον ελπιδοφόρες εναλλακτικές δυνατότητες για να παρουσιαστούν στους ενδιαφερόμενους φορείς. Λαμβάνει συνήθως τρεις επαναλήψεις με ενδιαφερομένους πριν υπάρχει συναίνεση σχετικά με το τελικό σχέδιο στο Web. Πρόκειται για μια δημιουργική και επαναληπτική διαδικασία.

Στοιχεία του Web Design

Στοιχεία web design είναι τα διακριτικά εικονίδια που συναρμολογούνται για την κατασκευή των σελίδων Web για μια τοποθεσία. Η τοποθεσία Web πρέπει να παρέχει μια συνεπή εμπειρία χρήστη. Για να εξασφαλιστεί αυτό, στο σχέδιο πρέπει να χρησιμοποιηθούν με συνέπεια ένα σύνολο τυποποιημένων γραφικών στοιχείων σε όλη την περιοχή. Αυτά τα στοιχεία θα πρέπει να αναπτυχθούν, στα πρώτα στάδια του έργου, κατά μήκος με τις κατευθυντήριες γραμμές για τη χρήση τους, για να εξασφαλιστεί ότι όλα τα μέλη της ομάδας κατανοούν πότε και πώς να χρησιμοποιούν αυτά τα συστατικά.

Παραδείγματα Web στοιχείων του σχεδίου περιλαμβάνουν γραφικά στοιχεία, όπως συσκευές πλοήγησης και η σελίδα υπόβαθρο. Η επαναχρησιμοποίηση των υψηλών προτύπων ποιότητας γραφικών στοιχείων σε ολόκληρη την τοποθεσία εξασφαλίζει τη συνοχή, μειώνει το χρόνο στην αγορά και μειώνει το κόστος της ανάπτυξης.

Αρχές Web User Interface Πρωτότυπο

Το πρωτότυπο UI βασίζεται στην υπογραφή του off-comp και είναι κατασκευασμένο κατά την Ορθολογική ενοποιημένη δραστηριότητα. Το πρωτότυπο αυτό χρησιμοποιεί τα στοιχεία σχεδιασμού που αναφέρονται παραπάνω. Οι αρχές web UI υποστηρίζουν κατά κανόνα μόνο τμήματα του συστήματος, με βάση την πιο σημαντική και αντιπροσωπευτική περίπτωση χρήσης. Η ανάπτυξη της Αρχής web UI επιτρέπει στους χρήστες και στους σχεδιαστές την καλύτερη επικοινωνία τόσο με την εμφάνιση και την αίσθηση του χώρου όσο και για την λειτουργικότητα που πρέπει να παραδοθεί.

UI Οδηγίες

Όταν οι αρχές του web είναι πλήρης, είναι καιρός να αναπτυχθούν λεπτομερείς κατευθυντήριες γραμμές για το σχεδιασμό του user interface. Το στυλ του οδηγού αυτού μεταξύ άλλων θα πρέπει να διευκρινίζει πως και πότε τα σχεδιαστικά στοιχεία πρέπει να χρησιμοποιηθούν, τους συνδυασμούς χρωμάτων, τις γραμματοσειρές και λεπτομέρειες για τον τρόπο πλοήγησης.

Πλήρες Web UI Πρωτότυπο

Το πλήρες Web UI Πρωτότυπο αναλύει τις αρχές του για να καλύψει όλες τις περιπτώσεις χρήσης. Το Πρωτότυπο αυτό θα πρέπει να αποδείξει πλήρη πλοήγηση μεταξύ των οθονών και όλων των οπτικών στοιχείων για την τοποθεσία.

Είναι αναμενόμενο ότι οι σελίδες που αναπτύσσονται στο πλαίσιο του πλήρους Web UI Πρωτότυπο υποβάλλεται σε περαιτέρω επεξεργασία κατά τη διάρκεια κάθε μία από τις επαναλήψεις κατασκευής του έργου. Ο στόχος της ανάπτυξης της πλήρους Web UI Πρωτότυπο είναι να έρθει σε συμφωνία με τα ενδιαφερόμενα μέρη σχετικά με το πεδίο και τον ειδικό χαρακτήρα του κάθε χρήστη των διεπαφών.

Πλήρης Χάρτης πλοήγησης

Μετά την ολοκλήρωση του πλήρους Web UI Πρωτότυπο, πρέπει να δημιουργηθεί ο πλήρης Χάρτης Πλοήγησης. Ο χάρτης αυτός πρέπει να βασίζεται στον χάρτη περιοχών αλλά και στην πλήρως καθορισμένη χρήση των

υποθέσεων για την λύση στο Web. Αυτός ο χάρτης θα πρέπει να περιλαμβάνει όλες τις γνωστές σελίδες/ οθόνες που προσδιορίζονται στο Web UI Πρωτότυπο.

Καλύτερες Πρακτικές της RUP

Τα ακόλουθα αντιπροσωπεύουν τις έξι καλύτερες πρακτικές της ανάπτυξης λογισμικού που ενσωματώνονται στην Rational Unified Διαδικασία, μαζί με τις λεπτομέρειες εφαρμογής τους για την ανάπτυξη λύσεων στο Web:

Ανάπτυξη · επαναληπτικό:

Η επαναληπτική ανάπτυξη βασίζεται σε μια προσέγγιση που βασίζεται σε συνεχή ανακάλυψη, εφεύρεση, και εφαρμογή. Είναι δυνάμεις για τον προσδιορισμό των έργων κινδύνων στα πρώτα στάδια της ζωής ανάπτυξης του κύκλου ζωής, καθιστώντας δυνατή τη διαχείριση και την επίθεση τους σε μια συντονισμένη, με έγκαιρο και αποτελεσματικό τρόπο.

Διαχείριση Απαιτήσεων:

Η διαχείριση των απαιτήσεων είναι μια συστηματική προσέγγιση για να προκαλέσουν, την οργάνωση, την επικοινωνία, και διαχείριση μεταβαλλόμενων απαιτήσεων της έντασης λογισμικού συστήματος ή της εφαρμογής. Δεδομένου ότι οι απαιτήσεις εφαρμογής Web μπορεί να αλλάζουν τόσο συχνά όσο τα γούστα και οι τάσεις της αγοράς, παρακολουθείτε η εξέλιξή τους και ο εντοπισμό τους μέσα από την ανάπτυξη ζωής κύκλου που γίνεται όλο και πιο επιτακτική.

Χρησιμοποιώντας Σύνθετες Αρχιτεκτονικές:

Η αρχιτεκτονική αυτή περιγράφει τη δομή της εφαρμογής από την άποψη των στοιχείων, τον τρόπο που είναι ολοκληρωμένες και τους βασικούς μηχανισμούς και τις μορφές με τις οποίες αλληλεπιδρούν. Δεδομένου ότι μια εφαρμογή Web είναι ουσιαστικά ανοιχτή και επεκτάσιμη, και μπορεί να αλλάξει σε συνεχή βάση, η χρησιμοποίηση των αρχιτεκτονικών στοιχείων αποκτά ζωτική σημασία.

Οπτικό Μοντέλο:

Ένα μοντέλο είναι μια απλοποίηση της πραγματικότητας που χρησιμοποιείται για την κατανόηση πολύπλοκων συστημάτων. Μας βοηθά να κατανοήσουμε και να διαμορφώσουμε τόσο το πρόβλημα όσο και τη λύση.

Επαλήθευση Ποιότητας:

Η ποιότητα επικεντρώνεται τόσο στο προϊόν όσο και στην διαδικασία. Οι web εφαρμογές που προορίζονται για την έκθεση του κοινού, καθώς και το κόστος της αποτυχίας μπορεί να είναι δυσβάσταχτο. Σε περιπτώσεις κακής επίδοσης ή αξιοπιστία, η διατήρηση των πελατών είναι σε κίνδυνο.

Αλλαγές ελέγχου:

Οι web εφαρμογές περιέχουν πολλά αντικείμενα και εξαρτήματα που μπορούν να δημιουργηθούν και να τροποποιηθούν από πολλούς ανθρώπους, συχνά παράλληλα. Ο έλεγχος και οι αλλαγές σε αυτή τη συνεχή ανάπτυξη και εγκατάσταση περιβάλλοντος είναι απαραίτητη.[6]

Τελειώνοντας, το συμπέρασμα όλων των παραπάνω είναι ότι η RUP αποτελεί ένα καλό θεμέλιο για την ανάπτυξη Ιστού.

Κεφάλαιο 3

ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΗ

Στο κεφάλαιο αυτό παρουσιάζονται τα έγγραφα ανάλυσης και σχεδίασης που δημιουργήθηκαν πριν την ανάπτυξη της εφαρμογής. Για τον λόγο αυτό χρησιμοποιήθηκε η γλώσσα UML. Η UML είναι μια Ενοποιημένη Γλώσσα Προγραμματισμού(Unified Modeling Language) η οποία χρησιμοποιείται για απεικόνιση, προδιαγραφή, τεκμηρίωση και κατασκευή δομικών συστατικών ενός συστήματος (λογισμικού ή όχι). Αναπαριστά με οπτικό τρόπο τμήματα λογισμικού. Περιλαμβάνει πληροφορίες τόσο για τη στατική δομή όσο και την δυναμική συμπεριφορά ενός συστήματος. Η στατική δομή καθορίζει τα είδη των αντικειμένων που είναι σημαντικά για το σύστημα καθώς και τις συσχετίσεις μεταξύ τους. Η δυναμική συμπεριφορά προσδιορίζει την εξέλιξη των αντικειμένων σε σχέση με τον χρόνο και την επικοινωνία μεταξύ τους. Στόχος της uml είναι η μοντελοποίηση συστημάτων με βάση της αρχές των αντικειμενοστραφών μοντέλων καθώς και η δημιουργία μιας μοντελοποιημένης γλώσσας που μπορεί να χρησιμοποιηθεί τόσο από τον άνθρωπο όσο και από μηχανές.

Η UML λειτουργεί ανεξάρτητα από τις μεθόδους ανάπτυξης λογισμικού και σε πληθώρα πεδίων εφαρμογής. Είναι ανεξάρτητη από πλατφόρμες και από γλώσσες προγραμματισμού. Είναι γλώσσα γραφικής αναπαράστασης με πλούσια σημασιολογία (ο συμβολισμός της είναι κρίσιμος).

3.1 Διάγραμμα Περίπτωσης Χρήσης

Τα διαγράμματα περιπτώσεων χρήσεων περιγράφουν την λειτουργικότητα του συστήματος όπως αυτή γίνεται αντιληπτή από τον εξωτερικό χρήστη. Σκοπός τους είναι να παρουσιάσουν μια γραφική επισκόπηση των λειτουργιών που παρέχονται από ένα σύστημα από την άποψη των φορέων, τους στόχους τους, καθώς και οποιεσδήποτε εξαρτήσεις μεταξύ των περιπτώσεων χρήσης. Διαμερίζουν τη λειτουργικότητα του συστήματος σε συναλλαγές που έχουν νόημα για τους χαρακτήρες (actors) του συστήματος. Τα επιμέρους τμήματα της λειτουργικότητας ονομάζονται περιπτώσεις χρήσης (use cases). Ουσιαστικά περιγράφουν πως αλληλεπιδρούν οι εξωτερικοί χρήστες με το υπό ανάπτυξη σύστημα. Κατά την αλληλεπίδραση αυτή ο χρήστης παράγει συμβάντα προς το σύστημα, ζητώντας ουσιαστικά την απόκριση λειτουργιών του συστήματος. Για

παράδειγμα, όταν ο ταμίας εισάγει τον κωδικό ενός προϊόντος, ζητάει από το POS σύστημα να καταχωρήσει το συγκεκριμένο προς πώληση προϊόν. Το αιτούμενο συμβάν ενεργοποιεί μια λειτουργία του συστήματος.

Τα βήματα για να δημιουργηθεί ένα διάγραμμα περιπτώσεων χρήσης είναι τα εξής:

- Καθορισμός των ορίων του συστήματος
- Εύρεση των χαρακτήρων (actors) και των περιπτώσεων χρήσης
- Κατασκευή γενικού(καθολικού) διαγράμματος περιπτώσεων χρήσης
- Περιγραφή των περιπτώσεων χρήσης με απλό κείμενο ή διαγράμματα
- Εύρεση των συσχετίσεων μεταξύ των περιπτώσεων χρήσης
- Επικύρωση του μοντέλου των περιπτώσεων χρήσης

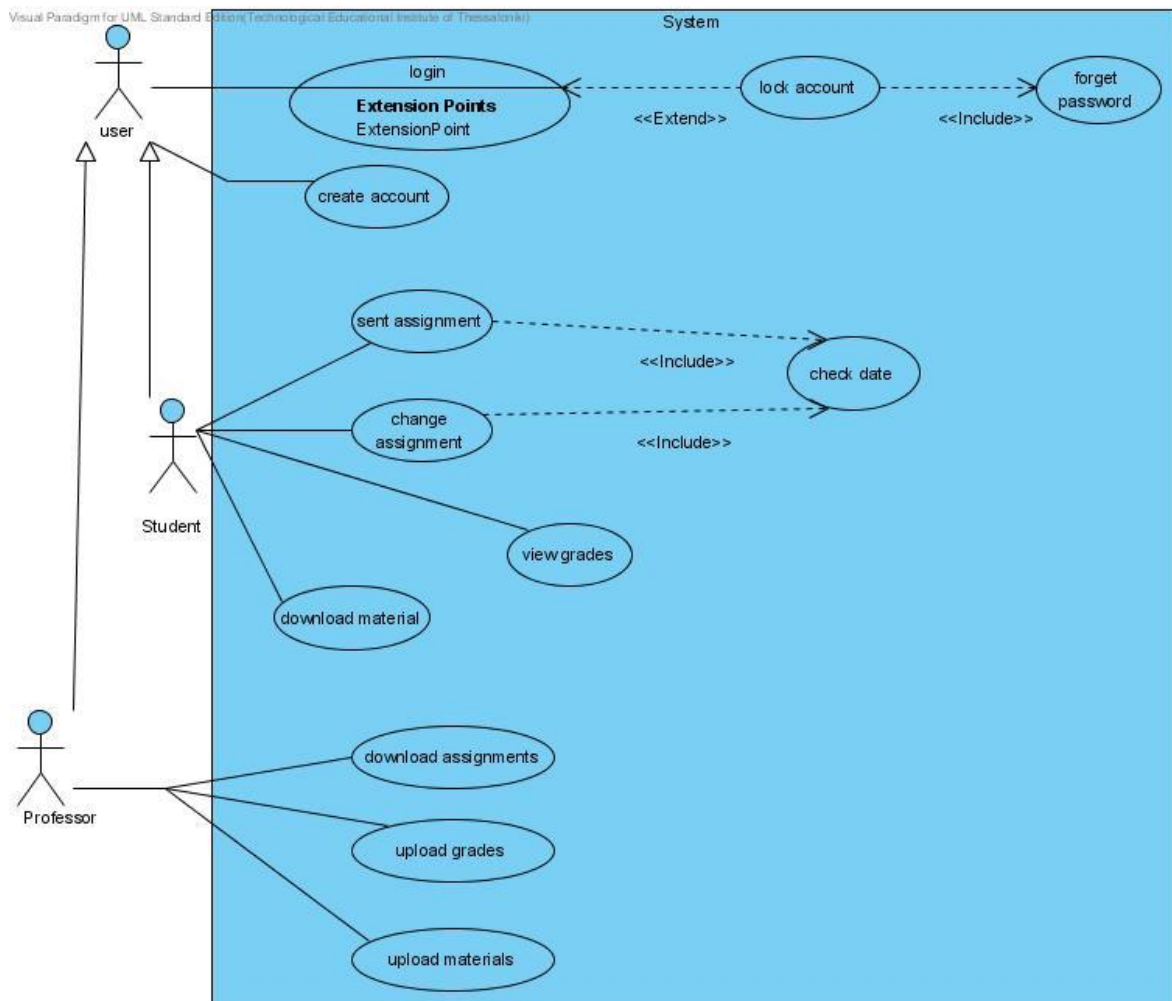
Σε κάθε διάγραμμα περίπτωσης χρήσης απεικονίζεται ένας χρήστης του συστήματος ως ένα σχηματικό ανθρωπάκι. Ο χρήστης ενεργοποιεί μια περίπτωση χρήσης αναμένοντας την εκτέλεση κάποιας συμπεριφοράς. Η συσχέτιση μεταξύ χρήστη και περίπτωσης χρήσης απεικονίζεται με μια ακμή μεταξύ τους ενώ η φορά της ενεργοποίησης υποδηλώνεται με τη χρήση προσανατολισμένης ακμής.

Οι χαρακτήρες(actors) αλληλεπιδρούν με το σύστημα χρησιμοποιώντας το σύστημα, δηλαδή ξεκινώντας την εκτέλεση των περιπτώσεων χρήσης ή με το να χρησιμοποιούνται από το σύστημα, δηλαδή παρέχοντας την λειτουργικότητα για να πραγματοποιούνται οι περιπτώσεις χρήσης. Η σχέση επικοινωνίας τους είναι μη κατευθυνόμενη. Οι χαρακτήρες αναπαριστούν ρόλους. Οι πραγματικοί χρήστες μπορούν να έχουν περισσότερους από έναν ρόλους σε κάθε χρονική στιγμή και πρέπει να βρίσκονται πάντα εκτός των ορίων του συστήματος. Μπορούν να επικοινωνούν με το σύστημα στέλνοντας και λαμβάνοντας μηνύματα.

Οι δυο κύριες σχέσεις μεταξύ των περιπτώσεων χρήσης οι οποίες χρησιμοποιούνται και στην εφαρμογή μας είναι η extend και η include. Μια σχέση extend ουσιαστικά σημαίνει επεκτείνει. Μια περίπτωση χρήσης <<επεκτείνει>> μια άλλη περίπτωση χρήσης σημαίνει ότι η πρώτη περιλαμβάνει κάποια από την

συμπεριφορά της κλάσης που επεκτείνει. Αντίθετα η σχέση include σημαίνει περιλαμβάνει και χρησιμοποιείται όταν υπάρχει κοινή συμπεριφορά μεταξύ use cases. Ακολουθεί η περίπτωση χρήσης που δείχνει τους χρήστες του συστήματος τις διάφορες εφαρμογές και τις σχέσεις μεταξύ τους.

Στο παρακάτω διάγραμμα φαίνονται οι χρήστες τους συστήματος, οι οποίοι είναι ο φοιτητής (Student), ο καθηγητής (Professor) και ο user. Ο χρήστης user περιλαμβάνει τις κοινές λειτουργίες των δυο χρηστών. Οι λειτουργίες αυτές είναι η δημιουργία λογαριασμού (create account), η είσοδος στο σύστημα (login), η οποία επεκτείνεται από το κλείδωμα λογαριασμού (lock account) η οποία περιλαμβάνει την υπενθύμιση κωδικού (forget password). Ο χρήστης Student εκτελεί τις λειτουργίες, αποστολή άσκησης (sent assignment), αλλαγή άσκησης (change assignment) οι οποίες περιλαμβάνουν τον έλεγχο για την ημερομηνία (check date), την κατάσταση βαθμολογίας (view grades) και το κατέβασμα ασκήσεων (download material). Τέλος ο χρήστης Professor εκτελεί τις λειτουργίες: κατέβασμα ασκήσεων (download assignments), ανέβασμα βαθμών (upload grades) και ανέβασμα υλικού (upload materials).



Σχήμα I: Διάγραμμα περίπτωσης χρήσης για την εφαρμογή.

Όλα τα παραπάνω συνοψίζονται στον παρακάτω πίνακα Χρηστών – Στόχων.

Πίνακας I: Πίνακας Χρηστών Στόχων

Χρήστης	Στόχος	Προτεραιότητα
Student, Professor	Create account	1
Student, Professor	Login	1
Student	Sent Assignment	1
Student	Change Assignment	1
Student	View Grades	2
Student	Download Material	3
Professor	Download Assignments	1
Professor	Upload Grades	1
Professor	Upload Material	2

3.2 Δημιουργώντας περιπτώσεις χρήσης

Οι περιπτώσεις χρήσεις που παρουσιάζονται παρακάτω περιγράφουν την επικοινωνία των χαρακτήρων με το σύστημα. Το κείμενο τους απαιτείται να είναι σαφές, πλήρες και συνεπές για να μπορεί να κατανοηθεί. Σύνθετες και περίπλοκες προτάσεις αποφεύγονται. Οι περιπτώσεις χρήσεις σε μορφή κειμένου θα πρέπει να περιλαμβάνουν τα παρακάτω:

- Τον αντικειμενικό στόχο των περιπτώσεων χρήσης (ο στόχος που επιτυγχάνει κάθε περίπτωση χρήσης θα πρέπει να είναι ολοφάνερος).
- Την ροή των μηνυμάτων ανάμεσα στους χαρακτήρες και στις περιπτώσεις χρήσεις (ποια μηνύματα εναλλάσσονται μεταξύ τους).
- Εναλλακτική ροή στην περίπτωση χρήσης (μια περίπτωση χρήσης μπορεί να παρουσιάζει εναλλακτικές εκτελέσεις αναλόγως με κάποιες συνθήκες ή εξαιρέσεις).
- Η περίπτωση χρήσης να δίνει κάποιο αποτέλεσμα στον χαρακτήρα.

Οι παρακάτω περιπτώσεις χρήσης έγιναν με την βοήθεια του εργαλείου Visual Paradigm. Τα πεδία που χρησιμοποιούνται περιγράφονται παρακάτω:

Use Case Id: είναι το όνομα της περίπτωσης χρήσης ή σύντομη περιγραφή της. Συνήθως γράφεται UC από το όνομά της ακολουθούμενο από έναν αριθμό.

Primary Actors: αφορά τους κύριους χαρακτήρες που εμπλέκονται στην περίπτωση χρήσης.

Brief Description: είναι τα προαπαιτούμενα στοιχεία ώστε να γίνει σωστά η εκτέλεση των γεγονότων.

Flow of Events: είναι η ροή των γεγονότων. Τα βήματα δηλαδή που εκτελεί ο χρήστης και το σύστημα, ουσιαστικά το κύριο σενάριο.

Post-Conditions: είναι η κατάσταση του συστήματος μετά από επιτυχή εκτέλεση του σεναρίου. Οι εναλλακτικές ροές, δηλαδή αυτό που θα συμβεί αν κάτι δεν εκτελεστεί σωστά γράφονται στο τέλος αρκεί στην αρχή να γράψουμε τον αριθμό του βήματος στο οποίο εμφανίζεται το λάθος.

Πίνακας II: Η περίπτωση χρήσης UC01 περιγράφει την διαδικασία δημιουργίας λογαριασμού.

Use Case Id	UC01		
Super Use Case			
Primary Actor	User		
Secondary Actor(s)			
Brief Description			
Flow of Events		Actor input	System response
	1	Ο χρήστης ξεκινάει τη διαδικασία εγγραφής στο σύστημα, πατώντας το κουμπί “Δημιουργία λογαριασμού”.	
	2		Τα σύστημα εμφανίζει τη φόρμα “Register”.
	3	Ο χρήστης εισάγει τα στοιχεία του (όνομα, επίθετο, username, φύλλο, χώρα, έτος γεννήσεως)και πατάει “αποδοχή”.	
	4		Το σύστημα στέλνει ένα email στο χρήστη ότι ο λογαριασμός του ενεργοποιήθηκε καθώς και το password του.
Post-Conditions	Ο χρήστης θα μπορεί να δημιουργήσει δικό του λογαριασμό.		
Alternative flows and exceptions			

Non-behavior requirements			
Assumptions			
Issue			
Source			
Author	Dora		
Date	29 Ιουνίου 2009		
5 α. Τα στοιχεία του χρήστη δεν είναι σωστά.		Actor input	System response
	1		Το σύστημα ενημερώνει τον χρήστη για το ποια στοιχεία είναι λανθασμένα.
			Το σύστημα επιστρέφει στο βήμα 2.

Πίνακας III: Η περίπτωση χρήσης UC02 περιγράφει την διαδικασία εισαγωγής στο σύστημα.

Use Case Id	UC02		
Super Use Case			
Primary Actor	User		
Secondary Actor(s)			
Brief Description			
Flow of Events		Actor input	System response
	1	Ο χρήστης ξεκινάει τη διαδικασία εισόδου στο σύστημα, πατώντας το κουμπί «Είσοδος».	
	2		Το σύστημα εμφανίζει τη φόρμα "Login".
	3		Το σύστημα ζητάει το όνομα

			χρήστη και τον κωδικό πρόσβασης.
	4	Ο χρήστης πληκτρολογεί το όνομα χρήστη και τον κωδικό πρόσβασης.	
	5		Το σύστημα ελέγχει την εγκυρότητα του username και password.
	6		Το σύστημα ελέγχει τον τύπο του χρήστη.
	7		Το σύστημα εμφανίζει την φόρμα «Student Menu».
Post-Conditions	Ο χρήστης θα μπορεί να κάνει είσοδο στην σελίδα.		
Alternative flows and exceptions			
Non-behavior requirements			
Assumptions			
Issue			
Source			
Author	Dora		
Date	29 Ιουνίου 2009		
6.a το username ή το password δεν είναι έγκυρα(πλήθος αποτυχιών>3).		Actor input	System response
	1		Το σύστημα ενημερώνει τον χρήστη ότι τα στοιχεία είναι λάθος.
	2		Καλείται η UC02 "Login".
6b. Το		Actor input	System response

username ή το password δεν είναι έγκυρα (πλήθος αποτυχιών=3).	1		Το σύστημα ενημερώνει τον χρήστη ότι τα στοιχεία είναι λάθος.
	2		Καλείτε η UC12 "Lock Account".
6c. Ο λογαριασμός είναι κλειδωμένος.		Actor input	System response
	1		Το σύστημα ενημερώνει τον χρήστη ότι ο λογαριασμός είναι κλειδωμένος.
7a. Ο χρήστης είναι τύπου καθηγητής.	2		Καλείτε η UC11 "Forget Password".
		Actor input	System response
			Το σύστημα εμφανίζει την φόρμα «Professor Menu».

ΠίνακαςIV: Η περίπτωση χρήσης UC03 περιγράφει την διαδικασία αποστολής ασκήσεων του φοιτητή.

