Διατμηματικό Πρόγραμμα Μεταπτυχιακών Σπουδών (Π.Μ.Σ.)

«Πληροφορική και Υπολογιστική Βιοϊατρική»

Ροή: Πληροφορικής

Διπλωματική εργασία

Ποιμενίδη Δέσποινα

ΘΕΜΑ

Δημιουργία διαδικτυακής εφαρμογής για τη διαχείριση και ασφάλεια μεγάλου όγκου δεδομένων σε δημόσια επιχείρηση.

Επιβλέπον Καθηγητής: κ. Γεώργιος Σταμούλης
Επιστημονικός Σύμβουλος: κ. Ευάγγελος – Δωρόθεος Αγγέλης

(Λαμία 2018 ®)

«Υπεύθυνη Δήλωση μη λογοκλοπής και ανάληψης προσωπικής ευθύνης»

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, και γνωρίζοντας τις συνέπειες της λογοκλοπής, δηλώνω υπεύθυνα και ενυπογράφως ότι η παρούσα εργασία με τίτλο «Δημιουργία διαδικτυακής εφαρμογής για τη διαχείριση και ασφάλεια μεγάλου όγκου δεδομένων σε δημόσια επιχείρηση» αποτελεί προϊόν αυστηρά προσωπικής εργασίας και όλες οι πηγές από τις οποίες χρησιμοποίησα δεδομένα, ιδέες, φράσεις, προτάσεις ή λέξεις, είτε επακριβώς (όπως υπάρχουν στο πρωτότυπο ή μεταφρασμένες) είτε με παράφραση, έχουν δηλωθεί κατάλληλα και ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο/Η ΔΗΛΩΝ/-ΟΥΣΑ

Ημερομηνία

Υπογραφή

Πίνακας περιεχομένων

Εικόνες

Πίνακες

ΠΕΡΙΛΗΨΗ

Στην παρούσα διπλωματική εργασία παρουσιάζεται η διαδικασία σχεδιασμού και ανάπτυξης μίας διαδικτυακής εφαρμογής την οποία θα εκμεταλλευτεί μια Δημόσια Επιχείρηση. Σκοπός του συγγραφέα είναι να περιγράψει τη διαδικασία ανάπτυξης και σχεδιασμού συστηματικά η οποία θα καταλήξει στην παραγωγή μιας λειτουργικής και αξιόπιστης εφαρμογής που θα καλύπτει όσο το δυνατόν τις απαιτήσεις που ορίζονται.

Οι εφαρμογές ιστού έχουν γίνει πολύ κρίσιμες για τις περισσότερες επιχειρήσεις στη σημερινή κοινωνία ιδιαίτερα στο ανταγωνιστικό περιβάλλον. Πολύ μακριά είναι εκείνες οι ημέρες, όταν ο μόνος τρόπος για να αποκτήσει πρόσβαση και να χρησιμοποιήσει κάποιος τις εφαρμογές ήταν μέσω της εγκατάστασης στον υπολογιστή του. Εδώ είναι μια βαθύτερη ματιά στις εφαρμογές ιστού για να διαπιστώσει τα οφέλη τους στις αναπτυσσόμενες επιχειρήσεις.

Στο πλαίσιο της εργασίας θα περιγράφουν έννοιες που συμβάλουν στην αποτελεσματικότερη ανάπτυξη και την βελτιστοποίηση μιας εφαρμογής web. Θα γίνει πρακτική εφαρμογή της ασφάλειας στα δεδομένα και θα αναλυθεί βήμα προς βήμα η δημιουργία μιας εφαρμογής ιστού.

Λέξεις κλειδιά: Εφαρμογές Ιστού, PHP, AJAX, Ασφάλεια, MD5, AES-256-BCB, Ασφάλεια λογισμικού, Αλγόριθμοι ασφάλειας, Κύκλος ζωής λογισμικού

ABSTRACT

This diplomatic thesis presents the process of designing and developing an online application that will be exploited by the Public Power Services Corporation. The aim of the author is to describe the process of systematically development and design, which will result in the production of a functional and reliable application, covering as much as possible the defined requirements.

Web applications have become very critical for most of the businesses in today's society, especially in the competitive environment. Far are those days which the only way to access and use the apps were by installing them on your computer. Here is a deeper look at web applications, seeing their benefits in developing businesses.

This work describes concepts that contribute to more effective development and optimization of a web application. A practical implementation of data security will be developed and a web application will be analyzed step-by-step.

Keywords: Web Application, PHP, AJAX, Security, MD5, AES-256-BCB, Security Algorithms, Web Security, Software life cycle

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα διπλωματική εργασία με θέμα «Δημιουργία διαδικτυακής εφαρμογής για τη διαχείριση και ασφάλεια μεγάλου όγκου δεδομένων σε δημόσια επιχείρηση», πραγματοποιήθηκε στο πλαίσιο της διπλωματικής εργασίας του Τμήματος Πληροφορικής με Εφαρμογές στη Βιοϊατρική (ΤΠΕΒ) του Πανεπιστημίου Θεσσαλίας (ΠΘ) σε συνεργασία με το Τμήμα Πληροφορικής (ΤΠ) της Σχολής Θετικών Επιστημών (ΣΘΕ) του Πανεπιστημίου Θεσσαλίας με τίτλο «Πληροφορική και Υπολογιστική Βιοϊατρική» στον τομέα της «Ασφάλειας Υπολογιστικών και Τηλεπικοινωνιακών Συστημάτων, της Διαχείρισης Μεγάλου Όγκου Δεδομένων και της Προσομοίωσης (ροή Πληροφορικής)».

Στο σημείο αυτό αισθάνομαι την ανάγκη να εκφράσω τις ειλικρινείς και θερμές ευχαριστίες μου σε όσους συνέβαλαν στην ολοκλήρωση αυτής της προσπάθειας:

Πρώτα απ' όλα, θα ήθελα να εκφράσω τις ευχαριστίες μου στον επιβλέποντα καθηγητή κ. Γεώργιο Σταμούλη για την ευκαιρία που μου έδωσε να παρακολουθήσω το συγκεκριμένο μεταπτυχιακό. Του είμαι ευγνώμων επειδή παρακολουθώντας τα μαθήματα εξελίχθηκα γνωστικά και αυτό έχει αντίκτυπο και στην επαγγελματική μου πορεία. Την δυνατότητα που μου έδωσε να πραγματοποιήσω την διπλωματική εργασία και την ευκαιρία που μου έδωσε να ασχοληθώ με το αντικείμενο της ανάπτυξης λογισμικού και την εφαρμογή της ασφάλειας για την προστασία των πληροφοριών.

Εν συνεχεία, θα ήθελα να ευχαριστήσω τον επιστημονικό μου σύμβουλο κ. Ευάγγελο – Δωρόθεο Αγγέλη, για τις εύστοχες παρατηρήσεις και τις πολύτιμες επιστημονικές συμβουλές του καθ' όλη τη διάρκεια της διατριβής. Πέρα από αυτό όμως, είναι ένας καθηγητής που με βοήθησε γενικότερα στην πορεία του μεταπτυχιακού μου βίου να κατανοήσω σε βάθος έννοιες που αφορούσαν το αντικείμενο των σπουδών.

Τέλος, θα ήθελα να ευχαριστήσω, όλους τους καθηγητές του Μεταπτυχιακού Διπλώματος Ειδίκευσης της ροής στην Πληροφορική για τις πολύτιμες γνώσεις που μου προσέφεραν.

Ποιμενίδη Δέσποινα

Κεφάλαιο 1 - Εισαγωγή

1.1 Γενικά

Μια εφαρμογή ιστού (web application) είναι ένα λογισμικό πελάτη – διακομιστή (client-server) που χρησιμοποιεί ένα πρόγραμμα περιήγησης (web browsers) για την εκτέλεση εργασιών μέσω του Διαδικτύου [1]. Οποιοσδήποτε μπορεί να έχει πρόσβαση στην εφαρμογή Ιστού από οποιονδήποτε υπολογιστή συνδεδεμένο στο διαδίκτυο χρησιμοποιώντας τυπικά προγράμματα περιήγησης. Πολλοί άνθρωποι χρησιμοποιούν web εφαρμογές στην καθημερινότητά τους.

Από τη άλλη πλευρά, εκατομμύρια είναι και οι επιχειρήσεις που χρησιμοποιούν το Διαδίκτυο ως ένα οικονομικά αποδοτικό κανάλι επικοινωνίας. Τους επιτρέπει να ανταλλάσσουν πληροφορίες και να πραγματοποιούν γρήγορες και ασφαλείς συναλλαγές. Ωστόσο, η αποτελεσματική δέσμευση είναι δυνατή μόνο όταν η επιχείρηση είναι σε θέση να καταγράψει και να αποθηκεύσει όλα τα απαραίτητα δεδομένα και να έχει ένα μέσο για την επεξεργασία αυτών των πληροφοριών και την παρουσίαση των αποτελεσμάτων στον χρήστη.

Οι εφαρμογές Web χρησιμοποιούν ένα συνδυασμό scripts από την πλευρά του διακομιστή (PHP και ASP) για να διαχειριστούν την αποθήκευση και την ανάκτηση των πληροφοριών και client-side-scripts (JavaScript και HTML) για την παρουσίαση πληροφοριών στους χρήστες [2]. Αυτό επιτρέπει στους χρήστες να αλληλεπιδρούν με τη εταιρεία χρησιμοποιώντας ηλεκτρονικές φόρμες, συστήματα διαχείρισης περιεχομένου και άλλα.

1.1.1 Πως λειτουργεί μια εφαρμογή στο Web

Οι εφαρμογές Web κωδικοποιούνται συνήθως, σε γλώσσα που υποστηρίζεται από προγράμματα περιήγησης, όπως JavaScript και HTML, καθώς, αυτές οι γλώσσες βασίζονται στο πρόγραμμα περιήγησης για να καταστήσουν εκτελέσιμο το πρόγραμμα. Ορισμένες από τις εφαρμογές είναι δυναμικές, απαιτώντας επεξεργασία από την πλευρά του διακομιστή.

Η διαδικτυακή εφαρμογή απαιτεί έναν διακομιστή ιστού (web server) να διαχειρίζεται αιτήματα από τον πελάτη, έναν διακομιστή εφαρμογών (application server) για την εκτέλεση των ζητούμενων εργασιών και ορισμένες φορές μια βάση δεδομένων (database) για την

αποθήκευση των πληροφοριών. Η τεχνολογία διακομιστή εφαρμογών (application server technology) κυμαίνεται από ASP.NET, ASP και ColdFusion έως PHP και JSP [2].

Μία τυπική ροή εφαρμογών ιστού είναι:

- Ο χρήστης ενεργοποιεί ένα αίτημα στον διακομιστή ιστού (web server) μέσω του Διαδικτύου, είτε μέσω ενός προγράμματος περιήγησης ιστού είτε μέσω της διεπαφής χρήστη (application's user interface) της εφαρμογής.
- Ο διακομιστής Web διαβιβάζει αυτό το αίτημα στον κατάλληλο διακομιστή εφαρμογών ιστού.
- Ο Web application server εκτελεί την απαιτούμενη εργασία, για παράδειγμα ερώτηση στην βάση δεδομένων ή επεξεργασία στη βάση δεδομένων και στη συνέχεια, δημιουργεί τα αποτελέσματα των ζητούμενων δεδομένων.
- Ο Web application server στέλνει τα αποτελέσματα στον web server με τις ζητούμενες πληροφορίες ή τα επεξεργασμένα δεδομένα.
- Ο web server ανταποκρίνεται στον πελάτη (client) με τις ζητούμενες πληροφορίες που εμφανίζονται στην οθόνη του χρήστη.

1.1.2 Οφέλη από μια εφαρμογή Web

Τα πλεονεκτήματα που προσφέρει μία web εφαρμογή είναι [3]:

- Οι εφαρμογές Web εκτελούνται σε πολλαπλές πλατφόρμες ανεξάρτητα από το λειτουργικό σύστημα ή την συσκευή, εφόσον το πρόγραμμα περιήγησης είναι συμβατό.
- Όλοι οι χρήστες έχουν πρόσβαση στην ίδια έκδοση, εξαλείφοντας τυχόν προβλήματα συμβατότητας.
- Δεν είναι εγκατεστημένο στον σκληρό δίσκο, εξαλείφοντας έτσι τους περιορισμούς χώρου.
- Μειώνουν την πειρατεία λογισμικού σε εφαρμογές ιστού που βασίζονται σε συνδρομές.
- Μειώνουν το κόστος για την επιχείρηση όσο και για τον τελικό χρήστη, καθώς απαιτείται λιγότερη υποστήριξη και συντήρηση από την επιχείρηση και χαμηλότερες απαιτήσεις για τον υπολογιστή του τελικού χρήστη.

1.2 Κίνητρα για την διεξαγωγή της Διπλωματικής Εργασίας

Η ανάπτυξη των τεχνολογιών της πληροφορικής οδήγησε τους πάσης φύσεως και μεγέθους οργανισμούς στο να αναπτύξουν πληροφοριακά συστήματα που να βασίζονται σε αυτές. Οι εφαρμογές που υπάρχουν και συνεχώς πληθαίνουν στο internet, είναι ένα μέσο για την ικανοποίηση των αναγκών που κάθε επιχείρηση αλλά και άνθρωπος θέλει να καλύψει. Συνεπώς, η παρούσα εφαρμογή θα δημιουργηθεί για να καλύψει κάποιες από τις ανάγκες μιας επιχείρησης. Ο στόχος και ο σκοπός της εργασίας αναλύονται παρακάτω.

1.3 Στόχος και Σκοπός της Εργασίας

Στόχος της παρούσας εργασίας είναι η ανάλυση, η σχεδίαση και η ανάπτυξη μίας διαδικτυακής εφαρμογής για την διαχείριση μεγάλου όγκου πληροφοριών μιας επιχείρησης. Συγκεκριμένα, πρόκειται για μια εφαρμογή η οποία θα λαμβάνει ένα εύρος πληροφοριών από τους πιστοποιημένους χρήστες, οι οποίοι είναι οι υπάλληλοι της εταιρείας αυτής. Η εφαρμογή αυτή θα ονομάζεται StaticISU και θα επιτρέπει στους υπαλλήλους να ενημερώνουν την βάση του συστήματος καθημερινά με απώτερο σκοπό την επεξεργασία των πληροφοριών. Στην απλούστερη μορφή η εφαρμογή θα λειτουργεί σαν ένα μέρος αποθήκευσης (backup), ώστε να διασφαλίζονται από τυχόν εισβολή τα δεδομένα.

Ένα τμήμα του τομέα του δημόσιου οργανισμού για τον οποίο αναπτύσσεται η εφαρμογή είναι υπεύθυνο για την εκτέλεση καθημερινών, ανά μήνα ή μια φορά τον χρόνο κάποιων εργασιών (αλυσίδων) που είναι απαραίτητες για την λειτουργία του. Μετά από συνεννόηση των διοικητικών μελλών αποφασίστηκε πως είναι χρήσιμη η γέννηση μιας εφαρμογής που να αποθηκεύει, να διαχειρίζεται και να προβάλει τις πληροφορίες που είδη έχουν εκτελεστεί από το πρόγραμμα Sap ISU[1]. Αυτό σημαίνει πως καθημερινά πιστοποιημένοι χρήστες θα επισκέπτονται την εφαρμογή και θα την ενημερώνουν. Μετέπειτα θα μπορούν να ανατρέξουν στις αποθηκευμένες πληροφορίες και να εξάγουν στατιστικά αποτελέσματα με σκοπό να συγκρίνουν δεδομένα (επόμενο στάδιο). Οι εργαζόμενοι θα έχουν το δικαίωμα να συνδεθούν από οποιοδήποτε υπολογιστή και η πιστοποίηση θα γίνεται με βάση το όνομα χρήστη και τον κωδικό πρόσβασης.

