



ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΗΠΕΙΡΟΥ

ΣΧΟΛΗ: ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ: ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ
ΗΛΕΚΤΡΟΝΙΚΟ ΕΡΓΑΣΤΗΡΙ**

Ελευθέριος Αντ. Καραγιαννάκης

Computer Engineering, BSc

Επιβλέπων: Φώτης Βαρτζιώτης

Καθηγητής Εφαρμογών, Ph. D.

Καρδίτσα, 2018

**DESIGN AND DEVELOPMENT OF WEB APPLICATION
ELECTRONIC LABORATORY**

ΕΓΚΡΙΘΗΚΕ ΑΠΟ ΤΡΙΜΕΛΗ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ
--

Αρτα, Δευτέρα 5 Φεβ 2018

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Επιβλέπων καθηγητής

Φώτης Βαρτζιώτης,

Καθηγητής Εφαρμογών, Ph. D.

2. Μέλος επιτροπής

Όνομα Επίθετο,

τίτλος, Ph. D.

3. Μέλος επιτροπής

Όνομα Επίθετο,

τίτλος, Ph. D.

Ο Προϊστάμενος του Τμήματος

Χρυσόστομος Στύλιος,

Καθηγητής, Ph. D.

Υπογραφή

© Καραγιαννάκης, Ελευθέριος, 2017, 2018.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα πτυχιική εργασία είναι εξ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

Καραγιαννάκης, Ελευθέριος

Υπογραφή

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέπων καθηγητή μου, κ. Βαρτζιώτη Φώτιο για την άψογη συνεργασία και την πολύτιμη βοήθεια που μου πρόσφερε σε όλη την διάρκεια της πτυχιακής μου εργασίας. Επίσης ένα ευχαριστώ θα ήθελα να απευθύνω στους καθηγητές μου για την γνώση που μου προσέφεραν όλα αυτά τα χρόνια. Τέλος, το πιο μεγάλο ευχαριστώ, θα ήθελα να το προσφέρω στην οικογένεια μου για την στήριξη που μου πρόσφερε όλα αυτά τα χρόνια, σε αυτό το μεγάλο αγώνα προς το πτυχίο μου.

ΠΕΡΙΛΗΨΗ

Η παρούσα εργασία αναφέρεται στη σχεδίαση και στην ανάπτυξη της διαδικτυακής εφαρμογής ηλεκτρονικό εργαστήριο. Αρχικά γίνεται μια μικρή εισαγωγή στις διαδικτυακές εφαρμογές όπου αναφέρεται ο σκοπός και το πως εξελίσσονται με την πάροδο του χρόνου. Στη συνέχεια, βλέπουμε την πλήρη ανάλυση της υπάρχουσας κατάστασης με βάση τα σημερινά δεδομένα. Ξεχωρίζει μια μικρή ιστορική αναδρομή στο χρόνο για να γνωρίσουμε την εξέλιξη της γλώσσας προγραμματισμού PHP, καθώς και τον τρόπο σύνταξης της. Στο τρίτο κεφάλαιο γίνεται μια πλήρης ανάλυση απαιτήσεων από τους χρήστες οι οποίοι χρησιμοποιούν το πληροφοριακό σύστημα που αναπτύχθηκε, οι οποίες διαχωρίζονται σε λειτουργικές και σε μη λειτουργικές απαιτήσεις. Μετέπειτα, ακολουθεί ο σχεδιασμός της εφαρμογής με χρήση σχεδιαγραμμάτων της γλώσσας UML. Στη συνέχεια, παρουσιάζεται η εφαρμογή και δίνονται οδηγίες χρήσης. Επιπλέον, παρουσιάζεται ο τρόπος που έχει προγραμματιστεί το συγκεκριμένο πληροφοριακό σύστημα μέσω της εφαρμογής Notepad Plus Plus. Τέλος, στα τελευταία κεφάλαια επισημαίνονται οι παρατηρήσεις της εφαρμογής, στις οποίες αναλύεται η παραμετροποίηση και ακολουθούν οι μελλοντικές αναβαθμίσεις.

Λέξεις-κλειδιά: διαδικτυακή, εφαρμογή, ηλεκτρονικό, εργαστήριο

ABSTRACT

This project refers to design and development of a web application, that is called “electronic laboratory”. In the first chapter, there is a small introduction to online applications about the purpose and how they have evolved over time. In the next chapter, we provide a complete analysis with the state-of-the-art in the domain of web applications, based on today’s data. A historical breakthrough in time stands out in order to get to know the evolution of the PHP programming language. The third chapter is an important analysis of users’ requirements. An analysis of the functional and non-function “Users Requirements” is given in the furthermore, the UML diagrams are used to depict the design flow of the targeted web application. The application then is presented and instructions for use are given. In addition, it’s presented how this information system has been programmed through Notepad Plus Plus. Finally, in the last chapters there are guidelines notes for parameterization and future upgrades.

Keywords: web, application, electronic, lab

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Ευχαριστίες.....	6
Περίληψη.....	7
Abstract.....	8
ΚΕΦΑΛΑΙΟ 1. Εισαγωγή.....	12
1.1 Εισαγωγή στις διαδικτυακές εφαρμογές.....	12
1.2 Λόγοι Χρήσης Εφαρμογής.....	12
1.3 Στόχοι της πτυχιακής.....	14
ΚΕΦΑΛΑΙΟ 2. Ανάλυση Υπάρχουσα Κατάσταση.....	16
2.1 Η Γλώσσα HTML.....	16
2.1.1 Ιστορική Αναφορά.....	16
2.1.2 Δομή HTML.....	17
2.2 Ο μηχανισμός CSS.....	19
2.2.1 Σύνταξη CSS.....	20
2.3 Γλώσσα Προγραμματισμού PHP.....	22
2.3.1 Ιστορική Αναφορά.....	23
2.3.2 Δομή.....	24
ΚΕΦΑΛΑΙΟ 3. Ανάλυση Απαιτήσεων.....	28
3.1 Κατηγορίες Απαιτήσεων.....	28
3.1.1 Η Απαίτηση.....	28
3.1.2 Λειτουργικές Απαιτήσεις και Μη Λειτουργικές Απαιτήσεις.....	30
3.2 Περιπτώσεις Χρηστών.....	31
3.2.1 Σύνδεση Χρήστη.....	31
3.2.2 Σύνδεση Χρήστη στο TCEXAM.....	33
3.2.3 Εγγραφή Χρήστη σε εργαστήριο.....	34
3.2.4 Επικοινωνία Χρήστη.....	35
3.2.5 Αλλαγή Γλώσσας εφαρμογής.....	36
3.2.6 Αποσύνδεση Χρήστη.....	37
ΚΕΦΑΛΑΙΟ 4. Σχεδιασμός Μηχανισμών Πλατφόρμας	39
4.1 Σχεδιασμός Αντικειμένων.....	39
4.1.1 Διάγραμμα Δραστηριότητας (Activity).....	41
4.1.2 Διάγραμμα Ακολουθίας (Sequence).....	42

ΚΕΦΑΛΑΙΟ 5. Σχεδίαση και Υλοποίηση Εφαρμογής.....	43
5.1 Οδηγίες Χρήσης Εφαρμογής	43
5.2 Περιβάλλον Ανάπτυξης Εφαρμογής	44
ΚΕΦΑΛΑΙΟ 6. Παρατηρήσεις.....	44
6.1 Συμπεράσματα.....	44
6.2 Μελλοντικές Αναβαθμίσεις.....	46
ΚΕΦΑΛΑΙΟ 7. Παραρτήματα	48
Παράρτημα Α. Κώδικας Αρχικής Σελίδας Ηλεκτρονικό Εργαστήρι	48
Βιβλιογραφία.....	55

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1.1 [Χρονοδιάγραμμα HTML].....	17
Πίνακας 1.2 [Παράδειγμα επεξήγησης ορισμένων εννοιών].....	18
Πίνακας 1.3 [Παράδειγμα επεξήγησης σύνταξης html].....	18
Πίνακας 1.4 [Παράδειγμα επεξήγησης σύνταξης CSS μέσα στην ιστοσελίδα].....	20
Πίνακας 1.5 [Παράδειγμα επεξήγησης σύνταξης CSS με μορφή κλάσης].....	21
Πίνακας 1.6 [Χρονοδιάγραμμα PHP].....	23
Πίνακας 1.7 [Παράδειγμα επεξήγησης σύνταξης PHP].....	25
Πίνακας 1.8 [Παράδειγμα επεξήγησης εντολή Echo].....	25

ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ/ΕΙΚΟΝΩΝ

Εικόνα 1. [Επικοινωνία χρήστη με την βάση δεδομένων].....	14
Εικόνα 2. [Περιπτώσεις χρήσης συστήματος].....	29
Εικόνα 3. [Μοντέλο του καταρράκτη].....	30
Εικόνα 4. [Διαγράμματα της UML].....	39
Εικόνα 5. [Απεικόνιση διαγράμματος κλάσεων].....	40
Εικόνα 6. [Περιπτώσεις Χρηστών].....	40
Εικόνα 7. [Διάγραμμα δραστηριοτήτων User Login].....	41
Εικόνα 8. [Διάγραμμα ακολουθίας εισόδου φοιτητή].....	42
Εικόνα 9. [Αρχική σελίδα εφαρμογής].....	43
Εικόνα 10. [Επιλογές εφαρμογής].....	44
Εικόνα 11. [Σύνδεση Χρήστη].....	44
Εικόνα 12. [Σύνδεση Χρήστη σε εξωτερική εφαρμογή].....	45
Εικόνα 13. [Εγγραφή χρήστη σε εργαστήριο].....	45
Εικόνα 14. [Επικοινωνία χρήστη με υπεύθυνο εργαστηρίου].....	46
Εικόνα 15. [Αποσύνδεση χρήστη].....	46
Εικόνα 16. [Το λογισμικό της εφαρμογής Notepad++].....	49

1. Εισαγωγή

1.1 Εισαγωγή στις διαδικτυακές εφαρμογές

Οι διαδικτυακές εφαρμογές στις μέρες μας ολοένα και αυξάνονται. Με αυτό τον τρόπο φανερώνουν πως έχουν έρθει στην ζωή του ανθρώπου για να την κάνουν πιο εύκολη και ευέλικτη. Χιλιάδες άνθρωποι σε καθημερινή βάση χρησιμοποιούν τέτοιου είδους εφαρμογές που έχουν το ρόλο του σταθμού εξυπηρέτησης.

Με τον όρο διαδικτυακή εφαρμογή ή με τον αγγλικό όρο web application ονομάζεται η εφαρμογή που είναι διαθέσιμη σ' έναν ή περισσότερους χρήστες ταυτόχρονα μέσω του διαδικτύου ή του ενδοδικτύου μιας επιχείρησης ή ακόμη και ενός οργανισμού. Αντίστοιχα ο χρήστης χρησιμοποιεί ένα απλό φυλλομετρητή ιστοσελίδων για να αποκτήσει πρόσβαση στο διαδίκτυο και στη συνέχεια χρησιμοποιεί τη διαδικτυακή εφαρμογή που έχει επιλέξει για να εργαστεί. Συνήθως ο χρήστης δεν χρειάζεται να εγκαταστήσει κάποια άλλη εξωτερική εφαρμογή, αλλά τέτοιου είδους εφαρμογές παρέχουν αυτόματα την πρόσβαση σε υπηρεσίες καθώς και πληροφορίες.

Με αυτό τον τρόπο εξοικονομούμε χρόνο και έχουμε το ίδιο αποτέλεσμα. Σήμερα όλο και περισσότερες εφαρμογές κάνουν την εμφάνισή τους, καθώς δίνουν την δυνατότητα στους χρήστες να εντοπίσουν τόσο τα θετικά όσο και τα αρνητικά στοιχεία μιας διαδικτυακής εφαρμογής.

1.2 Λόγοι Χρήσης Εφαρμογής

Καθημερινά οι διαδικτυακές εφαρμογές λύνουν αρκετά προβλήματα. Προβλήματα που προκείμενου να επιλυθούν στο παρελθόν απαιτούσαν είτε περισσότερο χρόνο είτε μεγαλύτερη χρηματική δαπάνη. Έτσι δημιουργήθηκε το ηλεκτρονικό εργαστήρι. Πρόκειται για μια σύγχρονη ηλεκτρονική υπηρεσία του ΤΕΙ Ηπείρου η οποία έχει φέρει μια επαναστατική πρόοδο στις ακαδημαϊκές συνθήκες του Ιδρύματος και γι' αυτό αξίζει να αναφερθούμε αναλυτικά σε αυτό.

Όσο αφορά στους λόγους για τους οποίους ασχολήθηκα προσωπικά με τη διαδικτυακή εφαρμογή είναι ποικίλοι και σημαντικοί:

Εύκολη διαχείριση των δεδομένων της εφαρμογής.

Ασφάλεια και προστασία δεδομένων του χρήστη.

Συγχρονισμός εφαρμογής σε όλες τις «έξυπνες» πλατφόρμες.

Εύκολη πλοήγηση και Νέος σχεδιασμός με διαδραστικά στοιχεία.

Η συγκεκριμένη εφαρμογή εξυπηρετεί καθημερινές ανάγκες σε ακαδημαϊκά πλαίσια. Συγκεκριμένα οι ανάγκες που καλύπτει σήμερα η εφαρμογή είναι οι παρακάτω.

Είσοδος χρήστη με την υπηρεσία CAS.

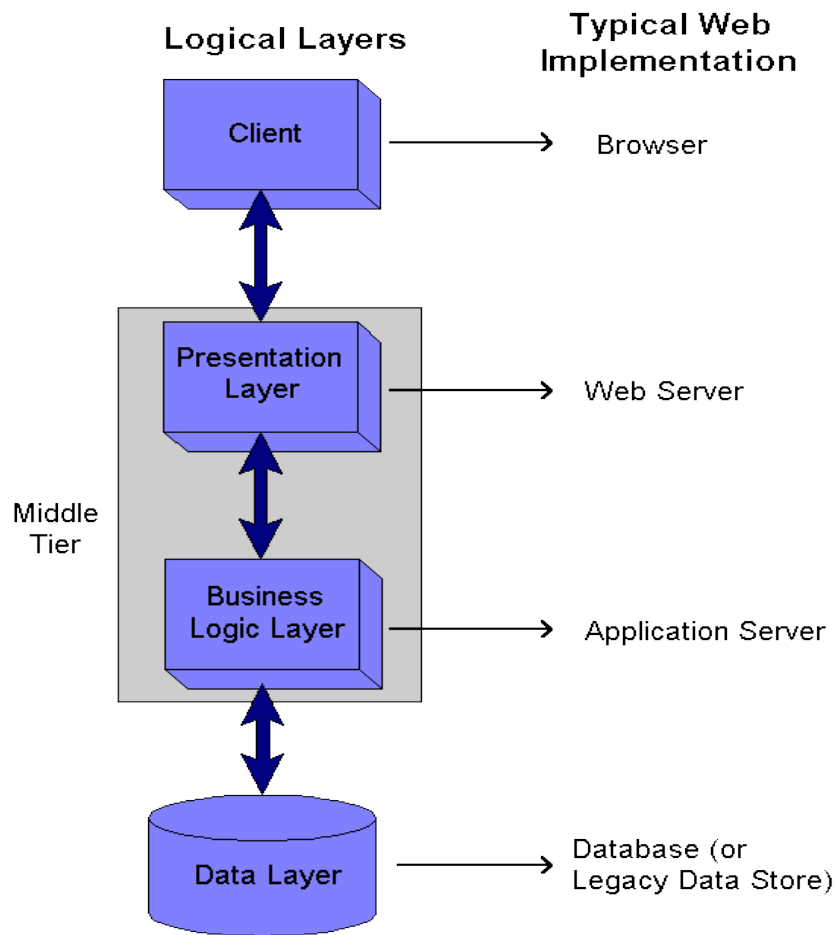
Εγγραφή χρήστη σε εργαστήριο μαθήματος.