Use Case Id	UC03		
Super Use Case			
Primary Actor	Student		
Secondary Actor(s)			
Brief Description	1. Να έχει κάνει login. 2. Να υπάρχει το αρχείο που θέλει να αποστείλει.		
Flow of Events		Actor input	System response
	1	Ο χρήστης ξεκινάει τη διαδικασία αποστολής άσκησης, μέσω του πλήκτρου «Αποστολή Άσκησης».	
	2		Το σύστημα εμφανίζει τη

		φόρμα "Upload Assignment".
	3	Ο χρήστης πατάει το κουμπί "Browse" και επιλέγει την άσκηση που θέλει να αποστείλει καθώς και την εβδομάδα στην οποία αναφέρεται η άσκηση.
	4	Το σύστημα ελέγχει την εγκυρότητα της άσκησης και την ανεβάζει στο δίσκο του server.
	5	Ο χρήστης επαναλαμβάνει το βήμα 3 για όλες ασκήσεις θέλει εκείνος να αποστείλει.
	6	Το σύστημα ελέγχει την εγκυρότητα όλων των ασκήσεων που επιθυμεί ο χρήστης.
	7	Το σύστημα επιβεβαιώνει την ολοκλήρωση της διαδικασίας.
Post-Conditions	Ο χρήστης Student θα μπορεί να αποστείλει ασκήσεις μαθήματος.	
Alternative flows and exceptions		
Non-behavior requirements		
Assumptions		
Issue		
Source		

Author	Dora		
Date	29 Ιουνίου 2009		
6.a		Actor input	System response
Εκπρόθεσμη αποστολή άσκησης.	1		Το σύστημα ενημερώνει ότι η προθεσμία για την υποβολή ασκήσεων έχει λήξει.
	2		Το σύστημα εμφανίζει την φόρμα "Student Form".

Πίνακας V: Η περίπτωση χρήσης UC04 περιγράφει την διαδικασία αντικατάστασης μιας ήδη απεσταλμένης άσκησης.

Use Case Id	UC04		
Super Use Case			
Primary Actor	Student		
Secondary Actor(s)			
Brief Description	1. Να έχει κάνει login. 2. Να έχει αποστείλει άσκηση.		
Flow of Events		Actor input	System response
	1	Ο φοιτητής επιλέγει το κουμπί «Διόρθωση Ασκήσεων» για να αντικαταστήσει μια ήδη απεσταλμένη άσκηση με μια καινούργια.	
	2		Το σύστημα εμφανίζει την φόρμα "sent assignment" με όλες τις μέχρι τώρα απεσταλμένες εργασίες.
	3	Ο χρήστης επιλέγει ποια εργασία θέλει να	

		αντικαταστήσει καθώς και για ποια εβδομάδα.	
	4	Ο χρήστης πατάει το κουμπί “browse” και επιλέγει τη διορθωμένη εργασία.	
	5		Το σύστημα ελέγχει την εγκυρότητα της νέας άσκησης και την ανεβάζει στο δίσκο του server.
	6	Ο χρήστης επαναλαμβάνει τα βήμα 4 για όσες ασκήσεις θέλει εκείνος να αντικαταστήσει.	
	7		Το σύστημα ελέγχει την εγκυρότητα όλων των ασκήσεων που έχει αντικαταστήσει ο χρήστης.
	8		Το σύστημα επιβεβαιώνει την ολοκλήρωση της διαδικασίας.
Post-Conditions	Ο χρήστης Student θα μπορεί να αντικαταστήσει μια ήδη απεσταλμένη άσκηση με μια καινούρια.		
Alternative flows and exceptions			
Non-behavior requirements			
Assumptions			
Issue			
Source			
Author	Dora		
Date	29 Ιουνίου 2009		

5.a Εκπρόθεσμη αποστολή νέας άσκησης.		Actor input	System response
	1		Το σύστημα ενημερώνει ότι η προθεσμία για την υποβολή ασκήσεων έχει λήξει.
	2		Εμφανίζεται η φόρμα “Student Form”.

Πίνακας VI: Η περίπτωση χρήσης UC05 περιγράφει την διαδικασία όπου ο φοιτητής θα μπορεί να δει τους βαθμούς που έχει πάρει για κάθε άσκηση.

Use Case Id	UC05		
Super Use Case			
Primary Actor	Student		
Secondary Actor(s)			
Brief Description	1.Να έχει κάνει login.		
Flow of Events		Actor input	System response
	1	Ο φοιτητής επιλέγει να ενημερωθεί για τους βαθμούς των διορθωμένων του ασκήσεων μέσω του κουμπιού «Κατάσταση Εργασιών».	
	2		Το σύστημα εμφανίζει την φόρμα “Assignments Status”, η οποία περιέχει μια λίστα με τις ήδη απεσταλμένες ασκήσεις του φοιτητή αντιστοιχισμένες με

		τους βαθμούς που έχουν ήδη καταχωρηθεί
Post-Conditions	Ο χρήστης Student θα μπορεί να δει τους βαθμούς του για κάθε άσκηση.	
Alternative flows and exceptions		
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	Dora	
Date	29 Ιουνίου 2009	

Πίνακας VII: Η περίπτωση χρήσης UC06 περιγράφει την διαδικασία όπου ο φοιτητής θα μπορεί να κατεβάσει κάποιο υλικό που έχει ανεβάσει ο καθηγητής.

Use Case Id	UC06		
Super Use Case			
Primary Actor	Student		
Secondary Actor(s)			
Brief Description	1. Να έχει κάνει login. 2. Ο καθηγητής να έχει εισάγει υλικό μαθήματος		
Flow of Events		Actor input	System response
	1	Ο φοιτητής επιλέγει να κατεβάσει το υλικό του	

	μαθήματος.	
	2	Το σύστημα επιλέγει μια λίστα με τις εβδομάδες μαθημάτων που έχουν γίνει, συνοδευμένες από το θέμα της διάλεξης και ένα link προς συνοδευτικό υλικό της καθώς και κάποια σχόλια.
	3	. Ο χρήστης επιλέγει το υλικό της διάλεξης που επιθυμεί.
	4	Το σύστημα αποθηκεύει στον τοπικό δίσκο του χρήστη το υλικό που επέλεξε.
	5	Ο χρήστης εκτελεί το βήμα 3 για όσα υλικά μαθήματος επιθυμεί να δει.
	6	Το σύστημα επαναλαμβάνει το βήμα 4 για όσα υλικά μαθήματος επιθυμεί να δει ο χρήστης.
Post-Conditions	Ο χρήστης Student θα μπορεί να ανοίξει ή να αποθηκεύσει υλικό του μαθήματος.	
Alternative flows and exceptions		
Non-behavior requirements		
Assumptions		
Issue		
Source		

Author	Dora
Date	29 Ιουνίου 2009

Πίνακας VIII: Η περίπτωση χρήσης UC07 περιγράφει την διαδικασία όπου ο καθηγητής θα μπορεί να κατεβάσει τις ασκήσεις των φοιτητών.

Use Case Id	UC07		
Super Use Case			
Primary Actor	Professor		
Secondary Actor(s)			
Brief Description	1. Να έχει κάνει login. 2. Να έχουν στείλει τις ασκήσεις οι φοιτητές.		
Flow of Events		Actor input	System response
	1	Ο καθηγητής επιλέγει να κατεβάσει τις ασκήσεις πατώντας το κουμπί “Κατέβασμα ασκήσεων”.	
	2		Το σύστημα εμφανίζει την φόρμα “download assignments”.
	3		Το σύστημα εμφανίζει μια λίστα με το όνομα του φοιτητή, τον αριθμό μητρώου και την κάθε άσκηση που έχει στείλει ο εκάστοτε φοιτητής.
	4	Ο καθηγητής επιλέγει να κατεβάσει μια άσκηση ενός φοιτητή.	
5		Το σύστημα αποθηκεύει στον τοπικό δίσκο του	

			χρήστη τις ασκήσεις που έχει επιλέξει.
	6	Ο καθηγητής επαναλαμβάνει το βήμα 4 για όσες ασκήσεις επιθυμεί να κατεβάσει.	
Post-Conditions	Ο χρήστης Professor θα μπορεί να κατεβάσει τις ασκήσεις των φοιτητών.		
Alternative flows and exceptions			
Non-behavior requirements			
Assumptions			
Issue			
Source			
Author	Dora		
Date	29 Ιουνίου 2009		
5.a		Actor Input	System response
	1		Το σύστημα ανοίγει το αρχείο που έχει επιλέξει ο χρήστης.

Πίνακας IX: Η περίπτωση χρήσης UC08 περιγράφει την διαδικασία όπου ο καθηγητής θα μπορεί να ανεβάζει τους βαθμούς των φοιτητών για κάθε άσκηση.

Use Case Id	UC08
Super Use Case	
Primary Actor	Professor
Secondary Actor(s)	
Brief	1.Να έχει κάνει login.

Description	2.Να έχει βαθμολογήσει τις ασκήσεις.		
Flow of Events	Actor input	System response	
	1	Ο καθηγητής ξεκινάει την διαδικασία ανεβάσματος βαθμών πατώντας το κουμπί "Grades".	
	2		Το σύστημα εμφανίζει την φόρμα "Upload Grades".
	3		Το σύστημα εμφανίζει μια λίστα με το όνομα του φοιτητή, τον αριθμό μητρώου του και την εβδομάδα της άσκησης.
	4	Ο καθηγητής βαθμολογεί τις ασκήσεις για κάθε φοιτητή που έχει στείλει.	
	5		Το σύστημα επιβεβαιώνει την ολοκλήρωση της διαδικασίας.
Post-Conditions	Ο χρήστης Professor θα μπορεί να ανεβάζει τους βαθμούς των φοιτητών.		
Alternative flows and exceptions			
Non-behavior requirements			
Assumptions			
Issue			
Source			
Author	Dora		
Date	29 Ιουνίου 2009		

Πίνακας X: Η περίπτωση χρήσης UC09 περιγράφει την διαδικασία όπου ο καθηγητής έχει την δυνατότητα να ανεβάσει υλικό για το μάθημα ώστε να μπορούν να το κατεβάσουν και διαβάσουν οι φοιτητές.

Use Case Id	UC09		
Super Use Case			
Primary Actor	Professor		
Secondary Actor(s)			
Brief Description	1. Να έχει κάνει Long in. 2. Να υπάρχουν στον δίσκο τα υλικά που θέλει να ανεβάσει.		
Flow of Events		Actor input	System response
	1	Ο χρήστης ξεκινάει την διαδικασία ανεβάσματος υλικού πατώντας το κουμπί “Ανέβασμα υλικού”.	
	2		Το σύστημα εμφανίζει την φόρμα “Upload materials”.
	3	Ο χρήστης επιλέγει το υλικό που θέλει να ανεβάσει πατώντας το κουμπί Browse.	
	4	Ο χρήστης συμπληρώνει την εβδομάδα, το θέμα του υλικού και τα σχόλια που επιθυμεί.	
	5		Το σύστημα ανεβάζει το υλικό στον δίσκο του server.
6	Ο χρήστης επαναλαμβάνει τα βήματα 3 και 4 αν επιθυμεί να ανεβάσει περισσότερα από ένα		

	υλικά.	
	7	Το σύστημα επαναλαμβάνει το βήμα 5 για όσα υλικά επιθυμεί να ανεβάσει ο χρήστης.
	8	Το σύστημα επιβεβαιώνει την διαδικασία ολοκλήρωσης.
Post-Conditions	Ο χρήστης Professor θα μπορεί να ανεβάσει υλικό μαθήματος στην σελίδα.	
Alternative flows and exceptions		
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	Dora	
Date	29 Ιουνίου 2009	

Πίνακας XI: Η περίπτωση χρήσης UC10 περιγράφει την διαδικασία κατά την οποία ο χρήστης αν πληκτρολογήσει τρεις φορές λάθος κωδικό πρόσβασης να κλειδώνεται ο λογαριασμός του.

Use Case Id	UC10
Super Use Case	
Primary Actor	User
Secondary Actor(s)	
Brief	

Description		
Flow of Events		Actor input
	1	System response
		Το σύστημα κλειδώνει το λογαριασμό του χρήστη.
Post-Conditions	Θα γίνετε κλείδωμα του λογαριασμού.	
Alternative flows and exceptions		
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	Dora	
Date	29 Ιουνίου 2009	

Πίνακας XII: Η περίπτωση χρήσης UC11 περιγράφει την διαδικασία όπου ο χρήστης έχει ξεχάσει τα στοιχεία πρόσβασής του.

Use Case Id	UC11	
Super Use Case		
Primary Actor	User	
Secondary Actor(s)		
Brief Description		
Flow of Events		Actor input
	1	System response
		Ο χρήστης έχει ξεχάσει τα στοιχεία του και πατάει το κουμπί “Υπενθύμιση Κωδικού”.
	2	Το σύστημα στέλνει στον

		χρήστη τα στοιχεία του.
	3	Καλείται η UC02 "Login".
Post-Conditions	Θα στέλνονται τα στοιχεία στον χρήστη εκ νέου σε περίπτωση που τα έχει ξεχάσει.	
Alternative flows and exceptions		
Non-behavior requirements		
Assumptions		
Issue		
Source		
Author	Dora	
Date	29 Ιουνίου 2009	

3.3 Διάγραμμα Ακολουθίας Συστήματος

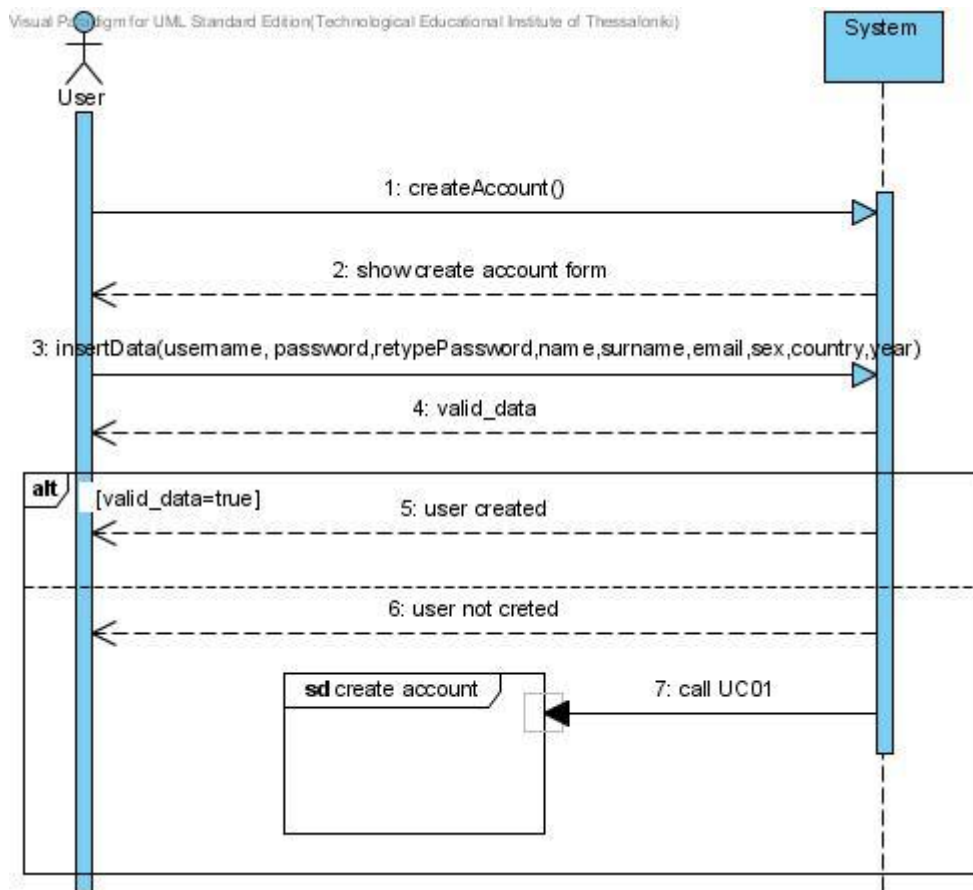
Ένα διάγραμμα Ακολουθίας Συστήματος (SSD – System Sequence Diagram) είναι μια εικόνα που δείχνει, για ένα συγκεκριμένο σενάριο μιας περίπτωσης χρήσης, τα συμβάντα που παράγουν οι εξωτερικοί χρήστες, την σειρά τους και τα συμβάντα μεταξύ των συστημάτων. Τα συστήματα τα απεικονίζουμε σαν μαύρα κουτιά. Το διάγραμμα δίνει έμφαση στα σύμβαντα των χρηστών που διαπερνούν τα όρια τους συστήματος, από τους χρήστες στα συστήματα. Ένα διάγραμμα ακολουθίας θα πρέπει να δημιουργείτε για το βασικό σενάριο επιτυχίας της περίπτωσης χρήσης.

Στα διαγράμματα ακολουθίας εμφανίζεται μια κάθετη γραμμή η οποία αντιστοιχεί στην κλίμακα του χρόνου και μια οριζόντια στην οποία συμβολίζονται ανεξάρτητα αντικείμενα. Τα αντικείμενα συμβολίζονται ως ορθογώνια παραλληλόγραμμα μέσα στα οποία μπορεί να σημειωθεί το όνομα του αντικειμένου που συμμετέχει στο σενάριο που απεικονίζεται. Αν στο σενάριο συμπεριλαμβάνονται αντί για αντικείμενα χρήστες τότε αυτοί συμβολίζονται ως σχηματικά ανθρωπάκια. Συνήθως η επικοινωνία στα διαγράμματα ακολουθίας συστήματος γίνεται σε χρήστες και στο σύστημα.

Σε κάθε αντικείμενο αντιστοιχεί μια κάθετη που ονομάζεται γραμμή ζωής. Για όσο χρόνο ένα αντικείμενο υφίσταται, η γραμμή αυτή είναι διακεκομμένη, ενώ για όσο χρόνο μια διαδικασία του εν λόγω αντικειμένου είναι ενεργή, η γραμμή ζωής σχεδιάζεται ως μια διπλή γραμμή. Ένα μήνυμα που αποστέλλεται μεταξύ αντικειμένων συμβολίζεται ως μια ακμή από τη γραμμή ζωής ενός αντικειμένου προς τη γραμμή ζωής ενός άλλου. Σε κάθε μήνυμα σημειώνεται το όνομά του το οποίο μπορεί να περιλαμβάνει και τυχόν παραμέτρους που να εμπλέκονται σε αυτό. Κάποιες φορές μπορεί να χρειαστεί η επιστροφή μιας μεθόδου και αν χρειαστεί επιστρέφεται και η τιμή της. Η επιστροφή της γίνεται με μια διακεκομμένη ακμή μεταξύ της γραμμής ζωής του καλούμενου αντικειμένου και τη γραμμή ζωής του καλούντος αντικειμένου. [3]

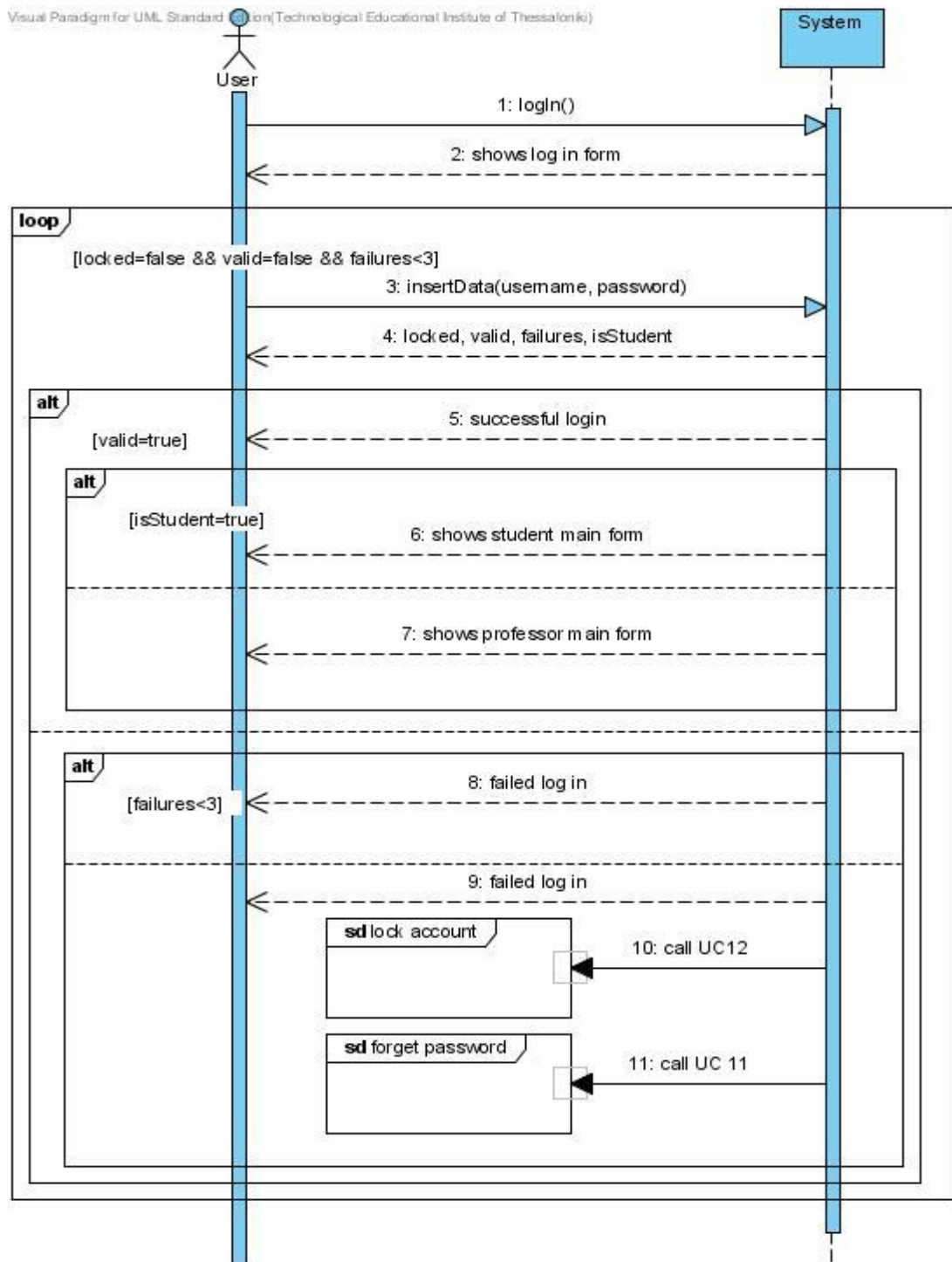
Παρακάτω φαίνονται τα διαγράμματα ακολουθίας συστήματος για όλες τις περιπτώσεις χρήσεις.

Στο παρακάτω διάγραμμα περιγράφεται η αλληλεπίδραση μεταξύ χρήστη και συστήματος κατά την κλήση της μεθόδου `createAccount()`. Αρχικά, το σύστημα εμφανίζει την φόρμα για την δημιουργία λογαριασμού. Ο χρήστης εισάγει τα απαραίτητα στοιχεία τα οποία είναι `username,password,name,surname,e mail, sex,country, year`. Το σύστημα ελέγχει την εγκυρότητα των στοιχείων. Αν αυτά είναι σωστά τότε ο χρήστης έχει δημιουργήσει τον δικό του λογαριασμό. Αν τα στοιχεία δεν είναι έγκυρα τότε το σύστημα ειδοποιεί τον χρήστη ποια στοιχεία είναι λάθος και καλείται η περίπτωση χρήσης `create account`.



Σχήμα II: Διάγραμμα ακολουθίας συστήματος για την δημιουργία λογαριασμού.

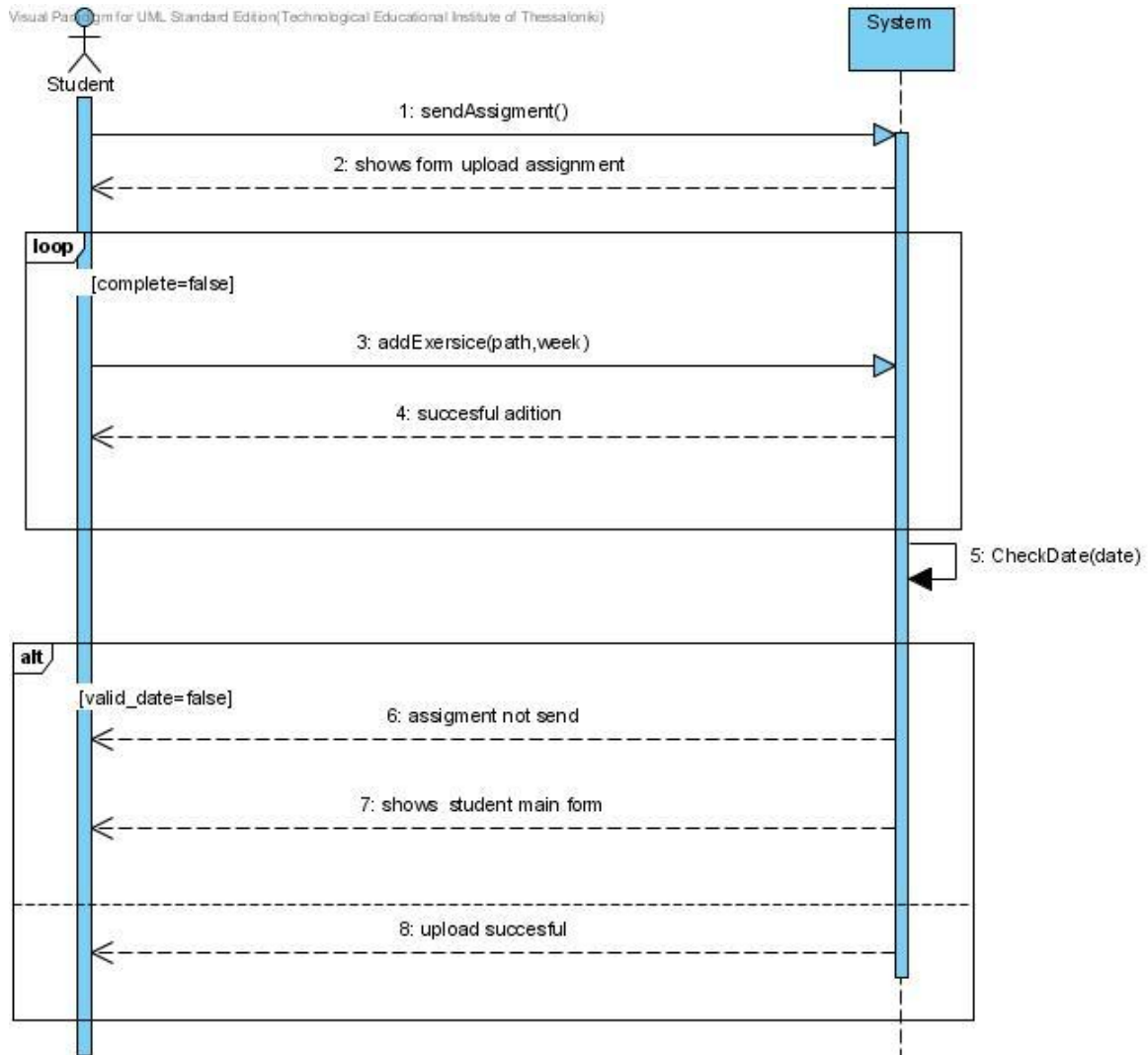
Στο σχήμα III φαίνεται το διάγραμμα που περιγράφει την αλληλεπίδραση χρήστη και συστήματος κατά την κλήση της μεθόδου login(). Το σύστημα εμφανίζει την φόρμα για την είσοδο στην σελίδα. Ο χρήστης πληκτρολογεί το username και το password του και το σύστημα ελέγχει αν τα στοιχεία αυτά είναι έγκυρα. Αν τα στοιχεία είναι έγκυρα τότε ο χρήστης έχει κάνει είσοδο στην σελίδα. Στην συνέχεια το σύστημα ελέγχει αν ο χρήστης είναι τύπου φοιτητής ή τύπου καθηγητής. Αν είναι φοιτητής τότε ανοίγει την αρχική φόρμα του φοιτητή ενώ αν είναι καθηγητής ανοίγει την κεντρική φόρμα του καθηγητή. Αν ο χρήστης πληκτρολογήσει λάθος τα στοιχεία του τότε το σύστημα τον ειδοποιεί με κατάλληλο μήνυμα για την λάθος πληκτρολόγηση του. Αν αυτή η διαδικασία γίνει τρεις φορές λάθος τότε το σύστημα κλειδώνει τον λογαριασμό του χρήστη και καλείται η περίπτωση χρήσης lock account. Ο χρήστης θα έχει την δυνατότητα επίσης να πατήσει τη λειτουργία υπενθύμιση κωδικού. Σε αυτή την περίπτωση το σύστημα καλεί την περίπτωση χρήσης forget password.