[1] Το SAP IS-U είναι η συγκεκριμένη λύση της SAP για τη βιομηχανία κοινής ωφελείας.

Αναλυτικότερα, οι πληροφορίες που θα καταχωρούν οι χρήστες θα αποθηκεύονται σε έναν διακομιστή. Οι χρήστες θα εκτελούν ασύγχρονα αιτήματα στον διακομιστή, μέσω Ajax request, για να υλοποιηθούν οι ενέργειες που επιθυμούν.

Ο σκοπός αφορά την ενημέρωση της διοίκησης, την παρακολούθηση βασικών λειτουργιών και τη ελαχιστοποίηση σφαλμάτων. Ειδικά, ο σκοπός εστιάζει στη λειτουργικότητα, στην διασφάλιση των δεδομένων από εξωγενής επιθέσεις και την εύκολη χρήση της εφαρμογής.

1.4 Η Δομή της Εργασίας

Η δομή της εργασίας θα περιγραφεί αναφορικά με βάση τα κεφάλαια που αναλύονται σε αυτή:

- Κεφάλαιο **1 -** είναι εισαγωγικό κεφάλαιο που παρουσιάζεται ο στόχος, ο σκοπός και η δομή που θα πραγματοποιηθεί.
- Κεφάλαιο **2 -** ασχολείται με την εκτενή αναφορά των χρησιμοποιούμενων τεχνολογιών που απαιτούνται για την ανάπτυξη της εφαρμογής web.
- Κεφάλαιο **3 -** αναφέρεται στην κρυπτογραφία και την ασφάλεια του λογισμικού. Περιγράφονται οι χρησιμοποιούμενες στην εφαρμογή μέθοδοι κρυπτογράφησης.
- Κεφάλαιο **5** - καταγράφει την ανάλυση απαιτήσεων της υπό ανάπτυξη εφαρμογής. Προσδιορίζονται οι ρόλοι των χρηστών καθώς και οι λειτουργικές απαιτήσεις του συστήματος.
- Κεφάλαιο **6** - παρουσιάζεται ο σχεδιασμός της εφαρμογής. Αρχικά, σχεδιάζονται οι λειτουργίες του συστήματος και ύστερα η διεπαφή του συστήματος με το χρήστη και τον διαχειριστή.
- Κεφάλαιο **7** - περιγράφει την υλοποίηση της υπό ανάπτυξη εφαρμογής και την αλληλεπίδραση των λειτουργιών του συστήματος μεταξύ τους. Εδώ γίνεται εφαρμογή των κρυπτογραφικών αλγορίθμων.
- Κεφάλαιο **8** - πραγματοποιείται η εγκατάσταση της εφαρμογής σε έναν εξυπηρετητή και εξετάζεται η λειτουργικότητά της.
- Κεφάλαιο **9** - συνοψίζει το περιεχόμενο της παρούσας εργασίας.

Κεφάλαιο 2 - Εργαλεία Ανάπτυξης Λογισμικού

1.1 Εισαγωγή

Για την διεξαγωγή της διπλωματικής εργασίας χρησιμοποιήθηκαν διάφορες τεχνολογίες λογισμικού.

Η ανάπτυξη της εφαρμογής βασίστηκε σε online server. Το σύστημα αναπτύχτηκε σε υπολογιστή με λειτουργικό σύστημα Windows, ωστόσο το τελικό προϊόν είναι ανεξάρτητο από λειτουργικό σύστημα.

Για την ευκολότερη ανάπτυξη της εφαρμογής χρησιμοποιήθηκε για web design framework το bootstrap[2]. Το Bootstrap είναι ένα **αρθρωτό**, ελεύθερο και ανοικτού κώδικα front-end πλαίσιο (βιβλιοθήκη) για το σχεδιασμό ιστοσελίδων και εφαρμογών web [4]. Περιέχει πρότυπα σχεδίασης HTML και CSS για τυπογραφία, καθώς και προαιρετικές επεκτάσεις JavaScript.

Το framework περιλαμβάνει εκατοντάδες στοιχεία για την σχεδίαση μιας διαδικτυακής σελίδας, τα οποία ως επί το πλείστων είναι responsive, περιλαμβάνοντας grid system, buttons, forms, panels, navigation bars και πολλά ακόμα. Με τον όρο responsive, εννοείται ότι η διάταξη των ιστοσελίδων προσαρμόζεται δυναμικά, λαμβάνοντας υπόψη τα χαρακτηριστικά της συσκευής που χρησιμοποιείται είτε αυτό είναι υπολογιστής, είτε tablet είτε κινητό τηλέφωνο. Με αυτό τον τρόπο το περιβάλλον της εφαρμογής είναι φιλικό προς τον χρήστη ανεξάρτητα από τη συσκευή που χρησιμοποιεί.

1.2 PHP

Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής είναι η PHP, η οποία είναι μία διαδεδομένη **γλώσσα προγραμματισμού γενικής χρήσης**, που είναι κατάλληλη για προγραμματισμό **διαδικτυακών εφαρμογών** και μπορεί να εισαχθεί εύκολα σε **HTML**. Αρχικά η PHP ήταν ακρώνυμο του Personal Home Page (προσωπική αρχική σελίδα), αλλά άλλαξε σύμφωνα με τη σύμβαση GNU και τώρα είναι ακρώνυμο του PHP Hypertext Preprocessor (προεπεξεργαστής κειμένου PHP). Η PHP δημιουργήθηκε αρχικά από τον Rasmus Lerdorf το 1994 [5].

[2] Learn the bootstrap grid https://tutorialzine.com/2015/10/learn-the-bootstrap-grid-in-15-minutes

Είναι μια ευρέως διαδεδομένη γλώσσα προγραμματισμού με μεγάλη δημοτικότητα που συνεχώς αυξάνεται η χρησιμότητά της. Είναι πλούσια σε χαρακτηριστικά γνωρίσματα που καθιστούν το σχεδιασμό και τον προγραμματισμό του Web ευκολότερο. Μια σελίδα PHP επεξεργάζεται από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (για παράδειγμα Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, το οποίο θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML [6].

Η γλωσσική σύνταξη της PHP είναι παρόμοια με τη σύνταξη της C, έτσι όποιος έχει εμπειρία με τη C, θα είναι άνετος με την PHP. Η PHP είναι πραγματικά απλούστερη από τη C επειδή δεν χρησιμοποιεί μερικές από τις δυσκολότερες έννοιες της C. Επίσης, η PHP δεν περιλαμβάνει τις χαμηλού επιπέδου ικανότητες προγραμματισμού της C επειδή έχει σχεδιαστεί για προγραμματισμό ιστοσελίδων και δεν απαιτεί εκείνες τις ικανότητες.

Η PHP είναι ιδιαίτερα ισχυρή στη δυνατότητά της να αλληλεπιδρά με τις βάσεις δεδομένων και μπορεί να χειρίζεται τη σύνδεση με τη βάση δεδομένων και την επικοινωνία με αυτήν. Για παράδειγμα, σε μια ιστοσελίδα που είναι απαραίτητη η εγγραφή των χρηστών, η PHP μπορεί να αποθηκεύει τα ονόματα και τους κωδικούς τους σε μια βάση δεδομένων.

Χρησιμοποιείται όχι για την αισθητική διαμόρφωση μιας σελίδας, αλλά για τον **χειρισμό** των **λειτουργιών** και **εργασιών** που θα διεκπεραιώνει. Συνεπώς, ο κώδικας που γράφεται για μια ιστοσελίδα σε γλώσσα PHP δεν γίνεται **άμεσα αντιληπτός** αλλά μετά από την **παρέμβαση** του χρήστη στην ιστοσελίδα.

Δεν είναι αναγκαίο να γνωρίζουμε τις τεχνικές λεπτομέρειες για τη σύνδεση με μια βάση δεδομένων ή για την ανταλλαγή των μηνυμάτων με αυτή. Λέμε στην PHP το όνομα της βάσης δεδομένων και που βρίσκεται, και η PHP χειρίζεται τις λεπτομέρειες [1] [5]. Συνδέεται με τη βάση δεδομένων, περνά τις οδηγίες μας σε αυτή και επιστρέφει την απάντηση της βάσης δεδομένων σε μας.

1.2.1 Πλεονεκτήματα της PHP

Η PHP είναι πολύ αποτελεσματική και μπορεί να εξυπηρετήσει εκατομμύρια επισκέψεις καθημερινά για αυτό και θεωρείται ένα από τα πιο δημοφιλή λογισμικά στην υλοποίηση web sites. Κάποιοι από τους βασικούς ανταγωνιστές της PHP είναι η Perl, η ASP (Microsoft Active Server Pages) και η JSP (Java Server Pages) [7].

Πλεονεκτήματα PHP:

- Το βασικό πλεονέκτημα της PHP είναι ότι λειτουργεί δυναμικά.
- Τα αποτελέσματα που παράγει, αλλάζουν σύμφωνα με τις ανάγκες του χρήστη.
- Ο δυναμικός τρόπος λειτουργίας εφαρμόζεται ακόμα και μέσα στο εσωτερικό της PHP.
- Έχει τη δυνατότητα να αλλάζει τον τύπο των μεταβλητών δυναμικά, σύμφωνα με τα δεδομένα που κάθε χρονική στιγμή είναι αποθηκευμένα σε αυτές.
- Διασυνδέσεις με πολλά διαφορετικά συστήματα βάσεων δεδομένων όπως PostgreSQL, mSQL, Oracle, dbm, filePro, Informix, InterBase, Sybase, κ.α.
- Έχει ενσωματωμένες βιβλιοθήκες για πολλές συνηθισμένες διαδικασίες διαδικτύου.
- Χαμηλό κόστος - Παρέχεται δωρεάν.
- Ευκολία μάθησης και χρήσης. Η σύνταξη της Php βασίζεται σε άλλες γλώσσες προγραμματισμού, βασικά στη C και στην Perl.
- Φορητότητα - Είναι διαθέσιμη σε πολλά λειτουργικά συστήματα.
- Διαθεσιμότητα του κώδικα προέλευσης.
- Είναι γρήγορη. Ο χρόνος απόκρισης είναι μικρός διότι βασίζεται σε HTML κώδικα.
- Είναι ασφαλής. Ο χρήστης δεν βλέπει τον PHP κώδικα.

1.3 MariaDB

Η **MariaDB** είναι ένα σύστημα διαχείρισης σχεσιακής βάσης ανοικτού κώδικα (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]. Η MariaDB μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Το πρόγραμμα τρέχει έναν εξυπηρετητή (server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων.

Επειδή είναι ανοικτού κώδικα (open source), οποιοσδήποτε μπορεί να κατεβάσει την **MariaDB** και να την διαμορφώσει σύμφωνα με τις ανάγκες του και σύμφωνα πάντα με την γενική άδεια που υπάρχει. Ξεκίνησε σαν κλάδος (fork) της MySQL από τον ιδιοκτήτη της, όταν αυτή πουλήθηκε στην Oracle. Μέχρι την έκδοση 5.5 παρείχε όλες τις δυνατότητες που είχαν και οι αντίστοιχες εκδόσεις της MySQL.

Επιτρέπει την αναζήτηση, την ανάκτηση, τη ταξινόμηση και την αποθήκευση των δεδομένων αποτελεσματικά. Ο **MariaDB** διακομιστής ελέγχει τη πρόσβαση στα δεδομένα

για να μπορούν να δουλεύουν πολλοί χρήστες ταυτόχρονα, για να παρέχει γρήγορη πρόσβαση και να διασφαλίζει ότι μόνο πιστοποιημένοι χρήστες μπορούν να έχουν πρόσβαση [9]. Συνεπώς η **MariaDB** είναι ένας πολυνηματικός διακομιστής πολλαπλών χρηστών.

1.4 HeidiSQL

Ο τρόπος σύνδεσής και επεξεργασίας μιας ή περισσοτέρων βάσεων δεδομένων που χρησιμοποιήθηκε στην εφαρμογή έγινε μέσο του προγράμματος HeidiSQL [10]. Αντίστοιχα προγράμματα για αυτό τον σκοπό είναι το phpMyAdmin [11], το MySQL Workbench [12] και το SSH [13]. Άλλα γνωστά εργαλεία είναι τα Toad for MySQL και το Sequel Pro.

Το "Heidi" επιτρέπει την προβολή και επεξεργασία δεδομένων και δομών από υπολογιστές που εκτελούν ένα από τα συστήματα βάσεων δεδομένων MySQL, MariaDB, Microsoft SQL ή PostgreSQL.

Το HeidiSQL[3], είναι ένα πρόγραμμα διαχείρισης και επεξεργασίας βάσεων δεδομένων, που έχουν αναπτυχθεί με την MySQL. Είναι μικρό, ελαφρύ και χάρη στο λειτουργικό και φιλικό του περιβάλλον επιτρέπει να εκτελούνται γρήγορα και εύκολα οι αναγκαίες ενέργειες.

1.4.1 Περιβάλλον εργασίας

Το περιβάλλον εργασίας του προγράμματος είναι λειτουργικό και οργανωμένο. Στα αριστερά του υπάρχει η λίστα με τους πίνακες και τα διάφορα στοιχεία της βάσης δεδομένων, με την οποία γίνετε η σύνδεση.

Στα δεξιά εμφανίζονται οι καρτέλες, τα δεδομένα του κάθε πίνακα που έχει επιλεγεί και στο κάτω μέρος, υπάρχει ένα παράθυρο εμφάνισης με τις εντολές που εκτελούνται κάθε φορά.

1.4.2 Χαρακτηριστικά και λειτουργίες

[3] https://www.heidisql.com/

Το πρόγραμμα HeidiSQL δίνει την δυνατότητα σύνδεσης σε πολλαπλούς servers ταυτόχρονα από ένα παράθυρο ή χρησιμοποιώντας την γραμμή εντολών. Παρέχει τη δυνατότητα επεξεργασίας πινάκων, προβολών, αποθηκευμένων ρουτινών triggers και προγραμματισμένων events καθώς και τη βελτιστοποίηση ή τη διόρθωση πινάκων.

Η πλοήγηση στα δεδομένα της βάσης καθώς και η επεξεργασία τους, γίνεται απευθείας μέσα στα κελιά του κάθε πίνακα. Με την λειτουργία Find υπάρχει η δυνατότητα αναζήτησης και εντοπισμού συγκεκριμένων κειμένων σε όλους τους πίνακες μιας βάσης. Γράφοντας queries και χρησιμοποιώντας σύνταξη SQL διευκολύνεται η χρήση syntax highlighting και code completion.

Με την βοήθειά του εξάγονται δεδομένα χρησιμοποιώντας την σύνταξη SQL, καθώς μπορούν να μεταφερθούν δεδομένα από μία βάση δεδομένων, απευθείας σε μία άλλη. Επίσης είναι δυνατή η εξαγωγή μεμονωμένων γραμμών ενός πίνακα σε μορφή csv, html, xnl ,sql, LaTex και Wiki Markup.

Όσον αφορά στον τομέα της εισαγωγής δεδομένων, είναι δυνατή η εισαγωγή αρχείων σε μορφή ascii ή binary κατευθείαν σε κάποιον πίνακα.

Η HeidiSQL έχει τις παρακάτω λειτουργίες και δυνατότητες GUI [10]:

Σύνδεση διακομιστή (server connection)

- Πολλές αποθηκευμένες περιόδους σύνδεσης με συνδέσεις και διαπιστευτήρια αποθηκευμένα μέσα.
- Συγκεκριμένο πρωτόκολλο πελάτη / διακομιστή για συμβατούς διακομιστές.
- Διασύνδεση με διακομιστές μέσω TCP / IP , ονομαστικές σωληνώσεις (υποδοχές) ή πρωτόκολλο σήραγγας (SSH).
- Πολλές παράλληλες περιόδους λειτουργίας σε ένα παράθυρο.
- Διαχειριστείτε τους χρήστες στο διακομιστή: προσθέστε, αφαιρέστε και επεξεργαστείτε τους χρήστες και τα διαπιστευτήρια τους.
- Διαχειριστείτε τα δικαιώματα χρήστη παγκοσμίως και ανά βάση δεδομένων.
- Εξάγετε βάσεις δεδομένων σε αρχεία SQL ή σε άλλους διακομιστές.
- Πολλαπλές καρτέλες επερωτήσεων, με το καθένα να έχει πολλαπλές υποκατηγορίες για αποτελέσματα παρτίδας.