Είσοδος χρήστη σε εξωτερική εφαρμογή TCEXAM.

Τροποποίηση γλώσσας εφαρμογής σε Αγγλική.

Επικοινωνία χρήστη με τον αρμόδιο καθηγητή εργαστηρίου.

Τέλος, το «ηλεκτρονικό εργαστήριο» γεννήθηκε για να διευκολύνει τις σχέσεις φοιτητή και καθηγητή, να ενισχύσει την απόδοση του φοιτητή, όπως ο υπολογισμός των απουσιών του και η καταχώρηση της βαθμολογίας του, καθώς κάνει πιο εύκολη την δουλειά του καθηγητή.



Εικόνα 1. Επικοινωνία Χρήστη με την Βάση Δεδομένων (<https://goo.gl/auS12D>)

1.3 Στόχοι της πτυχιακής εργασίας

Στόχος της πτυχιακής εργασίας είναι η δημιουργία μιας σύγχρονης διαδικτυακής εφαρμογής δήλωσης εργαστηριακών μαθημάτων εξ' αποστάσεως. Πρόκειται για μια διαδικτυακή εφαρμογή που έχει σαν στόχο την διευκόλυνση των φοιτητών – συναδέλφων και των καθηγητών της σχολής μας. Αναπτύχθηκε με σκοπό να μπορεί ο φοιτητής – συνάδελφος να δηλώσει το εργαστήριο του μαθήματος που επιθυμεί σε πραγματικό χρόνο οπουδήποτε κι αν βρίσκετε με ένα κλικ.

Η εφαρμογή περιλαμβάνει διαδικασία ελέγχου σε πραγματικό χρόνο καθώς και πιστοποίηση της ταυτότητας του χρήστη. Στην συνέχεια δίνει την δυνατότητα στον φοιτητή – συνάδελφο μέσα από διάφορες λειτουργίες που έχουν σχεδιαστεί και έχουν

αναπτυχθεί στο σύστημα, να εγγραφεί σε κάποιο εργαστήριο μαθήματος της σχολής. Μία ακόμη λειτουργία είναι ότι του παρέχει πρόσβαση στην εξωτερική εφαρμογή TCEXAM. Τέλος η πύλη ηλεκτρονικό εργαστήριο παρέχει ακόμη μια δυνατότητα η οποία είναι η επικοινωνία του φοιτητή με τον αρμόδιο καθηγητή εργαστηρίου. Επίσης θα είναι διαθέσιμη σε όλες τις « έξυπνες » συσκευές και η δήλωση θα γίνεται σε πραγματικό χρόνο και σε περιορισμένο χρονικό διάστημα που έχει επιλέξει ο κάθε καθηγητής.

2. Ανάλυση Υπάρχουσας Κατάστασης

2.1 Η Γλώσσα HTML

Η γλώσσα HTML¹ ή αλλιώς με τον ελληνικό όρο «Γλώσσα Σήμανσης Υπερκειμένου» είναι μια μη – υπολογιστική γλώσσα, που στις μέρες μας κατά κύριο λόγο είναι η επικρατέστερη γλώσσα για τη δημιουργία ιστοσελίδων ή μικρών διαδικτυακών εφαρμογών. Αυτό οφείλεται σε δυο λόγους:

i) στην απλή κατανόηση λογικής, ii) στον εύκολο τρόπο σύνταξης.

Ως έννοια στον όρο *γλώσσα* ορίζουμε ένα αλφάβητο Σ . Γλώσσα επί του αλφαβήτου Σ ονομάζουμε κάθε σύνολο συμβολοσειρών του Σ , δηλαδή κάθε υποσύνολο του Σ^* . Οι γλώσσες συμβολίζονται συνήθως με κεφαλαία λατινικά γράμματα, όπως *L*, *S*, *T*, *D*, κ.λπ. Πολλές φορές παραλείπουμε τον καθορισμό του αλφαβήτου της γλώσσας, αν αυτό εννοείται. Κάθε γλώσσα ενός αλφαβήτου Σ είναι και γλώσσα ενός αλφαβήτου Σ' τέτοιου ώστε $\Sigma \subseteq \Sigma'$. (2002, σ. 26)

Η γλώσσα υλοποιείται σε οποιοδήποτε επεξεργαστεί κειμένου όπως το Windows Notepad ή το Notepad++ με την βασική μορφή επέκτασης (.html) . Στην συνέχεια πρώτα γράφετε το κείμενο που επιθυμούμε και μετά προσθέτουμε *ετικέτες* «**markup tags**» γύρω από τις λέξεις ή τις ολοκληρωμένες προτάσεις, ώστε να κάνουμε τα αποτελέσματα στην οθόνη του χρήστη πιο ευανάγνωστα. “*Ετικέτες ονομάζουμε ονόματα στοιχείων που περιβάλουν γωνιακές παρενθέσεις*” (βλ. https://www.w3schools.com/html/html_intro.asp). Τα στοιχεία της HTML συνήθως αποτελούνται από μια *ετικέτα έναρξης* καθώς και μια τελική *ετικέτα* με το περιεχόμενα να περιλαμβάνεται ανάμεσα. Πιο συγκεκριμένα θα αναφερθούμε στα επόμενα υποκεφάλαια.

¹ Hyper Text Markup Language

2.1.1 Ιστορική Αναδρομή

Στα αρχές τις δεκαετίας 1990 επινοήθηκε μια μη υπολογιστική γλώσσα που αργότερα θα γινότανε ένας τρόπος επικοινωνίας και διαφήμισης. “ *Πατέρας του διαδικτύου είναι ο Tim Berners-Lee είναι ο άνθρωπος που οδήγησε στην ανάπτυξη του Παγκόσμιου Ιστού, τον ορισμό της γλώσσας HTML (hypertext markup language) που χρησιμοποιείται για τη δημιουργία ιστοσελίδων, HTTP (HyperText Transfer Protocol) και URLs (Universal Resource Locators)* ” (Mary Bellis, Sept 11, 2017). Παράλληλα την ίδια δεκαετία αναπτύχθηκε και ο φυλλομετρητής ή αλλιώς web browser, που με απλά λόγια θα λέγαμε ότι είναι ένα λογισμικό που εμφανίζει τις πληροφορίες της ιστοσελίδας. Ο φυλλομετρητής δεν εμφανίζει τον κώδικα της ιστοσελίδας αλλά μόνο τα αποτελέσματα. Η εξέλιξη όμως δεν σταματάει εδώ. Στις μέρες μας η HTML είναι από τις πιο διαδεδομένες γλώσσες καθώς η δομή της αποτελεί τον βασικό κορμό για την δημιουργία μιας ιστοσελίδας. Συντακτικά έχει πολύ απλή δομή δόμησης που γίνεται εύκολα κατανοητή απ’ όλους. Στο παρακάτω πίνακα αναφέρετε ένα απλό χρονοδιάγραμμα εξέλιξης της γλώσσας στο πέρασμα του χρόνου από την ημέρα δημιουργίας της έως το σήμερα.

ΕΞΕΛΙΞΗ	ΧΡΟΝΟΣ
HTML	1991
HTML+	1993
HTML 2.0	1995
HTML 3.2	1997
HTML 4	1999
HTML 5	2014

Πίνακας 1.1 [Χρονοδιάγραμμα HTML] (Ελευθέριος Καραγιαννάκης, 2018)

2.1.2 Δομή της HTML

Όπως προ αναφέραμε η σύνταξη της δομή ενός έγγραφου είναι εύχρηστη και κατανοητή. Η γλώσσα ξεκινά με συγκεκριμένες λέξεις κλειδιά καθώς τα έγγραφα HTML πρέπει να ξεκινούν με μια δήλωση τύπου εγγράφου. Επίσης κάθε έγγραφο αρχίζει και τελειώνει με την ίδια *ετικέτα* όμως το κύριος μέρος του εγγράφου

αναγράφετε μέσα στις ετικέτες **<body>** και **</body>**. Πιο κάτω αναφέρουμε ένα παράδειγμα επεξηγήσεις ορισμένων εννοιών της γλώσσας.

-
- **<!DOCTYPE html>** ορίζεται ότι το έγγραφο είναι HTML5.
 - **<html>** ορίζεται το στοιχείο ρίζα HTML
 - </html>**
 - **<head>** ορίζεται το στοιχείο που περιέχει μετά-πληροφορίες για το έγγραφο.**</head>**
 - **<title>** ορίζεται το στοιχείο που καθορίζει ένα τίτλο για το έγγραφο.
 - </title>**
 - **<body>** ορίζεται το περιεχόμενο ή αλλιώς το σώμα της σελίδας.
 - </body>**
 - **<p>** ορίζεται μια παράγραφος.
 - </p>**
 - **// Αυτό είναι το σχόλιο στην html**
-

Πίνακας 1.2 [Παράδειγμα επεξήγησης ορισμένων εννοιών] (Ελευθέριος Καραγιαννάκης, 2018)

Στη συνέχεια, θα αναφερθούμε σε ένα απλό εισαγωγικό παράδειγμα δημιουργίας web ιστοσελίδας βλέποντας βήμα-βήμα την δομή καθώς και τον τρόπο σύνταξης.

```
<!DOCTYPE html>

<html>

<head><title>1st page</title></head>

<body> <p>HELLO WORLD.</p> </body>

</html>
```

Πίνακας 1.3 [Παράδειγμα επεξήγησης σύνταξης html] (Ελευθέριος Καραγιαννάκης, 2018)

Όπως προαναφέραμε μια ιστοσελίδα ξεκινά με την δήλωση έγγραφου, στην συνέχεια ακολουθεί η επικεφαλίδα μαζί με τον τίτλο σελίδας, μετέπειτα αναγράφεται το κυρίως κείμενο μέσα στο σώμα της κεντρικής σελίδας και τέλος προσθέτουμε την παράγραφο. Σε δεύτερη φάση, αποθηκεύουμε το αρχείο με την ονομασία που επιθυμούμε και στο τέλος προσθέτουμε την μορφή επέκτασης *.html* στην επιφάνεια εργασίας. Σε τρίτη φάση, ανοίγουμε το αρχείο με τον φυλλομετρητή του ηλεκτρονικού υπολογιστή και βλέπουμε τα αποτελέσματα στην οθόνη. Η έξοδος που θα τυπωθεί στην οθόνη του χρήστη είναι το σχετικό μήνυμα:

HELLO WORD.

2.2 Μηχανισμός CSS

Το CSS² ή αλλιώς τα διαδοχικά φύλλα στυλ, είναι ένας μηχανισμός στυλ που το ορίζουμε για τις μορφοποιήσεις των ιστοσελίδων HTML. Με την χρήση CSS μπορούμε να ορίσουμε χρώματα, μεγέθη γραμματοσειρών, πίνακες, φόντο αρχικής σελίδας, τρόποι γραφής κάνοντας την ιστοσελίδα μας πιο ζωντανή καθώς και πιο διασκεδαστική στο πέρασμα του χρόνου.

Για να πετύχει αυτό υπάρχουν τρεις τρόποι. Πρώτος τρόπος είναι ο ορισμός του στυλ να γίνει μέσα στην ιστοσελίδα, δεύτερος τρόπος να υλοποιηθεί με εξωτερικό αρχείο καθώς ο τρίτος τρόπος είναι ο ορισμός να γίνει μέσα στις ετικέτες της ιστοσελίδας. Πιο αναλυτικά θα αναφερθούμε στο επόμενο υποκεφάλαιο.

Με άλλα λόγια, ο μηχανισμός CSS είναι απλώς ένα αρχείο κειμένου. Περικλείεται από έναν ή περισσότερους κανόνες και αποτελείται από δυο βασικά στοιχεία. Ως πρώτο στοιχείο θα ορίζουμε τον *επιλογέα (selector)* όπου αναφέρεται το όνομα του στοιχείου που θέλουμε να μετατρέψουμε. Στην συνέχεια ακολουθεί η *ιδιότητα (attributes)* του επιλογέα μαζί με τις *τιμές (values)* των ιδιοτήτων του επιλογέα.

² Cascading Style Sheets

Τέλος, η τεχνολογία CSS διαθέτει δυναμικές ιδιότητες για την δημιουργία δυναμικών μενού και αλληλεπιδραστικών στοιχείων κάνοντας την εμφάνιση της ιστοσελίδας πιο επαγγελματική και πιο εύχρηστη στον χρήστη.

2.2.1 Σύνταξη CSS

Η σύνταξη της Cascading Style Sheets δομικά είναι πολύ απλή, όπως προαναφέραμε στο προηγούμενο υποκεφάλαιο. Συντακτικά χωρίζεται από τον επιλογέα, την ιδιότητα και τις τιμές. Πιο αναλυτικά θα αναφερθούμε στα επόμενα παραδείγματα. Ας ξεκινήσουμε μ' ένα απλό παράδειγμα σύνταξης CSS προσθέτοντας την γλώσσα HTML.

```
<!DOCTYPE html>

<html>

<head>

<title>2nd page</title>

<style type="text/css">

    p {

        font-family: "Times New Roman";

    }

</style>

</head>

<body> <p>HELLO WORLD.</p> </body>

</html>
```

Πίνακας 1.4 [Παράδειγμα επεξήγησης σύνταξης CSS μέσα στην ιστοσελίδα] (Ελευθέριος Καραγιαννάκης, 2018)

Στην οθόνη του χρήστη θα εμφανιστεί το σχετικό μήνυμα της παραγράφου HELLO WORD, στην μορφή γραμματοσειράς Times New Roman. Κάθε φορά που θα χρησιμοποιούμε τις ετικέτες `<p></p>`, ορίζουμε το κείμενο ώστε να έχει την συγκεκριμένη γραμματοσειρά. Σ' ένα CSS μπορούμε να προσθέσουμε όσες ιδιότητες επιθυμούμε. Στο επόμενο παράδειγμα ακολουθεί άλλη μια υλοποίηση μηχανισμού CSS με διαφορετικό τρόπο. Γενικά, η CSS έχει αρκετούς τρόπους να ορίσουμε ιδιότητες ώστε να καταφέρουμε να υλοποιήσουμε τις απαιτήσεις του χρήστη.

```
<!DOCTYPE html>

<html>

<head>

<title>3rd page</title>

<style type="text/css">

    .color { COLOR: #228B22; }

    .b{ font-weight: bold; }

</style>

</head>

<body>

<p class="color b">HELLO WORLD. </p>

</body>

</html>
```

Πίνακας 1.5 [Παράδειγμα επεξήγησης σύνταξης CSS με μορφή κλάσης] (Ελευθέριος Καραγιαννάκης, 2018)

Στο παράδειγμα 1.5 αναφέρεται ο μηχανισμός CSS με την μορφή κλάσης. Η μορφή κλάσης δίνει την δυνατότητα στον προγραμματιστή να προσθέσει δυο ή περισσότερες ετικέτες σε μια κλάση, κάνοντας το αποτέλεσμα πιο διαδραστικό στην οθόνη του χρήστη. Ο τρόπος σύνταξης είναι απλός και αρκετά διαδεδομένος στις ημέρες μας.