Σχήμα III: Διάγραμμα ακολουθίας συστήματος για είσοδο στην σελίδα.

Στο επόμενο διάγραμμα η αλληλεπίδραση χρήστη- συστήματος γίνεται με την κλήση της μεθόδου `sendAssignment()`. Το σύστημα εμφανίζει την φόρμα για την αποστολή άσκησης. Ο χρήστης φοιτητής επιλέγει την άσκηση και την εβδομάδα για την οποία αντιστοιχεί η συγκεκριμένη άσκηση. Το σύστημα ελέγχει αν η αποστολή άσκησης έχει γίνει στο σωστό χρονικό περιθώριο. Αν η άσκηση

ξεπερνάει την ημερομηνία προθεσμίας το σύστημα εμφανίζει κατάλληλο μήνυμα στον χρήστη που τον ειδοποιεί ότι δεν μπορεί να αποστείλει την άσκηση και εμφανίζει την αρχική φόρμα του χρήστη. Ενώ αν η άσκηση δεν ξεπερνάει την ημερομηνία που έχει οριστεί από τον καθηγητή τότε γίνεται επιτυχές ανέβασμα της άσκησης.

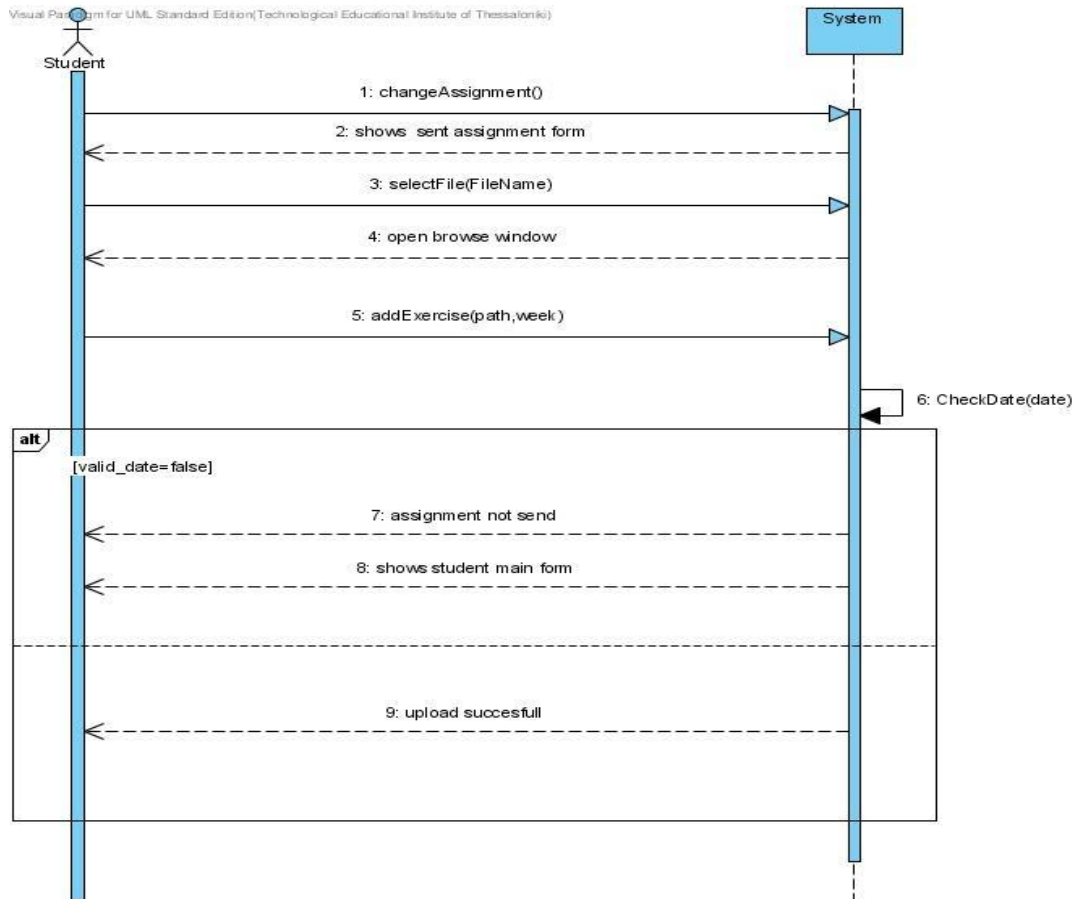


Σχήμα IV: Διάγραμμα ακολουθίας συστήματος για ανέβασμα ασκήσεων από τον χρήστη φοιτητή.

Στο σχήμα V απεικονίζεται το διάγραμμα στο οποίο φαίνεται η αλληλεπίδραση μεταξύ χρήστη και συστήματος κατά την κλήση της μεθόδου changeAssignment(). Το σύστημα εμφανίζει την φόρμα για την αντικατάσταση μιας ήδη απεσταλμένης άσκησης. Ο φοιτητής επιλέγει την άσκηση που θέλει να αντικαταστήσει καθώς και το την εβδομάδα για την οποία προορίζεται η άσκηση. Το σύστημα ελέγχει την εγκυρότητα της άσκησης όσο αφορά τα χρονικά

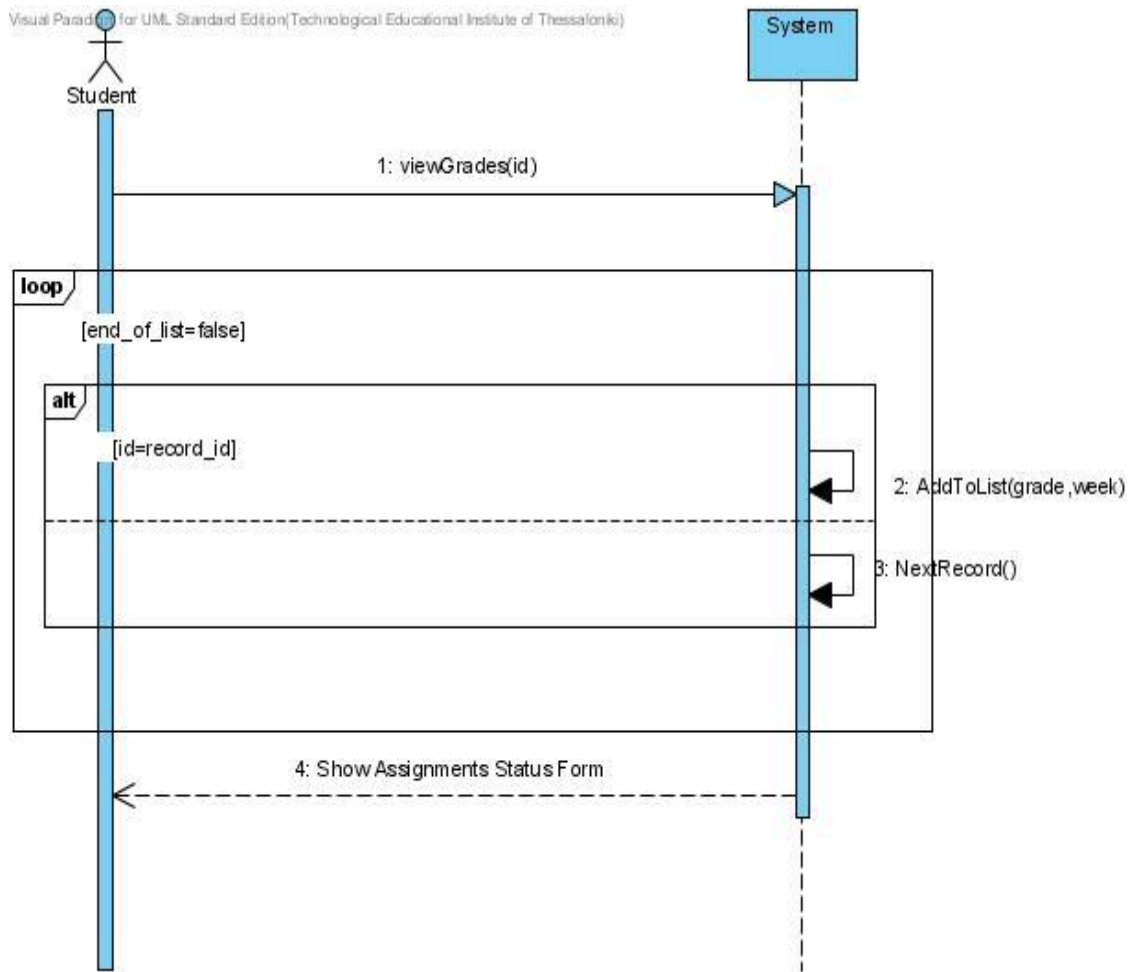
<<Πτυχιακή εργασία της φοιτήτριας Σκευούλη Θεοδώρας>>

περιθώρια ημερομηνίας που ισχύουν. Αν ξεπερνάει το χρονικό περιθώριο που ορίζεται τότε το σύστημα ειδοποιεί τον χρήστη ότι δεν είναι δυνατή η αποστολή της άσκησης και εμφανίζει στον χρήστη την αρχική του φόρμα. Αντίθετα, αν είναι μέσα στο χρονικό περιθώριο που ορίζεται το σύστημα ειδοποιεί τον χρήστη ότι η αποστολή της άσκησης έγινε με επιτυχία.



Σχήμα V: Διάγραμμα ακολουθίας συστήματος για την αντικατάσταση μιας ήδη απεσταλμένης άσκησης.

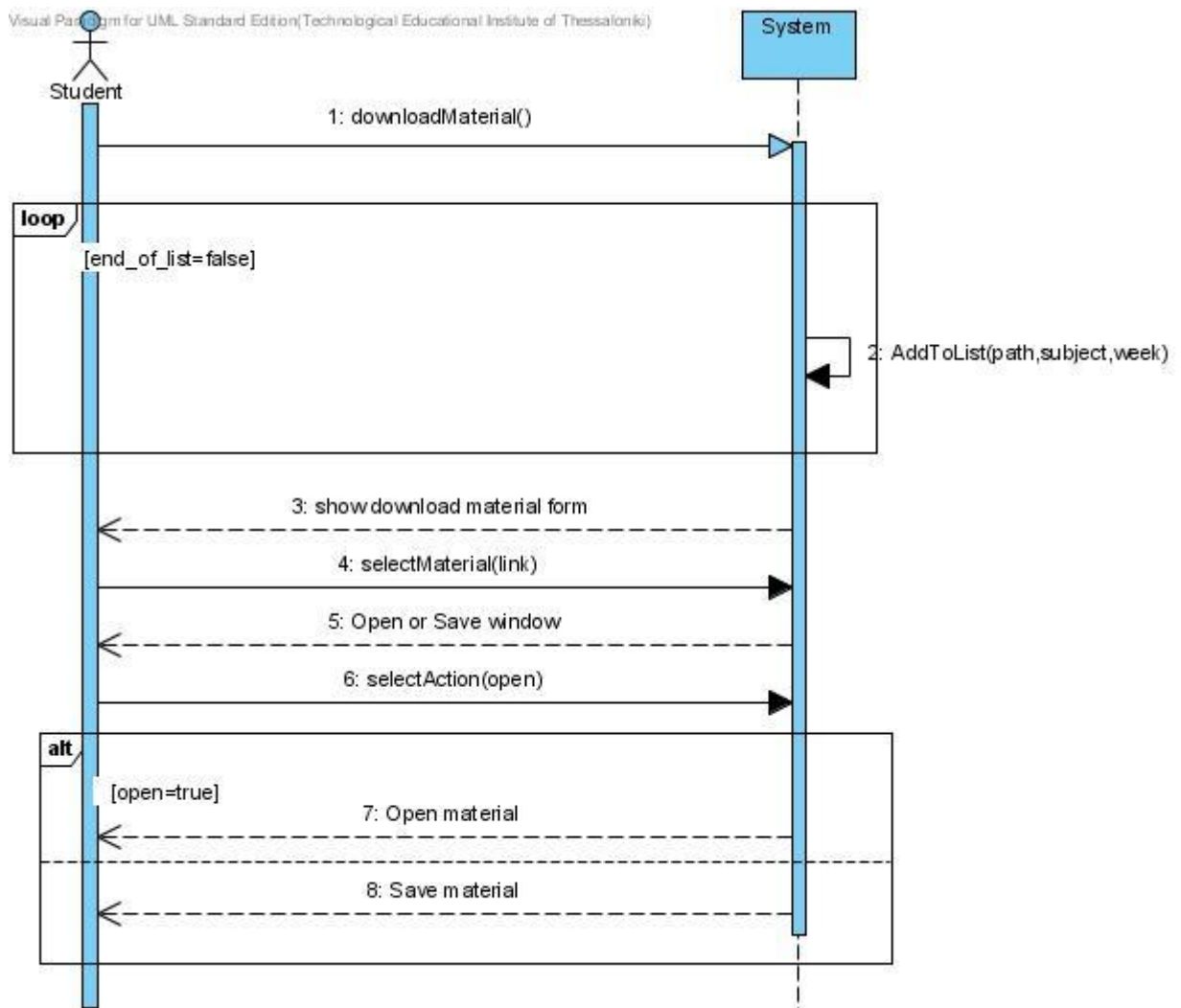
Στο σχήμα VI φαίνεται το διάγραμμα που περιγράφει την αλληλεπίδραση χρήστη και συστήματος που γίνεται κατά την κλήση της μεθόδου viewGrades(id). Το σύστημα εμφανίζει μια φόρμα με τον βαθμό και την βδομάδα της κάθε άσκησης και πηγαίνει στην επόμενη εγγραφή. Ο χρήστης έχει να την δυνατότητα να δει τους βαθμούς του σε μια λίστα για κάθε άσκηση.



Σχήμα VI: Διάγραμμα ακολουθίας συστήματος για εμφάνιση βαθμών.

Στο παρακάτω διάγραμμα φαίνεται η αλληλεπίδραση χρήστη και συστήματος κατά την κλήση της μεθόδου `downloadMaterial()`. Το σύστημα προσθέτει στην λίστα το υλικό, το θέμα του καθώς και την εβδομάδα και στην συνέχεια ανοίγει την φόρμα με όλα τα διαθέσιμα υλικά της λίστας. Ο χρήστης επιλέγει ποιο υλικό θέλει να κατεβάσει και το σύστημα τον ρωτάει αν θέλει απλά να το ανοίξει ή να το αποθηκεύσει στον υπολογιστή του. Αν ο χρήστης επιλέξει άνοιγμα το σύστημα του ανοίγει το υλικό ενώ αν επιλέξει αποθήκευση το σύστημα το αποθηκεύει.

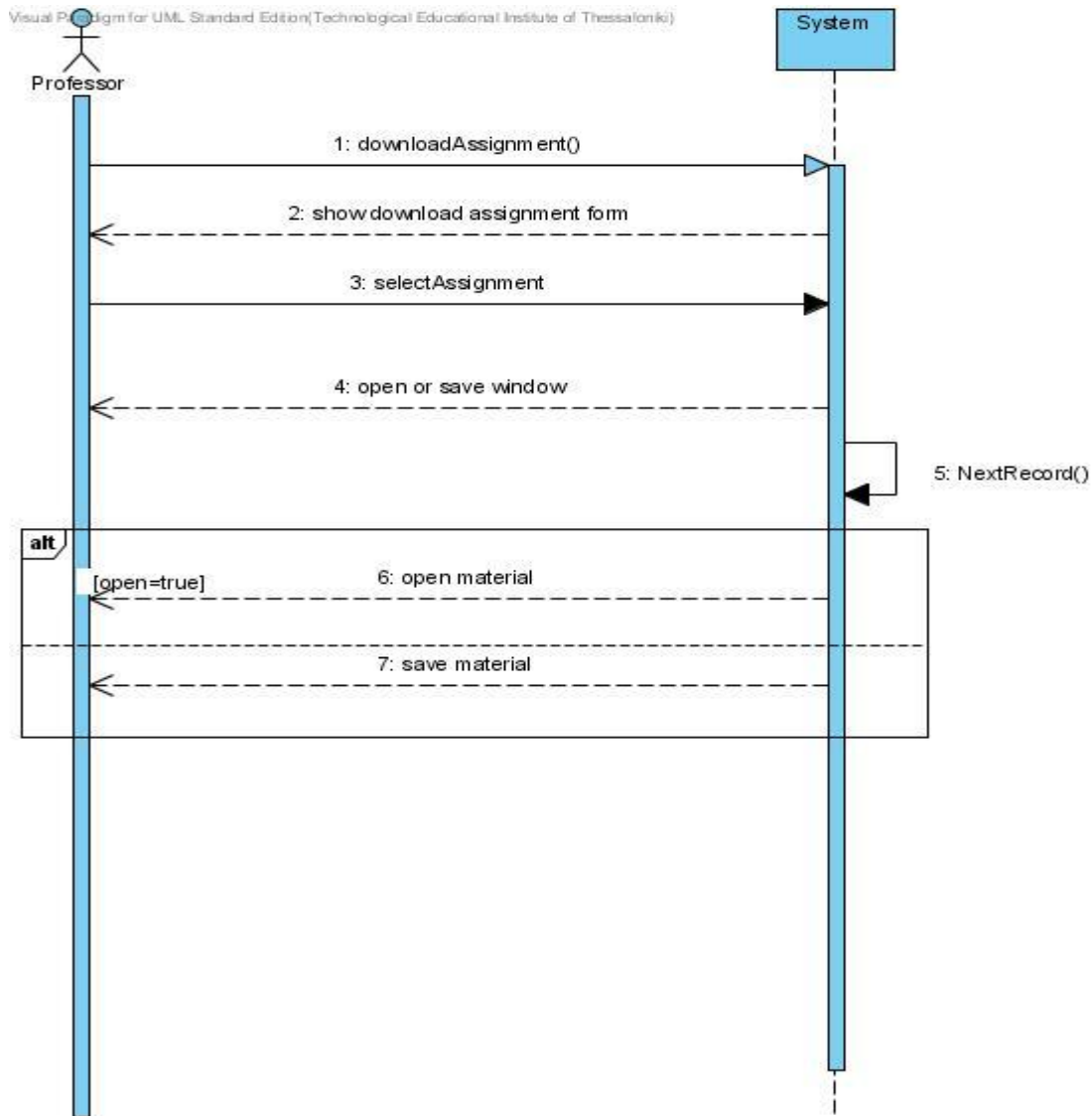
<<Πτυχιακή εργασία της φοιτήτριας Σκευούλη Θεοδώρας>>



Σχήμα VII: Διάγραμμα ακολουθίας συστήματος για κατέβασμα υλικού.

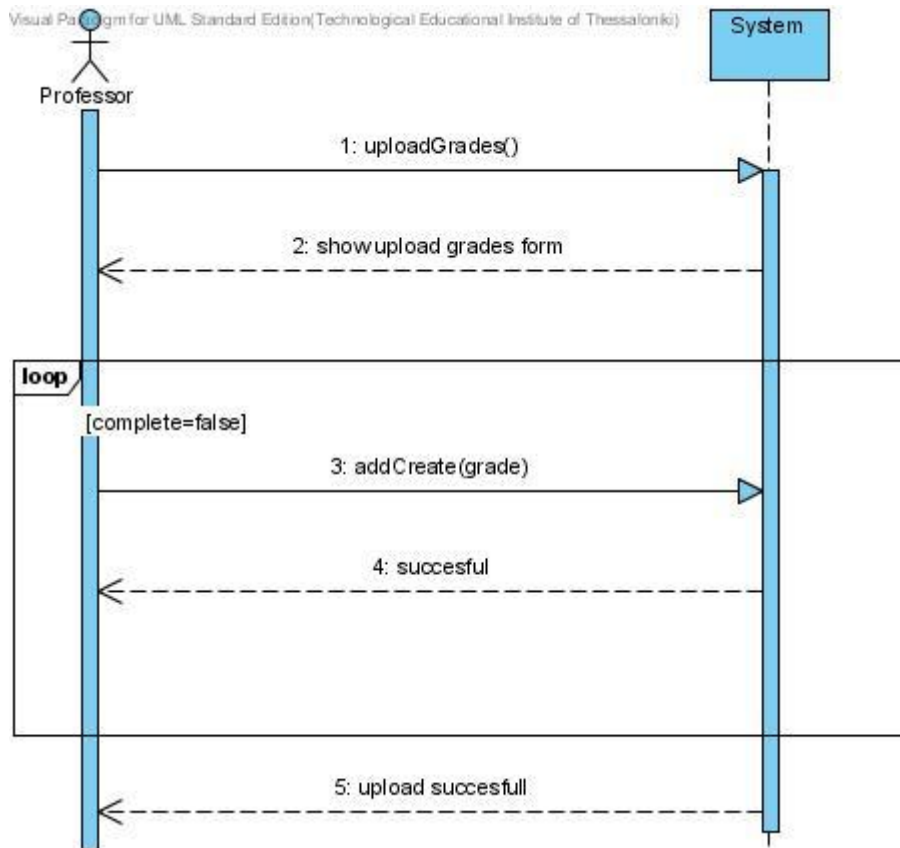
Στο σχήμα VIII περιγράφεται η αλληλεπίδραση μεταξύ χρήστη και συστήματος κατά την κλήση της μεθόδου `downloadAssignment()`. Αρχικά, το σύστημα εμφανίζει την φόρμα για κατέβασμα των ασκήσεων. Ο χρήστης καθηγητής επιλέγει να κατεβάσει την άσκηση ενός φοιτητή και το σύστημα τον ρωτάει αν θέλει να την αποθηκεύσει ή απλά να την ανοίξει. Αν ο χρήστης πατήσει άνοιγμα τότε το σύστημα ανοίγει απλά την άσκηση ενώ αν πατήσει αποθηκεύσει το σύστημα την αποθηκεύει στον υπολογιστή του. Ο καθηγητής επαναλαμβάνει την διαδικασία αυτή για όσες ασκήσεις θέλει να κατεβάσει.

<<Πτυχιακή εργασία της φοιτήτριας Σκευούλη Θεοδώρας>>



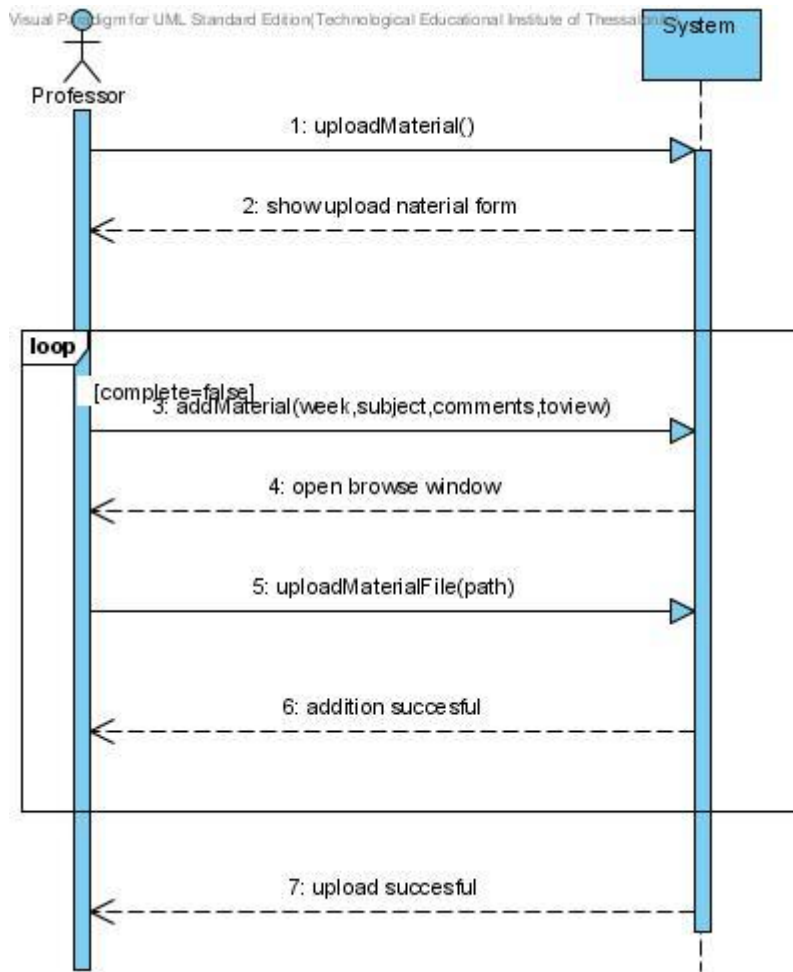
Σχήμα VIII: Διάγραμμα ακολουθίας συστήματος για κατέβασμα ασκήσεων.

Στο παρακάτω διάγραμμα περιγράφεται η αλληλεπίδραση χρήστη και συστήματος κατά την κλήση της μεθόδου `uploadGrades()`. Το σύστημα εμφανίζει την φόρμα για ανέβασμα βαθμών. Ο χρήστης καθηγητής πληκτρολογεί το βαθμό για κάθε φοιτητή που έχει στείλει άσκηση και πατάει ανέβασμα. Τέλος το σύστημα ενημερώνει το χρήστη ότι οι βαθμοί των φοιτητών έχουν ανέβει με επιτυχία.