Υποδοχή διακομιστή (server host)

- Προβάλετε και φιλτράρετε όλες τις μεταβλητές διακομιστή, όπως system_time_zone
- Επεξεργαστείτε όλες τις μεταβλητές διακομιστή, είτε για αυτή τη σύνοδο είτε με παγκόσμιο πεδίο
- Δείτε στατιστικές μεταβλητές διακομιστή και μέσες τιμές ανά ώρα και δευτερόλεπτο
- Αυτή τη στιγμή εκτελούνται διαδικασίες για την ανάλυση εκτελεσθείσας SQL και για την αποτροπή κακών διαδικασιών
- Προβολή στατιστικών στοιχείων εντολών με ποσοστιαίες γραμμές δείκτη ανά εντολή SQL

Βάσεις δεδομένων (databases)

- Προβάλετε όλες τις βάσεις δεδομένων του διακομιστή, συνδεθείτε σε μια ενιαία βάση δεδομένων για να εργαστείτε με τους πίνακες και τα δεδομένα
- Προβάλετε το συνολικό βάθος των βάσεων δεδομένων και το μέγεθος του πίνακα σε KB / MB / GB μέσα στη δομή δέντρου βάσης δεδομένων / πίνακα
- Δημιουργήστε νέα, αλλάξετε την υπάρχουσα βάση δεδομένων, το σύνολο χαρακτήρων και την ταξινόμηση, αποθέστε (διαγράψτε) βάσεις δεδομένων

Πίνακες, προβολές, διαδικασίες, ενεργοποιητές και συμβάντα (tables, views, procedures, triggers and events)

- Προβολή όλων των αντικειμένων μέσα στην επιλεγμένη βάση δεδομένων, κενά, μετονομασία και απόθεση (διαγραφή) αντικειμένων
- Επεξεργαστείτε τις στήλες, τα ευρετήρια και τα ξένα πλήκτρα. Υποστηρίζονται οι εικονικές στήλες στους διακομιστές MariaDB.
- Επεξεργασία ερωτήματος και ρυθμίσεων προβολής
- Επεξεργασία σώματος και παραμέτρους SQL διαδικασίας
- Επεξεργασία σκελετού και ρυθμίσεις SQL
- Επεξεργασία προγραμματισμένων ρυθμίσεων ώρας σώματος SQL συμβάντων

1.5 HTML

HTML είναι το ακρωνύμιο των λέξεων Hyper Text Markup Language (γλώσσα μορφοποίησης υπερκειμένου) και είναι βασική γλώσσα για την δόμηση σελίδων του World Wide Web (ή απλά Web) [14]. Είναι μία γλώσσα προγραμματισμού η οποία χρησιμοποιείται για να σημάνει ένα τμήμα κειμένου και να το κάνει να εμφανίζεται καλύτερα, και επιτρέπει την ενσωμάτωση ήχου και εικόνων στις web σελίδες. Αρχικά είχε κατασκευασθεί με σκοπό μόνο την μορφοποίηση κειμένου, αλλά μεγάλωσε και ενσωμάτωσε σχεδιαστικές τεχνικές και άλλα.

Η γλώσσα HTML χρησιμοποιεί έναν αριθμό από tags (ετικέτες) για την μορφοποίηση κειμένου, για την δημιουργία συνδέσμων (links) μετάβασης ανάμεσα στις σελίδες, με σκοπό την εισαγωγή εικόνων, ήχου και άλλα. Ο σκοπός ενός web browser είναι να διαβάζει τα έγγραφα HTML και να τα συνθέσει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να παρουσιάσει το περιεχόμενο της σελίδας. Η HTML έχει σχεδιαστεί ώστε να είναι διαλειτουργική όσο το δυνατόν με την ευρύτερη σειρά πλατφορμών και συσκευών με διαφορετικές δυνατότητες [14].

Στα στοιχεία της HTML μπορούν να ενσωματώνονται σενάρια εντολών σε γλώσσες όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML και από στατικές τις κάνουν διαδραστικές.

Η προαιρετική παράμετρος "charset" αναφέρεται στον χαρακτήρα κωδικοποίησης που χρησιμοποιείται για την αναπαραγωγή του εγγράφου HTML ως ακολουθία bytes [14]. Οποιοσδήποτε καταχωρημένος χαρακτήρας IANA μπορεί να χρησιμοποιηθεί, αλλά το UTF-8 είναι προνομιούχος. Παρόλο που αυτή η παράμετρος είναι προαιρετική, είναι ισχυρή συνέστησε να είναι πάντα παρούσα.

1.6 CSS

Το CSS (Cascading Style Sheets - Διαδοχικά Φύλλα Στυλ) ή (αλληλουχία φύλλων στυλ) είναι ένα είδος γλώσσας προγραμματισμού, το οποίο χρησιμοποιείται για να περιγράψει τη σημασιολογία μιας παρουσίασης (δηλαδή την εμφάνιση και τη μορφοποίηση) ενός εγγράφου γραμμένο σε μια γλώσσα σήμανσης [15]. Πιο κοινή εφαρμογή του είναι να

δίνει μορφή σε ιστοσελίδες γραμμένες σε HTML και XHTML, αλλά η γλώσσα μπορεί επίσης να εφαρμοστεί σε οποιοδήποτε είδος εγγράφου XML, όπως SVG και XUL.

Το CSS έχει σχεδιαστεί για να επιτρέπεται κυρίως ο διαχωρισμός του περιεχομένου ενός εγγράφου (γραμμένο σε HTML ή παρόμοια γλώσσα σήμανσης) από τη παρουσίαση του εγγράφου, συμπεριλαμβανομένων των στοιχείων όπως η διάρθρωση, τα χρώματα και οι γραμματοσειρές. Ο διαχωρισμός αυτός μπορεί να βελτιώσει την προσβασιμότητα του περιεχομένου, την παροχή περισσότερης ευελιξίας και έλεγχο των προδιαγραφών των χαρακτηριστικών παρουσίασης, ώστε πολλές ιστοσελίδες να έχουν την ίδια μορφοποίηση, καθώς και να μειώσουν την πολυπλοκότητα και την επανάληψη του διαρθρωτικού περιεχομένου (για παράδειγμα παρέχοντας τη δυνατότητα για tableless web design). Μπορεί επίσης να επιτρέψει την ίδια ιστοσελίδα να υποβληθεί σε διαφορετικά στυλ για τις διάφορες μεθόδους επεξεργασίας υποπροϊόντων, όπως στην οθόνη και σε έντυπη μορφή.

Το CSS ορίζει ένα σύστημα προτεραιότητας, τέτοιο ώστε να καθορίσει ποιοι κανόνες στυλ εφαρμόζονται αν υπάρχουν περισσότεροι από ένας κανόνας για κάποιο συγκεκριμένο στοιχείο. Οι προτεραιότητες αυτές υπολογίζονται με βάση κάποιους κανόνες, έτσι ώστε τα αποτελέσματα να είναι προβλέψιμα.

1.7 JavaScript

Η JavaScript (JS) είναι μια γλώσσα script που αναπτύχθηκε από την Netscape το 1995, αλλά υποστηρίζεται πλέον από την ECMA για αυτό και οι νεότερες εκδόσεις λέγονται ECMAScript [16].

Συγκεκριμένα, η JavaScript, είναι μια ελαφριά γλώσσα προγραμματισμού που χρησιμοποιεί γραμμές εκτελέσιμου κώδικα και εισάγεται σε ένα έγγραφο Html. Η JavaScript είναι μια γλώσσα "open scripting", και μπορεί να τη χρησιμοποιήσει οποιοσδήποτε χωρίς άδεια και δικαιώματα. Υποστηρίζεται από όλα τα γνωστά προγράμματα περιήγησης, όπως Firefox, Opera, Chrome, Safari και Internet Explorer και ενδείκνυται για την ανάπτυξη δυναμικών ιστοσελίδων. Είναι ένα βοηθητικό εργαλείο για την ανάπτυξη διαδικτυακών εφαρμογών αλλά και για την ανάπτυξη εφαρμογών τεχνολογίας Ajax.

Πιο συγκεκριμένα η JavaScript είναι μια γλώσσα σεναρίων που βασίζεται στα πρωτότυπα (prototype-based), είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Η σύνταξή της είναι επηρεασμένη από τη C ενώ παράλληλα

αντιγράφει πολλά ονόματα και συμβάσεις ονοματοδοσίας από τη Java. Οι βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme. Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm), υποστηρίζοντας αντικειμενοστραφές, προστακτικό και συναρτησιακό στυλ προγραμματισμού.

Αναλυτικότερα, όταν το script εισαχθεί σε ένα έγγραφο html, το πρόγραμμα περιήγησης θα διαβάσει τον κώδικα, θα μεταφράσει το script και θα εκτελέσει τις γραμμές κώδικα. Ένα script, μπορεί να εκτελεστεί τη στιγμή που το διαβάζει το πρόγραμμα περιήγησης ή μετά από ένα συμβάν (event) που κάποιος θα έχει ορίσει. Το script αυτό, μπορεί να ρυθμιστεί έτσι ώστε να εκτελείται μετά από κάποιο ή κάποια συγκεκριμένα συμβάντα, όπως για παράδειγμα, το πάτημα ενός πλήκτρου του ποντικιού [17].

Επιπλέον, μπορεί να διαβάσει αλλά και να αλλάξει τις ιδιότητες ενός στοιχείου html ή και τη δομή ενός εγγράφου προσθαφαιρώντας ετικέτες. Αυτό είναι μια από τις βασικές ιδιότητες που δίνει σε ένα έγγραφο δυναμικό χαρακτηριστικά. Μπορεί να διαβάσει και να αλλάξει τα στυλ (css) ενός στοιχείου html ή ακόμα, και να ελέγξει την εγκυρότητα δεδομένων στα πεδία μιας φόρμας, πριν αυτή σταλεί στον διακομιστή.

1.8 jQuery

Το jQuery είναι μια γρήγορη, μικρή αλλά πλούσια σε χαρακτηριστικά βιβλιοθήκη της JavaScript. Η βιβλιοθήκη αυτή δίνει τη δυνατότητα δημιουργίας διάφορων εφέ σε μια σελίδα, χωρίς όμως να χρειάζεται η ποσότητα κώδικα που θα χρησιμοποιούνταν στην παραδοσιακή JavaScript. Η jQuery είναι ελεύθερο λογισμικό, με άδεια MIT [18].

Το σύμβολο του **δολαρίου [$]** είναι ο επιλογέας της jQuery.

Η σύνταξη του jQuery έχει σχεδιαστεί για να διευκολύνει την πλοήγηση σε ένα έγγραφο, να επιλέγει στοιχεία DOM (Document Object Module) [19], να δημιουργεί κινούμενα σχέδια, να χειρίζεται συμβάντα και να αναπτύσσει εφαρμογές Ajax. Το jQuery παρέχει επίσης δυνατότητες για τους προγραμματιστές να δημιουργούν plug-ins πάνω από τη βιβλιοθήκη JavaScript. Αυτό επιτρέπει στους προγραμματιστές να δημιουργούν αφαιρέσεις για αλληλεπίδραση και κινούμενα σχέδια χαμηλού επιπέδου, εξελιγμένα εφέ και γραφικά widgets υψηλού επιπέδου. Η αρθρωτή προσέγγιση της

βιβλιοθήκης jQuery επιτρέπει τη δημιουργία ισχυρών δυναμικών ιστοσελίδων και εφαρμογών Web.

Οι αρχές ανάπτυξης με το jQuery είναι [18]:

- Διαχωρισμός του JavaScript και του HTML: Η βιβλιοθήκη jQuery παρέχει απλή σύνταξη για την προσθήκη χειριστών συμβάντων στο DOM χρησιμοποιώντας JavaScript, αντί να προσθέσει χαρακτηριστικά συμβάντος HTML για κλήσεις λειτουργιών JavaScript. Έτσι, ενθαρρύνει τους προγραμματιστές να ξεχωρίζουν πλήρως τον κώδικα JavaScript από τη σήμανση HTML.
- Εύρος (Brevity) και σαφήνεια (clarity): Το jQuery προάγει τη συντομία και τη σαφήνεια με χαρακτηριστικά όπως "αλυσιδωτές" λειτουργίες και ονόματα λειτουργιών στενογραφίας.
- Εξάλειψη των ασυμβατοτήτων μεταξύ των προγραμμάτων περιήγησης: Οι μηχανές JavaScript διαφορετικών προγραμμάτων περιήγησης διαφέρουν ελαφρώς, οπότε ο κώδικας JavaScript που λειτουργεί για ένα πρόγραμμα περιήγησης ενδέχεται να μην λειτουργεί για άλλο. Όπως και με άλλα εργαλεία JavaScript, το jQuery χειρίζεται όλες αυτές τις ασυνέπειες μεταξύ των περιηγητών και παρέχει μια συνεπή διασύνδεση που λειτουργεί σε διαφορετικά προγράμματα περιήγησης.
- Επεκτασιμότητα: Νέα συμβάντα, στοιχεία και μέθοδοι μπορούν να προστεθούν εύκολα και να επαναχρησιμοποιηθούν ως πρόσθετο.

1.9 WinSCP

Το WinSCP είναι ένα δημοφιλές δωρεάν πρόγραμμα-πελάτης SFTP και FTP για Windows, ένας ισχυρός διαχειριστής αρχείων που θα βελτιώσει την παραγωγικότητά σας. Προσφέρει ένα εύχρηστο GUI για την αντιγραφή αρχείων μεταξύ τοπικού και απομακρυσμένου υπολογιστή χρησιμοποιώντας πολλαπλά πρωτόκολλα: Amazon S3, FTP, FTPS, SCP, SFTP ή WebDAV. Οι χρήστες δύναμης μπορούν να αυτοματοποιήσουν το WinSCP χρησιμοποιώντας τη συναρμολόγηση .NET. Το WinSCP είναι διαθέσιμο στα Αγγλικά και σε πολλές άλλες γλώσσες.

Το WinSCP (Windows Secure Copy) [20] είναι ένα πρόγραμμα-πελάτης SFTP ανοικτού κώδικα, πρόγραμμα-πελάτης FTP, πελάτης WebDAV και πρόγραμμα-πελάτης SCP για Windows. Η κύρια λειτουργία του είναι η μεταφορά αρχείων μεταξύ τοπικού και

απομακρυσμένου υπολογιστή. Το WinSCP υποστηρίζει το SFTP (Πρωτόκολλο μεταφοράς αρχείων SSH) για ασφαλείς μεταφορές αρχείων και παλαιότερο SCP (Secure Copy Protocol). Μπορείτε να χρησιμοποιήσετε το WinSCP για τη μεταφορά αρχείων τόσο με το χέρι όσο και αυτόματα.

Το WinSCP εγκαθίσταται χωρίς κόπο μέσω του προγράμματος εγκατάστασης του, το οποίο σας επιτρέπει να επιλέξετε προεπιλογές ή να προσαρμόσετε το WinSCP στις προτιμήσεις σας. Για παράδειγμα, μπορείτε να επιλέξετε μια διασύνδεση Norton-Commander ή μια διεπαφή τύπου Explorer. Η διασύνδεση εντολών Norton παρέχει τόσο τοπικούς όσο και απομακρυσμένους πίνακες καταλόγων, ενώ η διασύνδεση τύπου Explorer παρέχει μόνο ένα απομακρυσμένο πλαίσιο.