Τέλος, μελετώντας το παράδειγμα 1.5 παρατηρούμε ότι η έξοδος που θα λάβει ο χρήστης είναι το προαναφερόμενο μήνυμα HELLO WORLD. , με χρώμα γραμματοσειράς πράσινο καθώς και με μια πιο έντονη γραφή.

2.3 Γλώσσα Προγραμματισμού PHP

Ως ορισμό στην έννοια *πρόγραμμα* ή *γλώσσα προγραμματισμού* ορίζουμε είτε ένα σύνολο γραμμένων οδηγιών που δημιουργείται από έναν προγραμματιστή, είτε ένα εκτελέσιμο τμήμα λογισμικού. Ακόμα ως έννοια στην λέξη *μεταγλωττιστή* θα μπορούσαμε να ορίσουμε ένα εργαλείο που μεταφράζει προγράμματα μιας αρχικής γλώσσας προγραμματισμού σε εκτελέσιμο κώδικα μηχανής. (2002, σ. ν)

Η PHP ή αλλιώς με τον αγγλικό όρο Hypertext Preprocessor συμπεριλαμβάνεται στις *CGI³ Script* η οποία είναι μέρος της δέσμης λογισμικού LAMP από το ακρόνυμο Linux/Apache/MySQL/PHP, ομοίως στην ίδια κατηγορία βρίσκονται η γλώσσα προγραμματισμού Perl αλλά και η Python. “ *Η Κοινή Διεπαφή Πύλη ή το (CGI) είναι ένα πρότυπο για προγράμματα εξωτερικής πύλης για διασύνδεση με διακομιστές πληροφοριών όπως διακομιστές HTTP* ” (Internet Archive WayBack Machine, Jan 27, 2010). Με άλλα λόγια τα αρχεία της PHP αποτελούνται αποκλειστικά και μόνο από ένα απλό κείμενο, δημιουργούνται με οποιαδήποτε συντάκτη κειμένων. Επίσης οι δημοφιλέστεροι συντάκτες κειμένου HTML παρέχουν και υποστήριξη στην γλώσσα προγραμματισμού PHP.

Συνεπώς οι γλώσσες PHP, Perl, Python συνεργάζονται με το διακομιστή ιστού για την μεταγλώττιση των αιτήσεων από τον παγκόσμιο ιστό, για την επεξεργασία τους και να λάβουν τα δεδομένα σε πραγματικό χρόνο, για την αλληλοεπίδραση τους με άλλες εφαρμογές που εκτελούνται στον διακομιστή ιστού, επίσης να παρουσιάσουν το περιεχόμενο που θα στείλουν στον φυλλομετρητή του πελάτη. Σαν γλώσσα προγραμματισμού θα μπορούσαμε να αναφέρουμε ότι περιέχει γρήγορη απόδοση, μεγάλη αξιοπιστία καθώς και φιλικό περιβάλλον στο χρήστη για την συγγραφή του κώδικα. Η κύρια χρήση της γλώσσας προγραμματισμού είναι η δημιουργία

³ Common Gateway Interface

ιστοσελίδας ιστού με δυναμικό περιεχόμενο. Στον επόμενο πίνακα παρατηρούμε ένα χρονοδιάγραμμα εξέλιξης της γλώσσας από το 1994 έως το 2005.

ΕΞΕΛΙΞΗ	ΧΡΟΝΟΣ
1 st version	1994
2 nd version	1995
PHP 3.0.x	1998
PHP 4.0.0	2000
PHP 5.0.0	2005

Πίνακας 1.6 [Χρονοδιάγραμμα PHP] (Ελευθέριος Καραγιαννάκης, 2018)

2.3.1 Ιστορική Αναδρομή

Η ιστορία της PHP ξεκινά πριν από μερικά χρόνια, περίπου το 1994 όταν ο τότε φοιτητής Rasmus Lerdorf δημιούργησε ένα προϊόν με την ονομασία PHP/FI από τα αρχικά των λέξεων (Personal Home Page / Form Interpreter). Η πρώτη έκδοση της PHP ήταν ένα απλό σύνολο δυαδικών αρχείων CGI γραμμένο σε γλώσσα προγραμματισμού C. “ *I wonder why people keep writing that PHP was ever written in Perl. It never was.* ” (Rasmus, Lerdorf, Jul 20, 2012)⁴ Αρχικά χρησιμοποιήθηκε για παρακολούθηση επισκέψεων σε ηλεκτρονικά βιογραφικά με την ονομασία PHP Tools, αλλά λόγω της μεγάλης ζήτησης ο δημιουργός αποφάσισε να ξαναγράψει την έκδοση κάνοντας μια καλύτερη και πλουσιότερη εφαρμογή. Το νέο μοντέλο ήταν ικανό για την αλληλεπίδραση με βάση δεδομένων παρέχοντας στους χρήστες την ανάπτυξη απλών δυναμικών διαδικτυακών εφαρμογών. Το 1995 δημοσιεύτηκε ο βασικός κώδικας δίνοντας τη δυνατότητα στους προγραμματιστές να κάνουν την παραμετροποίηση που χρειαζότανε και να το χρησιμοποιήσουν όπως θέλουν. Επίσης ενθάρρυνε τους προγραμματιστές να διορθώσουν τα σφάλματα που υπήρχαν και να παρέχουν νέες λύσεις για την βελτίωση του κώδικα.

⁴ Rasmus Lerdorf, (βλ. <https://twitter.com/rasmus/status/226405807305138176>)

Το Σεπτέμβριο του ίδιου χρόνου ο δημιουργός συνέχισε να εργάζεται στο έργο PHP για μικρό χρονικό διάστημα δίνοντας το όνομα PHP καθώς η νέα εφαρμογή περιλάμβανε κάποιες ορισμένες λειτουργίες που γνωρίζουμε ακόμη και σήμερα. Διατήρησε παρόμοια σύνταξη με την γλώσσα προγραμματισμού Perl στην δήλωση μεταβλητών και ενσωμάτωσε αυτόματη ερμηνεία μορφής HTML. Τον Οκτώβριο η γλώσσα προγραμματισμού σχεδιάστηκε σκόπιμα για να μοιάζει με την δομή της C, κάνοντας εύκολη την υιοθέτηση σε εξοικειωμένους προγραμματιστές. Τον επόμενο χρόνο Απρίλιο του '96 η γλώσσα πήρε ακόμη μια αλλαγή η οποία ήταν αφορμή που εξελίχθηκε από σειρά εργαλείων σε γλώσσα προγραμματισμού όπου περιλάμβανε την υποστήριξη βάσεις δεδομένων.

Η PHP 3.0 είναι η πιο κοντινή έκδοση της PHP που γνωρίζουμε σήμερα. Έκανε την εμφάνιση της το 1997. Σε μια προσπάθεια να αλλάξουν τον πυρήνα της PHP/FI αποφάσισαν να γίνει μια συνεργασία και να αναπτυχθεί μια νέα γλώσσα προγραμματισμού η οποία κυκλοφόρησε με νέο όνομα **PHP: Hypertext Preprocessor**. Προστέθηκε διεπαφή για πολλαπλή βάση δεδομένων, πρωτόκολλα, APIs και το βασικό χαρακτηριστικό ήταν που περιλάμβανε αντικειμενοστραφή υποστήριξη προγραμματισμού, καθώς ισχυρότερη και συνεπέστερη σύνταξη γλώσσας.

Τον χειμώνα του '98 η ομάδα άρχισε να εργάζεται στην επανεγγραφή του πυρήνα της PHP με στο στόχο το νέο σχεδιασμό να βελτιωθεί η απόδοση σύνθετων εφαρμογών. Ο στόχος ικανοποιήθηκε επιτυχώς με ονομασία «Zend Engine» και παρουσιάστηκε για πρώτη φορά το 1999. Η PHP 4.0 πλέον βασισμένη με αυτό το συνδυασμό και ένα ευρύ φάσμα πρόσθετων νέων δυνατοτήτων κυκλοφόρησε το 2000 με υποστήριξη σε περισσότερους διακομιστές ιστού, HTTP, εξόδους buffering καθώς και ασφάλεια.

Τέλος, το 2005 κυκλοφόρησε και η πέμπτη έκδοση της γλώσσας με την ονομασία PHP 5.0. Μετά από μεγάλη ανάπτυξη, πολλές δοκιμές και αρκετή προετοιμασία, βασίστηκε στο μοντέλο Zend Engine 2.0.⁵

⁵ (βλ. <http://php.net/manual/en/history.php.php>)

2.3.2 Δομή της PHP

Σ' αυτό το υποκεφάλαιο θα μελετήσουμε κάποια βασικά στοιχεία της γλώσσας PHP καθώς και μερικά απλά παραδείγματα. Με άλλα λόγια θα κατανοήσουμε κάποιες βασικές έννοιες και αρχές της γλώσσας καθώς θα καλύψουμε ορισμένες βασικές λειτουργίες, σε ότι αφορά την βασική σύνταξη. Αναλυτικά, για να γράψουμε κώδικα για μια ιστοσελίδα σε γλώσσα PHP πρέπει να περικλείεται από τις δυο ετικέτες `<?php` και `?>` που δηλώνουν την ετικέτα έναρξης καθώς και την ετικέτα τέλους του εκτελέσιμου κώδικα. Για να έχουμε μια έξοδο ενός μηνύματος θα πρέπει να χρησιμοποιήσουμε την εντολή **echo** αλλιώς την εντολή **print**. Στην συνέχεια ακολουθεί ένα παράδειγμα με τον τρόπο σύνταξης.

```
<!DOCTYPE html>

<html>

<head><title>4th page</title></head>

<body>

<?Php Echo “Hello World.”; ?>

</body>

</html>
```

Πίνακας 1.7 [Παράδειγμα επεξήγησης σύνταξης PHP] (Ελευθέριος Καραγιαννάκης, 2018)

Με μια ανάγνωση θα μπορούσαμε να καταλάβουμε ότι ο κώδικας της γλώσσας PHP αναγράφεται μέσα στις *ετικέτες* `<?php ?>` , καθώς περικλείονται μέσα στον αρχικό κώδικα HTML όπως προαναφέραμε στο προηγούμενο κεφάλαιο. Σαν στόχος της γλώσσας είναι να κάνει δυναμική την ιστοσελίδα HTML. Στο συγκεκριμένο παράδειγμα η εντολή **echo** θα τυπώσει το σχετικό μήνυμα που βρίσκεται μέσα στα εισαγωγικά, άρα η απάντηση που θα δούμε στην οθόνη του χρήστη είναι το μήνυμα:

Hello World.

Σε περίπτωση που θέλουμε να τυπώσουμε δυο μηνύματα διαφορετικά το ένα από το άλλο θα προσθέσουμε ακόμη μια εντολή **echo** καθώς και την ανάλογη ετικέτα.

Επίσης, από την στιγμή που ενσωματώσαμε μια *ετικέτα* κώδικα HTML μπορούμε να ενσωματώσουμε οτιδήποτε επιθυμούμε. Επομένως ο κώδικας θα τροποποιηθεί.

Echo “ Hello ”; Echo “<I> World. </I>”;

Πίνακας 1.8 [Παράδειγμα επεξήγησης εντολή Echo] (Ελευθέριος Καραγιαννάκης, 2018)

Επομένως, η έξοδος που θα λάβει ο χρήστης στην οθόνη του είναι το σχετικό μήνυμα:

Hello World.

Μια γλώσσα προγραμματισμού δεν σταματά εδώ. Οι γλώσσες προγραμματισμού περιλαμβάνουν ένα μεγάλο αριθμό λειτουργιών και δυνατοτήτων που μας βοηθούνε στην καθημερινή μας ζωή. Βασικές αρχές καθώς και ορισμένες έννοιες θα αναφερθούνε ονομαστικά σε αυτό το κεφάλαιο.

Ένα βασικό κόμματι στις γλώσσες προγραμματισμού είναι τα σχόλια και οι παρατηρήσεις. Όταν θέλουμε να σχολιάσουμε ένα συγκεκριμένο κομμάτι κώδικα χρησιμοποιούμε τα σύμβολα */** σχόλιο αρχής και τελειώνουμε με το σχόλιο τέλος **/* καθώς όταν θέλουμε να σχολιάσουμε μια συγκεκριμένη σειρά χρησιμοποιούμε το σύμβολο *//*.

Ωστόσο, οι γλώσσες προγραμματισμού περιλαμβάνουν μεταβλητές, τελεστές αλλά και δομές ελέγχου. Η μεταβλητή στην PHP αποτελείται από ένα λατινικό όνομα, καθώς περιλαμβάνει πάντα πρόθεμα το σύμβολο του δολαρίου (\$). Τα ονόματα των μεταβλητών περιλαμβάνουν *αριθμούς, γράμματα, χαρακτήρα της κάτω παύλας (_), πεζά γράμματα, κεφαλαία γράμματα*. Τέλος, οι μεταβλητές δεν περιλαμβάνουν κενά διαστήματα και ως εντολή τερματισμού χρησιμοποιούν το ελληνικό ερωτηματικό (;). Τυπικά η γλώσσα έχει τους ακόλουθους τύπους δεδομένων:

Integer, Double/Float, Boolean, String

Στην συνέχεια θα ασχοληθούμε με τους τελεστές που χωρίζονται σε τρεις κατηγορίες λογικούς, μαθηματικούς, συγκριτικούς. Αναλυτικά, οι λογικοί τελεστές ξεχωρίζουνε από τα σύμβολα *λογικό και (&&)*, από το *λογικό ή (||)*, από την *λογική άρνηση (!)*. Επίσης, στην γλώσσα PHP μπορούμε να χρησιμοποιήσουμε και τις λατινικές λέξεις **and, or, xor**. Οι μαθηματικοί τελεστές είναι ένα ακόμη σημαντικό κομμάτι για την γλώσσα PHP. Αναλυτικά, οι μαθηματικοί τελεστές ξεχωρίζουνε από

τα σύμβολα πρόσθεση(+), αφαίρεση(-), πολλαπλασιασμός(*), διαίρεση(/), υπόλοιπο
ακέραιας διαίρεσης(%). Τέλος, οι συγκριτικοί τελεστές ξεχωρίζουνε από τα σύμβολα
ισότητας(==), ταυτότητας(===), άρνηση ισότητας(!=), άρνηση ταυτότητας(!==).

Οι δομές ελέγχου, συναρτήσεις, πίνακες, στοίβες είναι η καρδιά κάθε γλώσσας
προγραμματισμού ομοίως και της PHP. Αρχικά χωρίζεται σε δυο κατηγορίες τις δομές
επιλογής, δομές ανακύκλωσης. Όλες οι δομές ελέγχου συμπεριλαμβάνουν την
έκφραση καθώς και την ομάδα επανάληψης.

a) Πρόταση If, Elseif, Else	a. Συναρτήσεις
b) Πρόταση Switch	b. Πίνακες
c) Πρόταση While	c. Δομή Foreach
d) Πρόταση Do/While	d. Στοίβες
e) Πρόταση For	

Τέλος, πιο πάνω στο αριστερό μέλος αναφέρουμε ονοματικά κάποιες βασικές
προτάσεις προγραμματισμού, καθώς στο δεξί μέλος αναφέρουμε τις πιο κύριες έννοιες
που περιλαμβάνει η συγκεκριμένη γλώσσα προγραμματισμού.