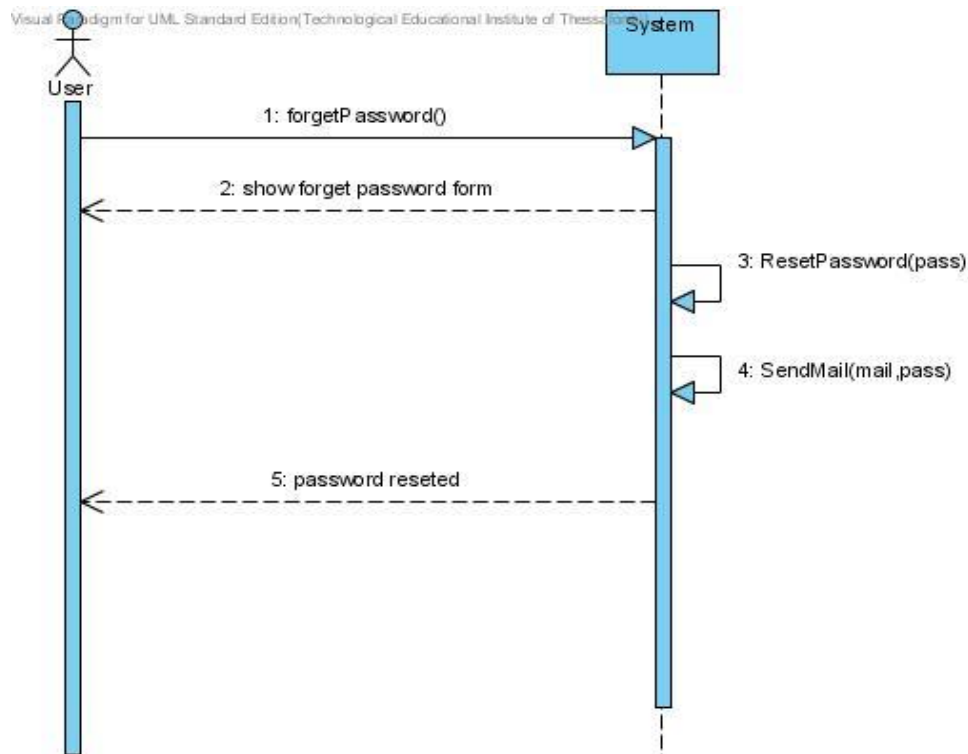
Σχήμα ΙΧ: Διάγραμμα ακολουθίας συστήματος για ανέβασμα βαθμών.

Στο παρακάτω διάγραμμα περιγράφεται η αλληλεπίδραση χρήστη και συστήματος κατά της κλήση της μεθόδου `uploadMaterial()`. Αρχικά το σύστημα εμφανίζει την φόρμα ανέβασμα υλικού. Ο χρήστης καθηγητής προσθέτει την εβδομάδα, το θέμα του υλικού, τα σχόλια και αν θέλει να εμφανίζεται το θέμα του ή όχι. Το σύστημα εμφανίζει το παράθυρο του `browse` και ο καθηγητής επιλέγει το υλικό που επιθυμεί να ανεβάσει. Το σύστημα επιβεβαιώνει την διαδικασία ανεβάσματος υλικού ενώ ο χρήστης καθηγητής μπορεί να επαναλάβει την διαδικασία αυτή για όσα υλικά επιθυμεί να ανεβάσει.



Σχήμα X: Διάγραμμα ακολουθίας συστήματος για ανέβασμα υλικού.

Τέλος, στο παρακάτω διάγραμμα περιγράφεται η αλληλεπίδραση χρήστη συστήματος που γίνεται κατά την κλήση της μεθόδου `forgetPassword()`. Αρχικά, το σύστημα εμφανίζει την φόρμα υπενθύμιση κωδικού. Το σύστημα βρίσκει τυχαία έναν κωδικό και στέλνει email στον χρήστη με το νέο κωδικό. Ο νέος κωδικός έχει σταλθεί με επιτυχία στον χρήστη.



Σχήμα XI: Διάγραμμα ακολουθίας συστήματος υπενθύμιση κωδικού.

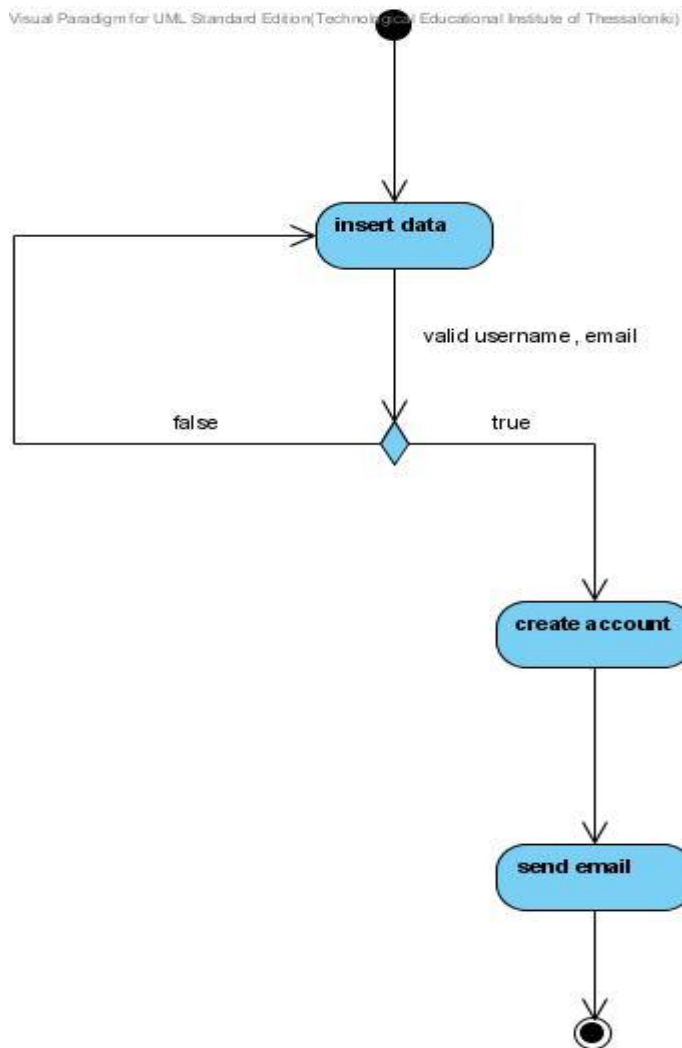
3.4 Διάγραμμα Δραστηριοτήτων

Τα διαγράμματα δραστηριοτήτων χρησιμοποιούνται για να περιγράψουν λειτουργίες, κλάσεις, περιπτώσεις χρήσης, καθώς επίσης και για να δείξουν τη ροή εργασιών. Μπορούν επίσης να χρησιμοποιηθούν για να μοντελοποιήσουν επιχειρησιακές διαδικασίες με εργαζόμενους και οργανωτικές δομές. Είναι χρήσιμα στη μοντελοποίηση βημάτων διαδικασιών σε εφαρμογές όπου αυτές οι διαδικασίες βασίζονται σε ανταλλαγή δεδομένων ή εμπεριέχουν πολύπλοκη λογική ροή. Είναι κατάλληλα για την ανάλυση μιας περίπτωσης χρήσης όταν πρέπει να γίνει κατανοητό ποιες ενέργειες πρέπει να πραγματοποιηθούν υπό διάφορες δυνατές συνθήκες. Επίσης είναι ακόμα χρήσιμα για την περιγραφή πολύπλοκων αλγορίθμων οι οποίοι πρόκειται να υλοποιηθούν από μια ή περισσότερες μεθόδους. Ουσιαστικά φανερώνουν την ακολουθία των γεγονότων και τις αποφάσεις που λαμβάνονται για το ποια δραστηριότητα θα επακολουθήσει.[3]

Ένα διάγραμμα δραστηριότητας περιγράφει τις συνθήκες που καθορίζουν ποιες δραστηριότητες θα εκτελεστούν σε κάθε σημείο του προγράμματος, ποιες δραστηριότητες μπορούν να λαμβάνουν χώρα παράλληλα καθώς και τυχόν επαναληπτικές δομές που περιλαμβάνονται. Μια δραστηριότητα συμβολίζεται ως

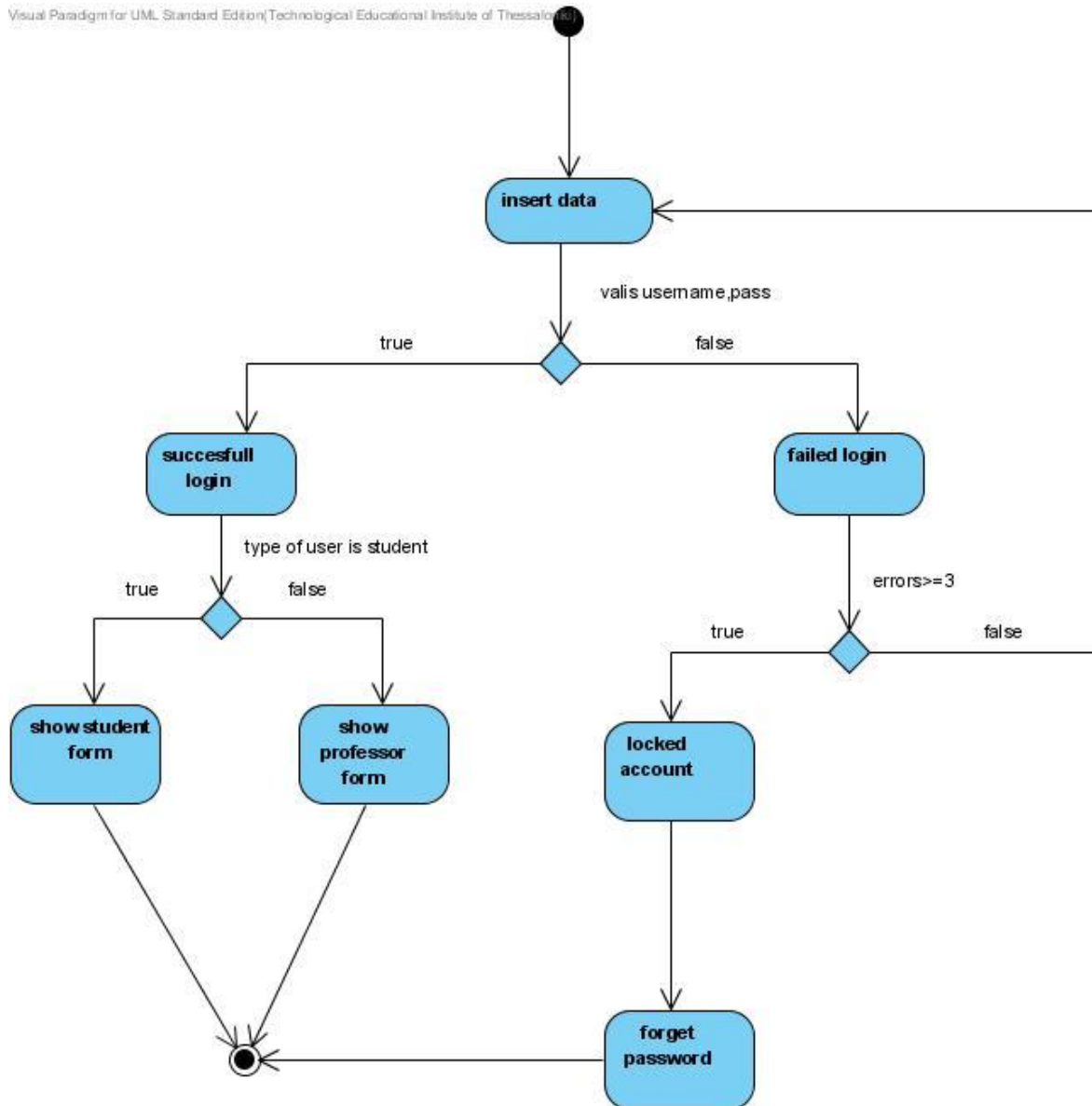
ένα ορθογώνιο με καμπύλες γωνίες. Η μετάβαση κατά την ολοκλήρωση μιας δραστηριότητας σε κάποια άλλη συμβολίζεται ως ακμή. Οι διακλαδώσεις συμβολίζονται είτε με συνθήκες φρουρούς επί των μεταβάσεων είτε με κόμβους απόφασης με πολλαπλές εξερχόμενες ακμές. Τέλος η μαύρη κουκίδα υποδηλώνει την αρχική κατάσταση του συστήματος ενώ η μαύρη κουκίδα με την γραμμή γύρω της υποδηλώνει την τελική κατάσταση του συστήματος.[3] Παρακάτω παρουσιάζονται τα διαγράμματα δραστηριότητας τα οποία χρησιμοποιήθηκαν στην εφαρμογή.

Στο παρακάτω διάγραμμα περιγράφονται οι ενέργειες του συστήματος κατά την εκτέλεση της λειτουργίας «Δημιουργία Λογαριασμού». Αρχικά, ο χρήστης πληκτρολογεί τα στοιχεία του (username, e-mail). Στη συνέχεια γίνεται έλεγχος για την ορθότητα των στοιχείων και αν αυτά είναι σωστά το σύστημα προχωράει στη δημιουργία του λογαριασμού αποστέλλοντας ένα e-mail επιβεβαίωσης στον χρήστη. Αν τα στοιχεία δεν είναι σωστά τότε ο χρήστης πρέπει να ξαναδώσει τα στοιχεία.



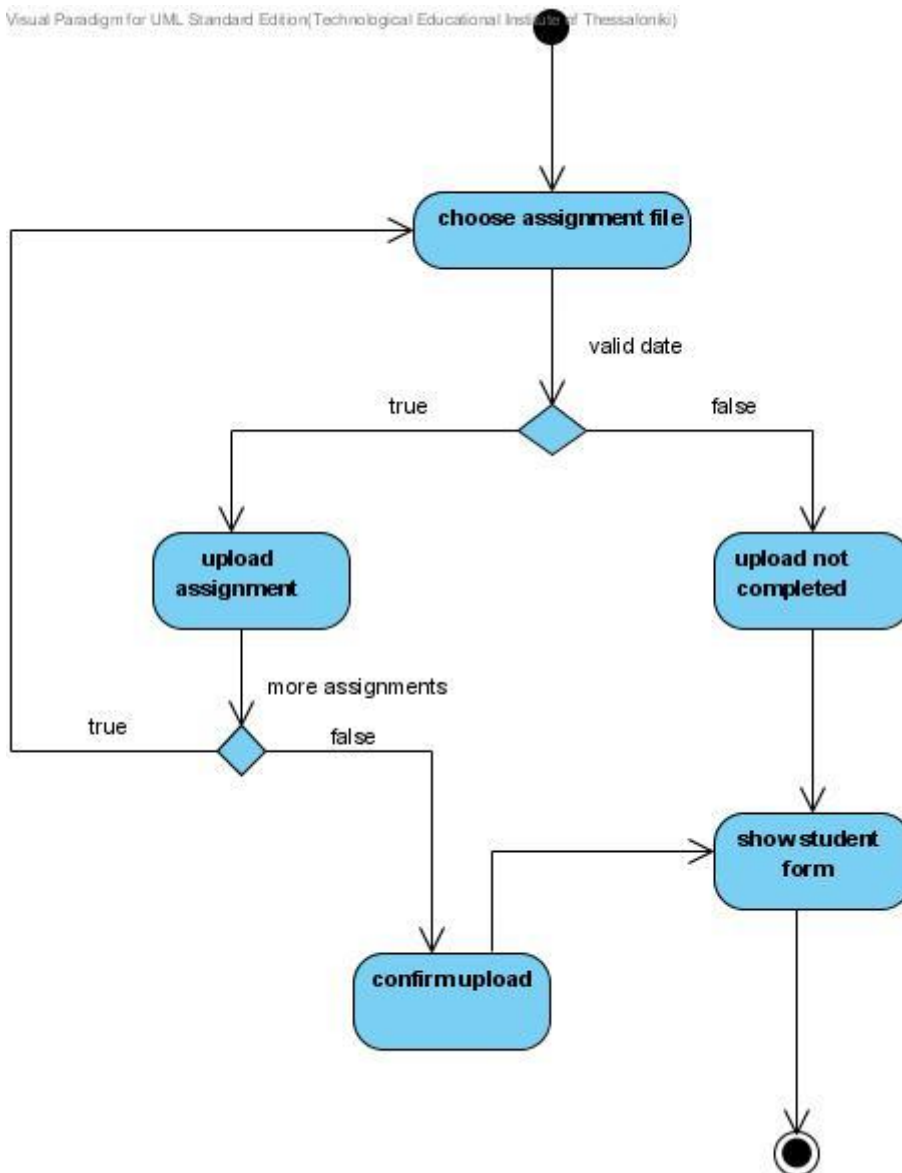
Σχήμα XII: Διάγραμμα δραστηριότητας για δημιουργία λογαριασμού.

Στο παρακάτω διάγραμμα περιγράφονται οι ενέργειες του συστήματος κατά την εκτέλεση της λειτουργίας «Είσοδος στη σελίδα». Ο χρήστης πληκτρολογεί το username και password του. Στην συνέχεια γίνεται έλεγχος των στοιχείων. Αν τα στοιχεία αυτά είναι σωστά τότε ελέγχεται τι τύπου είναι ο χρήστης. Αν είναι τύπου student τότε ανοίγει η κεντρική φόρμα του Student ενώ αν είναι τύπου professor ανοίγει η αντίστοιχη κεντρική φόρμα του Professor. Στην περίπτωση που ο χρήστης δώσει λάθος στοιχεία τότε πρέπει να τα ξανά πληκτρολογήσει. Αν τα πληκτρολογήσει λάθος τρεις φορές τότε ο λογαριασμός του κλειδώνεται και μπορεί να επιλέξει υπενθύμιση κωδικού.



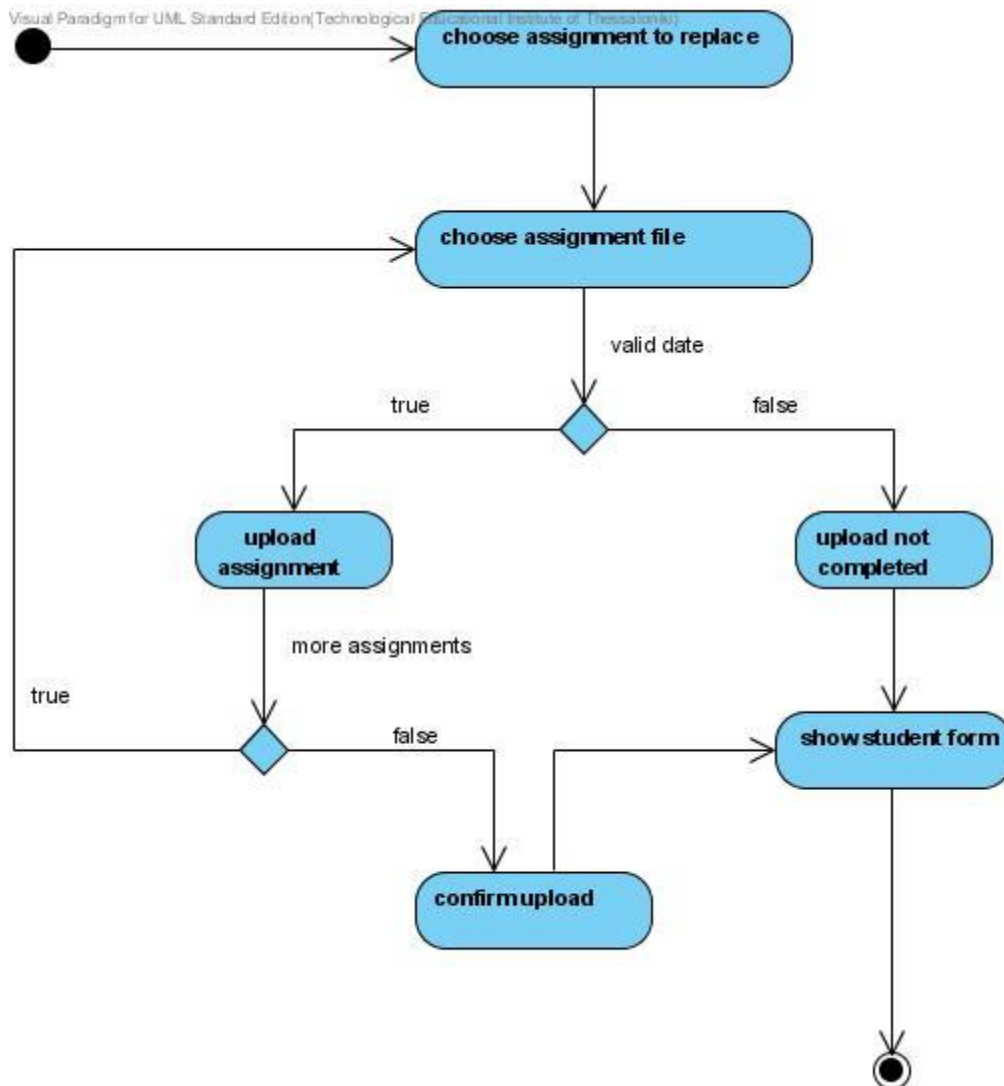
Σχήμα XIII: Διάγραμμα δραστηριότητας για είσοδο στην σελίδα.

Στο σχήμα XIV περιγράφονται οι ενέργειες του συστήματος κατά την εκτέλεση της λειτουργίας «Αποστολή Άσκησης». Αρχικά, ο χρήστης επιθυμεί να στείλει μια άσκηση. Ελέγχεται αν η αποστολή της άσκησης είναι μέσα στο σωστό χρονικό περιθώριο ημερομηνίας. Αν είναι μέσα σε αυτό τότε αποστέλλεται η άσκηση και ο χρήστης μπορεί να επιλέξει να στείλει κι άλλες ασκήσεις αν εκείνος το επιθυμεί. Αν δεν είναι μέσα στο σωστό χρονικό περιθώριο ημερομηνίας τότε γίνεται αποτυχία αποστολής και εμφανίζεται η κύρια φόρμα του χρήστη.



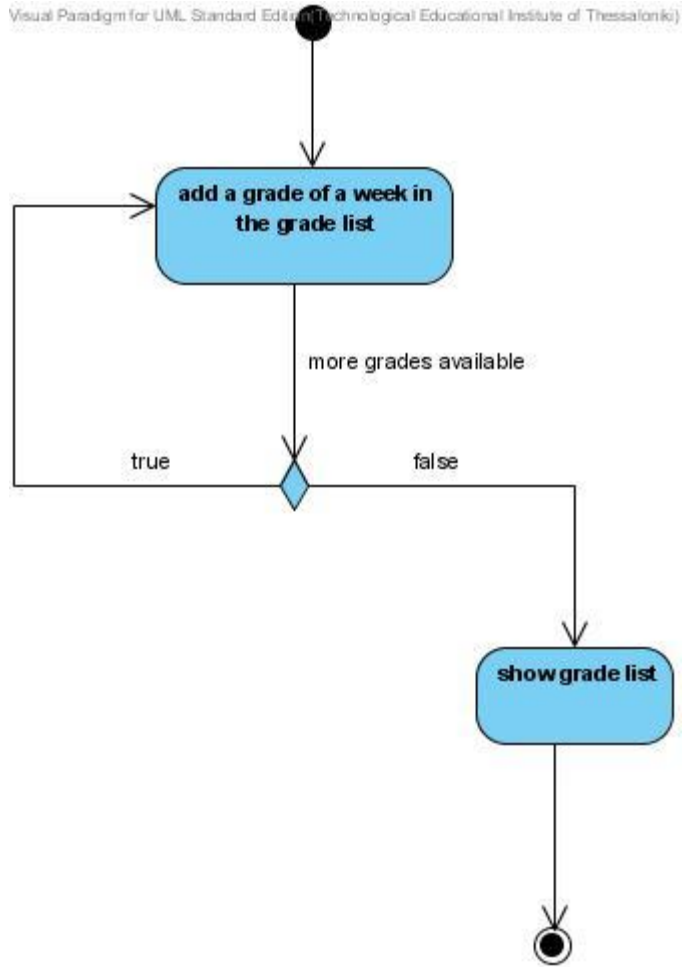
Σχήμα XIV: Διάγραμμα δραστηριότητας για αποστολή άσκησης.

Στο παρακάτω διάγραμμα περιγράφονται οι ενέργειες του συστήματος κατά την εκτέλεση της λειτουργίας «Αντικατάστασης Άσκησης». Ο χρήστης επιθυμεί να αντικαταστήσει μια ήδη απεσταλμένη άσκηση με μια διορθωμένη. Για την αποστολή της ελέγχεται και πάλι το χρονικό περιθώριο της ημερομηνίας. Αν ο χρήστης είναι μέσα σε αυτό γίνεται η ορθή αποστολή αντικατάστασης της άσκησης και έχει την δυνατότητα να αντικαταστήσει και άλλες ασκήσεις. Αν όχι τότε η άσκηση δεν μπορεί να σταλεί και εμφανίζεται η αρχική φόρμα του φοιτητή.



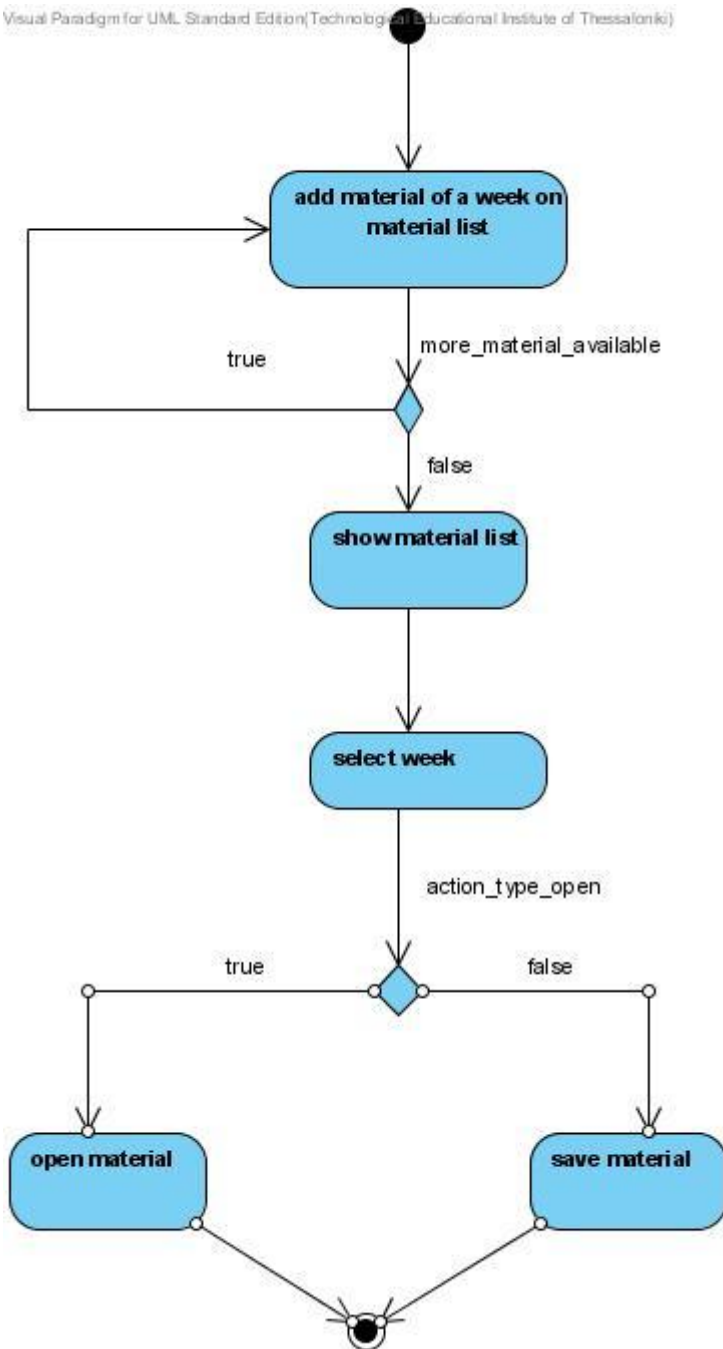
Σχήμα XV: Διάγραμμα δραστηριότητας για την αντικατάσταση μιας απεσταλμένης άσκησης.