Η ανάπτυξη του WinSCP ξεκίνησε γύρω στο Μάρτιο του 2000 και συνεχίζεται. Αρχικά φιλοξένησε το Οικονομικό Πανεπιστήμιο της Πράγας, όπου ο δημιουργός της εργάστηκε εκείνη τη στιγμή.

1.10 Visual Studio Code

Το Visual Studio Code είναι ένας επεξεργαστής πηγαίου κώδικα (editor) που αναπτύχθηκε από τη Microsoft για Windows, Linux και macOS. Περιλαμβάνει υποστήριξη για εντοπισμό σφαλμάτων (debugging), ενσωματωμένο έλεγχο Git (embedded Git cintrol), επισήμανση σύνταξης (syntax highlighting), έξυπνη ολοκλήρωση κώδικα (intelligent code completion), αποσπάσματα (snippets) και κώδικα refactoring. Είναι επίσης προσαρμόσιμη, ώστε οι χρήστες να μπορούν να αλλάζουν το θέμα του editor, τις συντομεύσεις του πληκτρολογίου και τις προτιμήσεις. Είναι δωρεάν και ανοικτού κώδικα, [21]αν και η επίσημη λήψη είναι υπό ιδιωτική άδεια [22].

Το Visual Studio Code βασίζεται στο Electron, ένα πλαίσιο που χρησιμοποιείται για την ανάπτυξη εφαρμογών Node.js για την επιφάνεια εργασίας που εκτελείται στη μηχανή εμφάνισης Blink. Παρόλο που χρησιμοποιεί το ηλεκτρονικό πλαίσιο, το λογισμικό δεν χρησιμοποιεί το Atom και χρησιμοποιεί την ίδια συνιστώσα επεξεργαστή (με την κωδική ονομασία "Monaco") που χρησιμοποιείται στο Visual Studio Team Services (πρώην Visual Studio Online)

Στο Survey Overflow 2018 Developer Survey, to Visual Code κατατάχθηκε ως το πιο δημοφιλές εργαλείο περιβάλλοντα ανάπτυξης, με το 34,9% των 75.398 ερωτηθέντων να ισχυρίζονται ότι το χρησιμοποιούν [23].

1.11 Apache HTTP

Το Apache HTTP Server Project είναι μια προσπάθεια να αναπτυχθεί και να διατηρηθεί ένας διακομιστής HTTP ανοιχτού κώδικα για σύγχρονα λειτουργικά συστήματα, συμπεριλαμβανομένων των UNIX και των Windows. Ο στόχος αυτού του έργου είναι να παρέχει έναν ασφαλή, αποδοτικό και επεκτάσιμο διακομιστή που παρέχει υπηρεσίες HTTP σε συγχρονισμό με τα τρέχοντα πρότυπα HTTP [24].

Το Apache είναι το πιο διαδεδομένο λογισμικό διακομιστή ιστού. Αναπτύχθηκε και συντηρείται από το Apache Software Foundation και είναι ένα λογισμικό ανοιχτού κώδικα διαθέσιμο δωρεάν. Λειτουργεί στο 67% όλων των web servers στον κόσμο. Είναι γρήγορο, αξιόπιστο και ασφαλές [24]. Μπορεί να είναι ιδιαίτερα προσαρμοσμένο για να καλύψει τις ανάγκες πολλών διαφορετικών περιβαλλόντων χρησιμοποιώντας επεκτάσεις και ενότητες.

1.12 AJAX

Ο όρος **AJAX** (Asynchronous JavaScript and XML) είναι ένα σύνολο από Web development τεχνικές που χρησιμοποιούν πολλές τεχνολογίες του διαδικτύου από την πλευρά του πελάτη (client-side) για να δημιουργήσουν ασύγχρονες Web εφαρμογές.

Με την χρήση AJAX οι διαδικτυακές εφαρμογές μπορούν να στέλνουν και να λαμβάνουν δεδομένα στο παρασκήνιο χωρίς να τους απασχολεί τι προβάλλεται στην σελίδα. Ο διαχωρισμός της ανταλλαγής δεδομένων με την προβολή τους επιτρέπει στην ιστοσελίδα να αλλάζει το περιεχόμενο της δυναμικά χωρίς να χρειάζεται ολοκληρωτική ανανέωση της [25].

Το AJAX δεν είναι από μόνο του μια τεχνολογία αλλά ένα **σύνολο τεχνολογιών**. Χρησιμοποιεί HTML και CSS για την σήμανση και την παρουσίαση των ιστοσελίδων και χρησιμοποιεί JavaScript και XMLHttpRequest για να κινεί δεδομένα ασύγχρονα ανάμεσα στον πελάτη και τον διακομιστή (client - server). Τα δεδομένα δύναται να μετακινούνται και σε μορφή JSON ή XML.

Το AJAX δεν απαιτεί την χρήση της XML για την λήψη των δεδομένων, καθώς επίσης δεν είναι απαραίτητο τα αιτήματα προς τον Διακομιστή να είναι ασύγχρονα.

Τεχνολογίες

Ο όρος *Ajax* έχει έρθει να αντιπροσωπεύσει μια ευρεία ομάδα των τεχνολογιών Web που μπορεί να χρησιμοποιηθούν για να φτιάξουν μια Web εφαρμογή που επικοινωνεί με ένα διακομιστή στο παρασκήνιο, χωρίς να παρεμβαίνει με την τρέχουσα κατάσταση της σελίδας. Στο άρθρο που αναλύεται ο όρος Ajax, ο Jesse James Garrett εξήγησε ότι οι απαιτούμενες τεχνολογίες είναι οι εξής [26]:

- HTML (ή XHTML) και CSS για την παρουσίαση
- Το Μοντέλο Αντικειμένου Εγγράφου (DOM- Document Object Model) για τη δυναμική επίδειξη της και την αλληλεπίδραση της με τα δεδομένα
- JSON ή XML για την ανταλλαγή των δεδομένων, και XSLT
- Το XMLHttpRequest αντικείμενο για την ασύγχρονη επικοινωνία
- Η JavaScript για να φέρει αυτές τις τεχνολογίες μαζί

Ωστόσο, με το πέρασμα του χρόνου γίνεται αντιληπτό πως υπάρχει μια σειρά από εξελίξεις στις τεχνολογίες που χρησιμοποιούνται σε μια Ajax εφαρμογή. Μια ποικιλία από δημοφιλή JavaScript βιβλιοθήκες, συμπεριλαμβανομένων και της JQuery, περιλαμβάνουν αφηρημένες έννοιες για να βοηθήσουν στην εκτέλεση αιτήσεων Ajax.

Η AJAX δίνει τη δυνατότητα εμφάνισης νέων στοιχείων στο site, χωρίς τη φόρτωση νέας σελίδα. Μπορεί δηλαδή ο web developer να δημιουργήσει ένα site με μία μόνο σελίδα, στην οποία θα φορτώνονται διαφορετικά δεδομένα ανάλογα με τις επιλογές του χρήστη. Έτσι καταργεί τους ατελείωτους φακέλους με τα html αρχεία, στα οποία επαναλαμβάνεται το ίδιο κομμάτι κώδικα, βελτιώνοντας παράλληλα και την ασφάλεια του site καθώς καταργεί την αλλαγή του url στη μπάρα διευθύνσεων.

ΚΕΦΑΛΑΙΟ 3 – Κρυπτογραφία και ασφάλεια λογισμικού

3.1 Εισαγωγή

Κρυπτογραφία (cryptography) είναι η μελέτη τεχνικών που βασίζονται σε μαθηματικά προβλήματα δύσκολο να λυθούν, με σκοπό την εξασφάλιση της ασφάλειας (εμπιστευτικότητα, ακεραιότητα, αυθεντικότητα) των δεδομένων [27].

Κρυπτανάλυση (cryptanalysis) είναι η μελέτη μαθηματικών τεχνικών για την προσβολή κρυπτογραφικών τεχνικών ή υπηρεσιών ασφάλειας και κρυπτολογία (cryptology) είναι ο συνδυασμός της κρυπτογραφίας και κρυπτανάλυσης σε ένα ενιαίο επιστημονικό κλάδο. Εφαρμογή της κρυπτογραφίας είναι η κρυπτογράφηση [28].

Κρυπτογράφηση είναι ο μετασχηματισμός δεδομένων σε μορφή που να είναι αδύνατον να διαβαστεί χωρίς τη γνώση της σωστής ακολουθίας bit. Η ακολουθία bit καλείται "κλειδί" και χρησιμοποιείται σε συνδυασμό με κατάλληλο αλγόριθμο / συνάρτηση. Η αντίστροφη διαδικασία είναι η αποκρυπτογράφηση και απαιτεί γνώση του κλειδιού. Σκοπός της κρυπτογράφησης είναι να εξασφαλίσει το απόρρητο των δεδομένων κρατώντας τα κρυφά από όλους όσους έχουν πρόσβαση σε αυτά. Η κρυπτογράφηση και η αποκρυπτογράφηση απαιτούν, όπως είπαμε, τη χρήση κάποιας μυστικής πληροφορίας, το κλειδί. Για μερικούς μηχανισμούς χρησιμοποιείται το ίδιο κλειδί και για την κρυπτογράφηση για άλλους όμως τα κλειδιά που χρησιμοποιούνται διαφέρουν. Η κρυπτογραφία παρέχει μηχανισμούς για διαδικασίες ασφάλειας, όπως η ψηφιακή υπογραφή, η οποία συνδέει ένα έγγραφο με τον κάτοχο ενός κλειδιού, έτσι ώστε όλοι όσοι είναι σε θέση να το αναγνώσουν, να είναι σίγουροι για το ποιος το έχει γράψει. Επίσης, μία ψηφιακή χρονοσφραγίδα (digital timestamp) συνδέει ένα έγγραφο με την ώρα δημιουργίας του. Τέτοιοι μηχανισμοί μπορούν να χρησιμοποιηθούν για έλεγχο πρόσβασης σε ένα σκληρό δίσκο, για ασφαλείς συναλλαγές μέσω του Διαδικτύου ή ακόμα και για σύνδεση με καλωδιακή τηλεόραση.

3.2 Είδη κρυπτογραφημάτων

Η κρυπτογραφία είναι η επιστήμη που ασχολείται με τη μελέτη, την ανάπτυξη και τη χρήση τεχνικών κρυπτογράφησης και αποκρυπτογράφησης με σκοπό την απόκρυψη του περιεχομένου των μηνυμάτων.

3.2.1 Ασύμμετρη Κρυπτογραφία

Η ασύμμετρη κρυπτογραφία (Public Key Cryptography) χρησιμοποιεί δύο διαφορετικά κλειδιά για την κρυπτογράφηση και αποκρυπτογράφηση. Κάθε χρήστης έχει στην κατοχή του ένα ζεύγος κλειδιών, το ένα καλείται **δημόσιο κλειδί** (public key) και το άλλο καλείται **ιδιωτικό κλειδί** (private key). Το δημόσιο κλειδί δημοσιοποιείται, ενώ το ιδιωτικό κλειδί κρατείται πάντοτε μυστικό. Το ιδιωτικό κλειδί δεν μεταδίδεται ποτέ στο δίκτυο και όλες οι επικοινωνίες βασίζονται στο δημόσιο κλειδί [29].

Η μόνη απαίτηση της ασύμμετρης κρυπτογραφίας είναι η διαπιστευμένη και επιβεβαιωμένη συσχέτιση των δημόσιων κλειδιών με τους κατόχους τους, ώστε να μην είναι δυνατή η σκόπιμη ή μη πλαστοπροσωπία. Η ασύμμετρη κρυπτογράφηση μπορεί να χρησιμοποιηθεί όχι μόνο για κρυπτογράφηση, αλλά και για παραγωγή ψηφιακών υπογραφών. Το ιδιωτικό κλειδί είναι μαθηματικά συνδεδεμένο με το δημόσιο κλειδί. Τυπικά, λοιπόν, είναι δυνατόν να "σπάσει" ένα τέτοιο κρυπτοσύστημα ανακτώντας το ιδιωτικό κλειδί από το δημόσιο.

Η κρυπτογράφηση με χρήση της ασύμμετρης κρυπτογραφίας γίνεται ως εξής: όταν ο χρήστης Α θέλει να στείλει ένα μυστικό μήνυμα στο χρήστη Β, χρησιμοποιεί το δημόσιο κλειδί του Β για να κρυπτογραφήσει το μήνυμα και έπειτα το στέλνει στον Β. Ο χρήστης Β, αφού παραλάβει το μήνυμα, κάνει χρήση του ιδιωτικού του κλειδιού για να το αποκρυπτογραφήσει [29].

Κάποιος που παρακολουθεί τη σύνδεση, δεν μπορεί να αποκρυπτογραφήσει το μήνυμα. Όποιος έχει το δημόσιο κλειδί του Β, μπορεί να του στείλει μήνυμα, ενώ μόνο ο Β μπορεί να το διαβάσει, γιατί είναι ο μόνος που γνωρίζει το ιδιωτικό κλειδί.

Όταν ο Α θέλει να χρησιμοποιήσει την ασύμμετρη κρυπτογραφία για να υπογράψει ένα μήνυμα, τότε πραγματοποιεί έναν υπολογισμό που απαιτεί το ιδιωτικό του κλειδί και το ίδιο το μήνυμα. Το αποτέλεσμα του υπολογισμού καλείται ψηφιακή υπογραφή και μεταδίδεται μαζί με το μήνυμα. Για να επαληθεύσει την υπογραφή ο Β πραγματοποιεί ανάλογο υπολογισμό χρησιμοποιώντας το δημόσιο κλειδί του Α, το μήνυμα και την υπογραφή. Εάν το αποτέλεσμα είναι θετικό, τότε η υπογραφή είναι αυθεντική. Διαφορετικά η υπογραφή είναι πλαστή ή το μήνυμα έχει τροποποιηθεί.

3.2.2 Συμμετρική κρυπτογραφία

Στη συμμετρική κρυπτογραφία (Symmetric Cryptography ή Secret –Key Cryptography) ο αποστολέας και ο παραλήπτης ενός μηνύματος γνωρίζουν και χρησιμοποιούν το ίδιο μυστικό

κλειδί (secret key). Ο αποστολέας χρησιμοποιεί το μυστικό κλειδί για να κρυπτογραφήσει το μήνυμα και ο παραλήπτης χρησιμοποιεί το ίδιο κλειδί για να αποκρυπτογραφήσει το μήνυμα. Αυτή η μέθοδος καλείται συμμετρική κρυπτογραφία ή κρυπτογραφία μυστικού κλειδιού [30]. Η συμμετρική κρυπτογραφία χρησιμοποιείται όχι μόνο για κρυπτογράφηση, άλλα και για πιστοποίηση ταυτότητας. Μία τέτοια τεχνική είναι η Message Authentication Code (MAC).

Τα βήματα συμμετρικής κρυπτογραφίας μπορούν να περιγραφούν ως:

- Ο αποστολέας ενός μηνύματος κρυπτογραφεί το μήνυμα χρησιμοποιώντας έναν αλγόριθμο που βασίζεται σε κλειδί.
- Το κρυπτογραφημένο μήνυμα στέλνεται μέσω κάποιου ανασφαλούς μέσου όπως το Διαδίκτυο.
- Το κλειδί μεταφέρεται με κάποιο ασφαλή τρόπο στον παραλήπτη.
- Ο παραλήπτης λαμβάνει το κλειδί και το χρησιμοποιεί για να αποκρυπτογραφήσει το μήνυμα που έλαβε.

Η κρυπτογράφηση συμμετρικού κλειδιού είναι πολύ αποδοτική όσον αφορά τους πόρους που απαιτούνται. Ωστόσο το κύριο πρόβλημα της συμμετρικής κρυπτογραφίας είναι η συνεννόηση του αποστολέα και του παραλήπτη στο κοινό μυστικό κλειδί που θα κρυπτογραφεί και αποκρυπτογραφεί όλη τη διακινούμενη πληροφορία, χωρίς κάποιον άλλο να λάβει γνώση αυτού. Υπάρχουν διάφοροι αλγόριθμοι συμμετρικού κλειδιού που χρησιμοποιούνται σήμερα όπως: *DES*, *Τριπλό DES*, *Blowfish*, *IDEA*, *RC2*, *RC4* και *RC5*.