3. Ανάλυση Απαιτήσεων

3.1 Κατηγορίες απαιτήσεων

Η ανάλυση απαιτήσεων ενός συστήματος, είναι η πρώτη διαδικασία που πραγματοποιείται πριν ξεκινήσουμε την δημιουργία οποιαδήποτε εφαρμογής. Υλοποιείται όταν ο χρήστης (actor) έρχεται σε συνάντηση με τον διευθυντή έργου και προσπαθεί να κατανοήσει τις απαιτήσεις του πελάτη καθώς να τις μετατρέψει σε λογισμικό.

Περιλαμβάνει 4 φάσεις. Πρώτη φάση είναι η αρχική **σύλληψη ιδέας και ο σχεδιασμός**. Δεύτερη φάση είναι ο **σχεδιασμός και η ανάπτυξη** του λογισμικού. Τρίτη φάση είναι η **υλοποίηση**. Τέταρτη φάση είναι η **λειτουργία και η παράδοση** λογισμικού στον πελάτη. (2002, σ. 52)

Μια ανοργάνωτη προσπάθεια καταγραφής απαιτήσεων είναι σίγουρο πως θα δημιουργήσει μεγάλα προβλήματα στο μέλλον, που θα κοστίσει περισσότερα χρήματα και ώρες εργασίας για την επίλυση του προβλήματος.

3.1.1 Η Απαίτηση

Η Απαίτηση κειμένου είναι το έγγραφο που συνδέει τους παράγοντες (actors) με τον διευθυντή έργου. Είναι η διεργασία που ξεκινά πρώτη και συνεχίζει να εξελίσσεται μέχρι την υλοποίηση της εφαρμογής. Με άλλα λόγια, εξηγεί στην ομάδα έργου τι ακριβώς κάνει το σύστημα καθώς και τι προβλήματα λύνει. Η απαίτηση λογισμικού δεν είναι έγγραφο που αφορά το σχεδιασμό του λογισμικού. Καταγράφεται και τεκμηριώνεται με τέτοιο τρόπο ώστε να είναι κατανοητή απ' όλα τα μέλη της ομάδας. Περιλαμβάνει τις απαιτήσεις του χρήστη για το λογισμικό καθώς και την απαραίτητη προδιαγραφή των απαιτήσεων συστήματος.

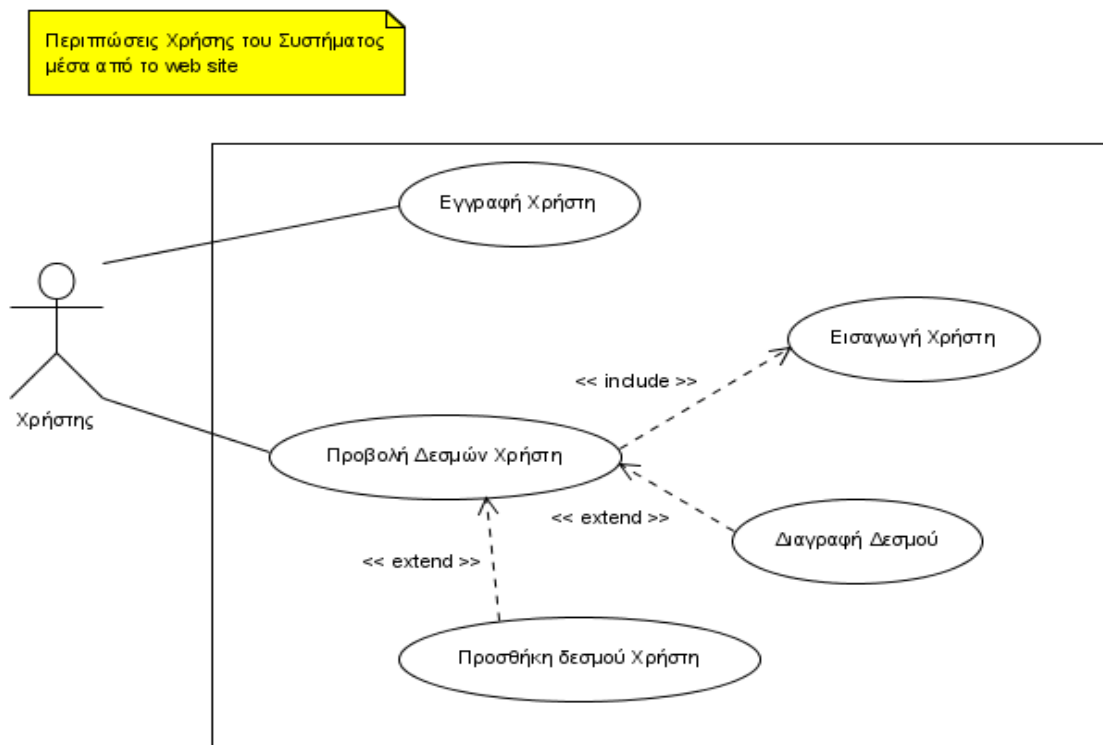
Το παρακάτω τμήμα είναι έγγραφο απαιτήσεων λογισμικού, ενός πληροφορικού συστήματος. Ενώ βλέπει τον κατάλογο με τα διαθέσιμα τμήματα εργαστηρίων: ο συνδεδεμένος χρήστης θα μπορεί να αποσυνδεθεί από το σύστημα.

Για την σύνδεση του χρήστη, ο χρήστης θα εισάγει τα δεδομένα σύνδεσης. Το λογισμικό θα ελέγχει αν τα δεδομένα σύνδεσης ταυτίζονται με τα δεδομένα CAS και σε περίπτωση ταύτισης, θα συνδέει τον χρήστη.

Ο χρήστης θα έχει την δυνατότητα να εγγραφεί σε κάποιο διαθέσιμο τμήμα της επιλογής του.

Ο κωδικός κρυπτογράφησης του κωδικού πρόσβασης θα κρυπτογραφείται σύμφωνα με τα νέα διεθνή πρότυπα.

Τέλος, το παρακάτω σχήμα δείχνει ένα μοντέλο περιπτώσεις χρήστη συστήματος καθώς στο επόμενο υποκεφάλαιο θα αναλύσουμε τους ορισμούς από τις δυο κατηγορίες, λειτουργικές και μη λειτουργικές απαιτήσεις.



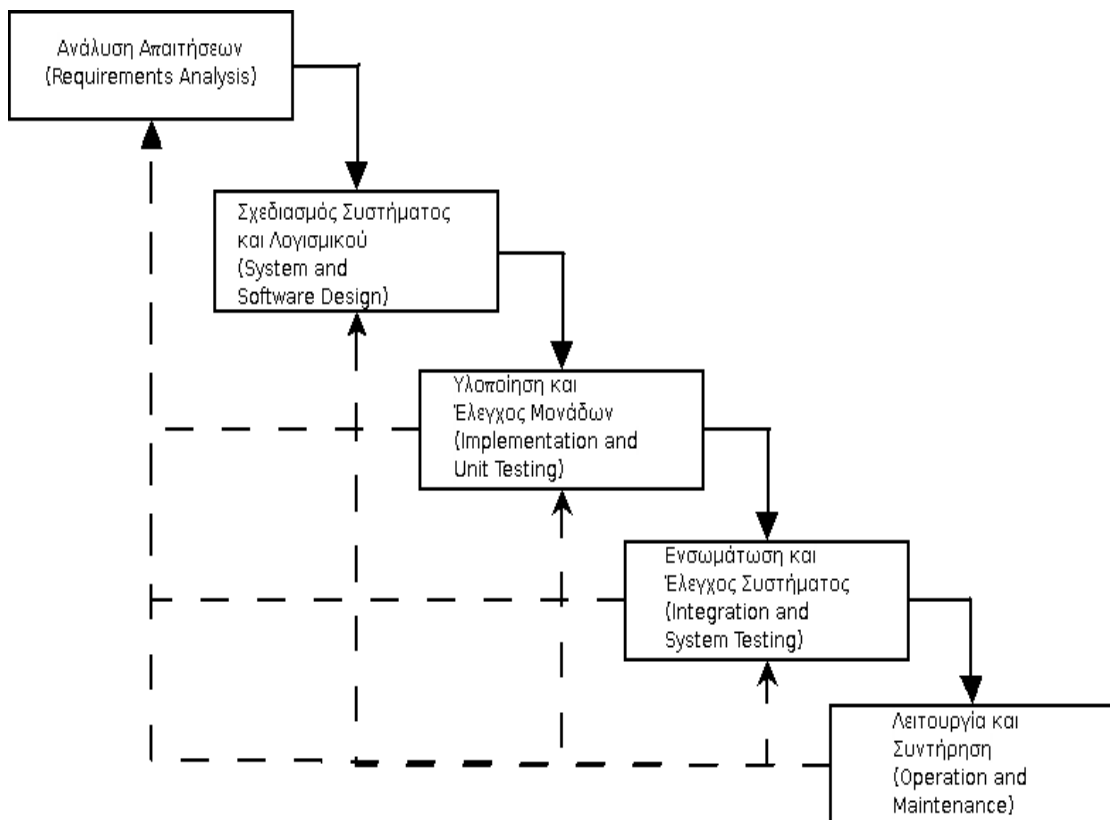
Εικόνα 2. Περιπτώσεις χρήσης συστήματος (<https://drive.draw.io/>)

3.1.2 Λειτουργικές Απαιτήσεις και Μη Λειτουργικές απαιτήσεις

Με τον όρο λειτουργικές απαιτήσεις, περιγράφουμε αναλυτικά την αλληλοεπίδραση του συστήματος βήμα προς βήμα με τους χρήστες (actors). Απαντά στο ερώτημα **τι** κάνει το σύστημα. Με άλλα λόγια οι λειτουργικές απαιτήσεις λειτουργούν και σαν είσοδο συστήματος.

Με τον όρο μη λειτουργικές απαιτήσεις ονομάζουμε τις απαιτήσεις οι οποίες δεν ανήκουν σε κάποια λειτουργία του συστήματος. Με άλλα λόγια οι μη λειτουργικές απαιτήσεις ανήκουν στην κατηγορία που περιγράφουν τις προδιαγραφές που πρέπει να έχει το σύστημα πιο ειδικά περιλαμβάνει τις συμπληρωματικές προδιαγραφές.

Στο παρακάτω σχήμα βλέπουμε το μοντέλου του καταρράκτη, που αποτελεί την παλαιότερη και την ωριμότερη μέθοδο ανάπτυξη λογισμικού από το 1956. Αποτελείται από πέντε φάσεις καθώς πρέπει να τελειώσει οριστικά η κάθε φάση για να περάσει στην επόμενη και αυτό είναι πολύ χρονοβόρο για τον πελάτη.



Εικόνα 3. Μοντέλου του καταρράκτη (goo.gl/6qMF5f)

3.2 Περιπτώσεις Χρηστών

Με βάση της προηγούμενης απαίτησης λογισμικού, δημιουργηθήκανε οι παρακάτω περιπτώσεις χρήστη (Use Case). Οι περιπτώσεις χρήστη παρέχουν μια εξιστόρηση του τρόπου χρήσης του συστήματος. Μια αλληλοεπίδραση παράγει ένα επιθυμητό αποτέλεσμα, με άλλα λόγια ικανοποιεί μια λειτουργική απαίτηση που ονομάζεται Περίπτωση Χρήστη. Οι επόμενες έξι διαδικασίες δημιουργηθήκανε με βάση την απαίτηση λογισμικού, όπου κάθε διαδικασία μοντελοποιείται με ένα συγκεκριμένο use case.

Use case 3.2 Αυθεντικοποίηση Χρήστη:

Use Case 3.2.1 «Σύνδεση Χρήστη»

Use Case 3.2.2 «Σύνδεση Χρήστη στο TCEXAM»

Use Case 3.2.3 «Εγγραφή Χρήστη σε εργαστήριο»

Use Case 3.2.4 «Επικοινωνία Χρήστη»

Use Case 3.2.5 «Αλλαγή γλώσσας Εφαρμογής»

Use Case 3.2.6 «Αποσύνδεση Χρήστη»

3.2.1 Use Case Σύνδεση Χρήστη

Actor: Χρήστης

Αρχική Συνθήκη: Έχει εμφανιστεί ο κατάλογος κατηγορίας και περιέχει μια τουλάχιστον κατηγορία.

Βασική Ροή:

B1α. Ο Χρήστης ζητά να εισαχθεί στο σύστημα, ως φοιτητής.

B1β. Το λογισμικό ζητά από το χρήστη να δώσει τα στοιχεία εισόδου. (υποχρεωτικής συμπλήρωσης) – βλ. Λεξικό Όρων.

Ο κωδικός πρόσβασης εισάγεται μια φορά.

B2α. Ο Χρήστης εισάγει τα παραπάνω δεδομένα.

B2β: Το λογισμικό κάνει το εξής:

B2β-1: Επικυρώνει τα δεδομένα εισαγωγής χρήστη, εφόσον οι επικυρώσεις είναι επιτυχείς. (* βλ. παρακάτω)

B2β-2: Το λογισμικό πραγματοποιεί την είσοδο στον χρήστη.

B2β-3: Το λογισμικό εμφανίζει μήνυμα επιτυχίας («Καλώς Όρισες»).

** Η παραπάνω επικύρωση των δεδομένων εισαγωγής περιλαμβάνει τα εξής:*

A) Το πεδίο (όνομα χρήστη) δεν είναι κενό.

B) Το πεδίο (κωδικός πρόσβασης) δεν είναι κενό.

Γ) Ο κωδικός πρόσβασης εισάγεται μια φορά και είναι συντακτικά ορθός.

Εναλλακτική Ροή:

Εναλλακτική Ροή 1.

E1.1β. Στο παραπάνω βήμα (B2β-1) μια τουλάχιστον επικύρωση είναι ανεπιτυχής.

Το λογισμικό κάνει το εξής:

E1.1β-1: Για κάθε ανεπιτυχής επικύρωση, ειδοποιεί το χρήστη με σχετικό μήνυμα. (** βλ. παρακάτω)

E1.1β-2: Ζητά από το χρήστη να ξαναδώσει τα δεδομένα εισαγωγής χρήστη.

*** Τα μήνυμα για τις ανεπιτυχείς επικυρώσεις είναι τα εξής:*

α) Το πεδίο (όνομα χρήστη) είναι κενό.

Μήνυμα: Το όνομα χρήστη είναι υποχρεωτικό πεδίο.

β) Το πεδίο (κωδικός πρόσβασης) είναι κενό.

Μήνυμα: Ο κωδικός χρήστη είναι υποχρεωτικό πεδίο.

γ) Ο κωδικός πρόσβασης δεν είναι συντακτικά ορθός.

Μήνυμα: Τα στοιχεία ταυτοποίησης που δώσατε δεν μπορούν να επαληθευθούν.

Εναλλακτική ροή 2:

E2.1α. Στο παραπάνω βήμα (B2α) εναλλακτικά ο χρήστης ζητά την ακύρωση της διαδικασίας εισόδου.

E2.1β. Το λογισμικό επαναφέρει την ροή στην αρχική σελίδα.

E2.1γ. Το λογισμικό δεν αποθηκεύει πουθενά τα στοιχεία εισόδου χρήστη.