Στο παρακάτω διάγραμμα περιγράφονται οι ενέργειες του συστήματος κατά την εκτέλεση της λειτουργίας «Εμφάνιση Βαθμών». Εμφανίζεται μια λίστα με τον βαθμό για κάθε άσκηση του φοιτητή. Γίνεται ο απαραίτητος έλεγχος αν υπάρχουν και άλλοι διαθέσιμοι βαθμοί. Αν υπάρχουν τότε προστίθενται και αυτοί στην λίστα διαφορετικά εμφανίζεται η τελική λίστα με τους υπάρχον βαθμούς.



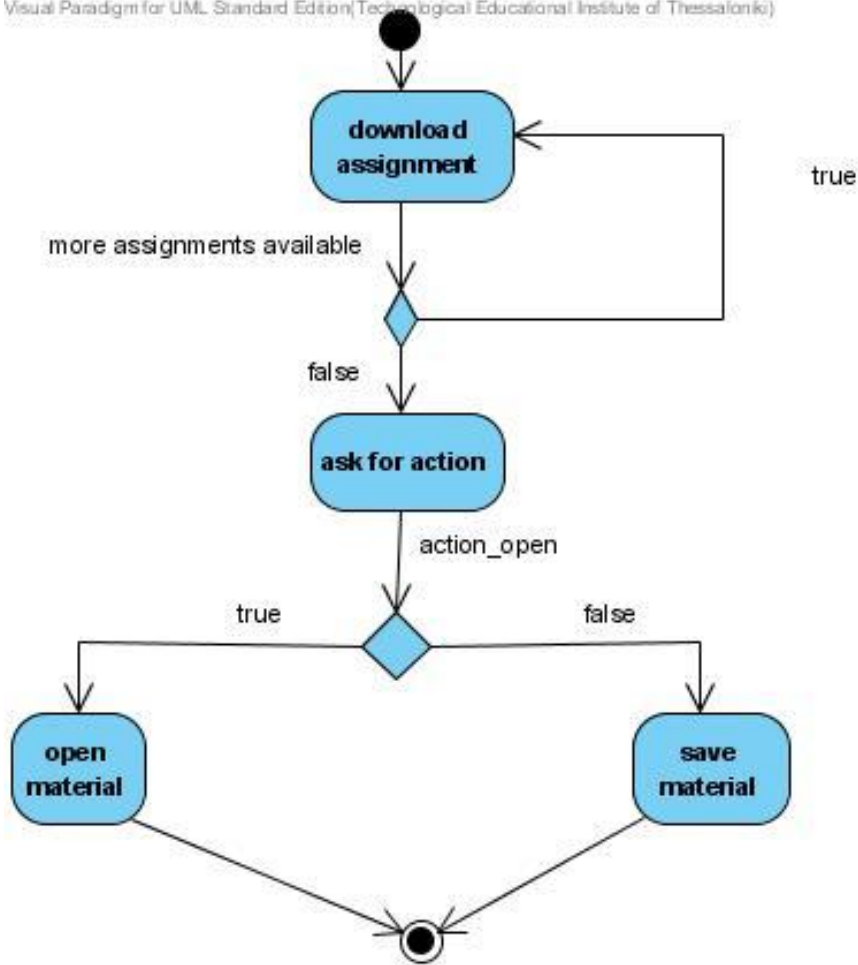
Σχήμα XVI: Διάγραμμα δραστηριότητας για εμφάνιση βαθμών.

Στο παρακάτω διάγραμμα περιγράφονται οι ενέργειες του συστήματος κατά την εκτέλεση της λειτουργίας «Κατέβασμα Υλικού». Ο χρήστης βλέπει τα διαθέσιμα υλικά που μπορεί να κατεβάσει. Αν υπάρχουν και άλλα διαθέσιμα υλικά προστίθενται στην λίστα διαφορετικά ο χρήστης επιλέγει το υλικό που θέλει να κατεβάσει. Τέλος επιλέγεται η αποθήκευση ή το άνοιγμα του υλικού.



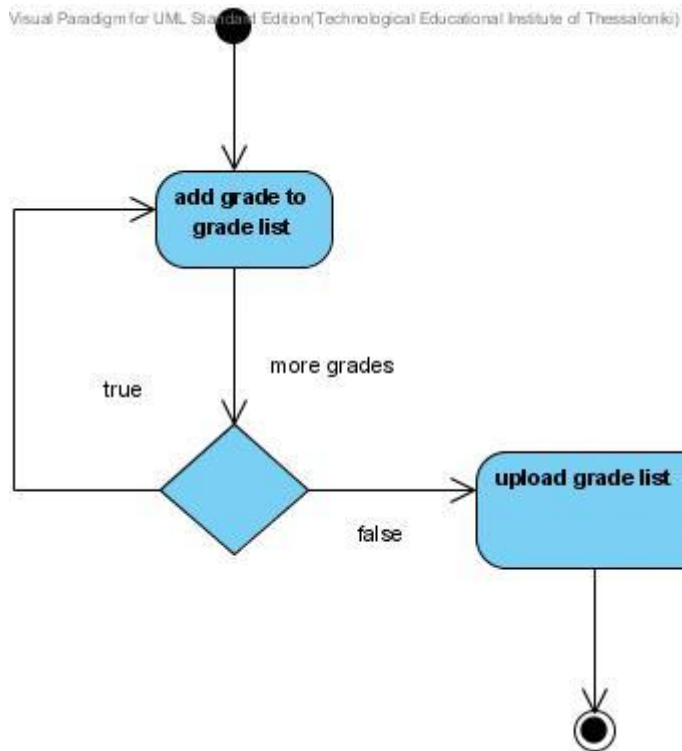
Σχήμα XVII: Διάγραμμα δραστηριότητας για κατέβασμα υλικού.

Στο παρακάτω διάγραμμα περιγράφονται οι ενέργειες του συστήματος κατά την εκτέλεση της λειτουργίας «Κατέβασμα Ασκήσεων». Εμφανίζεται η λίστα με όλες τις απεσταλμένες ασκήσεις που υπάρχουν. Αν υπάρχουν και άλλες διαθέσιμες ασκήσεις τότε αυτές προστίθενται στην λίστα αν όχι τότε γίνεται έλεγχος για το αν θα γίνει άνοιγμα ή αποθήκευση των ασκήσεων.



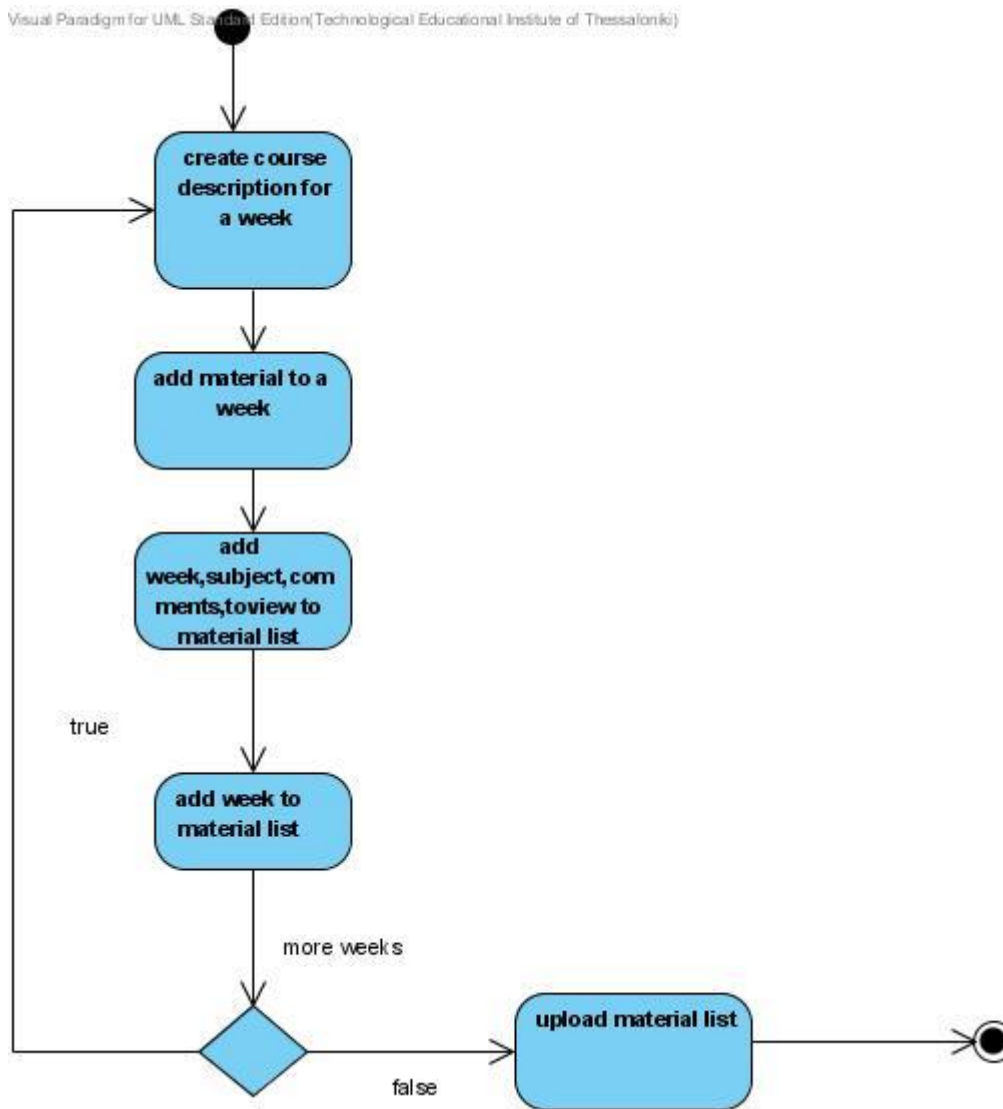
Σχήμα XVIII: Διάγραμμα δραστηριότητας για κατέβασμα ασκήσεων.

Στο παρακάτω διάγραμμα περιγράφονται οι ενέργειες του συστήματος κατά την εκτέλεση της λειτουργίας «Ανέβασμα Βαθμών». Ο χρήστης καθηγητής επιλέγει έναν βαθμό για έναν φοιτητή. Στην συνέχεια γίνεται έλεγχος αν υπάρχουν και άλλες ασκήσεις να βαθμολογήσει. Αν υπάρχουν κι άλλες διαθέσιμες τότε τις βαθμολογεί και αυτές προστίθενται στην λίστα το βαθμών. Τέλος γίνεται η επιτυχής αποστολή τους.



Σχήμα XIX: Διάγραμμα δραστηριότητας για ανέβασμα βαθμών.

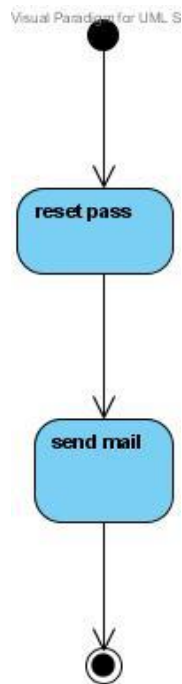
Στο παρακάτω διάγραμμα περιγράφονται οι ενέργειες του συστήματος κατά την εκτέλεση της λειτουργίας «Ανέβασμα Υλικού». Ο χρήστης προσθέτει στην λίστα το υλικό, την εβδομάδα, το θέμα και αν το θέμα θα εμφανίζεται ή όχι. Στην συνέχεια γίνεται έλεγχος αν ο χρήστης επιθυμεί να ανεβάσει και άλλα υλικά. Αν ναι τότε επαναλαμβάνει την ίδια διαδικασία αν όχι απλά ενημερώνεται για το ορθό ανέβασμα υλικού.



Σχήμα XX: Διάγραμμα δραστηριότητας για ανέβασμα υλικού.

<<Πτυχιακή εργασία της φοιτήτριας Σκευούλη Θεοδώρας>>

Τέλος στο επόμενο διάγραμμα περιγράφονται οι ενέργειες του συστήματος κατά την εκτέλεση της λειτουργίας «Υπενθύμιση Κωδικού». Επιλέγεται τυχαία ένας νέος κωδικός και στέλνεται στο email του χρήστη.



Σχήμα XXI: Διάγραμμα δραστηριότητας για υπενθύμιση κωδικού.

ΥΛΟΠΟΙΗΣΗ

Στο κεφάλαιο αυτό παρουσιάζεται η υλοποίηση και η εξήγηση κώδικα που χρησιμοποιήθηκε στην εφαρμογή. Περιγράφονται τα βασικά χαρακτηριστικά των γλωσσών html, css ,php και mysql και κάποια από τα κομμάτια κώδικα που χρησιμοποιήθηκαν για την υλοποίηση της σελίδας.

4.1 Ιστορική αναδρομή στο διαδικτυακό προγραμματισμό

Γλώσσα προγραμματισμού ονομάζεται μια τεχνητή γλώσσα που μπορεί να χρησιμοποιηθεί για τον έλεγχο μιας μηχανής, συνήθως ενός υπολογιστή. Οι γλώσσες προγραμματισμού (όπως άλλωστε και οι ανθρώπινες γλώσσες) ορίζονται από ένα σύνολο συντακτικών και εννοιολογικών κανόνων, που ορίζουν τη δομή και το νόημα, αντίστοιχα, των προτάσεων της γλώσσας. Οι γλώσσες προγραμματισμού χρησιμοποιούνται για να διευκολύνουν την οργάνωση και διαχείριση πληροφοριών αλλά και για την ακριβή διατύπωση αλγορίθμων. Ορισμένοι ειδικοί χρησιμοποιούν τον όρο γλώσσα προγραμματισμού μόνο για τυπικές γλώσσες που μπορούν να εκφράσουν όλους τους πιθανούς αλγορίθμους. Μη υπολογιστικές γλώσσες όπως η HTML δεν λέγονται συνήθως γλώσσες προγραμματισμού. Υπάρχουν χιλιάδες διαφορετικές γλώσσες προγραμματισμού και συνεχώς δημιουργούνται περισσότερες. Κάθε γλώσσα προγραμματισμού έχει το δικό της σύνολο τυπικών προδιαγραφών (ή κανόνων) που αφορούν το συντακτικό, το λεξιλόγιο και το όνομά της. Για τις περισσότερες γλώσσες που χρησιμοποιούνται ευρέως και έχουν χρησιμοποιηθεί για αρκετό χρονικό διάστημα, υπάρχουν ειδικοί οργανισμοί τυποποίησης οι οποίοι μέσα από τακτές συναντήσεις δημιουργούν, τροποποιούν ή επεκτείνουν τις τυπικές προδιαγραφές που διέπουν την χρήση μια γλώσσας προγραμματισμού.

4.1.1 Ιστορικά της html

Η HTML αποτελεί ένα από τα βασικά συστατικά του WWW υποστηρίζοντας την δημιουργία υπερκειμένου. Παρέχει την δυνατότητα προσδιορισμού των δομικών(structural) στοιχείων που συνθέτουν ένα έγγραφο όπως π.χ. είναι οι παράγραφοι, οι λίστες, οι επικεφαλίδες κα. Με βάση τον

προσδιορισμό των στοιχείων αυτών τα προγράμματα που είναι σε θέση να ερμηνεύσουν την HTML παράγουν έργα σε αναγνώσιμη μορφή.

Η HTML ορίστηκε με βάση την SGML(Standard Generalized Markup Language) η οποία αποτελεί διεθνές πρότυπο(ISO 8879: 1986, Information Processing – Text and Office Systems) για την επεξεργασία πληροφορίας σε μορφή κειμένου. Η SGML αποτελεί μεταγλώσσα (meta- language γλώσσα μέσω της οποίας μπορούν να οριστούν άλλες γλώσσες).Στόχο της SGML αποτελεί η μορφοποίηση πληροφορίας on-line για την διανομή, αναζήτηση και ανάκτηση με τρόπο που είναι ανεξάρτητος του τρόπου παρουσίασης του εγγράφου. Ένα έγγραφο που έχει δομηθεί πάνω στην SGML δεν περιέχει δεδομένα σχετικά με τον τρόπο απόδοσής του. Η φυσική μορφή και παρουσίαση του εγγράφου προκύπτει όταν προγράμματα παρουσίασης συνδυάσουν τον κώδικα της SGML του εγγράφου με πληροφορία μορφοποίησης.

Η SGML έχει χρησιμοποιηθεί για τον καθορισμό όλων των επιπέδων της HTML. Έτσι η HTML ακολουθεί την ίδια φιλοσοφία αναφορικά με την ανεξαρτησία δεδομένων, δομής και μορφής. Το περιεχόμενο της γλώσσας έχει οριστεί επίσημα από το IETF(Internet Engineering Task Force)και συγκεκριμένα από το HTML Working Group. Επίσης υπάρχουν σημαντικές βελτιώσεις της γλώσσας οι οποίες προτείνονται από διάφορους κατασκευαστές δεν έχουν όμως ακόμη συμπεριληφθεί στο πρότυπο. Οι επεκτάσεις αυτές επιχειρούν να προσδώσουν στην HTML ένα χαρακτήρα layout και όχι semantic markup γλώσσας. Τα επίπεδα 0 και 1 της HTML παρέχουν ένα βασικό σύνολο από constructs που είναι ερμηνεύσιμα από όλους. Στο επίπεδο 2 [2] ενσωματώνονται τα constructs εκείνα που επιτρέπουν την κατασκευή διαλογικών εφαρμογών(HTML forms). Το επίπεδο 3 [3] δεν έχει εξελιχθεί ακόμη σε πρότυπο, αυξάνει όμως σημαντικά το περιεχόμενο της γλώσσας σε θέματα παρουσίασης.[4]

4.1.2 Ιστορικά του css

Με την χρήση CSS ουσιαστικά επιχειρείται στην HTML η μετάβαση από τη σειριακή- διαδικαστική σήμανση στην γενική. Οι σημάνσεις HTML δημιουργήθηκαν αρχικά για να ορίσουν το περιεχόμενο ενός εγγράφου και να πετύχουν την μορφοποίηση του. Με την χρήση σημάνσεων για παράδειγμα μπορούμε να ορίσουμε μια επικεφαλίδα, μια παράγραφο, ένα πίνακα στοιχείων,

χρησιμοποιώντας τις κατάλληλες σημάνσεις. Η διάταξη του HTML εγγράφου αρχικά γινόταν απλά, χωρίς τη χρήση σημάνσεων μορφοποίησης.

Σιγά- σιγά στον αρχικό πυρήνα της HTML προστέθηκαν σημάνσεις όπως η σήμανση FONT που μπορούσε κάποιος να χρησιμοποιήσει για να μορφοποιήσει κείμενο. Επίσης σε διάφορες σημάνσεις προστέθηκαν ιδιότητες (attributes), όπως π.χ. οι ιδιότητες BGCOLOR, COLOR κλπ., για την μορφοποίηση του κειμένου. Έτσι τα μειονεκτήματα της διαδικαστικής σήμανσης έγιναν προφανή. Έτσι, καθώς οι σημάνσεις μορφοποίησης μπλέκονταν με τις σημάνσεις ορισμού κειμένου γινόταν ολοένα και δυσκολότερη η δημιουργία HTML εγγράφων και δικτυακών τόπων όπου το περιεχόμενο των εγγράφων HTML να μπορεί να ξεχωρίζει και να το διαχειρίζεται κάποιος ανεξάρτητα από την διάταξη παρουσίασης του εγγράφου.

Για να βρεθεί λύση σε αυτό το πρόβλημα, προστέθηκε το attributes STYLE και γενικά η έννοια των Styles και των Cascading Style Sheets. Με την εφαρμογή των styles είναι δυνατόν με πολύ πιο απλό και αποτελεσματικό τρόπο να μεταβάλλουμε συνολικά την μορφοποίηση μιας HTML ιστοσελίδας, άλλα ακόμη πιο σημαντικό τη μορφοποίηση ενός μεγάλου συνόλου από HTML ιστοσελίδες που αποτελούν π.χ. ένα μεγάλο web site.

4.1.3 Ιστορικά του PHP

Η Php είναι μια γλώσσα script από την πλευρά του διακομιστή, σχεδιασμένη ειδικά για το web. Μέσα σε μια HTML μπορεί να ενσωματωθεί PHP κώδικας, που θα εκτελείτε κάθε φορά που θα επισκέπτεστε την σελίδα. Ο PHP κώδικας μεταφράζεται στο web διακομιστή και δημιουργεί HTML ή άλλη έξοδο που θα δει ο επισκέπτης.

Η PHP δημιουργήθηκε το 1994 και ήταν αρχικά η δουλειά ενός ατόμου, του Rasmus Lerdorf. Υιοθετήθηκε και από άλλα ταλαντούχα άτομα και έχει περάσει από τρεις βασικές εκδόσεις. Τον Ιανουάριο του 2001 ήταν σε χρήση σχεδόν σε πέντε εκατομμύρια τομείς παγκόσμια και αυτός ο αριθμός μεγαλώνει γρήγορα.

Η PHP είναι ένα προϊόν ανοιχτού κώδικα. Θα έχετε πρόσβαση στον κώδικα προέλευσης. Μπορείτε να τον χρησιμοποιήσετε, να τον αλλάξετε και να τον αναδιανείμετε χωρίς χρέωση.[5]

Η PHP αρχικά σήμαινε Personal Home Page (προσωπική αρχική σελίδα) αλλά άλλαξε σύμφωνα με την σύμβαση GNU και τώρα σημαίνει PHP Hypertext Preprocessor (προεπεξεργαστής κειμένου PHP).

4.1.4 Ιστορικά της MySQL

Η MySQL είναι ένα πολύ γρήγορο και δυνατό σύστημα διαχείρισης βάσεων δεδομένων. Μια Βάση Δεδομένων σας επιτρέπει να αποθηκεύετε, να αναζητάτε, να ταξινομείτε και να ανακαλείται τα δεδομένα αποκλειστικά. Ο MySQL διακομιστής ελέγχει την πρόσβαση στα δεδομένα σας, για να μπορούν να δουλεύουν πολλοί χρήστες μαζί, για να παρέχει γρήγορη πρόσβαση και να διασφαλίζει ότι μόνο πιστοποιημένοι χρήστες μπορούν να έχουν πρόσβαση. Συνεπώς η MySQL είναι ένας πολυνηματικός διακομιστής πολλαπλών χρηστών. Χρησιμοποιεί την Sql(Structured Query Language) την τοπική γλώσσα ερωτημάτων για βάσεις δεδομένων παγκόσμια. Η MySQL είναι διαθέσιμη από το 1996 αλλά η ιστορία της ξεκινά από το 1979.

4.2 Προγραμματισμός σε html

Tag ονομάζουμε μία οδηγία γραμμένη σε HTML, την οποία πρέπει να αναγνωρίσει και ερμηνεύσει ο browser. Τα tags βρίσκονται πάντοτε μεταξύ των συμβόλων < και >. Παρακάτω παρουσιάζονται οι βασικότερες ετικέτες html που χρησιμοποιήθηκαν στην εφαρμογή.[2]

Ετικέτα <html> <HTML>...</HTML> Κάθε Web σελίδα αρχίζει με την οδηγία (tag) <HTML> και τελειώνει με την εντολή </HTML>. Η πρώτη πληροφορεί τον Web client πως το αρχείο είναι γραμμένο σε γλώσσα HTML και η τελευταία πως ο HTML κώδικας τελείωσε.

Ετικέτα <head> Η ετικέτα αυτή περιέχει πληροφορίες σχετικά με το HTML έγγραφο, συμπεριλαμβάνει τις λέξεις κλειδιά που περιγράφουν την ιστοσελίδα καθώς και συνδέσμους σε άλλα αρχεία τα οποία χρησιμοποιεί το έγγραφο όπως τα αρχεία CSS.[7]

Ετικέτα <base> Με την σήμανση αυτή ορίζεται ρητά ένα url που χρησιμοποιείται για να επιλύσει τις συνδέσεις και τις αναφορές στις εξωτερικές πηγές όπως html αρχεία, εικόνες κλπ όταν οι αναφορές-συνδέσεις δεν είναι πλήρης. Ένα URL είναι η διεύθυνση ενός αρχείου μέσα στο Internet και αποτελείται από 3 τμήματα: α) Το πρωτόκολλο που πρέπει να χρησιμοποιήσουμε

για να αποκτήσουμε αυτό το αρχείο. β) Ο κόμβος στον οποίο είναι τοποθετημένο το αρχείο. γ) Το directory (με το πλήρες path ή τμήμα αυτού κατά περίπτωση) που βρίσκεται το αρχείο και φυσικά το όνομα του αρχείου.

Ετικέτα <link> Με την σήμανση link μπορούμε να συνδέσουμε ένα html έγγραφο με άλλους πόρους. πχ να συνδέσουμε δυο σελίδες έτσι ώστε κάνοντας κλικ σε κείμενο ή εικόνα της μιας να μεταφερόμαστε στην άλλη.[7]

Ετικέτα <meta> Οι σημάνσεις meta μπαίνουν στο head μέρος της html σελίδας και δίνουν κάποιες πληροφορίες ή οδηγίες για την ιστοσελίδα. Βοηθούν τις μηχανές αναζήτησης στην ταξινόμηση του site. Χωρίζονται σε δυο κατηγορίες σε αυτές που γράφουμε μια σύντομη περιγραφή της σελίδας και σε αυτές που περιέχουν τις λέξεις κλειδιά οι οποίες περιγράφουν τις σελίδες.

Ετικέτα <style> Η ετικέτα αυτή δημιουργεί ένα εσωτερικό στυλ στο έγγραφο.

Ετικέτα <body> Η ετικέτα body αναφέρεται στο “σώμα” του κειμένου και πρέπει να γράφεται μετά την σήμανση <html> και να κλείνει πριν την σήμανση </html>.