3.2.3 Μειονεκτήματα και Πλεονεκτήματα

Πλεονεκτήματα συμμετρικών αλγορίθμων [31]:

- Χαμηλό υπολογιστικό κόστος
- Εύκολη υλοποίηση (hardware)
- Ταχύτητα κρυπτογράφησης. Κατά κανόνα, οι διαδικασίες κρυπτογράφησης και πιστοποίησης ταυτότητας με συμμετρικό κλειδί είναι σημαντικά ταχύτερες από την κρυπτογράφηση και ψηφιακή υπογραφή με ζεύγος ασύμμετρων κλειδιών.
- Δεδομένου ότι δεν υπάρχει κλειδί που μεταδίδεται με τα δεδομένα, οι πιθανότητες αποκρυπτογράφησης δεδομένων είναι μηδενικές.

- Μόνο το σύστημα που διαθέτει το μυστικό κλειδί μπορεί να αποκρυπτογραφήσει το μεταδιδόμενο μήνυμα.

Μειονεκτήματα συμμετρικών αλγορίθμων:

- Γνωστοποίηση κλειδιού. Η μετάδοση δια μέσω Διαδικτύου δεν είναι ασφαλής. Οποιοσδήποτε γνωρίζει για τη συναλλαγή μπορεί να καταγράψει την επικοινωνία μεταξύ του αποστολέα και του παραλήπτη και να υποκλέψει το κλειδί. Σε αυτή την περίπτωση, μπορεί να τροποποιηθούν ή και να πλαστογραφηθούν τα μηνύματα που ανταλλάσσουν οι δύο ανυποψίαστοι χρήστες. Έτσι ο μόνος ασφαλής τρόπος ανταλλαγής κλειδιών θα ήταν η ανταλλαγή προσωπικά.
- Δεν είναι δυνατή η παροχή ψηφιακών υπογραφών που δεν μπορούν να απορριφθούν.

Πλεονεκτήματα ασύμμετρων αλγορίθμων [31]:

- Υψηλή ασφάλεια. Δε χρειάζεται ποτέ να μεταδοθεί ή να αποκαλυφθεί το ιδιωτικό κλειδί.
- Αποτελεί μέθοδο για ψηφιακές υπογραφές. Κάποιος μπορεί να επιβεβαιώσει την ταυτότητά του μόνο με το ιδιωτικό του κλειδί. Ένα ιδιωτικό κλειδί αντιστοιχεί σε κάθε μοναδικό χρήστη και το σύστημα πιστοποίησης ταυτότητας παρέχει «ψηφιακή» εμπιστοσύνη στον κάτοχό του. Δεν υπάρχει κίνδυνος κλοπής του ιδιωτικού κλειδιού αφού κάθε χρήστης έχει αποκλειστική γνώση του ιδιωτικού του κλειδιού και είναι δικιά του ευθύνη η φύλαξή του.

Μειονέκτημα ασύμμετρων αλγορίθμων:

- Ταχύτητα κρυπτογράφησης.
- Ανάγκη για πιστοποίηση και επαλήθευση των δημόσιων κλειδιών από οργανισμούς Πιστοποίησης (Certificate Authority) ώστε να διασφαλίζεται η κατοχή των νόμιμων χρηστών. Όταν κάποιος επιτήδειος κατορθώσει και ξεγελάσει τον οργανισμό, μπορεί να συνδέσει το όνομά του με το δημόσιο κλειδί ενός νόμιμου χρήστη και να προσποιείται την ταυτότητα αυτού του νόμιμου χρήστη.

Τα δύο κρυπτοσυστήματα μπορούν να εφαρμοστούν μαζί, συνδυάζοντας τα καλά τους χαρακτηριστικά και εξαλείφοντας τα μειονεκτήματα τους. Ένα παράδειγμα τέτοιου συνδυασμού είναι οι ψηφιακοί φάκελοι.

3.3 Απαιτήσεις της κρυπτογραφίας

Η ασφάλεια δικτύων είναι απαραίτητη για την σωστή λειτουργίας οποιουδήποτε δικτύου υπολογιστών. Η διασφάλιση της γίνεται με την χρήση μεθόδων κρυπτογραφίας.

Η Υποδομή Δημόσιου Κλειδιού (PKI – Public Key Infrastructure) [32] είναι ένας συνδυασμός από προγράμματα, τεχνολογίες κρυπτογράφησης, διαδικασίες και υπηρεσίες οι οποίες χρησιμοποιούνται για την δημιουργία, διαχείριση, διανομή, χρήση και ανάκληση ψηφιακών πιστοποιητικών. Συγκεκριμένα πρόκειται για έναν τρόπο αντιστοίχησης δημόσιων κλειδιών με χρήστες, κάθε ένας εκ των οποίων έχει έναν συγκεκριμένο ρόλο και μία μοναδική ταυτότητα.

Τα απαραίτητα χαρακτηριστικά ασφάλειας μιας διαδικτυακής εφαρμογής σε διαδικασίες μπορούν να περιγραφούν σε [33]:

- **Εμπιστευτικότητα** (Confidentiality): Είναι η διαδικασία διασφάλισης της ανάγνωσης των δεδομένων μόνον από εξουσιοδοτημένους χρήστες. Η κρυπτογράφηση χρησιμοποιείται συχνά για να επιβάλλει προστασία της εμπιστευτικότητας. Η διαρροή ευαίσθητων πληροφοριών μπορεί να γίνει με πιο παραδοσιακές μεθόδους από την ψηφιακή υποκλοπή. Για παράδειγμα: με την κλοπή φορητών υπολογιστών από το κατάλληλο τμήμα μιας εταιρίας.

  Άλλες εκφάνσεις της εμπιστευτικότητας είναι η ιδιωτικότητα (privacy): προστασία των δεδομένων προσωπικού χαρακτήρα, δηλαδή αυτών που αφορούν συγκεκριμένα πρόσωπα, και η μυστικότητα (secrecy): προστασία των δεδομένων που ανήκουν σε έναν οργανισμό.
- **Ακεραιότητα** (Integrity): Είναι η διαδικασία διασφάλισης της τροποποίησης ή διαγραφής των δεδομένων μόνον από εξουσιοδοτημένους χρήστες. Η αναγνώριση της προσβολής της ακεραιότητας των δεδομένων, συνήθως παρέχεται με τη χρήση συναρτήσεων κατακερματισμού. Για παράδειγμα, μια εφημερίδα που δημοσιεύει τα άρθρα της και στο Διαδίκτυο θα ήθελε αυτά τα άρθρα να είναι ασφαλή από

μετατροπές ενός χάκερ που επιθυμεί να εισάγει λανθασμένες πληροφορίες στα κείμενα.

- **Διαθεσιμότητα** (Availability): Σημαίνει ότι οι πόροι (π.χ. υπολογιστικοί, αποθηκευτικοί, δικτυακοί) παραμένουν διαθέσιμοι όποτε τους χρειάζονται οι εξουσιοδοτημένοι χρήστες. Αυτό σημαίνει ότι οι εξουσιοδοτημένοι χρήστες των υπολογιστικών συστημάτων και των επικοινωνιακών μέσων δεν αντιμετωπίζουν προβλήματα άρνησης εξυπηρέτησης (denial of service) όταν επιθυμούν να προσπελάσουν τους πόρους του συστήματος. Η διαθεσιμότητα καλύπτει περιοχές πέρα από το φυσικό σκοπό της ασφάλειας. Για παράδειγμα, ένα μεγάλο μέρος της τεχνολογίας που απαιτείται για τη διασφάλιση της διαθεσιμότητας προέρχεται από άλλες περιοχές, όπως fault – tolerant computing. Για τους σκοπούς της ασφάλειας, μας απασχολεί βασικά η παρεμπόδιση κακόβουλων επιθέσεων που αποσκοπούν στο να παρακωλύσουν την πρόσβαση των νόμιμων χρηστών σε ένα πληροφοριακό σύστημα. Αυτές οι επιθέσεις ονομάζονται επιθέσεις άρνησης παροχής υπηρεσιών (denial of service attacks). Η άρνηση παροχής υπηρεσιών σημαίνει παρεμπόδιση της εξουσιοδοτημένης προσπέλασης πληροφοριών και πόρων ή πρόκληση καθυστέρησης των λειτουργιών που είναι κρίσιμες στο χρόνο (time - critical). Η αντιμετώπισή τους αποσκοπεί στο να υπερνικήσει την σκόπιμη (που προκαλείται από κακόβουλα μέρη) παρά τυχαία απώλεια της διαθεσιμότητας.
- **Αυθεντικοποίηση** (Authentication): Είναι η διαδικασία της επιβεβαίωσης της ταυτότητας των πελατών. Πελάτες μπορεί να είναι οι τελικοί χρήστες, άλλες υπηρεσίες, διαδικασίες, ή υπολογιστές.
- **Εξουσιοδότηση** (Authorization): Είναι η διαδικασία που διέπει τα μέσα και τις λειτουργίες ελέγχου πρόσβασης σε πόρους από αυθεντικοποιημένους πελάτες. Οι πόροι περιλαμβάνουν αρχεία, βάσεις δεδομένων, πίνακες, κ.ά., σε συνδυασμό με πόρους σε επίπεδο συστήματος, όπως κλειδιά μητρώου (registry keys) και δεδομένα ρυθμίσεων (configuration data).
- **Αδυναμία Αποποίησης** (Non-Repudiation): Ένα αποτελεσματικό σύστημα επιθεώρησης και καταγραφής μπορεί να είναι το κλειδί για μια υπηρεσία αδυναμίας αποποίησης. Μια υπηρεσία αδυναμίας αποποίησης εγγυάται ότι ο πελάτης δεν μπορεί να αποποιηθεί (αρνηθεί) την ευθύνη για την εκτέλεση μιας ενέργειας από μέρους του (π.χ. μιας ηλεκτρονικής συναλλαγής).

3.4 Τύποι επιθέσεων

Τα **είδη απειλών** για μια διαδικτυακή εφαρμογή μπορούν να ταξινομηθούν στις παρακάτω κατηγορίες [34]:

Πλαστογράφηση: Είναι η προσπάθεια του επιτιθέμενου να αποκτήσει πρόσβαση σε ένα σύστημα, χρησιμοποιώντας μια ψεύτικη ταυτότητα χρήστη. Για παράδειγμα, αυτό μπορεί να επιτευχθεί με τη χρήση κλεμμένων διαπιστευτηρίων ή με μια ψεύτικη διεύθυνση IP (IP spoofing).

Αλλοίωση: Αφορά τη μη εξουσιοδοτημένη τροποποίηση των δεδομένων της εφαρμογής. Για παράδειγμα, τροποποίηση μεταδιδόμενων δεδομένων μεταξύ δύο εξυπηρετητών.

Αποποίηση: Σχετίζεται με τη δυνατότητα των χρηστών να αρνούνται ότι έπραξαν συγκεκριμένες ενέργειες (π.χ. ηλεκτρονικές συναλλαγές). Χωρίς επαρκή έλεγχο και καταγραφή, οι επιθέσεις αποποίησης είναι δύσκολο να αποδειχθούν.

Δημοσιοποίηση πληροφοριών: Είναι η ανεπιθύμητη έκθεση ευαίσθητων δεδομένων. Τέτοια δεδομένα μπορεί να βρίσκονται αποθηκευμένα σε κρυφά πεδία φόρμας συμπλήρωσης στοιχείων, σε σχόλια ενσωματωμένα στον κώδικα ιστοσελίδων, τα οποία περιλαμβάνουν λεπτομέρειες σύνδεσης στη βάση δεδομένων ή λεπτομέρειες σχετικά με τη διαχείριση εξαιρέσεων και μπορεί να οδηγήσουν σε αποκάλυψη κρίσιμων λεπτομερειών για την εσωτερική δομή της εφαρμογής. Οποιαδήποτε από αυτές τις πληροφορίες μπορεί να είναι πολύ χρήσιμη για τον εισβολέα.

Άρνηση παροχής εξυπηρέτησης: Αφορά τη διαδικασία κατά την οποία ένα σύστημα ή μια εφαρμογή δεν είναι διαθέσιμη στους εξουσιοδοτημένους χρήστες της. Για παράδειγμα, μια επίθεση άρνησης εξυπηρέτησης θα μπορούσε να επιτευχθεί με «βομβαρδισμό» ενός εξυπηρετητή με αιτήματα τα οποία καταναλώνουν όλους τους διαθέσιμους πόρους του.

Κλιμάκωση δικαιωμάτων: Επιχειρείται όταν ένας χρήστης με περιορισμένα δικαιώματα χρησιμοποιεί την ταυτότητα ενός προνομιούχου χρήστη για να αποκτήσει μη εξουσιοδοτημένη πρόσβαση σε μια εφαρμογή. Για παράδειγμα, ένας εισβολέας με περιορισμένα δικαιώματα μπορεί να αναβαθμίσει το επίπεδο πρόσβασής του ή το επίπεδο των προνομίων του και να αναλάβει τον έλεγχο μιας εξαιρετικά κρίσιμης διεργασίας.

3.5 Κατηγορίες Απειλών

Σε σχέση με την προέλευσή τους, οι απειλές εντάσσονται στις τρεις ακόλουθες κατηγορίες [35]:

Φυσικές απειλές: Τέτοιου είδους καταστροφές (φωτιά, πλημμύρα κλπ.) δεν είναι πάντα δυνατόν να αποτραπούν. Όμως είναι σημαντικό η εκδήλωση παρόμοιων γεγονότων να διαπιστώνεται έγκαιρα, ώστε να ελαχιστοποιούνται οι πιθανότητες δραματικών ζημιών. Όπως επίσης σημαντικό είναι να αποφεύγονται ενέργειες που αυξάνουν την πιθανότητα εκδήλωσής τους (όπως για παράδειγμα, το κάπνισμα). Τέλος, η ετοιμότητα χρήσης εφεδρικού συστήματος, σε συνδυασμό με τη λήψη τακτικών εφεδρικών αρχείων (back - ups) για τα κρίσιμα δεδομένα, περιορίζει τις πιθανές δυσάρεστες συνέπειες.

Ακούσιες απειλές: Προκαλούνται είτε από αστοχίες υλικού ή λογισμικού (HW/SW failures), είτε από άγνοια ή αδιαφορία του ανθρώπινου παράγοντα. Σημαντικός παράγοντας πρόκλησης τέτοιων απειλών είναι η έλλειψη σωστής εκπαίδευσης, είτε πρόκειται για απλούς χρήστες είτε για διαχειριστές των συστημάτων. Να σημειωθεί ότι το ποσοστό των προβλημάτων που δημιουργούνται από άγνοια στα πληροφοριακά συστήματα είναι πολύ μεγαλύτερο από εκείνο που οφείλεται σε κακή πρόθεση.

Εκούσιες απειλές: Είναι αυτές που απασχολούν περισσότερο τη δημοσιότητα. Στην κατηγορία αυτή, οι κακόβουλοι χρήστες μπορεί να ανήκουν στο εσωτερικό του συστήματος (insiders), για παράδειγμα κάποιοι δυσαρεστημένοι υπάλληλοι. Είναι όμως πιθανό οι απειλές να προέρχονται από κάποιους επίδοξους εισβολείς που είναι εξωτερικοί χρήστες (outsiders). Στη περίπτωση αυτή η επιτυχία των επιθέσεων εξαρτάται κυρίως από τα μέσα που διαθέτουν δηλαδή το χρόνο, την υπολογιστική ισχύ, τις γνώσεις, τα άτομα, τα χρήματα, τις συσκευές και τα εξαρτήματα. Οι κακοήθεις χρήστες μπορεί να επιδιώκουν εκδίκηση, οικονομικό κέρδος, αναγνώριση ή λόγω ιδιοσυγκρασίας απλά τη δημιουργία προβληματικών καταστάσεων και τη διάπραξη βανδαλισμών.