Τελική Συνθήκη:

T1. Στο τέλος της Βασική Ροή, ο χρήστης έχει αποκτήσει πρόσβαση στο σύστημα, εφόσον έχουν επικυρωθεί τα στοιχεία εισαγωγής από την υπηρεσία CAS.

Λεξιικό όρων:

Δεδομένα σύνδεσης χρήστη =

Όνομα Χρήστη (αριθμός μητρώου) +

Κωδικός Πρόσβασης (ελάχιστο μήκος 6 χαρακτήρες)

3.2.2 Use Case Σύνδεση Χρήστη στο TCEXAM

Actor: Χρήστης

Αρχική Ροή: Έχει εμφανιστεί η επιλογή με ονομασία «Πλατφόρμα Εξέτασης» και περιέχει την εξωτερική εφαρμογή.

Βασική Ροή:

B1α. Ο Χρήστης επιλέγει να συνδεθεί στο TCEXAM.

B1β. Το λογισμικό κάνει το εξής:

B1β-1: Το λογισμικό εμφανίζει την εξωτερική εφαρμογή TCEXAM με τα πεδία:

a) «Όνομα Χρήστη»

b) «Κωδικός»

B1β-2: Το λογισμικό δίνει την δυνατότητα στον χρήστη να κάνει εισαγωγή των δεδομένων του στην πλατφόρμα TCEXAM.

Τελική Συνθήκη:

T1. Στο τέλος της Βασικής Ροής, ο χρήστης έχει πραγματοποιήσει είσοδο στο σύστημα TCEXAM.

3.2.3 Use Case Εγγραφή Χρήστη σε εργαστήριο**Actor: Χρήστης**

Αρχική Ροή: Έχει εμφανιστεί ο κατάλογος εργαστηρίων και περιέχει τουλάχιστον ένα μάθημα με ένα εργαστήριο.

Βασική Ροή:

B1α. Ο Χρήστης επιλέγει να εμφανιστεί ο κατάλογος εργαστηρίων.

B1β. Το λογισμικό κάνει το εξής:

B1β-1: Το λογισμικό εμφανίζει τον κατάλογο εργαστηρίων και τα διαθέσιμα τμήματα.

B1β-2: Το λογισμικό συσχετίζει τα δεδομένα του φοιτητή σε δύο πίνακες.

i. Πίνακας **rel_lab_student**

(id, am, lab_id, misses, grade, reg_month, reg_year, status)

ii. Πίνακας **rel_lab_group**

(lab_id, group_id)

B1β-3: Το λογισμικό εμφανίζει μήνυμα καταχώρησης εργαστηρίου και διαθέσιμο αριθμό ατόμων που έμειναν διαθέσιμα.

Εναλλακτική Ροή:

E1-1β: Στο παραπάνω βήμα (β1α Use Case 3.1.3 εγγραφή χρήστη σε εργαστήριο) μια τουλάχιστον επικύρωση είναι ανεπιτυχής.

E1.1β-1: Το λογισμικό κάνει το εξής:

E1.1β-2: Για κάθε ανεπιτυχής καταχώρηση ακυρώνει την διαδικασία καταχώρησης.

E1.1β-3: Επιτρέπει τον χρήστη να αποσυνδεθεί από το σύστημα.

Τελική Συνθήκη.

T1. Στο τέλος της Βασικής Ροής, ο χρήστης έχει εγγραφεί σε ένα εργαστήριο και αποσυνδέεται από το σύστημα.

3.2.4 Use Case Επικοινωνία Χρήστη

Actor: Χρήστης

Αρχική Συνθήκη: Έχει εμφανιστεί η επιλογή με ονομασία «Επικοινωνία» και περιέχει τουλάχιστον δύο λίστες «επιλογή εργαστηρίου», «επιλογή τμήματος» .

Βασική Ροή:

B1α. Ο χρήστης επιλέγει να επικοινωνήσει με τον καθηγητή.

B1β: Το λογισμικό κάνει το εξής:

B1β-1: Το λογισμικό εμφανίζει την πρώτη λίστα που ο χρήστης επιλέγει εργαστήριο για επικοινωνία με όνομα λίστας « --επιλογή εργαστηρίου-- » .

B1β-2: Το λογισμικό εμφανίζει τη δεύτερη λίστα που ο χρήστης επιλέγει τμήμα για επικοινωνία με όνομα λίστας « επιλογή τμήματος » .

B1β-3: Το λογισμικό εμφανίζει την επιλογή στοιχεία επικοινωνίας:

- a) Όνομα καθηγητή
- b) Βαθμίδα καθηγητή
- c) Τηλέφωνο καθηγητή
- d) Email καθηγητή

Εναλλακτική Ροή:

E1.1β: Στο παραπάνω βήμα (β1α Use Case 3.1.4 Επικοινωνία Χρήστη) μια τουλάχιστον προσπάθεια είναι ανεπιτυχής:

E1.β1-1: Το λογισμικό κάνει το εξής:

E1.β1-2: Για κάθε ανεπιτυχής προσπάθεια το λογισμικό ακυρώνει την διαδικασία.

Τελική Συνθήκη:

T1. Στο τέλος της Βασικής Ροής το λογισμικό εμφανίζει τα στοιχεία καθηγητή και έχει ολοκληρωθεί η επικοινωνία.

3.2.5 Use Case Αλλαγή Γλώσσα Εφαρμογής

Actor: Χρήστης

Αρχική Συνθήκη: Έχει εμφανιστεί ο κατάλογος μενού και περιέχει τουλάχιστον μια κατηγορία καθώς και την επιλογή αλλαγή γλώσσα.

Βασική Ροή:

B1α. Ο χρήστης επιλέγει να αλλάξει την λειτουργία της γλώσσας.

B1β. Το λογισμικό κάνει το εξής:

B1β-1: Το λογισμικό εμφανίζει την δυνατότητα επιλογής γλώσσας.

i.Επιλογή γλώσσας Ελληνική

ii.Επιλογή γλώσσας Αγγλική

B1β-2: Το λογισμικό εμφανίζει την αρχική σελίδα μετά την επιλογή της γλώσσας.

B1β-3: Το λογισμικό επιτρέπει στον χρήστη να συνεχίσει την διαδικασία σύνδεση χρήστη «3.1.1 Use Case Σύνδεση Χρήστη».

Τελική Συνθήκη:

T1. Στο τέλος της Βασική Ροή, ο χρήστης έχει πραγματοποιήσει πρόσβαση στο σύστημα, εφόσον έχουν επικυρωθεί τα στοιχεία του από την υπηρεσία CAS.

3.2.6 Use Case Αποσύνδεση Χρήστη

Actor: Χρήστης

Αρχική Συνθήκη: Έχει εμφανιστεί ο κατάλογος μενού και περιέχει τουλάχιστον μια κατηγορία καθώς και την επιλογή αποσύνδεση χρήστη.

Βασική Ροή:

B1α. Ο χρήστης ζητά να αποσυνδεθεί από το σύστημα, ως φοιτητής.

B1β. Το λογισμικό κάνει το εξής:

B1β-1: Το λογισμικό εμφανίζει το πλήκτρο μενού.

B1β-2: Το λογισμικό εμφανίζει τις επιλογές που έχει ο χρήστης.

B1β-3: Το λογισμικό εμφανίζει το πλήκτρο «Αποσύνδεση».

Εναλλακτική Ροή:

E1.1β: Στο παραπάνω βήμα (β1α Use Case 3.1.6 Αποσύνδεση Χρήστη) μια τουλάχιστον προσπάθεια είναι ανεπιτυχής:

E1.β1-1: Το λογισμικό κάνει το εξής:

E1.β1-2: Για κάθε ανεπιτυχής προσπάθεια το λογισμικό ακυρώνει την διαδικασία.

E1.β1-3: Επιτρέπει στον χρήστη να μη αποσυνδεθεί από το σύστημα.

Τελική Συνθήκη:

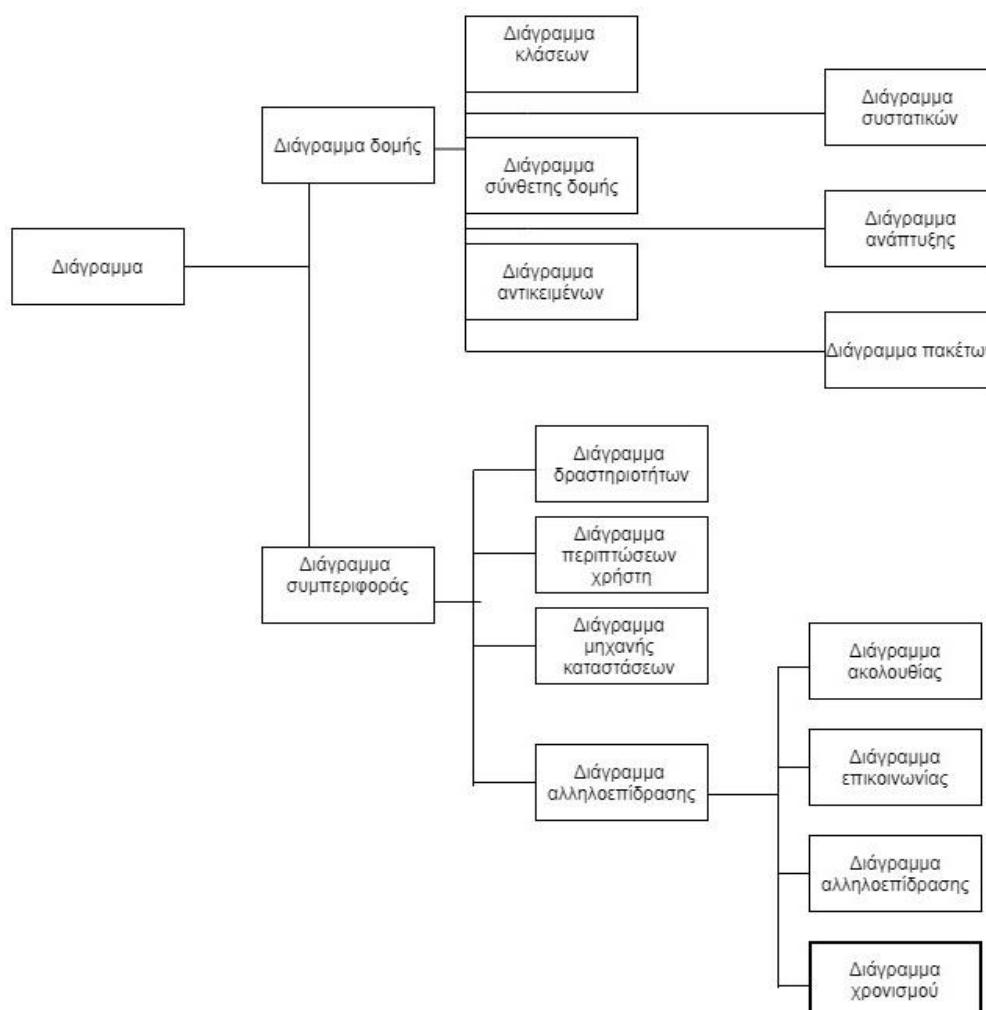
T1. Στο τέλος της Βασική Ροή, ο χρήστης έχει αποσυνδεθεί από το σύστημα καθώς η ροή διαδικασίας επιστρέφει στην αρχική σελίδα.

4. Σχεδιασμός Μηχανισμών Πλατφόρμας

4.1 Σχεδιασμός Αντικειμένων

Ως ορισμό στην έννοια **Ενοποιημένη Γλώσσα Μοντελοποίησης (UML)** ορίζουμε μια οικογένεια γραφικών συμβολισμών που υποστηρίζεται από ένα και μόνο μεταμοντέλο, το οποίο βοηθάει στην περιγραφή και στο σχεδιασμό συστημάτων λογισμικού, ιδιαίτερα σε αντικειμενοστραφή προγραμματισμό.

Η γλώσσα Μοντελοποίησης στην παρούσα φάση περιγράφεται με διάφορους τύπους διαγραμμάτων. Συγκεκριμένα στην Εικόνα 4. παρατηρούμε τις κατηγορίες διαγραμμάτων της UML.



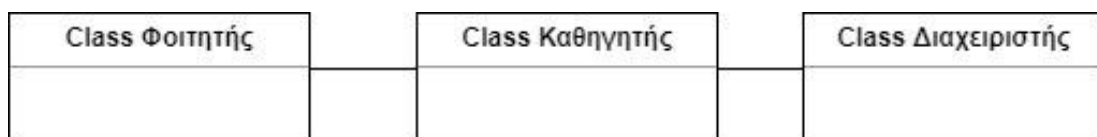
Εικόνα 4. Διαγράμματα της UML (<https://drive.draw.io/>)

Γενικά, τα διαγράμματα της UML διαφέρουν αρκετά μεταξύ τους. Υπάρχουν πολλοί παράγοντες που προσεγγίζουν την ανάπτυξη λογισμικού. Το είδος του συστήματος που αναπτύσσετε, η τεχνολογία που χρησιμοποιείται, το μέγεθος και η κατανομή της ομάδας, οι κίνδυνοι, η συνέχεια της αποτυχίας καθώς και το κλίμα της

ομάδας. Ως αποτέλεσμα, καταλαβαίνουμε ότι δεν υπάρχει μια διαδικασία κατάλληλη για όλα τα έργα. Με άλλα λόγια, πάντα πρέπει να προσαρμόζουμε μια εργασία ώστε να ταιριάζει στο δικό μας συγκεκριμένο περιβάλλον. Για να πετύχει κάτι τέτοιο πρέπει να υπάρχει:

1. Ανάλυση Απαιτήσεων
2. Σχεδιασμός
3. Τεκμηρίωση
4. Κατανόηση του Κώδικα

Από τις πιο βασικές έννοιες στην γλώσσα Μοντελοποίησης είναι τα διαγράμματα κλάσεων. “ Γενικά ένα **διάγραμμα κλάσεων** περιγράφει τους τύπους των αντικειμένων στο σύστημα και τα διάφορα είδη των στατικών σχέσεων που υπάρχουν μεταξύ τους. Τα διαγράμματα κλάσεων απεικονίζουν επίσης τις ιδιότητες και τις λειτουργίες μιας κλάσης, καθώς και τους περιορισμούς που εφαρμόζονται στους τρόπους με τους οποίους συνδέονται τα αντικείμενα. ” (2006, σ. 73) Στην Εικόνα 5 απεικονίζεται ένα απλό διάγραμμα κλάσεων Φοιτητή, Καθηγητή, Διαχειριστή του συστήματος μας.



Εικόνα 5. Απεικόνιση διαγράμματος κλάσεων (<https://drive.draw.io>)

Κάθε περίπτωση χρήστη παριστάνεται με μια έλλειψη, όπου στο κάτω μέρος αναγράφεται ο τίτλος της. (βλ. Εικόνα 6). Αντίθετα ένας Χρήστης παριστάνεται με σκίτσο ανθρώπου. Επίσης στο κάτω μέρος του σκίτσου αναγράφεται η περιγραφή του Actor.