Ετικέτα επικεφαλίδας headings Η οδηγία επικεφαλίδας καθορίζει το μέγεθος των γραμμάτων της επικεφαλίδας και ισχύει ανεξάρτητα από την γραμματοσειρά που χρησιμοποιεί ο browser για να διαβάσει την σελίδα. Η μεγαλύτερη σε μέγεθος επικεφαλίδα είναι η <H1>, ακολουθούμενη από τις <H2>...<H6>.

Ετικέτα <p> Η ετικέτα <p> ομαδοποιεί το κείμενο σε μια παράγραφο, αφήνοντας αυτόματα μια κενή γραμμή πριν την αρχή της παραγράφου και μετά το τέλος αυτής.

**Ετικέτα και ** Η ετικέτα εισάγει μια μη αριθμημένη λίστα στην σελίδα μας. Η ετικέτα προσθέτει γραμμές στη λίστα. Κάθε γραμμή της λίστας πρέπει να περικλείεται μέσα στα και , ενώ όλες οι γραμμές θα πρέπει να περικλείονται μέσα στα και .

**Ετικέτα
** Η ετικέτα
 χρησιμοποιείται όταν θέλουμε μια γραμμή κειμένου και να αρχίζουμε μια καινούρια.

Ετικέτα <div> Η ετικέτα <div> ορίζει ένα τμήμα στο HTML έγγραφο. Ένα τμήμα ομαδοποιεί τα περιεχόμενα που είναι μέσα στις ετικέτες <div>.....</div> ώστε να μπορούμε να ορίζουμε ένα ενιαίο στυλ (χρώμα, γραμματοσειρά, στοίχιση) με την ιδιότητα style.

**Ετικέτα ** Η ετικέτα χρησιμοποιείται για να δηλώσει την εισαγωγή ενός αρχείου εικόνας στο HTML έγγραφο και απαιτεί την χρήση της ιδιότητας src για να καθοριστεί η θέση του αρχείου.

Επίσης απαιτείται η ιδιότητα alt η οποία παρέχει ένα εναλλακτικό κείμενο το οποίο προσδιορίζει την εικόνα. Το κείμενο αυτό εμφανίζεται στην περίπτωση που δεν είναι διαθέσιμη η εικόνα και δεν είναι δυνατόν να εμφανιστεί στον browser. Επίσης, χρησιμοποιείται για να αναγνωστεί από συσκευές ανάγνωσης οθόνης σε άτομα με προβλήματα όρασης.

Ετικέτα <a> Με την ετικέτα αυτή λέμε στον browser να μεταφερθεί σε μια διαφορετική τοποθεσία από την τρέχουσα όταν ο χρήστης κάνει κλικ πάνω στον σύνδεσμο που εμφανίζεται ανάμεσα στα <a>..... Αυτή είναι και η κύρια χρήση της ετικέτας <a>, δηλαδή η δημιουργία συνδέσμου (link) προς ένα άλλο έγγραφο χρησιμοποιώντας την ιδιότητα href.

Ετικέτα <href> Η ετικέτα <href> ορίζει τη URL, είτε διεύθυνση (<http://www.....>) είτε μια άλλη ιστοσελίδα(file.html).

Ετικέτα <table> Την ετικέτα <table> την χρησιμοποιούμε κυρίως για να κατασκευάσουμε πίνακες σε περίπτωση που θέλουμε να εμφανίσουμε κάποια δεδομένα σε σειρές και σε στήλες, δηλαδή για την οργάνωση και παρουσίαση πληροφορίας σε μορφή πίνακα. Το στοιχείο ενός πίνακα ξεκινάει με την σήμανση <table> και τελειώνει με την σήμανση </table>. Ανάμεσα στις σημάνσεις αυτές μπορούμε να καθορίσουμε την αρχή και το τέλος μιας γραμμής και μιας στήλης του πίνακα με τη βοήθεια των σημάνσεων <tr></tr> και <td></td> αντίστοιχα. Ενώ με την σήμανση <th> ορίζουμε επικεφαλίδες σε έναν πίνακα. Το κείμενο που βρίσκεται μέσα στις ετικέτες <th>....</th> εμφανίζεται με bold χαρακτήρες.

Ετικέτα <form> Οι φόρμες αποτελούν τον μηχανισμό για την αποστολή δεδομένων από τον πελάτη web client στον εξυπηρετητή web server. Η βασική ιδέα πίσω από τις φόρμες είναι ότι ο χρήστης στον client συμπληρώνει δεδομένα στα στοιχεία μιας φόρμας που εμφανίζονται μπροστά του και μόλις συμπληρωθούν τα δεδομένα ο χρήστης με κάποιο τρόπο αποστέλλει τα δεδομένα στον web server. Ο web server δέχεται τα δεδομένα και συνήθως τα επεξεργάζεται ή τα αποθηκεύει κάπου. Οι φόρμες είναι και αυτές html αρχεία και κατά συνέπεια σχηματίζονται χρησιμοποιώντας κατάλληλες html σημάνσεις.

Για να δημιουργήσουμε μια φόρμα χρειαζόμαστε την σήμανση <form>. Μια βασική ιδιότητα της φόρμας είναι η METHOD η οποία λαμβάνει δυο πιθανές τιμές

την get και την post και καθορίζει πως τα δεδομένα που πληκτρολογεί ο χρήστης στον web browser θα αποσταλούν στον server. Όταν η τιμή της ιδιότητας αυτής είναι get τα δεδομένα αποστέλλονται μέσω της URL με την μορφή string το οποίο απαρτίζεται από ζεύγη ονόματος/ τιμής για κάθε δεδομένο. Όταν η τιμή της ιδιότητας method είναι post, τα δεδομένα αποστέλλονται απευθείας στον χειριστή φόρμας της εφαρμογής για επεξεργασία στον εξυπηρετητή. Η πρώτη χρησιμοποιείται συνήθως όταν απαιτούνται στατικά δεδομένα από τον εξυπηρετητή για προσωρινή χρήση ή όταν αυτά θα ξαναχρησιμοποιηθούν σύντομα. Ενώ η δεύτερη χρησιμοποιείται συνήθως όταν τα δεδομένα που αποστέλλονται πρέπει να αποθηκευτούν για μελλοντική χρήση ή όταν για λόγους ασφαλείας τα δεδομένα δεν πρέπει να είναι εμφανή. Επίσης, υπάρχει και η ιδιότητα ACTION, η οποία καθορίζει τη URL διεύθυνση του αρχείου που θα κληθεί για να λάβει τα δεδομένα που θα αποσταλούν.

Ετικέτα <input> Με την σήμανση <input> μπορούμε να ορίσουμε στοιχεία μιας φόρμας όπως text, boxes, check boxes, radio buttons, buttons. Το είδος του στοιχείου καθορίζεται από την ιδιότητα type της σήμανσης <input>. Η type καθορίζει το είδος των δεδομένων που θα πάρει η φόρμα από τον χρήστη. Αν δεν υπάρχει, τότε θεωρείται πως έχει την τιμή text που είναι η προκαθορισμένη. Η άλλη ιδιότητα που πρέπει να έχει η σήμανση <input> είναι η name που μας δίνει την δυνατότητα αναφοράς σε κάθε πεδίο. Η αναφορά μπορεί να θέλουμε να γίνει για έλεγχο των δεδομένων αλλά και από το script που τελικά θα λάβει τα δεδομένα για επεξεργασία ή αποθήκευση. Επίσης, η ιδιότητα maxlength καθορίζει το μέγιστο αριθμό χαρακτήρων που μπορούν να εισαχθούν σε ένα κείμενο. Η ιδιότητα αυτή υπάρχει μόνο για στοιχεία κειμένου και κωδικού.

Ετικέτα <select> Η ετικέτα <select> δημιουργεί ένα menu επιλογών από τις οποίες ο χρήστης μπορεί να επιλέξει μια που αντιστοιχεί σε μια τιμή. Το menu αυτό μπορεί να εμφανίζεται είτε οριζόντια είτε κάθετα. Αυτή θα σταλεί στο script μαζί με την τιμή της NAME. Οι επιλογές που υπάρχουν είναι για την δημιουργία απλών και drop-down list boxes αλλά και multi-select list boxes. Όλα αυτά καθορίζονται από τις ιδιότητες που λαμβάνει η σήμανση <select>. Πρέπει να σημειωθεί ότι η ιδιότητα name είναι απαραίτητη για το στοιχείο <select> έτσι ώστε να μπορεί να προσδιοριστεί. Τέλος, επειδή δεν μπορεί να είναι άδειο, πρέπει να περιέχει τουλάχιστον μια τιμή.

Ετικέτα <option> Ανάμεσα στις σημάνσεις <select> και </select> υπάρχει μια σειρά από σημάνσεις <option> οι οποίες αφορούν την κάθε επιλογή της λίστας. Κάθε επιλογή έχει και μια τιμή η οποία αποστέλλεται αν γίνει η τελική επιλογή του συγκεκριμένου στοιχείου. Το στοιχείο αυτό δεν μπορεί να είναι κενό αλλά μπορεί να περιέχει μόνο μια ετικέτα κειμένου.

Ετικέτα <fieldset> Με την ετικέτα αυτή μπορούμε να σχεδιάσουμε μια γραμμή τετραγώνου γύρω από ένα κείμενο ή άλλο περιεχόμενο που περιέχεται μέσα στο fieldset. Ουσιαστικά ομαδοποιεί σε μια λογική ενότητα όλα τα στοιχεία τα οποία εμπεριέχει.

Ετικέτα <label> Η ετικέτα <label> ορίζει ένα κείμενο για ένα μηχανισμό μιας φόρμας (π.χ. ένα κουμπί επιλογής). Αν πατήσουμε μέσα σε αυτό το κείμενο θα ενεργοποιηθεί αυτός ο μηχανισμός. Η ιδιότητα for της ετικέτας αναφέρεται στο id του μηχανισμού.

4.3 Μορφοποίηση σε css

Η σύνταξη των css αποτελείται από τρία μέρη: έναν επιλογέα (selector), μια ιδιότητα (property), και μια τιμή (value):

```
Επιλογέας {ιδιότητα: τιμή}  
selector {property: value}
```

Ο επιλογέας είναι συνήθως το στοιχείο-σήμανση που θέλουμε να ορίσουμε. Η σκοπιά του μπορεί να είναι μεγάλου εύρους, επηρεάζοντας την μορφή κάθε στοιχείου ή ακόμα και πολύ μικρή και συγκεκριμένη, επηρεάζοντας για παράδειγμα ένα μόνο στοιχείο.

Η ιδιότητα είναι το χαρακτηριστικό που θέλουμε να αλλάξουμε όπως το χρώμα, το πλάτος, το ύψος η γραμματοσειρά ή ακόμα και η θέση όπου πρόκειται να εμφανιστεί το στοιχείο.

Η τιμή της ιδιότητας προσδίδει το συγκεκριμένο στυλ το οποίο πρόκειται να εφαρμοστεί στο επιλεγμένο στοιχείο. Οι τιμές οι οποίες είναι αποδεκτές εξαρτώνται από την κάθε ιδιότητα και μερικές ιδιότητες δέχονται πολλαπλές τιμές οι οποίες χωρίζονται με κόμμα. Η ιδιότητα και η τιμή ξεχωρίζουν από τον χαρακτήρα άνω και κάτω τελεία : και περικλείονται από τους χαρακτήρες { }, ενώ η δήλωση τους τελειώνει με ελληνικό ερωτηματικό. Το ερωτηματικό αυτό είναι σημαντικό για τον διαχωρισμό κάθε δήλωσης άλλα αν υπάρχει μόνο μια δήλωση

στον κανόνα ή στην περίπτωση της τελευταίας δήλωσης τότε το ερωτηματικό είναι προαιρετικό. Καλό είναι πάντως αυτά να τοποθετούνται κάθε φορά.

4.3.1 Επιλογέας κλάσης (class selector)

Η λέξη τάξη σημαίνει μια κατηγορία του στυλ που γράφετε στο CSS που πρέπει να εφαρμόζονται σε κάθε στοιχείο στο έγγραφο της σήμανσης, ανεξάρτητα από το είδος του. Με άλλα λόγια, ο επιλογέας τάξη είναι ένας τρόπος για να εφαρμόσει τις ιδιότητες στυλ για διάφορα στοιχεία σήμανσης που θα χρησιμοποιεί το όνομα της κλάσης ως τιμή για το χαρακτηριστικό κλάσης. Έτσι μπορείτε να έχετε μια κατηγορία που ονομάζεται επιλογέας FontClass ότι θέτει ορισμένες ιδιότητες γραμματοσειράς και να το χρησιμοποιούν με ορισμένα στοιχεία HTML H4, p, όλη και ούτω καθ' εξής, μέσω της χρήσης του χαρακτηριστικού κλάσης του στοιχείου του.

Για να δημιουργήσετε ένα επιλογέα κατηγορίας, θα πρέπει να δώσετε ένα όνομα και να προηγούνται με μια τελεία (.), Και στη συνέχεια χρησιμοποιήστε το όνομα αυτό από τον κώδικα HTML με το χαρακτηριστικό κλάσης για κάθε στοιχείο. Για παράδειγμα ο παρακάτω κώδικας:

```
.info {font-size: 12px;  
      line-height: 2.2;}
```

θα μορφοποιήσει κάθε στοιχείο που ανήκει στην κλάση info.

4.3.2 Επιλογέας Ταυτότητας (ID Selector)

Ένας id selector επιλέγει μόνο το στοιχείο που φέρει το αντίστοιχο id. Συγκεκριμένα κάθε στοιχείο μπορεί να έχει κάποιο id αλλά μόνο ένα id του ίδιου ονόματος μπορεί να υπάρχει σε όλο το έγγραφο. Ο id selector στοχεύει σε ένα μόνο στοιχείο σε κάθε σελίδα και αυτό τον κάνει πολύ πιο συγκεκριμένο από τον επιλογέα τάξης. Για την δημιουργία ενός id selector, χρησιμοποιείται το σήμα hash (#) αντί για την τελεία(.) και στην συνέχεια να οριστεί η τιμή του χαρακτηριστικού HTML ταυτότητας στο όνομα της επιλογής id. Για παράδειγμα ο παρακάτω κώδικας θα μορφοποιήσει το στοιχείο με id userfile και θα του δώσει τις κατάλληλες επιλογές.

```
#userfile {  
font-size: 12px;
```


<<Πτυχιακή εργασία της φοιτήτριας Σκευούλη Θεοδώρας>>

```
border: 1px solid #CCC;
background-color: #F6F6F6;
width: 150px;
height: 25px;
margin: 0px 0px 10px 5px;
color: #666;}
```

4.3.3 Ομαδοποίηση επιλογών(Grouping Selectors)

Εδώ υπάρχει η δυνατότητα ομαδοποίησης επιλογών σε έναν κανόνα έτσι ώστε το ίδιο σύνολο δηλώσεων να μπορεί να εφαρμοστεί σε πλήθος στοιχείων χωρίς να απαιτείται η επανάληψη συγγραφής του ίδιου κανόνα. Για παράδειγμα ο παρακάτω κανόνας:

```
html, p, form { color:blue; }
```

θα εφαρμόσει το ίδιο χρώμα, δηλαδή μπλε, για τα στοιχεία html,p και form.

4.4 Υλοποίηση σε php

Ο κώδικας php στα παραδείγματα μας και γενικά αρχίζει με <? και τελειώνει με ?>. Αυτό είναι παρόμοιο με τις ετικέτες HTML, επειδή όλες ξεκινάνε με ένα σύμβολο μικρότερο "<" και τελειώνουν με το σύμβολο μεγαλύτερο ">". Αυτά τα σύμβολα ονομάζονται PHP ετικέτες και λένε στον web διακομιστή που ξεκινά ο κώδικας PHP και που τελειώνει. Το κενό μεταξύ των ετικετών θα μεταφραστεί σαν PHP. Το κείμενο έξω από αυτές τις ετικέτες θα αντιμετωπιστεί σαν κανονική HTML. Οι ετικέτες PHP μας επιτρέπουν να διαφεύγουμε από την HTML.

4.4.1 Εντολή echo

Η χρήση της εντολής echo είναι ότι τυπώνει την συμβολοσειρά που έχει περάσει σε αυτήν, στον browser. π.χ. η εντολή echo "<p> Η χρήση της PHP."; έχει ως αποτέλεσμα το κείμενο «Η χρήση της PHP.» να εμφανίζεται στο παράθυρο του browser. Το ερωτηματικό που εμφανίζεται στο τέλος της εντολής echo χρησιμοποιείται για να ξεχωρίζει τις εντολές PHP, όπως μια τελεία χρησιμοποιείται για να ξεχωρίζει τις προτάσεις της γλώσσας. Γι αυτό είναι απαραίτητη η χρήση του και είναι συνηθισμένο συντακτικό λάθος η παράλειψή του.

4.4.2 Μεταβλητές και σταθερές

Η μεταβλητή και η σταθερά που συνενώνουμε σε κάθε μια από τις εντολές echo, είναι διαφορετικοί τύποι πραγμάτων. Οι μεταβλητές είναι σύμβολο για δεδομένα. Η συμβολοσειρά είναι η ίδια η δεδομένα. Όταν χρησιμοποιούμε απλά δεδομένα σε ένα πρόγραμμα, το ονομάζουμε σταθερά για να το ξεχωρίζουμε από μια μεταβλητή.

4.4.3 Τύποι μεταβλητών

Για να δώσουμε τιμή σε μια μεταβλητή αρκεί να χρησιμοποιήσουμε τον τελεστή εκχώρησης ίσων (=). Η PHP υποστηρίζει τους παρακάτω τύπους δεδομένων:

- Integer(Ακέραιος)
- Double(Διπλής Ακρίβειας)
- String(Συμβολοσειράς)
- Array(Πίνακα)
- Object(Αντικείμενα)

4.4.4 Τελεστές Σύγκρισης

Οι τελεστές σύγκρισης χρησιμοποιούνται για να συγκρίνουμε δυο τιμές. Οι προτάσεις που χρησιμοποιούν αυτούς τους τελεστές επιστρέφουν λογικές τιμές true-false, ανάλογα με το αποτέλεσμα σύγκρισης. Οι τελεστές σύγκρισης είναι:

- == (Ισότητα)
- === (Ταυτότητα)
- != (Ανισότητα)
- <> (Ανισότητα)
- < (Μικρότερο)
- > (Μεγαλύτερο)
- <= (Μεγαλύτερο ή ίσο)
- >= (Μικρότερο ή ίσο)

4.4.5 Εντολές if

Μπορούμε να χρησιμοποιήσουμε μια εντολή if για να πάρουμε μια απόφαση. Θα πρέπει να δώσουμε στην εντολή if μια συνθήκη για να την χρησιμοποιήσουμε. Αν η εντολή είναι true θα εκτελεστεί το επόμενο τμήμα του κώδικα. Οι συνθήκες με εντολές if πρέπει να περιβάλλονται με παρενθέσεις. Συνήθως έχουμε περισσότερες από μια εντολές που θέλουμε να εκτελέσουμε μέσα σε μια εντολή if. Έτσι μπορούμε να ομαδοποιήσουμε πολλές εντολές if σαν ένα block.

4.4.6 Εντολές elseif

Για πολλές από τις αποφάσεις που παίρνουμε, υπάρχουν περισσότερες από δυο επιλογές. Μπορούμε να δημιουργήσουμε μια σειρά από πολλές επιλογές χρησιμοποιώντας την εντολή elseif. Η εντολή αυτή είναι συνδυασμός της else και if.

4.4.7 Βρόχοι while και for

Το απλούστερο είδος βρόχων στην PHP είναι ο βρόχος while ο οποίος βασίζεται και αυτός σε μια συνθήκη. Ο βρόχος while εκτελεί το τμήμα επαναλαμβανόμενα, εφόσον η συνθήκη είναι αληθής. Γενικά τον χρησιμοποιούμε όταν δεν ξέρουμε πόσες επαναλήψεις θα απαιτηθούν για να γίνει η συνθήκη αληθής. Την ίδια χρήση έχει και ο βρόχος for με την διαφορά ότι εδώ ξέρουμε τον αριθμό των επαναλήψεων.

4.4.8 Πίνακες

Ένας πίνακας είναι μια μεταβλητή που περιέχει ένα σύνολο ή σειρά από τιμές. Ένας πίνακας μπορεί να περιέχει πολλά στοιχεία. Κάθε στοιχείο μπορεί να περιέχει μια τιμή, όπως κείμενο, αριθμούς ή άλλο πίνακα. Η PHP υποστηρίζει πίνακες με αριθμητικούς δείκτες και συσχετιζόμενους πίνακες. Αφού έχουμε τις πληροφορίες σε ένα πίνακα μπορούμε να κάνουμε διάφορα πράγματα. Χρησιμοποιώντας τους βρόχους, μπορούμε να εξοικονομήσουμε τόπο εκτελώντας τις ίδιες ενέργειες σε κάθε τιμή του πίνακα. Οι πίνακες στις περισσότερες γλώσσες προγραμματισμού έχουν αριθμητικούς δείκτες. Η PHP υποστηρίζει αυτούς τους τύπους πινάκων.

4.4.9 Συναρτήσεις στην Php

Χρησιμοποιούνται για να ξεχωρίσουν κώδικα που εκτελεί μια καλά ορισμένη διαδικασία. Αυτό κάνει ευκολότερο τον κώδικα ευκολότερο να διαβαστεί και μας επιτρέπει να ξαναχρησιμοποιήσουμε τον κώδικα κάθε φορά που θέλουμε να κάνουμε την ίδια διαδικασία. Μια συνάρτηση είναι ανεξάρτητη λειτουργική μονάδα κώδικα που περιγράφει το περιβάλλον που κάνει την κλήση, εκτελεί κάποια διαδικασία και προαιρετικά επιστρέφει ένα αποτέλεσμα. Για να κληθεί μια συνάρτηση αρκεί να γράψουμε το όνομά της συνοδευόμενη από παρενθέσεις. Δηλαδή `function name()`; Αυτή είναι η πιο απλή κλήση μιας συνάρτησης. Υπάρχουν συναρτήσεις οι οποίες απαιτούν μια ή περισσότερες παραμέτρους. Δηλαδή πληροφορίες που δίνονται σε μια συνάρτηση όταν καλείται και που επηρεάζουν το αποτέλεσμα της εκτέλεσης της συνάρτησης.

4.4.10 Συνάρτηση trim()

Το πρώτο βήμα για την μορφοποίηση είναι η αποκοπή των επιπλέον από μια συμβολοσειρά. Αν και αυτό δεν απαιτείται ποτέ, μπορεί να είναι χρήσιμο αν πρόκειται να αποθηκεύσουμε την συμβολοσειρά σε ένα αρχείο ή βάση δεδομένων ή αν πρόκειται να τις συγκρίνουμε με άλλες συμβολοσειρές.

Η συνάρτηση `trim()` κόβει τον κενό χώρο από την αρχή και το τέλος της συμβολοσειράς και επιστρέφει την τελική συμβολοσειρά. Οι χαρακτήρες που αποκόπτεται είναι οι αλλαγές γραμμών και οι χαρακτήρες επιστροφής, οι οριζόντιοι και οι κατακόρυφοι στηλοθέτες, οι χαρακτήρες τέλους συμβολοσειράς και τα κενά.

4.4.11 Συνάρτηση Strlen()

Με την συνάρτηση `Strlen()` μπορούμε να ελέγξουμε το μέγεθος μιας συμβολοσειράς. Αν περάσουμε μια συμβολοσειρά, αυτή η συνάρτηση θα επιστρέψει το μήκος της. Για παράδειγμα `strlen("hello")` επιστρέφει 5.

4.4.12 Εντολές require() και include()

Η `php` έχει δυο πολύ απλές εντολές, για να μπορούμε να ξαναχρησιμοποιήσουμε οποιοδήποτε είδος κώδικα. Χρησιμοποιώντας μια εντολή `require()` ή `include()` μπορούμε να φορτώσουμε ένα αρχείο στο `php script` μας. Το αρχείο μπορεί να περιέχει οτιδήποτε που θα πληκτρολογήσαμε σε ένα `script`,

συμπεριλαμβανόμενων PHP εντολών, κειμένου,HTML ετικετών,PHP συναρτήσεων.

4.4.13 Μεταβλητές \$_get και \$_post

Το ενσωματωμένο \$_get χρησιμοποιείται για την συλλογή τιμών από ένα έντυπο που στάλθηκε με τη μέθοδο ="get". Πληροφορίες που στέλνονται με την μέθοδο get είναι ορατές από όλους(αυτό θα εμφανίζεται στην γραμμή διευθύνσεων του προγράμματος περιήγησης) και έχει όρια για τον όγκο πληροφοριών για την αποστολή της (μέχρι 100 χαρακτήρες). Πρέπει να σημειωθεί ότι η μέθοδος αυτή δεν πρέπει να χρησιμοποιείται κατά την αποστολή κωδικών πρόσβασης ή για άλλες ευαίσθητες πληροφορίες. Η php παίρνει από την διεύθυνση ότι υπάρχει μετά το ? και το τοποθετεί σε ένα πίνακα που είναι μια υπερκαθολική μεταβλητή(έτσι ονομάζεται), η \$_get. Είναι πολύ χρήσιμο κατά το πέρασμα url με php να χρησιμοποιούμε μια μεταβλητή για να πάρουμε την τοποθεσία url όπου βρισκόμαστε. Για παράδειγμα ο κώδικας \$id=\$_get['id']; θα πάρει την μεταβλητή \$id από την url και θα την αποθηκεύσει σε μια τοπική μεταβλητή \$id. Με αυτόν τον τρόπο μπορούν να αποσταθούν τιμές ανάμεσα στις ιστοσελίδες.

Η χρήση της \$_post είναι πιο απλή και πιο ασφαλής από της \$_get. Οι πληροφορίες που στέλνονται με την μεταβλητή αυτή είναι αόρατες από όλους και δεν έχει όρια για τον όγκο πληροφοριών για την αποστολή. Για παράδειγμα ο κώδικας \$name=\$_post['username']; θα αποθηκεύσει στην μεταβλητή \$name την τιμή που εισήχθηκε το πεδίο της φόρμας με id ορισμένο με "username".