3.6 Κρυπτογραφικά Εργαλεία

Μέχρι τώρα αναφέρθηκαν τα δύο σημαντικότερα κρυπτοσυστήματα που ευρέως εφαρμόζονται σήμερα. Περιγράφθηκαν οι αρχές που τα διέπουν και το είδος των κλειδιών

που χρησιμοποιούν (συμμετρικά ή ασύμμετρα). Τώρα θα περιγραφούν οι συναρτήσεις συμμετρικού κατακερματισμού και συγκεκριμένα θα αναλυθεί ο αλγόριθμος MD5 ο οποίος χρησιμοποιήθηκε στον κώδικα για την ασφάλεια των δεδομένων των χρηστών.

3.6.1 Συναρτήσεις Κατακερματισμού

Μια συνάρτηση κατακερματισμού (hash function) είναι οποιαδήποτε λειτουργία που μπορεί να χρησιμοποιηθεί για τη χαρτογράφηση δεδομένων αυθαίρετου μεγέθους σε δεδομένα σταθερού μήκους. Οι τιμές που επιστρέφονται από μια συνάρτηση κατακερματισμού ονομάζονται **τιμές hash** (hash value)**, κωδικοί κατακερματισμού** (hash codes)**, digests** ή απλά **hashes** [36] .

Μια συνάρτηση κατακερματισμού h υποδηλώνει ένα μετασχηματισμό που παίρνει ως είσοδο ένα μήνυμα m οποιουδήποτε μήκους και επιστρέφει στην έξοδο μία ακολουθία χαρακτήρων h(m) περιορισμένου μήκους την τιμή κατακερματισμού (hash value). Οι συναρτήσεις κατακερματισμού είναι συναρτήσεις με τις εξής ιδιότητες:

- Η είσοδος είναι οποιουδήποτε μήκους.
- Η έξοδος έχει περιορισμένο μήκος.
- Δεδομένου του m, ο υπολογισμός του h(m) είναι εύκολος.
- Η h είναι μη αντιστρέψιμη.
- Η h δεν είναι αμφιμονοσήμαντη (ένα προς ένα συνάρτηση).

Οι λειτουργίες Hash χρησιμοποιούνται συχνά σε συνδυασμό με έναν πίνακα κατακερματισμού (hash table), μια κοινή δομή δεδομένων (data structure) που χρησιμοποιείται στο λογισμικό υπολογιστών για γρήγορη αναζήτηση δεδομένων. Οι λειτουργίες Hash επιταχύνουν την αναζήτηση πίνακα ή βάσης δεδομένων ανιχνεύοντας διπλότυπες εγγραφές σε ένα μεγάλο αρχείο. Μια κρυπτογραφική συνάρτηση κατακερματισμού επιτρέπει σε κάποιον να επαληθεύει εύκολα ότι ορισμένοι χάρτες δεδομένων εισόδου αντιστοιχούν σε μια δεδομένη τιμή κατακερματισμού, αλλά αν τα δεδομένα εισόδου είναι άγνωστα, είναι σκόπιμα δύσκολο να την αναδημιουργήσει (ή οποιαδήποτε ισοδύναμη εναλλακτική λύση) γνωρίζοντας την αποθηκευμένη τιμή hash. Αυτό χρησιμοποιείται για τη διασφάλιση της ακεραιότητας των μεταδιδόμενων δεδομένων και

αποτελεί το δομικό στοιχείο για τα HMAC, τα οποία παρέχουν έλεγχο ταυτότητας μηνυμάτων.

Οι hash functions σχετίζονται συχνά με το άθροισμα ελέγχου (checksums), τα ψηφία ελέγχου (check digits), τα δακτυλικά αποτυπώματα (fingerprints), την συμπίεση απώλειας (lossy compression), τις λειτουργίες ταυτοποίησης (randomization functions), τους κώδικες διόρθωσης λαθών (error correcting codes) και τις ψηφιακές κρυπτογραφίες (ciphers).

Η τιμή κατακερματισμού παρουσιάζει συνοπτικά το μεγαλύτερο μήνυμα ή έγγραφο, για αυτό καλείται και σύνοψη μηνύματος (message digest). Μπορεί να ταυτιστεί με τη σύνοψη του μηνύματος σαν "ψηφιακό αποτύπωμα" ("digital fingerprint") του εγγράφου. Παραδείγματα γνωστών συναρτήσεων κατακερματισμού είναι οι MD2, MD5 και SHA [37].

3.6.2 MD5

Ο **αλγόριθμος αφομοίωση μηνυμάτων MD5** (Message-Digest algorithm 5) είναι μια ευρέως χρησιμοποιούμενη λειτουργία κατακερματισμού που παράγει μια τιμή κατακερματισμού 128 bit. Παρόλο που ο MD5 σχεδιάστηκε αρχικά για να χρησιμοποιηθεί ως κρυπτογραφική λειτουργία κατακερματισμού, μετέπειτα έχει αποδειχθεί ότι υποφέρει από εκτεταμένες ευπάθειες. Μπορεί ακόμα, να χρησιμοποιηθεί ως έλεγχος για την επαλήθευση της ακεραιότητας των δεδομένων, αλλά μόνο ενάντια στην ακούσια διαφθορά.

Ο MD5 αναπτύχθηκε το 1991 από τον Ron Rivest RCF 1321 [38], καθηγητής στο Τεχνολογικό Ινστιτούτο της Μασαχουσέτης. Σχεδιάστηκε για να αντικαταστήσει τον MD4 και είναι μια κατά πολύ βελτιωμένη έκδοση του, γι' αυτό είναι και λίγο πιο αργός. Η μόνη διάφορα είναι η χρήση τεσσάρων επαναλήψεων κατά την επεξεργασία του κάθε τμήματος. Οι απαιτήσεις σε μέγεθος τμήματος και μήκος μηνύματος παραμένουν οι ίδιες. Η κρυπτανάλυση του MD5 συνεχίζεται ακόμα, αλλά οι πρώτες εκτιμήσεις δείχνουν ότι έχει αρκετές αδυναμίες. Το Υπουργείο Εσωτερικής Ασφάλειας των ΗΠΑ (U. S. Department of Homeland Security) συστήνει τη μετάβαση στην οικογένεια των αλγορίθμων SHA, καθώς ο MD5 θεωρείται ότι δεν είναι ανθεκτικός σε συγκρούσεις (collision resistant) και δεν ενδείκνυται για σοβαρές εφαρμογές, όπως οι ψηφιακές υπογραφές. Ο MD5 χρησιμοποιείται, πλέον, κυρίως για τον έλεγχο της ακεραιότητας αρχείων που διακινούνται μέσω του Διαδικτύου [38].

Για εφαρμογές που βασίζονται σε OSI, το αναγνωριστικό αντικειμένου MD5 είναι:

md5 IDENTIFIER OBJECT :: =
iso (1) member-body (2) US (840) rsadsi (113549) digestAlgorithm (2) 5}

3.6.2.1 Ανάλυση Αλγόριθμου MD5

Παρακάτω περιγράφονται τα πέντε βήματα του αλγόριθμου κατακερματισμού MD5. Αυτή η περιγραφή έρχεται μέσω του **Ius Mentis** και οι λεπτομέρειες μπορούν να βρεθούν στο **IETF RFC 1321** [38].

Βήμα 1ο Πρόσθεση Bits γεμίσματος **(**Append Padding Bits)

Ο MD5 λειτουργεί σε κατάσταση 128 δυαδικών ψηφίων. Ο αλγόριθμος MD5 διαχωρίζει πρώτα την είσοδο σε μπλοκ των 512 bits. Εκ των οποίων τα 64 bit εισάγονται στο τέλος του τελευταίου τμήματος. Αυτά τα 64 bit χρησιμοποιούνται για την καταγραφή της διάρκειας της αρχικής εισόδου. Εάν το τελευταίο μπλοκ είναι μικρότερο από 512 bit, κάποια επιπλέον bits είναι «γεμισμένα» στο τέλος.

Ο αλγόριθμος MD5 πραγματοποιεί μια επέκταση της τελευταίας δέσμης του μηνύματος, έτσι ώστε να τη μετατρέψει σε δέσμη μήκους 512 bit. Η συμπλήρωση πραγματοποιείται ως εξής:

- Προστίθεται στα δεξιά ένα bit με τιμή 1, για να σηματοδοτήσει την αρχή του επιθέματος (postfix).
- Στη συνέχεια, προστίθενται τόσα bit με τιμή 0, όσα είναι απαραίτητα ώστε το μήκος της δέσμης να γίνει ίσο με 448 bit, modulo 512.
- Τέλος, προσαρτάται η τιμή (με μέγεθος 64-bit) του συνολικού μήκους του μηνύματος πριν την επέκτασή του.

Βήμα 2ο Μήκος προσάρτησης **(**Append Length**)**

Στη συνέχεια, κάθε μπλοκ χωρίζεται σε 16 λέξεις των 32 bits το καθένα. Αυτά ορίζονται ως M0 ... M15. Το μήνυμα είναι γεμισμένο έτσι ώστε το μήκος του να διαιρείται με το 512.

Σε αυτό το σημείο το μήκος του προκύπτοντος μηνύματος είναι ακριβές πολλαπλάσιο των 512 bits. Αντίστοιχα, αυτό το μήνυμα έχει μήκος που είναι ακριβές πολλαπλάσιο των 16 (32-bit) λέξεων. Έστω ότι το M [0 ... N-1] υποδηλώνει τις λέξεις του προκύπτοντος μηνύματος, όπου N είναι πολλαπλάσιο των 16.

Βήμα 3ο Αρχικοποίηση μνήμης MD **(**Initialize MD Buffer)

Ο MD5 χρησιμοποιεί ένα buffer που αποτελείται από τέσσερις λέξεις (A, B, C, D) για τον υπολογισμό του μεγέθους του μηνύματος. Καθένα από τα A, B, C, D έχει μήκος 32bit. Αρχικοποιούνται ως:

λέξη A: 01 23 45 67
λέξη B: 89 ab cd ef
λέξη C: fe dc ba 98
λέξη D: 76 54 32 10

Το MD5 χρησιμοποιεί περαιτέρω έναν πίνακα K που έχει 64 στοιχεία. Ο αριθμός στοιχείου i υποδεικνύεται ως Ki. Ο πίνακας υπολογίζεται εκ των προτέρων για να επιταχυνθούν οι υπολογισμοί. Τα στοιχεία υπολογίζονται χρησιμοποιώντας τη μαθηματική συνάρτηση sin:

Ki = abs (sin (i + 1)) * 232

Βήμα 4ο Μήνυμα επεξεργασίας σε μπλοκ 16 λέξεων

Τέσσερις βοηθητικές λειτουργίες

Επιπλέον, το MD5 χρησιμοποιεί τέσσερις βοηθητικές λειτουργίες, κάθε μία από τις οποίες λαμβάνει ως είσοδο τρεις λέξεις 32-bit και παράγουν ως έξοδο μία λέξη 32-bit. Εφαρμόζουν τους λογικούς χειριστές and, or, not και xor στα bits εισόδου.

F(X,Y,Z) = (X and Y) or (not(X) and Z)
G(X,Y,Z) = (X and Z) or (Y and not(Z))
H(X,Y,Z) = X xor Y xor Z
I(X,Y,Z) = Y xor (X or not(Z))

Σε κάθε θέση δυαδικού ψηφίου F ενεργεί ως conditional:

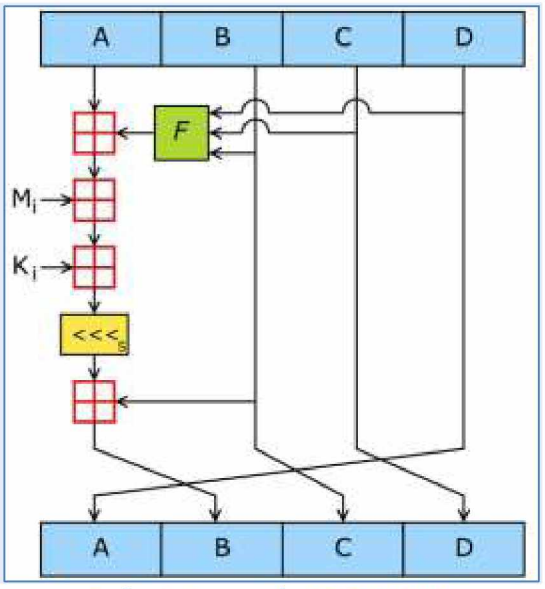
if X then Y else Z

Η συνάρτηση F θα μπορούσε να έχει καθοριστεί χρησιμοποιώντας το + αντί του v από το XY και όχι (X) Z δεν θα έχουν ποτέ 1 στην ίδια θέση bit. Είναι ενδιαφέρον να σημειωθεί ότι αν τα δυαδικά ψηφία των X, Y και Z είναι ανεξάρτητα και αμερόληπτα, κάθε bit του F (X, Y, Z) θα είναι επίσης ανεξάρτητο και αμερόληπτο.

Οι λειτουργίες G, H, και I είναι παρόμοιες με τη συνάρτηση F, από το ότι αυτές ενεργούν σε "bitwise parallel" για να παράγουν την έξοδο τους από τα bits του X, Y και Z, με τέτοιο τρόπο ώστε εάν τα αντίστοιχα δυαδικά ψηφία των X, Y, και Z είναι ανεξάρτητες και αμερόληπτες, τότε κάθε bit του G (X, Y, Z), H (X, Y, Z) και I (X, Y, Z) θα είναι ανεξάρτητα και αμερόληπτα.

Επεξεργασία των μπλοκ

Τα περιεχόμενα των τεσσάρων buffer (A, B, C και D) αναμιγνύονται τώρα με τις λέξεις της εισόδου, χρησιμοποιώντας τις τέσσερις βοηθητικές λειτουργίες (F, G, H και I). Υπάρχουν τέσσερις γύροι, ο καθένας περιλαμβάνει 16 βασικές λειτουργίες.



Εικόνα 1: Λειτουργία MD5 RFC 1321

(https://commons.wikimedia.org/wiki/File:MD5.png)

Το σχήμα δείχνει πως η βοηθητική λειτουργία F εφαρμόζεται στα τέσσερα buffer (A, B, C και D), χρησιμοποιώντας τη λέξη μηνύματος Mi και τη σταθερή Ki. Το στοιχείο "<<< s" υποδηλώνει μια δυαδική αριστερή μετατόπιση από s bits.

Βήμα 5ο Έξοδος

Μετά την εκτέλεση όλων των γύρων, τα buffer A, B, C και D περιέχουν το digest MD5 της αρχικής εισόδου. Δηλαδή, ξεκινούν με το byte χαμηλής τάξης του A και τελειώνουν με το byte υψηλής τάξης του D.

Έτσι, το MD5 έχει *πέντε βήματα* με *τέσσερις κύκλους* υπολογισμών που υπολογίζουν την κατακερματισμό της τιμής εισόδου.

Το MD5 χρησιμοποιείται από κοινού στο κέντρο δεδομένων για αρκετό καιρό. Όπως προαναφέρθηκε, δεν είναι η πρώτη ούτε η καλύτερη λειτουργία κατακερματισμού. Η σχετικά μικρή σύνταξη μηνυμάτων 128 bit κάνει γρήγορο τον υπολογισμό, αλλά δεν είναι κατάλληλο για χρήση στην κρυπτογραφία.

Στην κρυπτογραφία, μια "σύγκρουση" είναι όταν δύο ξεχωριστές τιμές εισόδου παράγουν το ίδιο hash. Αυτό είναι κακό, διότι αν υπάρξουν συγκρούσεις τότε ο αλγόριθμος μπορεί να παραβιαστεί. Το 1996, βρέθηκαν συγκρούσεις στο MD5. Ωστόσο, το MD5 εξακολουθεί να χρησιμοποιείται ευρέως για ορισμένες εφαρμογές, συμπεριλαμβανομένης της κρυπτογράφησης HTTPS.