Εικόνα 6. Περιπτώσεις Χρηστών (<https://drive.draw.io/>)

4.1.1 Διάγραμμα Δραστηριότητας

Τα διαγράμματα δραστηριότητας βρίσκονται στο τρίτο επίπεδο της κατάταξης των τύπων διαγραμμάτων της UML όπως παρατηρούμε στην Εικόνα 4. Σκοπός τους είναι μια τεχνική για την περιγραφή της διαδικασιακής λογικής (procedural logic), των επιχειρηματικών διεργασιών και της ροής εργασίας. Με άλλα λόγια έχουν μια βασική διαφορά μεταξύ των διαγραμμάτων δραστηριοτήτων και του συμβολισμού των διαγραμμάτων ροής είναι ότι τα διαγράμματα δραστηριοτήτων υποστηρίζουν παράλληλες συμπεριφορές.

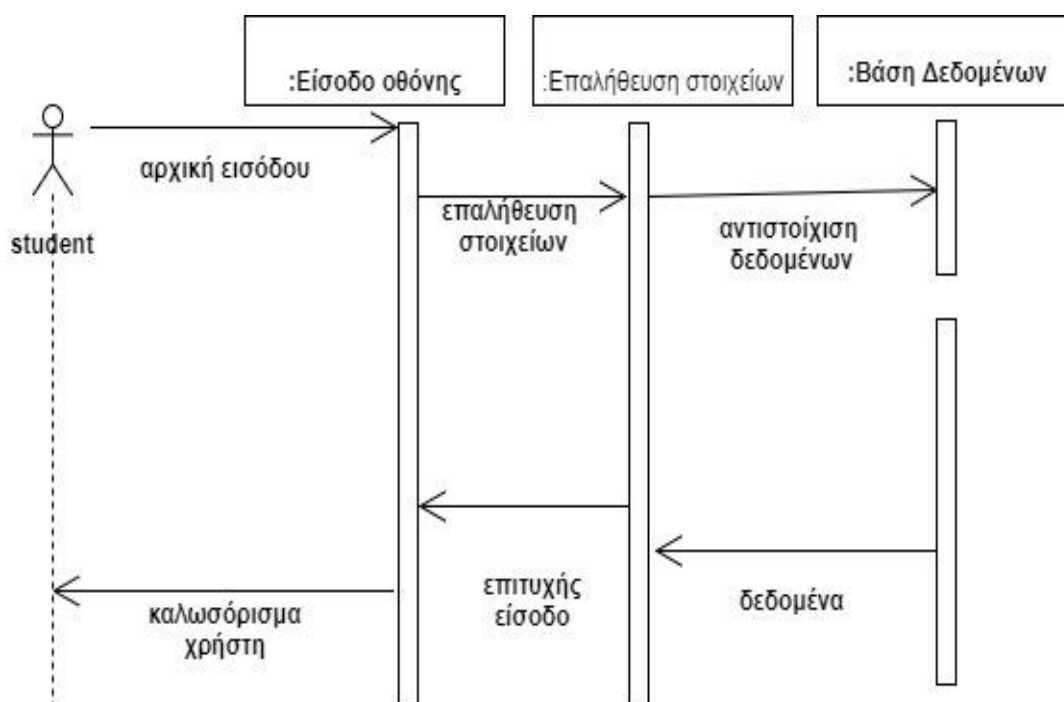
Στη συνέχεια βλέπουμε το διάγραμμα δραστηριοτήτων της εφαρμογής ηλεκτρονικό εργαστήρι. Το διάγραμμα απεικονίζει την είσοδο του χρήστη στην εφαρμογή καθώς και την περίπτωση λάθους.



Εικόνα 7. Διάγραμμα δραστηριοτήτων User Login (<https://drive.draw.io/>)

4.1.2 Διάγραμμα Ακολουθίας

Η UML αρκετές φορές ορίζει μορφές διαγραμμάτων αλληλεπίδρασης. Η πιο συνηθισμένη μορφή είναι το **διάγραμμα ακολουθίας**. Με άλλα λόγια, τα διαγράμματα ακολουθίας περιγράφουν τη συνεργασία μιας ομάδας αντικειμένων μέσα σε κάποια συμπεριφορά. Γενικά ένα διάγραμμα ακολουθίας περιγράφει την συμπεριφορά ενός και μόνου σεναρίου.



Εικόνα 8. Διάγραμμα ακολουθίας εισόδου φοιτητή (<https://drive.draw.io/>)

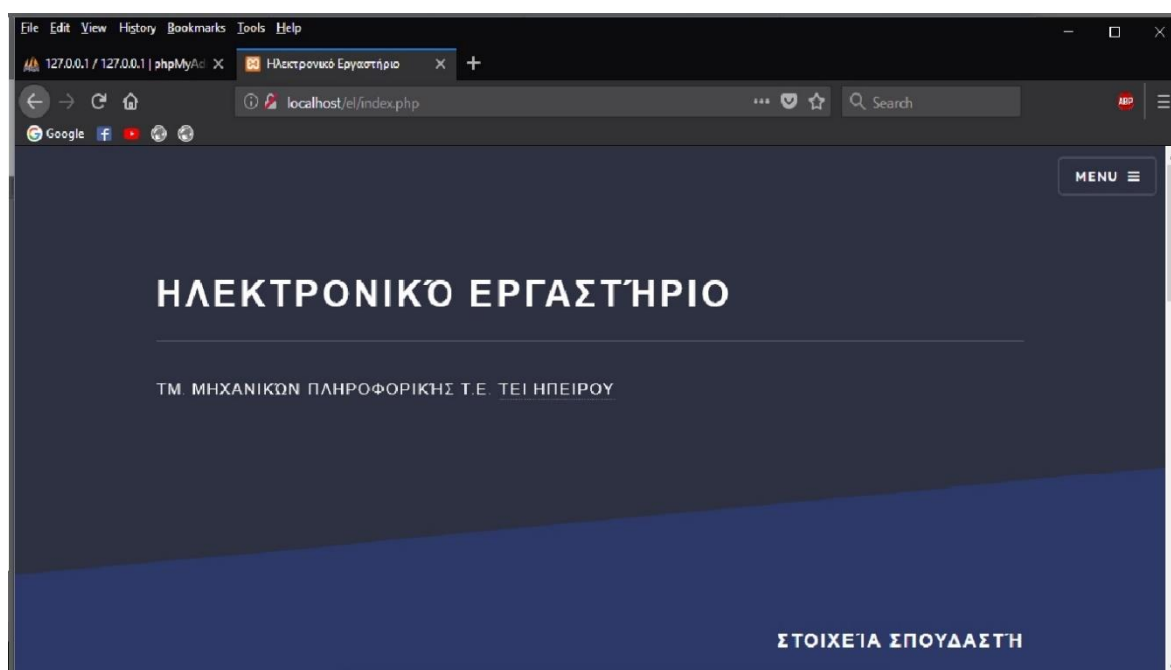
Παρόλα αυτά τα διαγράμματα ακολουθίας ξεχωρίζουν διότι έχουν σαν χαρακτηριστικό την άνω κάτω τελεία : **Όνομα Κλάσης**, όπως παρατηρούμε στην Εικόνα 8. Κάθε γραμμή ζωής έχει ένα πλαίσιο που δείχνει πότε ένας χρήστης (Actor) είναι ενεργός κατά την αλληλοεπίδραση.

Στην Εικόνα 8, παρατηρούμε την αλληλοεπίδραση που γίνεται ανάμεσα στον χρήστη (Actor) και στην προσπάθεια του να κάνει είσοδο στην εφαρμογή. Τα βέλη προς την δεξιά κατεύθυνση μας δείχνουν της κινήσεις του χρήστη καθώς τα αντίστοιχα βέλη την απάντηση που λαμβάνει από το λογισμικό σύστημα.

5. Σχεδίαση και Υλοποίηση Εφαρμογής

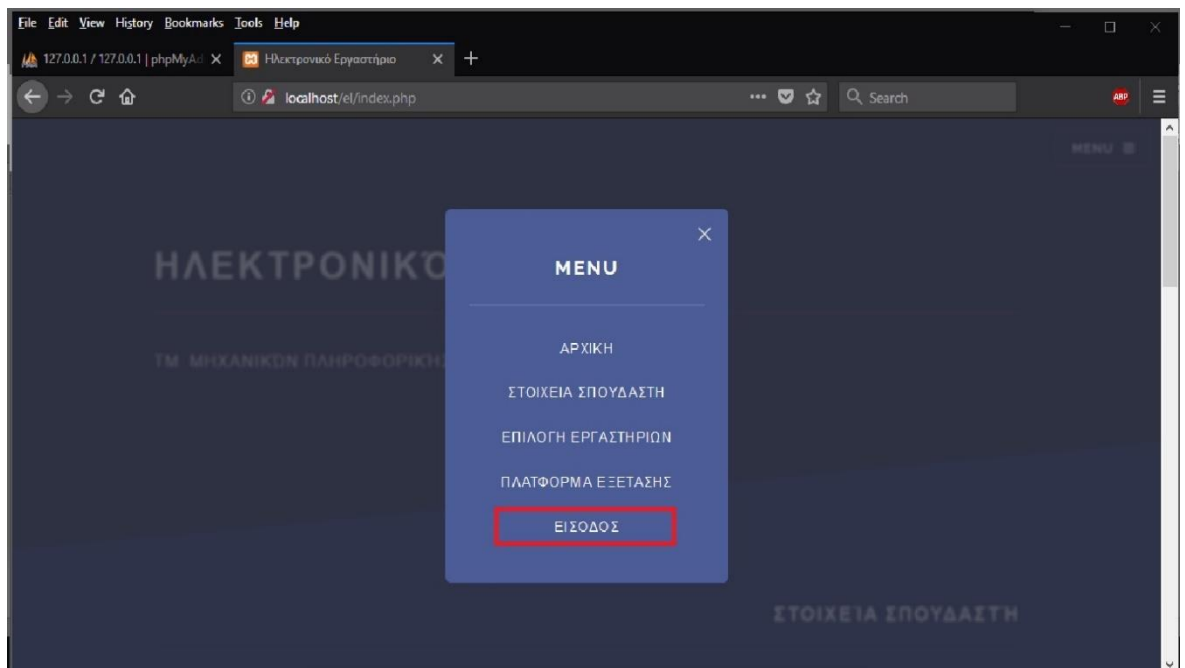
5.1 Οδηγίες Χρήσης Εφαρμογής

Η διαδικτυακή εφαρμογή, ηλεκτρονικό εργαστήριο στην βασική λειτουργία της αποτελείται από τρεις διαφορετικές οντότητες. Οι οντότητες αυτές αντιπροσωπεύουν τα τρία διαφορετικά επίπεδα χρηστών που χρησιμοποιούν την διαδικτυακή εφαρμογή.

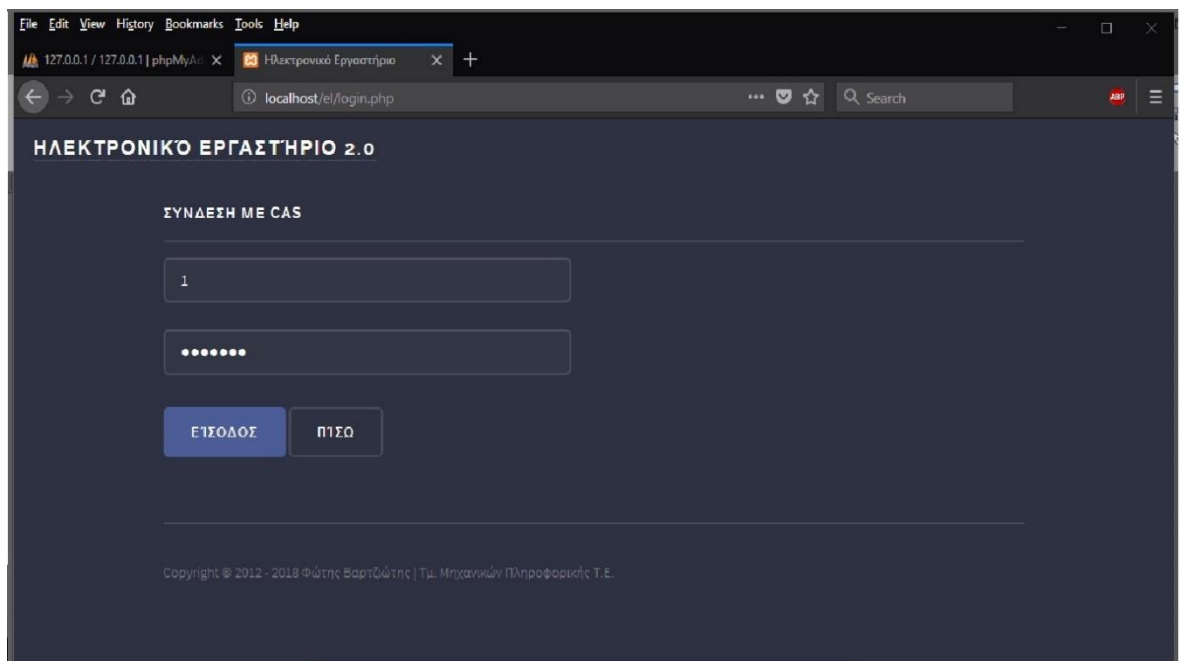


Εικόνα 9. Αρχική σελίδα εφαρμογής (Print Screen)

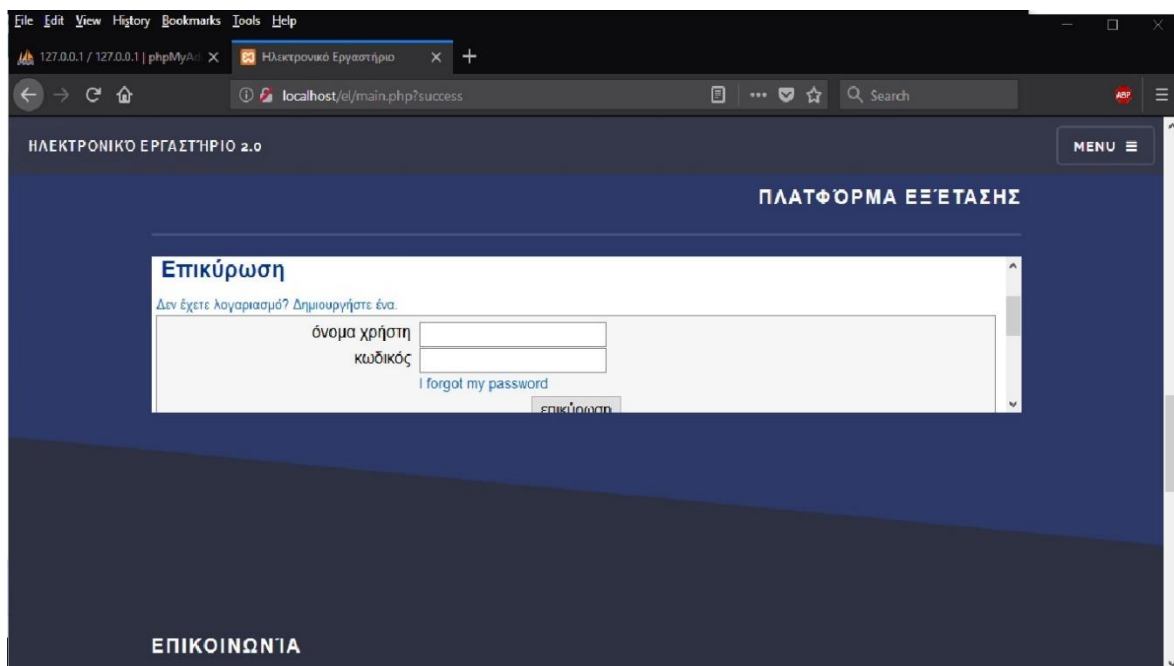
Πιο συγκεκριμένα τα επίπεδα της εφαρμογής είναι τα είναι τα εξής, Φοιτητές, Καθηγητές, Διαχειριστής. Στην Εικόνα 9, απεικονίζεται η αρχική σελίδα της εφαρμογής με την ακόλουθη (Εικόνα 10) να εμφανίζει τις επιλογές της εφαρμογής. Με κύρια επιλογή την Είσοδο στην εφαρμογή ηλεκτρονικό εργαστήριο.