4.4.14.Μεταβλητή \$_session

Όταν εργαζόμαστε με μια εφαρμογή μπορούμε να την ανοίξουμε, να κάνουμε κάποιες αλλαγές και στην συνέχεια να την κλείσουμε. Ο υπολογιστής γνωρίζει τότε θα ξεκινήσει η εφαρμογή και τότε θα τελειώσει. Αλλά στο διαδίκτυο υπάρχει ένα πρόβλημα, ο web server δεν γνωρίζει ποιο είμαστε και τι κάνουμε επειδή η διεύθυνση http δεν έχει κατάσταση. Το πρόβλημα αυτό το λύνει η session. Στην ουσία είναι ένα αρχείο το οποίο αποθηκεύεται στο web server. Μπορούμε να βρούμε ποιο αρχείο ανήκει σε κάθε χρήστη κάνοντας χρήση των cookies. Θέτουμε ένα ειδικό session cookie στον browser και μετά κοιτάμε σε αυτό το cookie για να βρούμε το μέρος όπου βρίσκεται το αρχείο στον server μας

κι έπειτα να ψάξουμε στο αρχείο αυτό για τις πληροφορίες που θέλουμε. Οι πληροφορίες αυτές δεν μπορούν να αλλαχθούν αλλά το μόνο που μπορούμε να κάνουμε είναι να αλλάξουμε το id number το οποίο είναι πολύ μεγάλο για λόγους ασφαλείας. Για να χρησιμοποιήσουμε μια session πρώτα δημιουργούμε ένα τέτοιο αρχείο και να θέσουμε το cookie στον υπολογιστή του χρήστη ή αν αυτό έχει ήδη γίνει να βρούμε αυτό το cookie και να βρούμε το αντίστοιχο αρχείο σε αυτόν τον υπολογιστή. Έτσι πρέπει να αρχίσουμε μια session_start(). Αυτή πρέπει να γίνει στην αρχή του αρχείου πριν εκτυπώσουμε οποιαδήποτε HTML ή ακόμα και κενό και πριν στείλουμε οποιαδήποτε πληροφορία στον browser του χρήστη. Έτσι η php βρίσκει από τον browser το id number του session cookie, βρίσκει το αρχείο και το ανοίγει ώστε να μπορεί να χρησιμοποιηθεί. Αν δεν το βρει το id number ή το session file τότε αρχίζει μια καινούρια session και διαμορφώνει το κατάλληλο cookie. Ας δούμε ένα παράδειγμα που χρησιμοποιήθηκε στην εφαρμογή

```
session_start();
function logged_in() {
    return isset($_SESSION['auth']);
}
function confirm_logged_in() {
    if (!logged_in()) {
        redirect_to('index.php');
    }
}
```

Το πρώτο πράγμα που γίνεται όταν ο χρήστης περιηγείται στις σελίδες είναι να γίνει κλήση της συνάρτησης confirm_logged_in(), η οποία ελέγχει αν το email του χρήστη βρίσκεται στον πίνακα \$_session. Το email του κάθε χρήστη είναι μοναδικό και αποθηκεύεται στον πίνακα αυτό κατά την είσοδό του. Έτσι, αν ο χρήστης είναι συνδεδεμένος τότε συνεχίζει την περιήγηση του στην εφαρμογή διαφορετικά καλείται η συνάρτηση redirect_to('index.php') όπου τον μεταφέρει στην αρχική σελίδα.

4.4.15 Σύνδεση με την Βάση Δεδομένων

Πριν κάνουμε οποιαδήποτε εργασία με την βάση δεδομένων θα πρέπει πρώτα να υπάρξει σύνδεση με τον διακομιστή. Για τον λόγο αυτό γίνεται χρήση

της `mysql_connect()` η οποία δεν απαιτεί κανένα όρισμα άλλα δέχεται τρία string και είναι το `hostname`, `username` και `password`. Αν δεν δοθεί κανένα από τα αυτά τα ορίσματα τότε υποτίθεται ότι το `host` είναι `localhost` και ότι το `username` και το `password` δεν έχουν οριστεί στο `mysqluser` πίνακα. Όταν γίνει η σύνδεση με τον διακομιστή της βάσης θα πρέπει να επιλεγεί η βάση με την οποία θα λειτουργεί η εφαρμογή. Η δυνατότητα αυτή γίνεται από την συνάρτηση `mysql_select_db()`. Η συνάρτηση αυτή απαιτεί το όνομα μιας βάσης δεδομένων και προαιρετικά έναν αναγνώστη συνδέσμου. Τέλος, αυτή επιστρέφει `true` αν υπάρχει και έτσι μπορεί να γίνει χειρισμός της, διαφορετικά επιστρέφει `false`. Όλα τα παραπάνω φαίνονται στον επόμενο κώδικα:

```
function db_connection() {
    $connection = mysqli_connect(SERVER, USER, MY_PASSWORD,
MY_DATABASE);
    $connection->query("SET NAMES 'utf8'");
    if (!connection) {
        die("Database connection failed: " .
mysqli_error());
    }
    return $connection;}
function db_selection() {
    $db_select = mysqli_select_db(MY_DATABASE,
db_connection());
    if (!$db_select) {
        die("Database selection failed: " .
mysqli_error());
    }
    echo "OK";
    return $db_select;
```

4.4.16 Συναρτήσεις της εφαρμογής

Για την δημιουργία λογαριασμού κάποιου χρήστη στην εφαρμογή χρησιμοποιείται η συνάρτηση `create_account()`. Η συνάρτηση αυτή κάνει τους διάφορους ελέγχους για να ελέγξει αν τα στοιχεία που πληκτρολογεί ο χρήστης είναι έγκυρα. Ο παρακάτω κώδικας ελέγχει τον αρχικό κωδικό με την επαλήθευση κωδικού που δίνει ο χρήστης για να δει αν αυτά συμφωνούν μεταξύ τους. Αν αυτά τα δυο πεδία είναι διαφορετικά τότε εμφανίζεται κατάλληλο μήνυμα λάθους.

<<Πτυχιακή εργασία της φοιτήτριας Σκευούλη Θεοδώρας>>

```
elseif ($password != $retryRegisterPassword) {
    $out .= '<p class="warning">Ο κωδικός που
πληκτρολογήσατε δε συμφωνεί με το πεδίο επαλήθευσης
κωδικού.</p>'; }
```

Για την δημιουργία λογαριασμού ο χρήστης πρέπει να εισάγει το κατάλληλο email. Ο παρακάτω κώδικας ελέγχει την εγκυρότητα του. Ουσιαστικά ελέγχει αν στο πεδίο που πληκτρολογεί ο χρήστης το email υπάρχει ο χαρακτήρας “@”. Αν δεν υπάρχει τότε εμφανίζεται μήνυμα λάθους.

```
elseif (!preg_match('/@/', $_POST['registerEmail'])) {
    $out .= '<p class="warning">Παρακαλώ εισάγετε ένα
έγκυρο email.</p>'; }
```

Επίσης ελέγχεται αν το email που δόθηκε χρησιμοποιείται ήδη. Αυτό επιτυγχάνεται ελέγχοντας την βάση δεδομένων εκτελώντας το παρακάτω κομμάτι κώδικα.

```
$query = "SELECT * FROM Students WHERE email =
'$email'";
$result = q($connection, $query);
$num_rows_email = mysqli_num_rows($result);
```

Η επόμενη γραμμή κώδικα μας δείχνει τον έλεγχο που γίνεται όταν ο χρήστης κάνει login στην σελίδα. Αν δώσει λάθος στοιχεία για τρίτη ή παραπάνω φορά τότε ο λογαριασμός του κλειδώνεται και πρέπει να τον ξεκλειδώσει αλλιώς δεν μπορεί να έχει πρόσβαση στην σελίδα.

```
if (wrong_password($connection, $username, $password)
>= 3)
```

Για να ανεβάσει ο χρήστης μια άσκηση χρησιμοποιείται η μέθοδος upload_assignment(). Ένας από τους ελέγχους που κάνει αυτή η μέθοδος είναι να δει αν ο χρήστης έχει επιλέξει κάποιο αρχείο για αποστολή. Αυτό γίνεται με την βοήθεια της συνάρτησης empty() η οποία ελέγχει αν έχει δοθεί συγκεκριμένη τιμή στο πεδίο της φόρμας. Η μέθοδος αυτή επιστρέφει true ή false. Σε περίπτωση που δεν έχει επιλέξει κάποιο αρχείο επιστρέφεται μήνυμα λάθους και ο κώδικας φαίνεται παρακάτω.

<<Πτυχιακή εργασία της φοιτήτριας Σκευούλη Θεοδώρας>>

```
if (empty($_FILES['userfile']['name'])) {  
    $out = '<p class="warning">Παρακαλώ επιλέξτε κάποιο  
αρχείο. </p>';  
    return $out;}  

```

Η συνάρτηση που χρησιμοποιείται για την αντικατάσταση μιας άσκησης είναι η `change_assignment()`. Εδώ ο χρήστης καλείται να επιλέξει ποια εργασία θέλει να αντικαταστήσει. Έτσι στον παρακάτω κώδικα γίνεται έλεγχος αν ο χρήστης έχει επιλέξει την εβδομάδα για την οποία προορίζεται η άσκηση που θέλει να αντικαταστήσει. Διαφορετικά εμφανίζεται μήνυμα λάθους.

```
if (empty($_POST['sentAssignment'])) {  
    $out = '<p class="warning">Παρακαλώ επιλέξτε την  
εργασία για την οποία προορίζετε το αρχείο.</p>';  
    return $out;}  

```

Στο παρακάτω τμήμα κώδικα ελέγχεται ποιος χρήστης κάνει είσοδο στην σελίδα. Αυτό φαίνεται από την τιμή που έχει το πεδίο `type`. Αν η τιμή του είναι 1 τότε σημαίνει ότι είναι τύπου `Student` ενώ αν είναι 2 σημαίνει ότι είναι τύπου `Professor`. Ανάλογα με το ποιος χρήστης κάνει είσοδο εμφανίζεται και η κατάλληλη κύρια οθόνη του.

```
if (isStudent($connection)) {  
    if ($_SESSION['type'] == 1) {  
        redirect_to("home.php?mod=viewgrades");  
    } elseif ($_SESSION['type'] == 2) {  
        redirect_to("home.php?mod=sentassign");  
    }  
}
```

Στο παρακάτω παράδειγμα μας ο χρήστης εισόδου είναι `Student` και μας δείχνει ποιες φόρμες μπορεί να προσπελάσει αυτός.

```
if ($_SESSION['type'] == 1) {  
    switch ($mod) {  
        case "sentassign" :  
            $out .= do_sentassign($connection);  
            break;  

```

```
case "changeassign" :
    $out .= do_changeassign($connection);
    break;
case "viewgrades" :
    $out .= do_viewgrades($connection);
    break;
case "downloadmat" :
    $out .= do_downloadmat($connection);
    break;
default :
    $out .= do_viewgrades($connection);
    break;}}
```

Το παρακάτω τμήμα κώδικα αναφέρεται στην εμφάνιση των βαθμών. Από τον πίνακα studentsAssings εμφανίζονται όλες οι ασκήσεις που έχουν βαθμολογηθεί για τον συγκεκριμένο φοιτητή.

```
$sId = $_SESSION['id'];
$query = "SELECT * FROM studentAssings WHERE sId = '$sId' AND
grade IS NOT NULL;";
```

Τέλος το παρακάτω κομμάτι κώδικα αναφέρεται στο ανέβασμα υλικού του καθηγητή. Αν δεν υπάρξει κανένα λάθος ο πίνακας λαθών θα έχει μήκος ίσο με 0 οπότε ο κώδικας θα συνεχίζει να εκτελείται. Διαφορετικά αν υπάρξει κάποιο από τα λάθη που παρουσιάζονται τότε θα εμφανιστεί το κατάλληλο μήνυμα.

```
if ($_FILES['userfile']['error'] > 0) {
    switch ($_FILES['userfile']['error']) {
        case 1 :
            $out = '<p class="warning">Το αρχείο
είναι πολύ μεγάλο για να το ανεβάσετε.</p>';
            break;
        case 2 :
            $out = '<p class="warning">Το αρχείο
είναι πολύ μεγάλο για να το ανεβάσετε.</p>';
            break;
        case 3 :
            $out = '<p class="warning">Αδυναμία
ανάρτησης ολόκληρου του αρχείου.</p>';
            break;
        case 4 :
            $out = '<p class="warning">Αδυναμία
ανάρτησης του αρχείου.</p>';
            break;
```

```
}  
return $out;
```

4.5 Βάσεις Δεδομένων

Η θεωρία της κανονικοποίησης αναπτύχθηκε με στόχο να ανιχνεύεται και να προλαμβάνεται προβληματικές καταστάσεις όταν η εφαρμογή είναι ακόμα στο στάδιο του σχεδιασμού της. Περιλαμβάνει τέσσερις μορφές τις οποίες θα περιγράψουμε με λίγα λόγια παρακάτω. Η πρώτη είναι η μη κανονική μορφή στην οποία δεν υπάρχει πίνακας με την σχεσιακή έννοια. Τα πεδία τοποθετούνται το ένα δίπλα στο άλλο με μόνο επιπλέον στοιχείο πληροφορίας την ομαδοποίηση πεδίων που αντιστοιχούν σε χαρακτηριστικά που παίρνουν πολλαπλές τιμές. Η ομαδοποίηση των τελευταίων αναπαριστάται με ζεύγη εσωτερικών παρενθέσεων στην περιγραφή της δομής του αντίστοιχου πίνακα. Η δεύτερη μορφή ονομάζεται Πρώτη Κανονική Μορφή(1NF) και για να έρθει ένας πίνακας σε αυτή τη μορφή πρέπει να απομακρυνθούν οι επαναλαμβανόμενες ομάδες πεδίων, έτσι ώστε η τομή μιας γραμμής και μιας στήλης του πίνακα να αντιστοιχεί πάντα σε μια απλή τιμή. Πρέπει η τιμή του κάθε πεδίου σε κάθε εγγραφή, να είναι ατομική, δηλαδή να μην μπορεί να διασπαστεί σε περισσότερες πληροφορίες. Η τρίτη μορφή ονομάζεται Δεύτερη Κανονική Μορφή(2NF) και για να έρθει ένας πίνακας σε αυτή τη μορφή, θα πρέπει να πρώτα να είναι σε πρώτη κανονική μορφή και στην συνέχεια να απομακρυνθούν όλες οι μερικές συναρτησιακές εξαρτήσεις που υφίστανται ανάμεσα στα πεδία του. Με άλλα λόγια, όταν όλα τα πεδία που δεν ανήκουν στο πρωτεύον κλειδί του πίνακα εξαρτώνται συναρτησιακά από μόνο από τα πεδία του πρωτεύοντος κλειδιού και μάλιστα, μέσω πλήρους συναρτησιακής εξάρτησης. Τέλος η τέταρτη μορφή ονομάζεται Τρίτη Κανονική Μορφή και για να φτάσουμε σε αυτήν θα πρέπει ο πίνακας να είναι σε δεύτερη κανονική μορφή και στην συνέχεια να απομακρυνθούν όλες οι μεταβατικές συναρτησιακές εξαρτήσεις που υφίστανται ανάμεσα στα πεδία του.

Παρακάτω γίνεται η ανάλυση του κάθε πίνακα της εφαρμογής και παρουσιάζεται η ολοκληρωμένη βάση δεδομένων στην Τρίτη κανονική μορφή της.

4.5.1 Σχεσιακό Μοντέλο

Το σχεσιακό μοντέλο αναπαράστασης δεδομένων μιας εφαρμογής είναι ένα γράφημα που απεικονίζει τις συσχετίσεις ανάμεσα στις οντότητες της βάσης. Σε αυτό το μοντέλο, τα δεδομένα μιας εφαρμογής, αναπαρίστανται ως ένα

σύνολο από σχέσεις οι οποίες μπορεί να είναι πίνακες ή αρχεία. Στις πιο πολλές περιπτώσεις είναι πίνακες που περιέχουν ένα πλήθος γραμμών και στηλών. Η κάθε μια από τις γραμμές ή πλειάδες όπως αλλιώς ονομάζονται περιέχει ένα σύνολο απλών πεδίων τα οποία συσχετίζονται μεταξύ τους. Κάθε πεδίο ή στήλη ενός πίνακα, δέχεται τιμές οι οποίες ανήκουν σε ένα συγκεκριμένο και καθορισμένο εκ των προτέρων σύνολο τιμών. Το είδος των τιμών αυτού το σύνολο καθορίζεται από το τύπο δεδομένων του πεδίου του πίνακα, ο οποίος με την σειρά του ορίζεται κατά το στάδιο της λογικής σχεδίασης της εφαρμογής.

Παρακάτω παρατίθεται το σχεσιακό μοντέλο της εφαρμογής.

Assignments (assignId, assignName, expireDate)

Στον πίνακα Assignments είναι οι ασκήσεις που στέλνουν οι φοιτητές.

Material (mId, fileName, subject, week, toView, comments, uploadDate)

Στον πίνακα Material καταχωρούνται τα υλικά που ανεβάζει ο καθηγητής.

Studentassigns (sId, assignId, fileName, grade, comments, uploadDate)

Στον πίνακα Studentassigns καταχωρούνται οι ασκήσεις που βαθμολογούν οι καθηγητές.

Student (sId, name, surname, username, password, email, sex, country, birthYear, type, tries, lastLogin)

Στον πίνακα Students αποθηκεύονται όλοι οι διαχειριστές της εφαρμογής οι οποίοι θα μπορούν να έχουν πρόσβαση στις λειτουργία της.

4.5.2 Τεκμηρίωση Πινάκων

Στην συγκεκριμένη εφαρμογή χρησιμοποιήθηκαν οι τέσσερις παραπάνω πίνακες και ας δούμε τώρα αυτοί πως φαίνονται στην βάση δεδομένων.

Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
<u>assignId</u>	int(11)			Όχι		auto_increment	[Icons]
<u>assignName</u>	varchar(100)	utf8_unicode_ci		Όχι			[Icons]
<u>expireDate</u>	datetime			Όχι			[Icons]

Σχήμα XXII: Πίνακας συστήματος assignments

<<Πτυχιακή εργασία της φοιτήτριας Σκευούλη Θεοδώρας>>

Διακομιστής: localhost ▶ Βάση: skeuouli ▶ Πίνακας : material "InnoDB free: 4096 kB"

Περιήγηση Δομή SQL Αναζήτηση Εισαγωγή Εξαγωγή Import Λειτουργίες Αδειασμα Διαγραφή

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
<input type="checkbox"/>	<u>mId</u>	int(11)			Όχι		auto_increment	
<input type="checkbox"/>	fileName	varchar(100)	utf8_unicode_ci		Όχι			
<input type="checkbox"/>	subject	varchar(50)	utf8_unicode_ci		Ναι	NULL		
<input type="checkbox"/>	week	int(11)			Όχι			
<input type="checkbox"/>	toView	set('y', 'n')	utf8_unicode_ci		Όχι	n		
<input type="checkbox"/>	comments	varchar(30)	utf8_unicode_ci		Όχι			
<input type="checkbox"/>	uploadDate	timestamp			Όχι	CURRENT_TIMESTAMP		

↑ Επιλογή όλων / Απεπιλογή όλων Με τους επιλεγμένους:

Σχήμα XXIII: Πίνακας συστήματος material

Διακομιστής: localhost ▶ Βάση: skeuouli ▶ Πίνακας : studentassigns "InnoDB free: 4096 kB"

Περιήγηση Δομή SQL Αναζήτηση Εισαγωγή Εξαγωγή Import Λειτουργίες Αδειασμα Διαγραφή

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
<input type="checkbox"/>	<u>sId</u>	int(11)			Όχι			
<input type="checkbox"/>	<u>assignId</u>	int(11)			Όχι			
<input type="checkbox"/>	fileName	varchar(200)	utf8_unicode_ci		Όχι			
<input type="checkbox"/>	<u>grade</u>	varchar(6)	utf8_unicode_ci		Ναι	NULL		
<input type="checkbox"/>	comments	varchar(20)	utf8_unicode_ci		Ναι	NULL		
<input type="checkbox"/>	uploadDate	timestamp			Όχι	CURRENT_TIMESTAMP		

















↑ Επιλογή όλων / Απεπιλογή όλων Με τους επιλεγμένους:

Σχήμα XXIV: Πίνακας συστήματος studentassigns

<<Πτυχιακή εργασία της φοιτήτριας Σκευούλη Θεοδώρας>>

Διακομιστής: localhost ▶ Βάση: skeuouli ▶ Πίνακας : students "InnoDB free: 4096 kB"

Περιήγηση Δομή SQL Αναζήτηση Εισαγωγή Εξαγωγή Import Λειτουργίες Άδεια Διαγραφή

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
<input type="checkbox"/>	sld	int(11)			Όχι		auto_increment	     
<input type="checkbox"/>	name	varchar(40)	utf8_unicode_ci		Όχι			     
<input type="checkbox"/>	surname	varchar(40)	utf8_unicode_ci		Όχι			     
<input type="checkbox"/>	username	varchar(20)	utf8_unicode_ci		Όχι			     
<input type="checkbox"/>	password	varchar(40)	utf8_unicode_ci		Όχι			     
<input type="checkbox"/>	email	varchar(40)	utf8_unicode_ci		Όχι			     
<input type="checkbox"/>	sex	set('m', 'f')	utf8_unicode_ci		Όχι			     
<input type="checkbox"/>	country	varchar(40)	utf8_unicode_ci		Όχι			     
<input type="checkbox"/>	birthYear	year(4)			Όχι			     
<input type="checkbox"/>	type	set('1', '2', '3')	utf8_unicode_ci		Όχι	1		     
<input type="checkbox"/>	tries	int(11)			Όχι	0		     
<input type="checkbox"/>	lastLogin	timestamp		ON UPDATE CURRENT_TIMESTAMP	Όχι	0000-00-00 00:00:00		     

Σχήμα XXV: Πίνακας συστήματος students

4.5.3 Περιγραφή πεδίων

Ακολουθεί η τεκμηρίωση των πινάκων με όλα τα πεδία τους.

Πίνακας:Assignments	Τύπος
<u>assgnId</u>	Ο κωδικός της κάθε άσκησης .
assignName	Το όνομα της άσκησης .
expireDate	Η ημερομηνία για την φραγή της αποστολής της άσκησης.

Πίνακας:Material	Τύπος
<u>mId</u>	Ο κωδικός του κάθε υλικού
fileName	Το όνομα του κάθε υλικού
subject	Το θέμα του κάθε υλικού
week	Η εβδομάδα του κάθε υλικού
tonview	Αν θα εμφανίζεται ή όχι το υλικό
comments	Τα σχόλια για κάθε υλικό
uploadDate	Η ημερομηνία ανεβάσματος του κάθε υλικού

Πίνακας : Studentassigns	Τύπος
<u>sld</u>	Ο κωδικός της άσκησης που στέλνει ο φοιτητής
<u>assignId</u>	Το id που συσχετίζεται με το id του πίνακα assignments
fileName	Το όνομα της κάθε άσκησης
grade	Ο βαθμός για την κάθε άσκηση του φοιτητή
comments	Τα σχόλια που θέλει να γράψει ο καθηγητής για κάθε άσκηση
uploadDate	Η ημερομηνία ανεβάσματος κάθε άσκησης

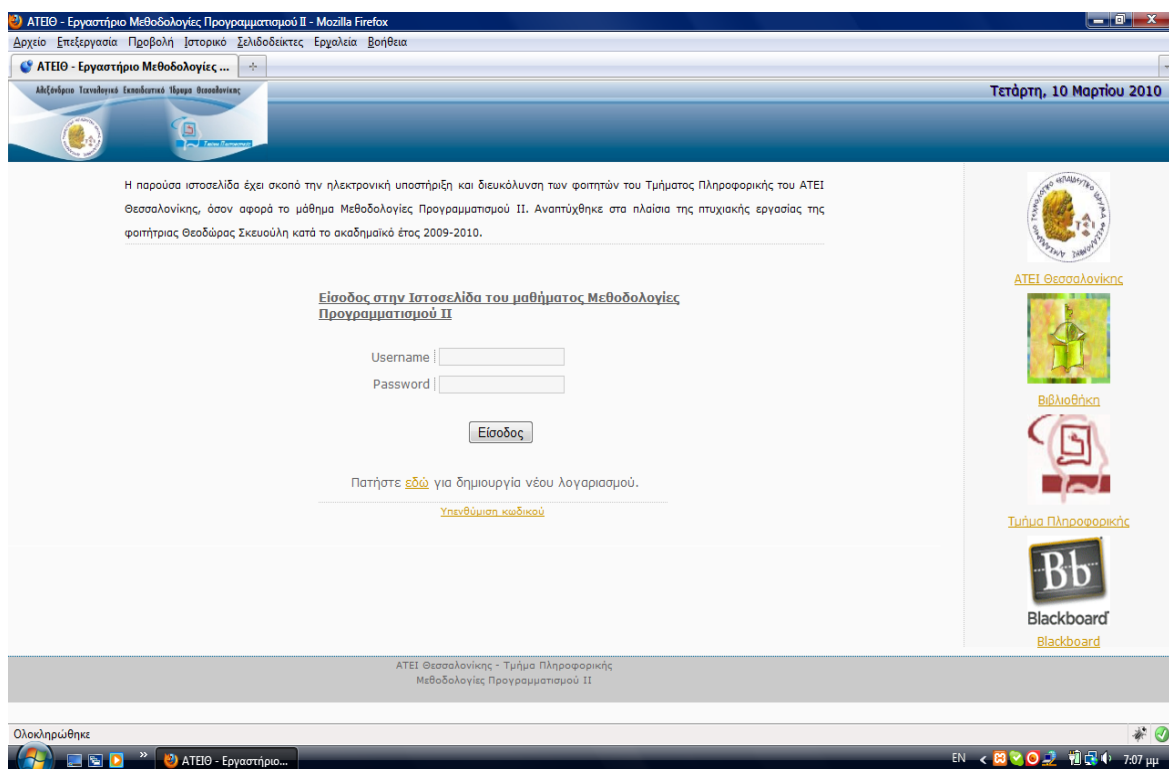
Πίνακας: Students	Τύπος
<u>sld</u>	Ο κωδικός του κάθε διαχειριστή
name	Το όνομα του κάθε διαχειριστή
surname	Το επίθετο του κάθε διαχειριστή
username	Το ψευδώνυμο του κάθε διαχειριστή
password	Ο κωδικός του κάθε διαχειριστή
email	Το email του κάθε διαχειριστή
sex	Το φύλλο του κάθε διαχειριστή
country	Η χώρα του κάθε διαχειριστή
birthYear	Η χρονολογία γέννησης του κάθε διαχειριστή
type	Ο τύπος του διαχειριστή
tries	Οι προσπάθειες εισαγωγής στο σύστημα για κάθε διαχειριστή
lastLogin	Η τελευταία σύνδεση του κάθε διαχειριστή

SCREENSHOTS ΕΦΑΡΜΟΓΗΣ

Στο κεφάλαιο αυτό παρουσιάζονται τα screenshots της εφαρμογής καθώς και οδηγίες σχετικά με την περιήγηση του χρήστη στην εφαρμογή.