3.7 Openssl

Το OpenSSL προέρχεται από τη βιβλιοθήκη SSLeay που ανέπτυξε ο Eric A. Young και Tim J. Hudson. Το πακέτο εργαλείων OpenSSL διατίθεται με άδεια χρήσης τύπου Apache. Χρησιμοποιείται για άδεια OpenSSL αλλά και άδεια SSLeay, πράγμα που σημαίνει ότι υπάρχει ελευθερία χρήσης για εμπορικούς και μη εμπορικούς σκοπούς εφόσον πληρούνται οι προϋποθέσεις και των δύο αδειών [39].

Το OpenSSL είναι μια βιβλιοθήκη κρυπτογράφησης για την υλοποίηση των πρωτοκόλλων SSL (Secure Sockets Layer) και TLS (Transport Layer Security). Το πρόγραμμα Openssl χρησιμοποιεί συναρτήσεις της βιβλιοθήκης OpenSSL για τη δημιουργία κλειδιών τόσο συμμετρικής όσο και ασύμμετρης κρυπτογράφησης, για την υλοποίηση διαδικασιών κρυπτογράφησης και αποκρυπτογράφησης καθώς και για τις διαδικασίες υπογραφής και επαλήθευσης.

Γενική μορφή σύνταξης [40]**:**

```
openssl command <command_options> <command_args>
```

3.7.1 Βασικές λειτουργίες Openssl

Στόχος της εφαρμογής είναι να δημιουργηθεί ένα περιβάλλον που να υποστηρίζει ασφαλείς συνδέσεις επικοινωνίας με τους πελάτες (secure website) [39].

- Δημιουργία πιστοποιητικώνX.509
- Δημιουργία αιτημάτων πιστοποίησης
- Πιστοποίηση κλειδιών χρηστών
- Δημιουργία λιστών ανάκλησης πιστοποιητικών (CRLs)
- Κρυπτογράφηση-αποκρυπτογράφηση

Λειτουργία Openssl

Δημιουργία ιδιωτικού κλειδιού

openssl genrsa –out <my.file.key>

Δημιουργία αίτησης για υπογραφή πιστοποιητικού (Certificate Signing Request – CSR)

openssl req -new -key – <my.file.key>

keyform PEM -out <my.file.csr>

Υπογραφή πιστοποιητικού από CA (Certification Authority)

openssl ca -in <my.file.csr> -out <my.file.crt>

OpenSSL encrypt	
Περιγραφή	
(string $data $aad = "" , string $aad = "" , string $aad = "" , string $aad = "" [, int $tag_length = 16]]]]]]) Κρυπτογραφεί τα δεδομένα δίνοντας μια μέθοδο και ένα κλειδί και επιστρέφει ένα ακατέργαστη (Raw) ή μία κωδικοποιημένα γραμματοσειρά (base64).	
Παράμετροι	
data	Τα δεδομένα πεπερασμένων στοιχείων που πρέπει να κρυπτογραφηθούν
method	Η μέθοδος κρυπτογράφησης. Για μια λίστα με τις διαθέσιμες μεθόδους κρυπτογράφησης χρησιμοποιείτε η openssl_get_cipher_methods()
key	Το κλειδί
Options	Είναι μια δυαδική διαίρεση των OPENSSL_RAW_DATA και OPENSSL_ZERO_PADDING
iv	Διάνυσμα αρχικοποίησης NULL
tag	Ετικέτα ελέγχου ταυτότητας
aad	Πρόσθετα δεδομένα ελέγχου ταυτότητας
Tag_length	Το μήκος της tag. Η τιμή του μπορεί να είναι μεταξύ 4 και 16 για GCM mode.
Επιστροφή τιμών	Επιστρέφει κρυπτογραφημένη συμβολοσειρά ή False σε περίπτωση αποτυχίας
Σφάλματα/ Εξαιρέσεις	• Εκπέμπει E Warning αν ένας άγνωστος αλγόριθμος κρυπτογράφησης διέρθει μέσω της παραμέτρου **method** • Εκπέμπει E Warning αν περάσει μια κενή τιμή μέσω της παραμέτρου **iv**

Πίνακας 1: OpenSSL encrypt-Encrypts data (PHP 5>=5.3.0, PHP 7) [41]

3.7.2 AES αλγόριθμος

Όλοι οι συμμετρικοί αλγόριθμοι κρυπτογράφησης μπλοκ (symmetric block cipher algorithms) έχουν κοινά χαρακτηριστικά και μεταβλητές, συμπεριλαμβανομένης της λειτουργίας, του μεγέθους του κλειδιού, των αδύναμων πλήκτρων, του μεγέθους του μπλοκ και τους γύρους.

Μέγεθος κλειδιού και αριθμός γύρων

Το AES υποστηρίζει τρία μεγέθη κλειδιών: 128 bits, 192 bits και 256 bits (αντίστοιχα AES-128, AES-192 και AES-256). Το προεπιλεγμένο μέγεθος κλειδιού είναι 128 bits και όλες οι υλοποιήσεις πρέπει να υποστηριχθούν από αυτό το μέγεθος κλειδιού [42]. Οι υλοποιήσεις μπορούν επίσης να υποστηρίζουν τα μεγέθη των κλειδιών 192bits και 256 bits. Όσο αυξάνεται το μέγεθος του κλειδιού τόσο αυξάνεται η ισχύς του αλγορίθμου. Η NSA που υιοθέτησε τον AES για την επικοινωνία των υπηρεσιών της Αμερικής, όρισε το μέγεθος 128

για επικοινωνίες που χαρακτηρίζονται “secret” και το 192 ή 256 μέγεθος κλειδιού για “top secret”.

Το AES χρησιμοποιεί διαφορετικό αριθμό γύρων για κάθε καθορισμένο μέγεθος κλειδιού.

Όταν χρησιμοποιείται:

- Ένα κλειδί 128-bit, οι υλοποιήσεις ΠΡΕΠΕΙ να χρησιμοποιούν 10 γύρους.

  AES-128: 9 encryption rounds + 1 final round

- Ένα κλειδί 192 bit, οι εφαρμογές ΠΡΕΠΕΙ να χρησιμοποιούν 12 γύρους.

  AES-192: 11 encryption rounds + 1 final round

- Ένα κλειδί 256-bit, οι υλοποιήσεις ΠΡΕΠΕΙ να χρησιμοποιούν 14 γύρους.

  AES-256: 13 encryption rounds + 1 final round

Στον πρώτο γύρο επεξεργασίας ως είσοδος είναι ένα plaintext μπλοκ και το αρχικό κλειδί, ενώ στους γύρους που ακολουθούν ως είσοδος είναι το μπλοκ που έχει προκύψει από τον προηγούμενο γύρο καθώς και ένα κλειδί που έχει παραχθεί από το αρχικό με βάση κάποια διαδικασία που ορίζει ο αλγόριθμος. Το τελικό προϊόν της επεξεργασίας είναι το κρυπτογραφημένο μπλοκ (ciphertext). Το μπλοκ αυτό έχει ακριβώς το ίδιο μέγεθος (128 bits) με το plaintext μπλοκ.

Το προηγμένο πρότυπο κρυπτογράφησης (AES) δημοσιεύθηκε ως FIPS 197 στις 26 Νοεμβρίου 2001. Η δοκιμή επικύρωσης για τη συμμόρφωση των εφαρμογών AES με το FIPS 197 ξεκίνησε τότε με το πρόγραμμα επικύρωσης αλγόριθμου κρυπτογράφησης [43].

Αναλυτική περιγραφή αλγορίθμου [42]:

- Τα KeyExpansion-round keys προέρχονται από το κλειδί κρυπτογράφησης χρησιμοποιώντας το βασικό πρόγραμμα του Rijndael. Το AES απαιτεί ξεχωριστό μπλοκ κλειδιού 128-bit για κάθε γύρο συν ένα ακόμη. Στην διαδικασία KeyExpansion, δημιουργούνται τα κλειδιά που θα χρησιμοποιηθούν στους επόμενους γύρους.
- Αρχική προσθήκη στρογγυλού κλειδιού:
- AddRoundKey - κάθε byte της κατάστασης συνδυάζεται με ένα μπλοκ του round key χρησιμοποιώντας bitwise xor (δηλαδή το αρχικό block γίνεται xor με το 1ο κλειδί).

- Εκτελούνται οι γύροι του αλγορίθμου (το πλήθος τους Nr εξαρτάται από το μέγεθος του κλειδιού).
- Τελευταίος γύρος (ίδιος με τους προηγούμενους, μόνο που τώρα δεν υπάρχει η διαδικασία MixColumns). Η αποκρυπτογράφηση γίνεται με τις αντίστροφες διαδικασίες.

Παράδειγμα κρυπτογράφησης ταυτότητας AES σε PHP

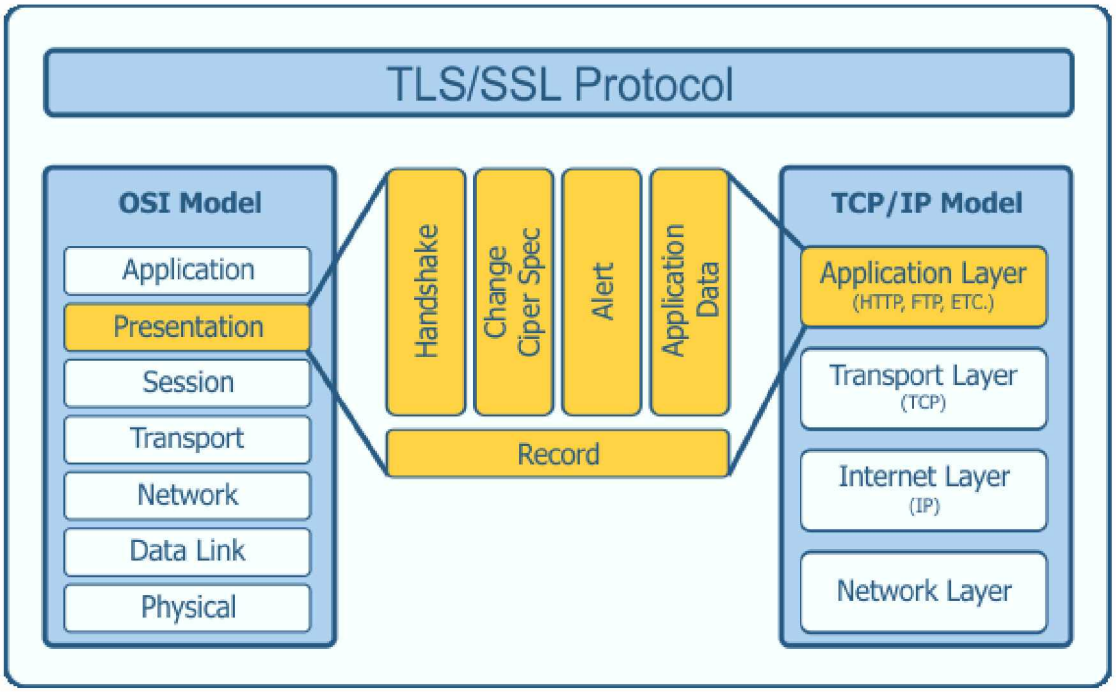
```
<?php

Το κλειδί $ θα πρέπει να έχει προηγουμένως δημιουργηθεί με
κρυπτογραφικό τρόπο, όπως το openssl_random_pseudo_bytes
$plaintext = "message to be encrypted" ;
$cipher = "aes-128-gcm" ;
if ( in_array ( $cipher , openssl_get_cipher_methods ()))
{
$ivlen = openssl_cipher_iv_length ( $cipher );
$iv = openssl_random_pseudo_bytes ( $ivlen );
$ciphertext = openssl_encrypt ( $plaintext , $cipher , $key , $options
= 0 , $iv , $tag );

Αποθήκευση $ cipher, $ iv και $ tag για αποκρυπτογράφηση μετέπειτα
$original_plaintext = openssl_decrypt ( $ciphertext , $cipher , $key ,
$options = 0 , $iv , $tag );
echo $original_plaintext . "\n" ;
}
?>
```

3.8 Κρυπτογραφικά Πρωτόκολλα TLS και SSL

Το Security Layer Security (TLS) και ο προηγούμενος προκάτοχός του, **Secure Sockets Layer (SSL)**, είναι κρυπτογραφικά πρωτόκολλα σχεδιασμένα να παρέχουν ασφάλεια επικοινωνιών μέσω ενός δικτύου υπολογιστών [44]. Ορισμένες εκδόσεις των πρωτοκόλλων βρίσκουν ευρεία χρήση σε εφαρμογές όπως περιήγηση στο web, ηλεκτρονικό ταχυδρομείο, ανταλλαγή άμεσων μηνυμάτων και φωνή μέσω IP (VoIP). Οι ιστότοποι μπορούν να χρησιμοποιήσουν το TLS για να εξασφαλίσουν όλες τις επικοινωνίες μεταξύ των διακομιστών τους και των προγραμμάτων περιήγησης ιστού.

Το TLS (Transport Layer Security) πρωτόκολλο, καθώς το πρωτόκολλο SSL (Secure Sockets Layer) λειτουργούν στο 6ο επίπεδο *μοντέλου αναφοράς ανοικτής διασύνδεσης*

συστημάτων **OSI**. Το επίπεδο αυτό ονομάζεται **επίπεδο παρουσίασης** και εξασφαλίζει ότι η πληροφορία από το επίπεδο εφαρμογής ενός συστήματος μπορεί να διαβαστεί από το επίπεδο εφαρμογής ενός άλλου συστήματος. Στο επίπεδο αυτό γίνεται στα δεδομένα κρυπτογράφηση, συμπίεση, κωδικοποίηση MIME, και όποια άλλη διαμόρφωση απαιτεί η μορφή δεδομένων ή ο σχεδιαστής του πρωτοκόλλου.



Εικόνα 2: Επίπεδο μοντέλου OSI που λειτουργούν τα TLS/SSL Protocols

(https://www.kenwalger.com/blog/iot/iot-security-ssltls/)

Επομένως, ο κύριος ρόλος του πρωτοκόλλου SSL είναι η λήψη πληροφοριών από τις εφαρμογές υψηλότερων επιπέδων, ώστε αυτές να κρυπτογραφηθούν και στη συνέχεια, η μετάδοσή τους στο διαδίκτυο προς τον ηλεκτρονικό υπολογιστή, που έθεσε το αίτημα. Από την άλλη μεριά, το TLS εγγυάται ότι κατά την επικοινωνία server - client μέσω του διαδικτύου δεν πρόκειται να μεσολαβήσει κάποιος άλλος χρήστης με σκοπό να υποκλέψει το περιεχόμενο της επικοινωνίας. Στην πραγματικότητα, τα δύο πρωτόκολλα έχουν τον ίδιο βασικό στόχο, απλά το SSL πρωτόκολλο αντικαταστάθηκε από το TLS, διότι οι αυξανόμενες ανάγκες στον τομέα της πληροφορικής οδήγησαν τα συστήματα σε πιο εξελιγμένες μορφής διάτρησης των συστημάτων με αποτέλεσμα να είναι απαραίτητος ο εκσυγχρονισμός του δικτύου για να επιτυγχάνεται ή όσο το δυνατό μέγιστη ασφάλεια για τον εκάστοτε χρήστη.

3.8.1 Πρωτόκολλο SSL

Το **Secure Sockets Layer SSL** πρωτόκολλο αναπτύχθηκε από την *Netscape Communications Corporation* για την ασφαλή επικοινωνία ευαίσθητων πληροφοριών όπως προσωπικά στοιχεία και αριθμούς πιστωτικών καρτών [45]. Η πρώτη σχεδίαση του πρωτοκόλλου έγινε τον Ιούλιο του 1994 και αποτελούσε την πρώτη έκδοση *(version 1.0)* και τον Οκτώβριο του ίδιου χρόνου δημοσιοποιήθηκε υπό την μορφή *RFC (Request For Comments)*.