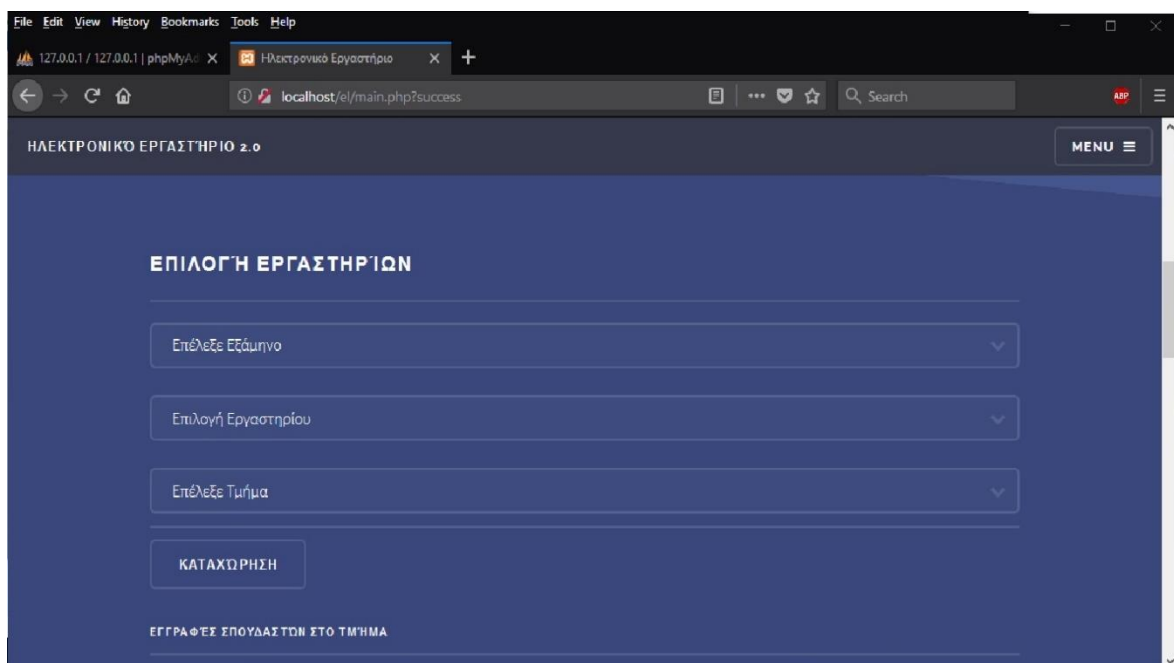
Εικόνα 10. Επιλογές εφαρμογής (Print Screen)



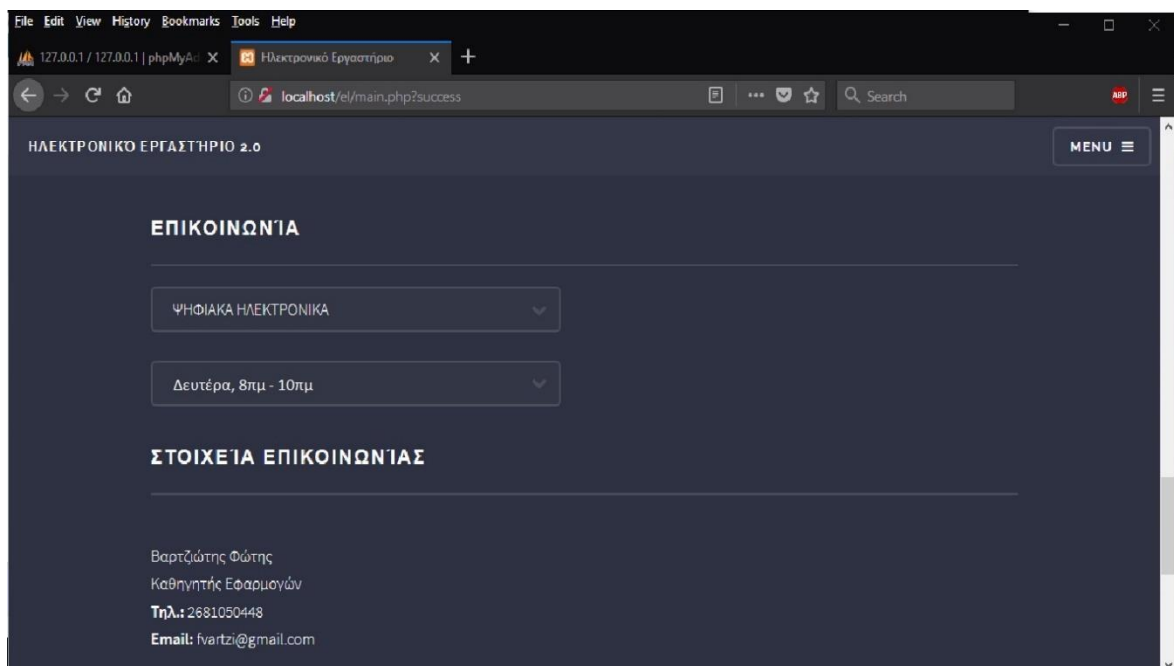
Εικόνα 11. Σύνδεση χρήστη (Print Screen)



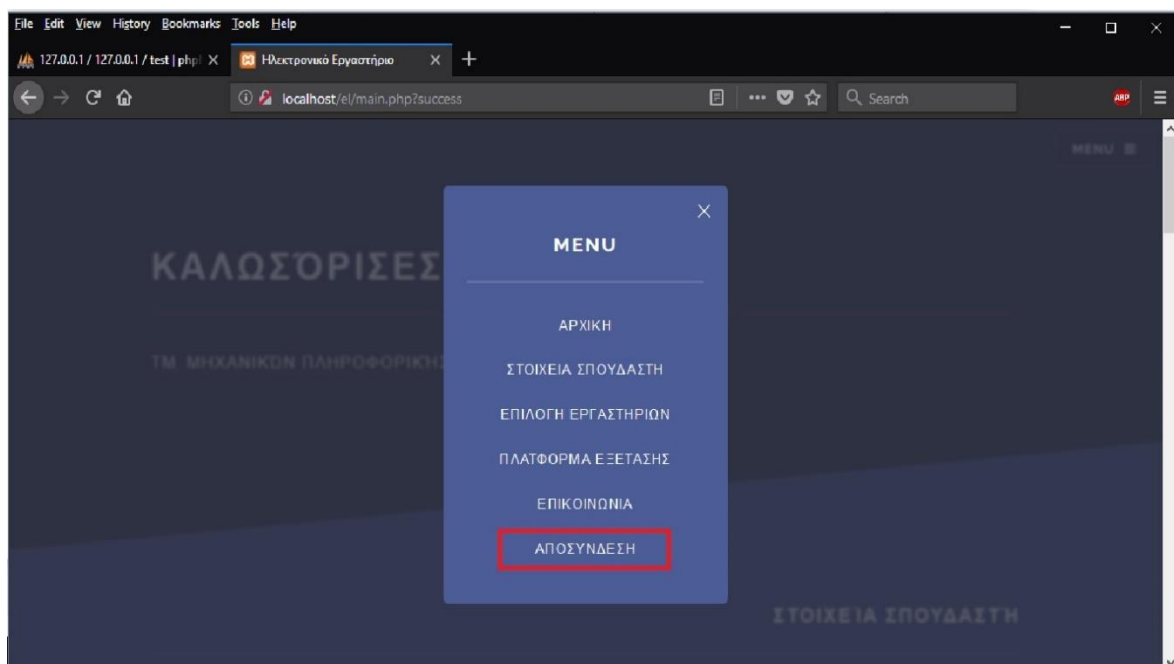
Εικόνα 12. Σύνδεση Χρήστη με εξωτερική εφαρμογή (Print Screen)



Εικόνα 13. Εγγραφή χρήστη σε εργαστήριο (Print Screen)



Εικόνα 14. Επικοινωνία με υπεύθυνο εργαστηρίου (Print Screen)



Εικόνα 15. Αποσύνδεση χρήστη (Print Screen)

«ΦΟΙΤΗΤΗΣ»

Στην κατηγορία φοιτητής, συμπεριλαμβάνονται όλοι οι φοιτητές του ιδρύματος οι οποίοι είναι εγγεγραμμένοι στην γραμματεία της σχολής και έχουν πρόσβαση στις ηλεκτρονικές υπηρεσίες του ΤΕΙ ΗΠΕΙΡΟΥ. Αναλυτικότερα, οι φοιτητές για την εισαγωγή στο ηλεκτρονικό εργαστήριο, χρησιμοποιούνε το ίδιο «Όνομα Χρήστη» και «Κωδικό Πρόσβασης» που χρησιμοποιούνε και στο Πληροφορικό Σύστημα Φοιτητών. Έπειτα γίνεται ο απαραίτητος έλεγχος δεδομένων και εφόσον επαληθευτεί από την κεντρική βάση τότε και μόνο πραγματοποιείται είσοδος στο σύστημα και εμφανίζεται οι προσωπικές πληροφορίες του φοιτητή.

Στην συνέχεια ο φοιτητής είναι ο μόνος υπεύθυνος για την εγγραφή του σε κάποιο διαθέσιμο τμήμα εργαστηρίου. Η επιλογή προς την διαθέσιμη εγγραφή εργαστηρίου είναι ανεξάρτητη από το τυπικό εξάμηνο που φοιτά ο φοιτητής εκείνη την συγκεκριμένη χρονική περίοδο. Σε περίπτωση λανθασμένης εγγραφής ο φοιτητής δεν έχει δικαίωμα διόρθωσης ή διαγραφής του εργαστηριακού τμήματος. Φυσικά τέτοια προβλήματα είναι επιλύσιμα μόνο από τον διαχειριστή του συστήματος και από κανένα άλλο χρήστη.

«ΚΑΘΗΓΗΤΗΣ»

Στην κατηγορία καθηγητής, συμπεριλαμβάνονται όλο το μόνιμο εκπαιδευτικό προσωπικό, οι πανεπιστημιακοί υπότροφοι, καθώς και το έκτακτο εκπαιδευτικό προσωπικό του ιδρύματος οι οποίοι διδάσκουν κάποιο εργαστήριο. Αναλυτικότερα, το διδακτικό προσωπικό του τμήματος για την εισαγωγή στο ηλεκτρονικό εργαστήριο χρησιμοποιεί το «Όνομα Χρήστη» και τον «Κωδικό Πρόσβασης» που τους παρέχει το διαχειριστής. Στην συνέχεια γίνεται ο έλεγχος στοιχείων και εφόσον τα στοιχεία επαληθευτούνε από την κεντρική βάση τότε πραγματοποιείται είσοδος στο σύστημα.

Ως κύριο μέλημα, το διδακτικό προσωπικό είναι υπεύθυνο για την δημιουργία εργαστηρίων στα μαθήματα που διδάσκουν. Οι καθηγητές έχουν την πλήρη πρόσβαση των εργαστηρίων. Με άλλα λόγια είναι υπεύθυνοι για την εισαγωγή διαθέσιμων τμημάτων καθώς και για την αφαίρεση αυτών, αν αυτή είναι απαραίτητη. Επίσης, ελέγχουν τις διαθέσιμες θέσεις των τμημάτων τους. Μετά την καταχώρηση του φοιτητή

στο διαθέσιμο εργαστήριο ο καθηγητής μπορεί να του καταχωρήσει την αντίστοιχη βαθμολογία καθώς και να ελέγχει τις απουσίες του αν επιθυμεί.

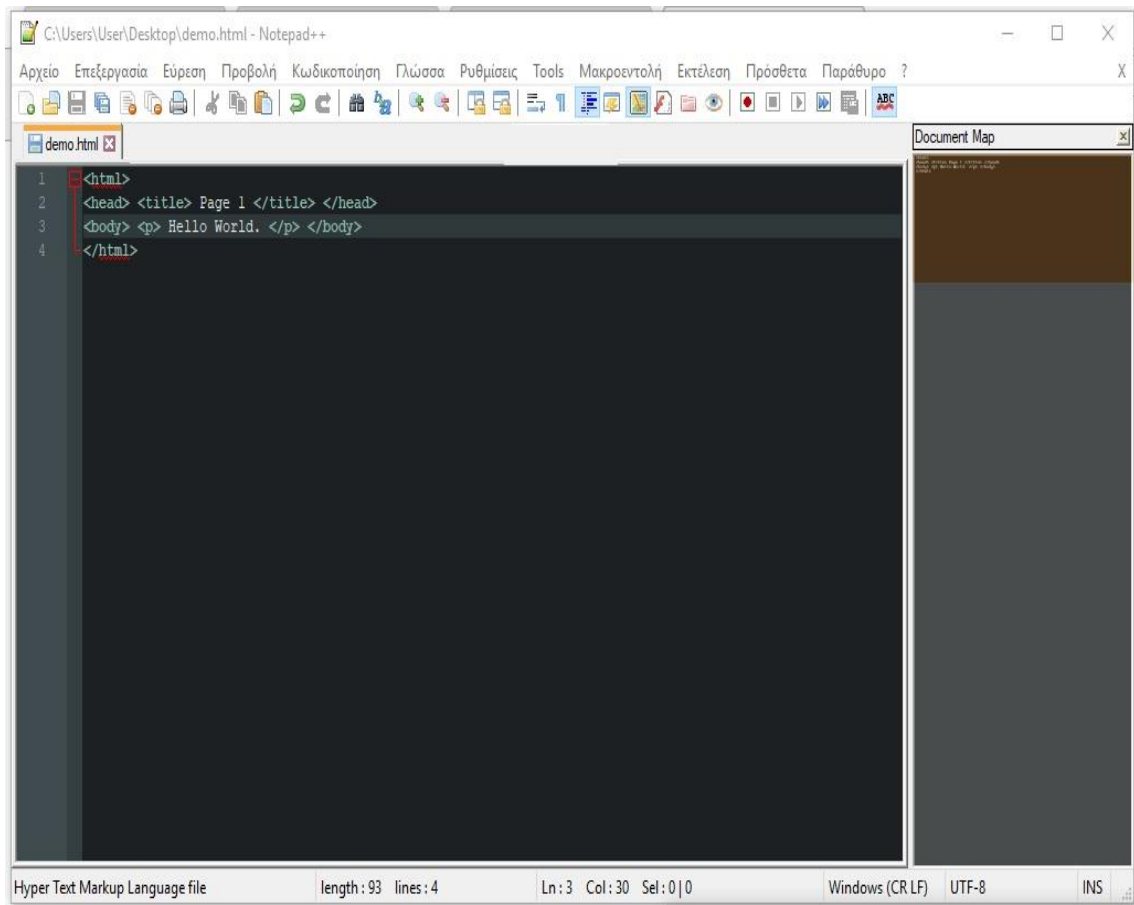
«ΔΙΑΧΕΙΡΙΣΤΗΣ»

Στην κατηγορία διαχειριστής, ανήκει η ομάδα που είναι υπεύθυνη για την βασική λειτουργία της διαδικτυακής εφαρμογής ηλεκτρονικό εργαστήριο. Αποτελείται από μια ομάδα χρηστών οι οποίοι είναι υπεύθυνη για το λειτουργικό κομμάτι της εφαρμογής. Επεμβαίνει σε ζητήματα που είναι απαραίτητα όπως διαγραφής φοιτητή από εσφαλμένη καταχώρηση τμήματος, γρήγορης αντιμετώπιση κάποιων προβλημάτων καθώς και στην συντήρηση του συστήματος.