5.1 Βασική οθόνη

Η οθόνη που εμφανίζεται όταν κάποιος εισέρχεται πρώτη φορά στο σύστημα είναι η παρακάτω:



5.2 Δημιουργία Χρήστη

Για να μπορέσει ο χρήστης να συμμετάσχει στις λειτουργίες που παρέχει η εφαρμογή θα πρέπει από την βασική οθόνη να πατήσει το κουμπί <<εδώ>> για να δημιουργήσει λογαριασμό. Η οθόνη που θα εμφανιστεί θα είναι η παρακάτω:

ATEIΘ - Εργαστήριο Μεθοδολογίας Προγραμματισμού II - Mozilla Firefox

Αρχείο Επεξεργασία Προβολή Ιστορικό Σελιδοδείκτες Εργαλεία Βοήθεια

ATEIΘ - Εργαστήριο Μεθοδολογίας...

Αθήνα Τεχνολογικό Εκπαιδευτικό Ίδρυμα Θεσσαλονίκης

Τετάρτη, 10 Μαρτίου 2010

Η παρούσα ιστοσελίδα έχει σκοπό την ηλεκτρονική υποστήριξη και διεκκάλυψη των φοιτητών του Τμήματος Πληροφορικής του ΑΤΕΙ Θεσσαλονίκης, όσον αφορά το μάθημα Μεθοδολογίας Προγραμματισμού II. Αναπτύχθηκε στα πλαίσια της πτυχιακής εργασίας της φοιτήτριας Θεοδώρας Σκευούλη κατά το ακαδημαϊκό έτος 2009-2010.

Εγγραφή στο μάθημα Μεθοδολογίας Προγραμματισμού II
[Επιστροφή στην αρχική σελίδα](#)

Username :

Password :

Retype Password :

Όνομα :

Επώνυμο :

Email :

Φύλο :

Χώρα :

Χρονολογία Γέννησης :

ATEI Θεσσαλονίκης - Τμήμα Πληροφορικής
Μεθοδολογίας Προγραμματισμού II

Ολοκληρώθηκε

ATEIΘ - Εργαστήριο...

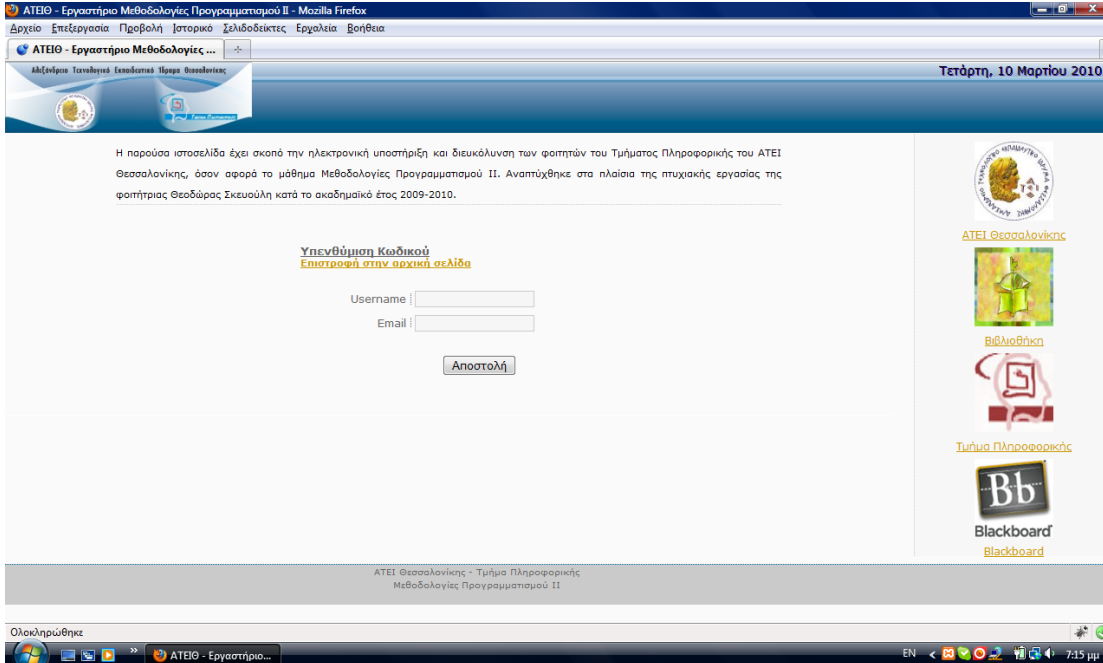
EN < > 7:40 μμ

Στην φόρμα αυτή ο χρήστης πρέπει να εισάγει τα στοιχεία του σε όλα τα πεδία και να πατήσει το κουμπί εγγραφή. Αν ο χρήστης εισάγει όλα του τα στοιχεία σωστά τότε ο χρήστης μπορεί να κάνει είσοδο στην σελίδα. Αντίθετα, αν ο χρήστης κάνει κάποιο λάθος στην αίτηση του τότε το σύστημα τον ενημερώνει με κατάλληλο μήνυμα και τον προτρέπει στο να διορθώσει τα σφαλμένα του πεδία. Όταν ο χρήστης ολοκληρώσει την υποβολή της αίτησης με επιτυχία, θα δεχθεί αυτόματα email στο email επικοινωνίας που δήλωσε.

<<Πτυχιακή εργασία της φοιτήτριας Σκευούλη Θεοδώρας>>

5.3 Υπενθύμιση κωδικού

Ο χρήστης έχει την δυνατότητα το σύστημα να του υπενθυμίσει τον κωδικό του σε περίπτωση που το έχει ξεχάσει. Πατώντας το κουμπί <<Υπενθύμιση κωδικού>> στην βασική οθόνη εμφανίζεται η παρακάτω φόρμα:



The screenshot shows a web browser window with the URL 'ATEIO - Εργαστήριο Μεθοδολογίας Προγραμματισμού II - Mozilla Firefox'. The page title is 'ATEIO - Εργαστήριο Μεθοδολογίας ...'. The date is 'Τετάρτη, 10 Μαρτίου 2010'. The main content area contains the following text: 'Η παρούσα ιστοσελίδα έχει σκοπό την ηλεκτρονική υποστήριξη και διευκόλυνση των φοιτητών του Τμήματος Πληροφορικής του ΑΤΕΙ Θεσσαλονίκης, όσον αφορά το μάθημα Μεθοδολογίας Προγραμματισμού II. Αναπτύχθηκε στα πλαίσια της πτυχιακής εργασίας της φοιτήτριας Θεοδώρας Σκευούλη κατά το ακαδημαϊκό έτος 2009-2010.' Below this text is a form titled 'Υπενθύμιση Κωδικού' with the subtitle 'Επιστροφή στην αρχική σελίδα'. The form has two input fields: 'Username' and 'Email', and a 'Αποστολή' button. On the right side of the page, there are several icons and links: 'ATEI Θεσσαλονίκης', 'Βιβλιοθήκη', 'Τμήμα Πληροφορικής', and 'Blackboard'.

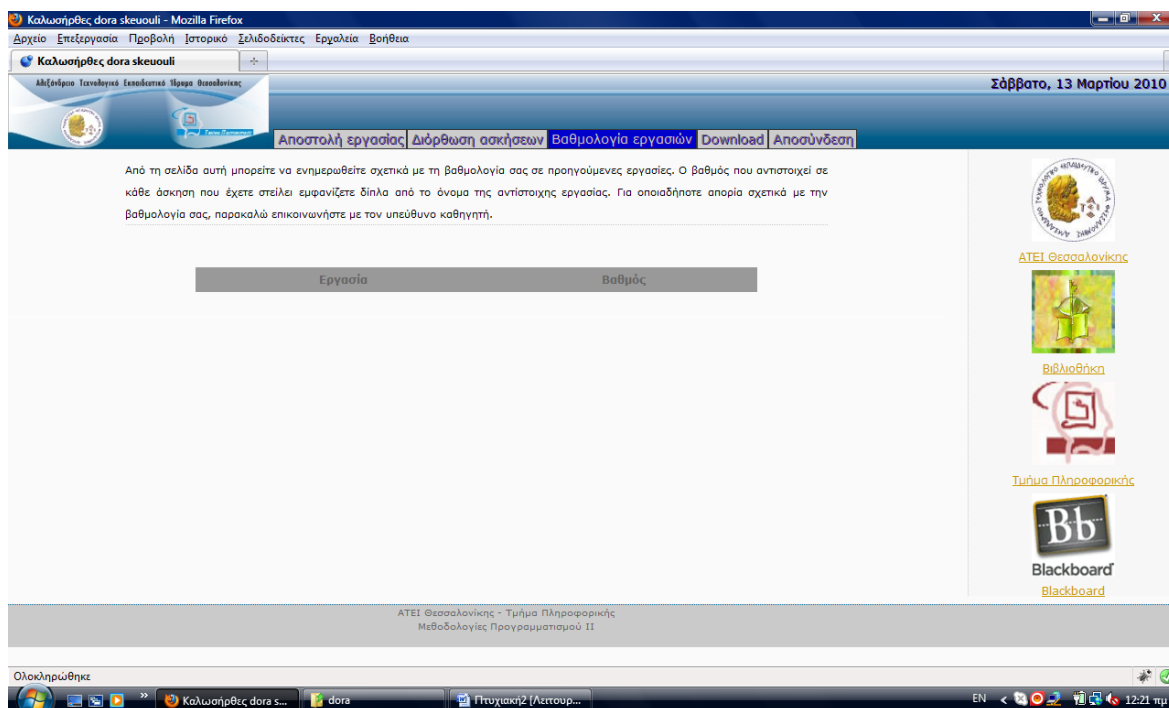
Έτσι εδώ απλά ο χρήστης εισάγει το username και το email του και το σύστημα του στέλνει το κωδικό του.

5.4 Δυνατότητες Φοιτητή

Οι δυνατότητες για τον χρήστη φοιτητή παρουσιάζονται παρακάτω.

5.4.1 Εμφάνιση Βαθμών

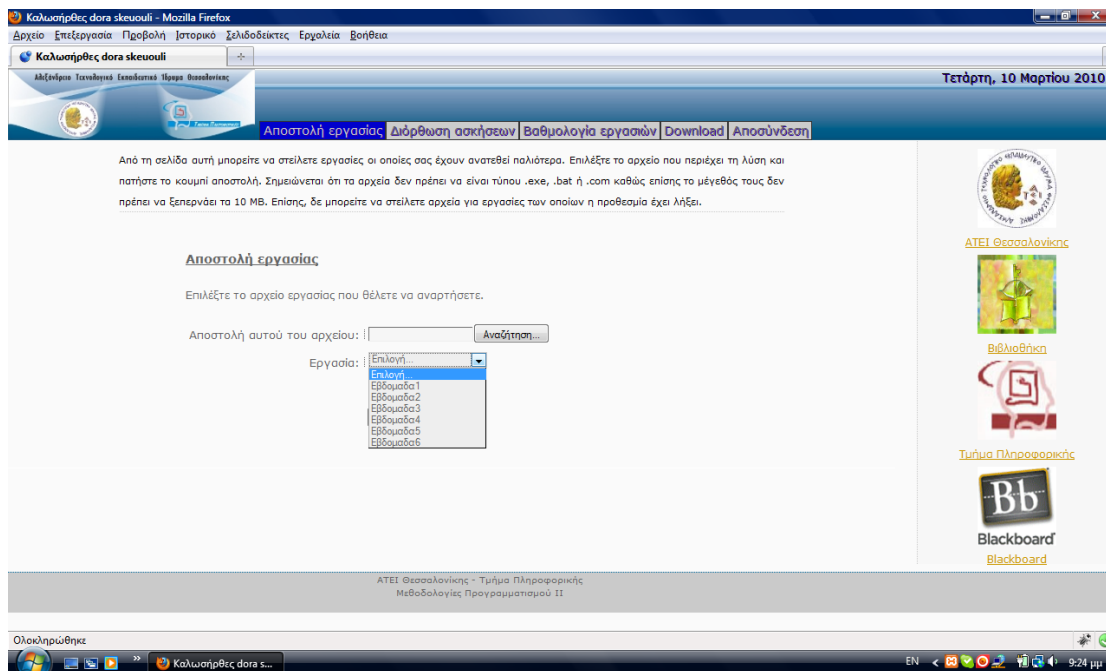
Η αρχική σελίδα που εμφανίζεται όταν ο χρήστης φοιτητής μπαίνει για πρώτη φορά στο σύστημα είναι η παρακάτω



Η φόρμα αυτή αναφέρεται στους βαθμούς που παίρνει ο φοιτητής στις εργασίες που στέλνει. Στην παρούσα φάση δεν έχει στείλει ασκήσεις οπότε και γι αυτό είναι κενό.

5.4.2 Αποστολή Άσκησης

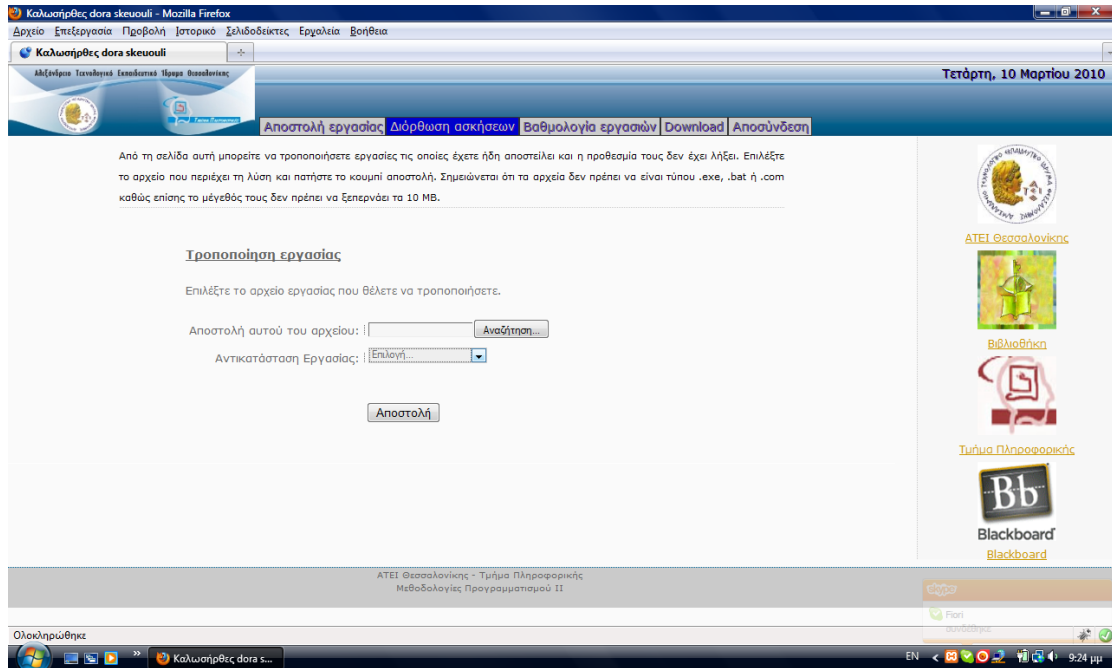
Ο χρήστης φοιτητής θα μπορεί να στείλει ασκήσεις από την παρακάτω φόρμα:



Αυτό που πρέπει να κάνει είναι να πατήσει το κουμπί αναζήτησης να βρει την άσκηση και να επιλέξει την εβδομάδα στην οποία ανήκει η άσκηση που θέλει να αποστείλει και να πατήσει αποστολή. Αυτό θα είναι δυνατόν μόνο αν είναι μέσα στο σωστό χρονικό περιθώριο που ορίζεται από το σύστημα.

5.4.3 Αποστολή διορθωμένης άσκησης

Ο χρήστης φοιτητής θα έχει την δυνατότητα να αντικαταστήσει μια ήδη σταλμένη άσκηση με μια άσκηση στην οποία έχει κάνει αλλαγές. Αυτό θα γίνεται μέσω της φόρμας:



Ο φοιτητής επιλέγει την άσκηση που θέλει να στείλει και επιλέγει και την εβδομάδα της άσκησης που θέλει να αντικαταστήσει. Βέβαια η άσκηση αυτή θα πρέπει να αποσταλθεί μέσα στο χρονικό περιθώριο που έχει ορίσει ο καθηγητής διαφορετικά η δυνατότητα ακυρώνεται.

5.4.4 Κατέβασμα υλικού

Τέλος ο χρήστης φοιτητής έχει την δυνατότητα να κατεβάσει χρήσιμο υλικό που έχει ανεβάσει ο χρήστης μέσω της φόρμας:



The screenshot shows a Mozilla Firefox browser window displaying a course page. The page title is "Καλωσήρθες dora skeouli". The browser's address bar shows "Καλωσήρθες dora skeouli". The page content includes a navigation menu with links for "Αποστολή εργασίας", "Διόρθωση ασκήσεων", "Βαθμολογία εργασιών", "Download", and "Αποσύνδεση". Below the menu, there is a text block stating: "Από τη σελίδα αυτή μπορείτε να κατεβάσετε υλικό το οποίο έχει παρουσιαστεί κατά τη διάρκεια των διαλέξεων. Πατήστε στο αντίστοιχο link και αποθηκεύστε το αρχείο στον υπολογιστή σας. Για οποιαδήποτε πρόβλημα, παρακαλώ επικοινωνήστε με τον υπεύθυνο καθηγητή." Below this text is a table with the following data:

Εβδομάδα	Αρχείο	Θέμα	Σχόλια
1	uml.docx	eisagogi stin uml	uml
2	use_cases.docx		use_case
3	rup.docx		rup
4	iconix.docx		iconix
5	protypa.docx		Protypa sxhediashs
6	activity.docx		activity diagram
7	Πρόλογος.doc	askisi	
8	-Προϊντώσεις-1.doc	periptwseis	

On the right side of the page, there are logos for "ΑΤΕΙ Θεσσαλονίκης", "Βιβλιοθήκη", and "Τμήμα Πληροφορικής", along with a "Blackboard" logo.

Η φόρμα αυτή περιγράφει την εβδομάδα στην οποία αναφέρεται το υλικό, το αρχείο, το θέμα του υλικού και σε σχόλια όπου επιθυμεί να γράψει ο χρήστης που το ανεβάζει. Ο χρήστης απλά επιλέγει το αρχείο που επιθυμεί να κατεβάσει και το ανοίγει ή το αποθηκεύει κατευθείαν στον υπολογιστή του.

5.5 Δυνατότητες Καθηγητή

Ας περιγράψουμε τώρα τις δυνατότητες του καθηγητή.

5.5.1 Κατέβασμα ασκήσεων

Ο χρήστης καθηγητής θα μπορεί να κατεβάσει τις ασκήσεις τις οποίες έχουν στείλει οι φοιτητές από την φόρμα:

Από τη σελίδα αυτή ο καθηγητής μπορεί να κατεβάσει τις ασκήσεις των φοιτητών που έχουν ανατεθεί παλαιότερα και οι οποίες ακόμα δεν έχουν βαθμολογηθεί.

Φοιτητής	AM	Αρχείο
eirini	13	13_16_metafrash.docx
giota	14	14_16_metafrash.docx
dionisis	15	15_16_metafrash.docx
anis	16	16_16_metafrash.docx
thodoris	18	18_16_metafrash.docx
efi	19	19_16_metafrash.docx
mixalis	17	17_16_metafrash.docx
anis	20	20_16_metafrash.docx
eirini	13	13_17_Πρόλογος.doc

[1]

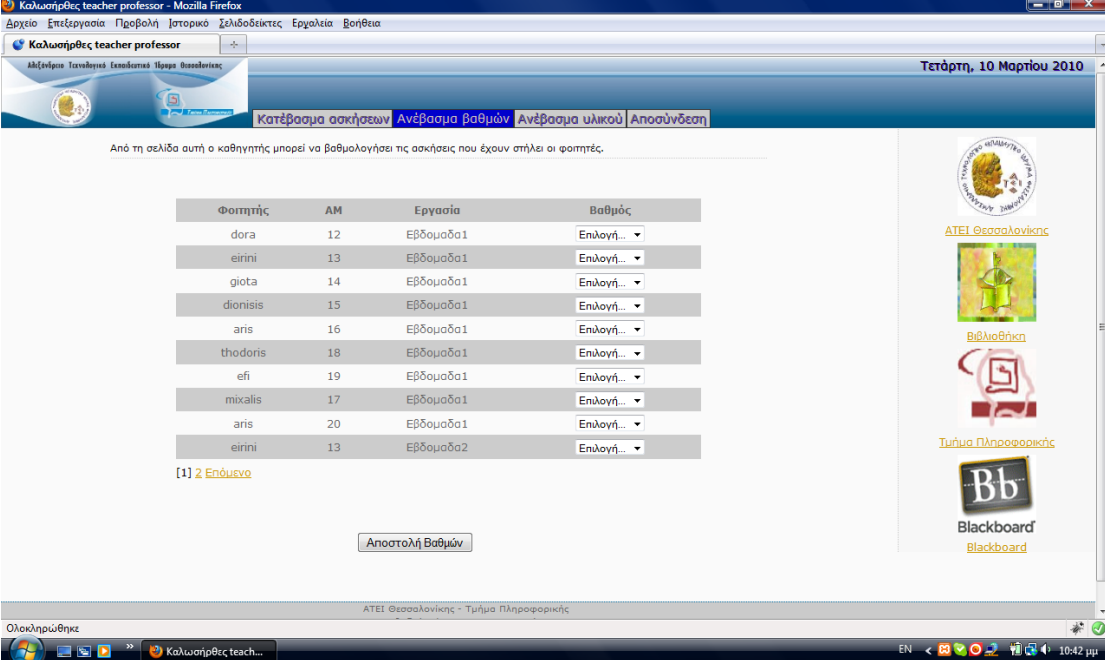
ATEI Θεσσαλονίκης - Τμήμα Πληροφορικής
Μεθοδολογίας Προγραμματισμού II

Blackboard

Η φόρμα αυτή περιγράφει το όνομα του φοιτητή, τον αριθμό μητρώου του καθώς και το όνομα της άσκησης. Ο χρήστης καθηγητής θα πατάει απλά πάνω στο link του κάθε αρχείου και ή θα το ανοίγει αμέσως ή θα το αποθηκεύει στον υπολογιστή.

5.5.2 Ανέβασμα Βαθμών

Η παρακάτω δείχνει πως ο καθηγητής θα μπορεί να βαθμολογεί τις ασκήσεις των φοιτητών.



The screenshot shows a web browser window with the address bar displaying 'Καλωσήρθες teacher professor - Mozilla Firefox'. The page title is 'Καλωσήρθες teacher professor'. The navigation menu includes 'Κατέβασμα ασκήσεων', 'Ανέβασμα βαθμών', 'Ανέβασμα υλικού', and 'Αποσύνδεση'. The main content area contains a table for entering grades and a sidebar with navigation links.

Από τη σελίδα αυτή ο καθηγητής μπορεί να βαθμολογήσει τις ασκήσεις που έχουν στείλει οι φοιτητές.

Φοιτητής	ΑΜ	Εργασία	Βαθμός
dora	12	Εβδομοδα1	Επιλογή...
eirini	13	Εβδομοδα1	Επιλογή...
giota	14	Εβδομοδα1	Επιλογή...
dionisis	15	Εβδομοδα1	Επιλογή...
aris	16	Εβδομοδα1	Επιλογή...
thodoris	18	Εβδομοδα1	Επιλογή...
efi	19	Εβδομοδα1	Επιλογή...
mixalis	17	Εβδομοδα1	Επιλογή...
aris	20	Εβδομοδα1	Επιλογή...
eirini	13	Εβδομοδα2	Επιλογή...

[1] 2 [Επόμενο](#)

ΑΤΕΙ Θεσσαλονίκης - Τμήμα Πληροφορικής

Η φόρμα αυτή δείχνει το όνομα του φοιτητή, τον αριθμό μητρώου, την εβδομάδα στην οποία ανήκει η άσκηση που έχει στείλει ο εκάστοτε φοιτητής και τέλος τον βαθμό τον οποίο πρέπει να εισάγει ο καθηγητής αφού έχει διορθώσει την κάθε άσκηση.

5.5.3 Ανέβασμα υλικού

Τέλος ο χρήστης καθηγητής θα μπορεί να ανεβάσει υλικό για να είναι διαθέσιμο στους φοιτητές μέσω της παρακάτω φόρμας:

Καλωσήρθες teacher professor - Mozilla Firefox

Αρχείο Επεξεργασία Προβολή Ιστορικό Σελιδοδείκτες Εργαλεία Βοήθεια

Καλωσήρθες teacher professor

Αθήνα Τετάρτη, 10 Μαρτίου 2010

Κατέβασμα ασκήσεων Ανέβασμα βαθμών **Ανέβασμα υλικού** Αποσύνδεση

Από τη σελίδα αυτή ο καθηγητής μπορεί να ανεβάσει υλικό ώστε να μπορούν να το κατεβάσουν οι φοιτητές.

Ανάρτηση υλικού

Επιλέξτε το αρχείο που περιέχει το υλικό που θέλετε να αναρτήσετε.

Ανάρτηση αυτού του αρχείου: Αναζήτηση...

Αριθμός Εβδομάδας: Επιλογή...

Να εμφανίζεται: Επιλογή...

Σχόλια:

Θέμα:

Αποστολή

ΑΤΕΙ Θεσσαλονίκης

Βιβλιοθήκη

Τμήμα Πληροφορικής

Blackboard

Blackboard

ΑΤΕΙ Θεσσαλονίκης - Τμήμα Πληροφορικής
Μεθοδολογίες Προγραμματισμού II

Ολοκληρώθηκε

Καλωσήρθες teach...

EN 10:43 μμ

Ο χρήστης πατώντας το κουμπί αναζήτηση επιλέγει το υλικό που επιθυμεί να ανεβάσει. Επίσης, επιλέγει τον αριθμό της εβδομάδας, τα σχόλια, το θέμα του υλικού και αν επιθυμεί να φαίνονται ή όχι.

Βιβλιογραφία

- [1] Κ. Κοντογιάννης, *Τεχνολογία Λογισμικού*(2007)
- [2] Δρ. Μιχάλης Σαλαμπάσης, *Προγραμματισμός διαδικτυακών εφαρμογών*,(2008), Θεσσαλονίκη, Ελλάδα.
- [3] Αλέξανδρος Ν. Χατζηγεωργίου, *Αντικειμενοστραφής Σχεδίαση*, (2008), Θεσσαλονίκη, Ελλάδα.
- [4] D.Martakos, *HyperText Markup Language (HTTP)*,(1998)
- [5] Welling, L., Laura, T., *PHP and MySQL Web Development*, (2005) USA.
- [6] Stan Ward, Perl Kroll, *Building Web Solutions with the Rational Unified Process* (1999)
- [7] www.eei.gr
- [8] www.el.wikipedia.gr
- [9] www.computerstudio.gr
- [10] www.devarticles.gr
- [11] www.netrino.gr
- [12] www.wlearn.gr