Ο διαχειριστής είναι πιστοποιημένος από το σύστημα καθαρά για λόγους ασφαλείας. Η ομάδα δεν έχει καμία σχέση με την κοινότητα των φοιτητών ή των καθηγητών του συστήματος. Η είσοδος του διαχειριστή γίνεται με διαφορετική εφαρμογή στο σύστημα καθώς ο ρόλος του είναι καθαρά συντονιστικός και συμβουλευτικός για την ομαλή λειτουργία του συστήματος.

5.2 Περιβάλλον Ανάπτυξης Εφαρμογής

Το περιβάλλον ανάπτυξης της εφαρμογής που χρησιμοποιήθηκε, ονομάζεται **Notepad++**. Ο επεξεργαστής κειμένου **Notepad++** είναι ένα ελεύθερο λογισμικό το οποίο υποστηρίζεται από επεξεργασία καρτελών. Δίνεται διαθέσιμο για εγκατάσταση στην ηλεκτρονική διεύθυνση (βλ. <https://notepad-plus-plus.org>). Το περιβάλλον του



Εικόνα 16. Το Λογισμικό εφαρμογής Notepad++

Δίνει την πρόσβαση στον προγραμματιστή να εργαστεί σε πολλά αρχεία ταυτόχρονα κάνοντας χρήση ενός παραθύρου. Το λογισμικό είναι γραμμένο σε γλώσσα προγραμματισμού C++ καθώς επιτυγχάνει μεγάλες επιδόσεις σε μικρό μέγεθος προγράμματος. Δημιουργός του κειμενογράφου είναι ο Don Ho. Η κυκλοφορία του λογισμικού πραγματοποιήθηκε στις 25 Σεπτεμβρίου 2003 ως εφαρμογή για το λειτουργικό σύστημα Windows.

Οι δυνατότητες που κάνουν τον κειμενογράφο ξεχωριστό είναι ότι διαθέτει “Εκτύπωση σύνταξης, αναδίπλωση κώδικα και περιορισμένη συμπλήρωση για προγραμματισμό, δέσμες ενεργειών και γλώσσες σήμανσης.” (βλ. <https://en.wikipedia.org/wiki/Notepad%2B%2B>)

Τέλος, υπάρχουν άλλες δυο επιλογές που κάνουν τον επεξεργαστή κειμένου **notepad++** ξεχωριστό. Το πρώτο στοιχείο επιλογής αναφέρεται στο απλό κείμενο “

Normal text ” καθώς η δευτερεύουσα επιλογή αναφέρεται στο στυλ κειμένου “ MS-DOS Style ” δίνοντας μια ψευδαίσθηση στο χρήστη ότι δουλεύει σε περιβάλλον κειμενογράφου της εποχής DOS.

6. Παρατηρήσεις

6.1 Συμπεράσματα

Αρχικά σκοπός της πτυχιακής εργασίας ήταν η δημιουργία μιας διαδικτυακής εφαρμογής όπου εξυπηρετεί τους φοιτητές στην δήλωση εργαστηρίων. Γι' αυτό το λόγο δημιουργήθηκε το ηλεκτρονικό εργαστήριο.

Πιο συγκεκριμένα, δημιουργήθηκε εξ' ολοκλήρου νέα διεπαφή χρήστη που κάνει την εφαρμογή προσιτή σε όλους. Επίσης, μέσω της εργασίας ασχολήθηκα με την βελτιστοποίηση του κώδικα. Με άλλα λόγια κατασκευάστηκε μια πιο γρήγορη και αποδοτική εφαρμογή για όλες τις πλατφόρμες. Αναλυτικότερα, έδωσα βάση στη δομή του συστήματος όπου και προστέθηκαν νέοι μηχανισμοί.

Στη συνέχεια ασχολήθηκα με τις κατηγορίες απαιτήσεων, ειδικότερα γράφτηκε εξ' αρχής μια νέα απαίτηση λογισμικού όπου και δημιουργήθηκαν νέες λειτουργικές και μη λειτουργικές απαιτήσεις που δημιουργήσανε νέες περιπτώσεις χρηστών.

Εν κατακλείδι, μελετήθηκε και σχεδιάστηκε ο νέος μηχανισμός αντικειμένων που βοήθησε στην υλοποίηση των διαγραμμάτων δραστηριότητας, ακολουθίας, σύνδεσης. Συμπερασματικά, η συγκεκριμένη εργασία ήταν μια ευκαιρία να διευρύνει το γνωστικό μου πεδίο σχετικά με το αντικείμενο που δεν είναι άλλο, αυτό του προγραμματισμού και την σχεδίαση της εφαρμογής.

6.2 Μελλοντικές Αναβαθμίσεις

Η διαδικτυακή εφαρμογή ηλεκτρονικό εργαστήριο, δεν αποτελεί εμπορικό προϊόν στην τρέχουσα έκδοση. Υπάρχει ένας ικανοποιητικός αριθμός βελτιώσεων που θα μπορούσε να δώσει στην εφαρμογή άλλη τροπή. Οι βελτιώσεις αφορούν την συντήρηση της εφαρμογής όσο και την εγκατάσταση του λογισμικού.

Όσο αναφορά την τρέχουσα έκδοση θα μπορούσε να υλοποιηθεί:

1. Χρήση γραφημάτων

Η χρήση γραφημάτων αποτελεί μια σημαντική βελτίωση της εφαρμογής διότι ο αρμόδιος καθηγητής – διαχειριστής θα μπορεί να βλέπει τα στατιστικά του εργαστηρίου καθώς και πιο κεφάλαιο της ύλης οι μαθητές δείχνανε περισσότερο ενδιαφέρον. Έτσι κάνοντας το μάθημα κάθε χρόνο και καλύτερο.

2. Χρήση φύλλο εργασίας Excel

Το φύλλο εργασίας Excel απευθύνεται στου καθηγητές του εργαστηρίου. Ειδοποιεί τους διαχειριστές του τμήματος ότι το διαθέσιμο τμήμα έχει γεμίσει καθώς αυτόματα τυπώνετε το αρχείο Excel που συμπεριλαμβάνει τα ονόματα των φοιτητών που έκαναν εγγραφή στο συγκεκριμένο τμήμα, όπως και ο αριθμός μητρώου κάθε φοιτητή.

3. Χρήση ίδια εμφάνιση στο admin panel

Η χρήση ίδιας εμφάνισης εφαρμογής, στο admin panel βοηθά στην καλύτερη διαχείριση εργαστηρίων, εξοικονομώντας χρόνο, καθώς μεγαλώνει την αποδοτικότητα της εφαρμογής.

4. Εργαλείο επαναφοράς αντιγράφου

Το εργαλείο επαναφοράς αντιγράφου ασφαλείας, βοηθά σε περίπτωση ανάγκης οι καθηγητές να μπορούν να κάνουν ανάκτηση των δεδομένων από την βάση δεδομένων. Μια τέτοια υλοποίηση ανεβάζει την αποδοτικότητα της εφαρμογής καθώς την κάνει πιο ευέλικτη στους διαχειριστές.

7. Παραρτήματα

Παράρτημα Α. Κώδικας Αρχικής Σελίδας Ηλεκτρονικό Εργαστήριο

```
<!DOCTYPE HTML>
<html>
<head>
<title>Ηλεκτρονικό Εργαστήριο</title>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <link rel="stylesheet" href="assets/css/main.css" />
</head>
<body>
<div id="page-wrapper">
  <header id="header" class="alt">
<h1><a href="index.php">Ηλεκτρονικό Εργαστήριο 2.0</a></h1>
<nav>
<a href="#menu">Menu</a>
  </nav>
</header>
  <nav id="menu">
<div class="inner">
<h2>Menu</h2>
  <ul class="links">
    <li><a href="index.php">ΑΡΧΙΚΗ</a></li>
    <li><a href="#one">ΣΤΟΙΧΕΙΑ ΣΠΟΥΔΑΣΤΗ</a></li>
    <li><a href="#three">ΠΛΑΤΦΟΡΜΑ ΕΞΕΤΑΣΗΣ</a></li>
    <li><a href="login.php">ΕΙΣΟΔΟΣ</a></li>
  </ul>
  <a href="#" class="close">Close</a>
</div>
</nav>
<section id="banner">
<div class="inner">
<h2>Ηλεκτρονικό Εργαστήριο</h2>
<p>Τμ. Μηχανικών Πληροφορικής Τ.Ε.<a href="http://www.teiep.gr">ΤΕΙ
ΗΠΕΙΡΟΥ</a></p>
</div>
</section>
<section id="wrapper">
<section id="one" class="wrapper alt spotlight style6">
<div class="inner">
<div class="content">
<h2 class="major">Στοιχεία Σπουδαστή</h2>
<p>Συνδεθείτε μέσω CAS για να ξεκινήσετε.</p> </div>
```

```
</div>
</section>
<section id="three" class="wrapper spotlight style6">
<div class="inner"> <div class="content">
<h2 class="major">Πλατφόρμα Εξέτασης</h2>
<iframe src="http://sonetto.teiep.gr/tcexam" width="100%" height="100%" /></iframe>
</div></div>
</section>
</section>
<section id="footer">
<div class="inner">
<ul class="copyright">
<li>Copyright &copy; 2012 - <?php echo date("Y");?> Φώτης Βαρτζιώτης | Τμ.
Μηχανικών Πληροφορικής Τ.Ε.</li> </ul></div>
</section></div>
</body>
</html>
```

BIBΛΙΟΓΡΑΦΙΑ

ΕΛΛΗΝΙΚΗ ΒΙΒΛΙΟΓΡΑΦΙΑ

- E.1) Παπασπύρου, Ν. & Σκορδαλάκης, Ε., 2002. *Μεταγλωττιστές*. Αθήνα: ΣΥΜΜΕΤΡΙΑ
- E.2) ΓΙΑΜΑΛΙΔΟΥ, Μ., 2002. *Διαχείριση Έργου*. Αθήνα: ΚΡΙΤΙΚΗ
- E.3) Μάντακας, Μ., 2008. *Πληροφορικά Συστήματα Επιχειρήσεων*. Άρτα:
- E.4) Μάντακας, Μ., 2015. *Τεχνολογία Λογισμικού*. Άρτα: mmantak@gmail.com

ΑΓΓΛΙΚΗ ΒΙΒΛΙΟΓΡΑΦΙΑ

- A.1) Jesse, Liberty, 1997. *Εγχειρίδιο της C++*. Αθήνα: Μ. ΓΚΙΟΥΡΔΑΣ
- A.2) Julie, Meloni, 2012. *Μάθετε Php, MySQL και Apache*. Αθήνα: Μ. ΓΚΙΟΥΡΔΑΣ
- A.3) Martin, Fowler, 2004. *Εισαγωγή στην UML*. Αθήνα: ΚΛΕΙΔΑΡΙΘΜΟΣ
- A.4) Deitel, P., Deitel, H., 2010. *Java Προγραμματισμός*. Αθήνα: Μ. ΓΚΙΟΥΡΔΑΣ
- A.5) Sommerville, I, 2009. *Βασικές αρχές τεχνολογίας λογισμικού*. Αθήνα: ΚΛΕΙΔΑΡΙΘΜΟΣ

ΗΛΕΚΤΡΟΝΙΚΗ ΒΙΒΛΙΟΓΡΑΦΙΑ

- H.1) Πρεζεράκος, Γ., 2014. *Καταχώρηση βλαβών και συμβάντων σε επίπεδο δήμου μέσω διαδικτυακής εφαρμογής*. 2014. Πτυχιακή εργασία. ΤΕΙ Πειραιά. Διαθέσιμο στο: <http://oceanis.lib.teipir.gr/xmlui/handle/123456789/2263> [Πρόσβαση στις 12 Οκτώβρη 2017]
- H.2) Wikipedia.org, 2017. Cascading Style Sheets [online] Διαθέσιμο στο: https://en.wikipedia.org/wiki/Cascading_Style_Sheets [Πρόσβαση στις 23 Οκτώβρη 2017]
- H.3) Wikipedia.org, 2017. CSS [online] Διαθέσιμο στο: <https://el.wikipedia.org/wiki/CSS> [Πρόσβαση στις 23 Οκτώβρη 2017]
- H.4) The Php Group, 2001-2017. *History of PHP*. [online] Διαθέσιμο στο: <http://php.net/manual/en/history.php.php> [Πρόσβαση στις 23 Οκτώβρη 2017]
- H.5) The PHP Group, 2001-2017. *Unsupported Historical Releases*. [online] Διαθέσιμο στο: <http://php.net/releases/> [Πρόσβαση στις 23 Οκτώβρη 2017]
- H.6) Dmoz.org, 2017. *PHP*. [online] Διαθέσιμο στο: <http://dmztools.net/Computers/Programming/Languages/PHP/> [Πρόσβαση στις 23 Οκτώβρη 2017]
- H.7) Techteam.gr, 2002-2017. *PHP*. [online] Διαθέσιμο στο: <http://www.techteam.gr/wiki/PHP> [Πρόσβαση στις 23 Οκτώβρη 2017]
- H.8) Wikipedia.org, 2017. *PHP* [online] Διαθέσιμο στο: <https://el.wikipedia.org/wiki/PHP> [Πρόσβαση στις 23 Οκτώβρη 2017]
- H.9) Τσούλος, Ι., 2016. *MySQL – PHP*. 2016. Ηλεκτρονικές Σημειώσεις. ΤΕΙ Ηπείρου. Διαθέσιμο στο: http://myweb.teleinfom.teiep.gr/gtsoulos/public_html//ecommerce/php1.pdf [Πρόσβαση στις 24 Οκτώβρη 2017]
- H.10) Manifesto, 2001. *Μανιφέστο για ευέλικτη ανάπτυξη λογισμικού*. [online] Διαθέσιμο στο: <http://agilemanifesto.org/iso/el/manifesto.html> [Πρόσβαση στις 13 Δεκεμβρίου 2017]

- H.11) IEEE, 1998. *Recommended practice for software requirements specification*. [online] Διαθέσιμο στο: <http://ieeexplore.ieee.org/document/720574/?reload=true> [Πρόσβαση στις 13 Δεκεμβρίου 2017]
- H.12) Wikipedia.org, 2017. *Notepad++*. [online] Διαθέσιμο στο: <https://en.wikipedia.org/wiki/Notepad%2B%2B> [Πρόσβαση στις 15 Ιανουαρίου 2018]
- H.13) Βεσκούκης, Β. 2000. *Τεχνολογία Λογισμικού*. [online] Διαθέσιμο στο: <ftp://150.140.208.250/share/ebooks/tecnologia/Greek%20Computer%20Ebooks/Tecnologia%20Logismikou%20II.pdf>