Γενικά, το SSL προσφέρει συνοπτικά τις εξής διαδικασίες, όπως είναι:

- η πιστοποίηση του server από τον client,
- η πιστοποίηση του client από τον server και
- η εγκαθίδρυση ασφαλούς κρυπτογραφημένου διαύλου επικοινωνίας μεταξύ των δύο μερών.

Οι κρυπτογραφικοί αλγόριθμοι, που υποστηρίζονται από το πρωτόκολλο είναι οι εξής:

- DES - Data Encryption Standard,
- DSA - Digital Signature Algorithm,
- KEA - Key Exchange Algorithm,
- MD5 - Message Dige

Το SSL είναι σχεδιασμένο, ώστε να παρέχει διαφανείς (transparent) υπηρεσίες στο χρήστη. Ένας SSL Web server δέχεται μία αίτηση για «ασφαλή» σύνδεση σε μια θύρα (443) διαφορετική από αυτήν των απλών αιτήσεων HTTP (port 80). Το URL για συνδέσεις στην port 443 είναι της μορφής: “https://www.server.com”. Όταν ο client συνδέεται σε αυτήν την θύρα, αρχικοποιεί τη σύνοδο SSL με τη μέθοδο, η οποία καλείται χειραψία (handsake).Το SSL δημιουργεί μια σύνοδο κατά τη διάρκεια της οποίας η χειραψία πραγματοποιείται μόνο μια φορά. Όταν ολοκληρωθεί η χειραψία, η επικοινωνία κρυπτογραφείται και οι έλεγχοι ακεραιότητας εκτελούνται, έως ότου εκπνεύσει η σύνοδος SSL. Η χειραψία του SSL ομοιάζει με αυτή του TLS πρωτοκόλλου. Η σημαντικότερη διαφορά είναι ότι στο SSL πρωτόκολλο οι δύο πλευρές συμφωνούν σε μία αξιόπιστη επικοινωνία χρησιμοποιώντας την ίδια μέθοδο κρυπτογράφησης και τα ίδια κλειδιά. Στη χειραψία του TLS ο πελάτης και ο εξυπηρετητής διαπραγματεύονται προτού καταλήξουν σε κάποιο είδος κρυπτογράφησης, σε Media Access Control (MAC) και σε κρυπτογραφικά κλειδιά, κι αυτό είναι φανερό στα διάφορα μηνύματα, που ανταλλάσσονται μεταξύ τους, όπως, τα Client Hello, Server Hello, Server Key Exchange, Server Hello Done, Client Key Exchange, Change Cipher Spec και Finished.

3.8.2 Πρωτόκολλο TLS

Το πρωτόκολλο Transport Layer Security στοχεύει στο να προσφέρει ιδιωτικότητα και ακεραιότητα δεδομένων μεταξύ server-client, ώστε να αποφεύγεται η υποκλοπή των μηνυμάτων και η τροποποίησή τους από τρίτους [44].

Από την στιγμή που τα πρωτόκολλα μπορούν να λειτουργήσουν είτε με την χρήση, είτε χωρίς TLS (ή SSL), είναι απαραίτητο ο πελάτης να ειδοποιήσει τον διακομιστή για το ξεκίνημα μιας συνεδρίας TLS. Υπάρχουν δύο βασικοί τρόποι για να γίνει αυτό. Ο πρώτος είναι η χρήση μιας διαφορετικής πόρτας για τις συνδέσεις TLS, όπως για παράδειγμα η 443 για το HTTPS. Ο άλλος τρόπος είναι ο πελάτης να χρησιμοποιήσει έναν μηχανισμό που υποστηρίζει το ίδιο το πρωτόκολλο επικοινωνίας, όπως για παράδειγμα το STARTTLS στο ηλεκτρονικό ταχυδρομείο, και να ζητήσει από τον διακομιστή να μεταχειριστεί την σύνδεση ως TLS.

Μόλις ο πελάτης και ο διακομιστής συμφωνήσουν να χρησιμοποιήσουν TLS, συμφωνούν σε μία σύνδεση με πολλαπλές καταστάσεις, χρησιμοποιώντας την διαδικασία χειραψίας (Handshaking) [44]. Κατά την διάρκεια της χειραψίας, ο πελάτης και ο διακομιστής συμφωνούν σε διάφορες παραμέτρους που θα χρησιμοποιηθούν για να επιβεβαιωθεί η ασφάλεια της σύνδεσης:

- Η χειραψία ξεκινάει όταν ένας πελάτης συνδέεται σε έναν διακομιστή που υποστηρίζει TLS και αιτείται μια ασφαλή σύνδεση, παρουσιάζοντας μια λίστα με κρυπτογραφήματα, καθώς και μία λίστα με κρυπτογραφικές συναρτήσεις κατακερματισμού που ο ίδιος υποστηρίζει.
- Από αυτήν την λίστα, ο διακομιστής επιλέγει έναν αλγόριθμο κρυπτογράφησης και μια κρυπτογραφική συνάρτηση κατακερματισμού που και ο ίδιος υποστηρίζει και ενημερώνει τον πελάτη για αυτήν την απόφαση.
- Ο διακομιστής συνήθως στέλνει πίσω και κάποιο στοιχείο για να επιβεβαιώσει την ταυτότητα του, συνήθως στην μορφή ενός ψηφιακού πιστοποιητικού. Το πιστοποιητικό αυτό περιέχει το όνομα του διακομιστή, την έμπιστη Αρχή Πιστοποίησης (CA), καθώς και το δημόσιο κλειδί του για χρήση σε πράξεις με κρυπτογράφηση δημόσιου κλειδιού.

- Ο πελάτης στην συνέχεια επιβεβαιώνει την εγκυρότατα του ψηφιακού πιστοποιητικού πριν προχωρήσει.
- Για την δημιουργία του **κλειδιού συνεδρίας** που θα χρησιμοποιηθεί για την ασφάλιση της σύνδεσης, ο πελάτης είτε:
    - κρυπτογραφεί έναν τυχαίο αριθμό με το δημόσιο κλειδί του διακομιστή και στέλνει το αποτέλεσμα στον διακομιστή (το οποίο μπορεί πλέον να δει μόνο ο διακομιστής, χάρη στο ιδιωτικό κλειδί). Έτσι, και τα δύο μέρη της σύνδεσης χρησιμοποιούν έπειτα αυτόν τον τυχαίο αριθμό για να δημιουργήσουν ένα μοναδικό κλειδί συνεδρίας το οποίο θα χρησιμοποιήσουν αργότερα για κρυπτογράφηση και αποκρυπτογράφηση των δεδομένων σε αυτήν την συνεδρία.
    - χρησιμοποιεί ανταλλαγή κλειδιών Diffie-Hellman για να δημιουργήσει με ασφάλεια ένα τυχαίο και μοναδικό κλειδί συνεδρίας για την κρυπτογράφηση και αποκρυπτογράφηση δεδομένων το οποίο έχει επίσης την ιδιότητα του forward secrecy: αν το ιδιωτικό κλειδί του διακομιστή κλαπεί στο μέλλον, δεν μπορεί να χρησιμοποιηθεί για να αποκρυπτογραφηθεί η τρέχουσα συνεδρία, ακόμα και αν αυτή καταγραφεί πλήρως από κάποιον τρίτο.

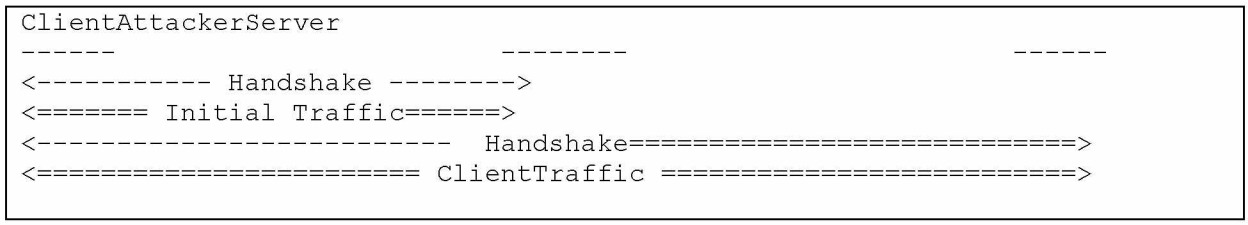
Σε αυτό το σημείο τελειώνει η χειραψία και αρχίζει η ασφαλής σύνδεση, της οποίας τα δεδομένα κρυπτογραφούνται και αποκρυπτογραφούνται με το κλειδί συνεδρίας μέχρι να τερματιστεί η σύνδεση. Αν οποιοδήποτε από τα παραπάνω βήματα αποτύχει, η χειραψία TLS αποτυγχάνει, και η σύνδεση δεν δημιουργείται

3.8.3 Επιθέσεις στο πρωτόκολλο SSL/TLS

Παρακάτω παρατίθενται προβλήματα ασφάλειας σχετικά με το πρωτόκολλο SSL/TLS.

3.2.1 Renegotiation Attack

Σε μία *Renegotiation Attack* [46] ο επιτιθέμενος προσπαθεί να εισάγει τα δικά του δεδομένα κατά την διάρκεια μιας επαναδιαπραγμάτευσης. Το TLS επιτρέπει σε ένα από τα δύο μέρη μιας σύνδεσης - server ή client - τις επαναδιαπραγματεύσεις των κρυπτογραφήσεων σε ήδη κρυπτογραφημένες συνδέσεις. Ο επιτιθέμενος αδυνατεί να αποκρυπτογραφήσει την σύνδεση, μπορεί όμως να εισάγει τα δεδομένα του. Παρακάτω παρουσιάζεται σχηματικά η επίθεση. (*Το == αναφέρεται στην κρυπτογραφημένη κίνηση*)



Πιο συγκεκριμένα, ο επιτιθέμενος δημιουργεί μια TLS σύνδεση με τον διακομιστή και εγχέει τα δεδομένα του πριν από τον πελάτη. Στη συνέχεια, επιτρέπει στον πελάτη να συνεχίσει την κρυπτογραφημένη ‘’χειραψία‘’ με τον διακομιστή. Αυτή η ‘’χειραψία‘’ δεν είναι κρυπτογραφημένη για τον επιτιθέμενο όπως είναι η ‘’χειραψία‘’ που έχει ξεκινήσει ο επιτιθέμενος με τον διακομιστή. Όταν ολοκληρωθεί η διαδικασία, ο πελάτης επικοινωνεί με τον διακομιστή χωρίς όμως να γνωρίζει για την επαναδιαπραγμάτευση και ο διακομιστής υποθέτει ότι όλη η κίνηση του δικτύου – ακόμα και η υποκλεμμένη – προέρχεται από τον πελάτη.

3.2.2 TLS Truncation Attack

Σκοπός μιας *Truncation Attack* είναι να αποτρέψει τους χρήστες να αποσυνδεθούν από μια εγκατεστημένη σύνδεση και χωρίς να το αντιληφθούν, να νομίζουν ότι έχουν αποσυνδεθεί. Χαρακτηριστικό παράδειγμα αυτής της επίθεσης είναι οι επιτυχημένες προσπάθειες των *Smyth* και *Pironti* ενάντια στο ηλεκτρονικό σύστημα ψήφων (Helios) και σε *web-based* εφαρμογές όπως email (Gmail, Hotmail) [46]. Κατά τον τερματισμό μιας TLS σύνδεσης ο πελάτης και ο διακομιστής πρέπει να στείλουν ένα μήνυμα τερματισμού (*close_notify*) καθώς οποιαδήποτε δεδομένα μετά από το σήμα τερματισμού αγνοούνται. Πιο συγκεκριμένα στο παράδειγμα της επίθεσης στην *web-based* εφαρμογή ηλεκτρονικού ταχυδρομείου της Google (Gmail) ο *Smyth* και ο *Pironti* περικόπτοντας την TLS σύνδεση χρησιμοποιώντας ένα TCP reset αίτημα, αποτρέψανε να τερματιστεί η συνεδρία μεταξύ του χρήστη και της υπηρεσίας. Εξαιτίας της κακής διαχείρισης τερματισμού του πρωτοκόλλου TLS το πρόγραμμα περιήγησης ενώ εμφάνισε ένα TEXT μήνυμα αποσύνδεσης δεν μπόρεσε να ειδοποιήσει τον χρήστη ότι δεν τερματίστηκε σωστά η σύνδεση. Έτσι, ενώ ο χρήστης

νόμιζε ότι πήρε μήνυμα επιτυχούς αποσύνδεσης ο επιτιθέμενος μπόρεσε, μέσω ενός κοινού υπολογιστή, να έχει πρόσβαση στα δεδομένα του χρήστη.

3.2.3 SSL Stripping Attack

Το *HTTPS* αναφέρεται στον συνδυασμό του *HTTP* πρωτοκόλλου και του *SSL* και δημιουργήθηκε για την προστασία έναντι HTTP επιθέσεων. Η επίθεση *SSL stripping* βασίζεται στην υποκλοπή στοιχείων σε μια επικοινωνία. Πιο συγκεκριμένα, κατά την διάρκεια μιας *SSL Stripping Attack* ο επιτιθέμενος, εκτελεί μια *MITM* (Man In The Middle) επίθεση για να βρεθεί μεταξύ του πελάτη και του διακομιστή [46].

Οι περισσότεροι χρήστες κατά την διάρκεια της πλοήγησής τους στο Διαδίκτυο, πληκτρολογούν μια URL διεύθυνση χωρίς την επιλογή του ασφαλούς πρωτοκόλλου HTTPS. Έτσι όταν σταλθεί κάποιο πακέτο, σε μια HTTP σύνδεση ο επιτιθέμενος προωθεί την κίνηση στον διακομιστή. Αν ο πελάτης επιλέξει κάποιο σύνδεσμο που χρησιμοποιούσε HTTPS, ο επιτιθέμενος ανακατευθύνει την κίνηση σε μια απλή HTTP σύνδεση και στη συνέχεια στέλνει την κίνηση στον πελάτη. Έτσι, όταν ο πελάτης πληκτρολογήσει κάποια ευαίσθητα δεδομένα (συνθηματικά, αριθμοί πιστωτικής κάρτας κλπ.) ο επιτιθέμενος μπορεί να υποκλέψει αυτές τις πληροφορίες αφού δεν υπάρχει κρυπτογράφηση. Στη συνέχεια, αφού ο επιτιθέμενος λάβει τα πακέτα από τον πελάτη, τα κωδικοποιεί και τα στέλνει στον διακομιστή. Έτσι ενώ ο φυλλομετρητής του πελάτη χρησιμοποιεί μια HTTP σύνδεση, ο διακομιστής χρησιμοποιεί HTTPS σύνδεση και δεν μπορεί να αντιληφθεί την επίθεση.

Σύνοψη

Με την χρήση MD5 κρυπτογραφήθηκαν τα στοιχεία του Login των χρηστών στην αρχική οθόνη. Ακόμα, και αν κάποιος επιτιθέμενος καταφέρει να σπάσει το σύστημα αυτό δεν θα κατορθώσει να πάρει τα προσωπικά στοιχεία των υπαλλήλων. Αυτά με την σειρά τους έχουν κρυπτογραφηθεί με openssl AES-256-CBC.

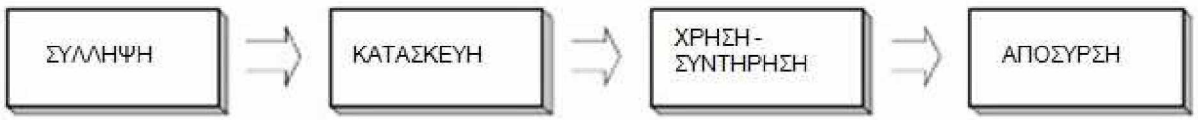
ΚΕΦΑΛΑΙΟ 4 – ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ

4.1 Εισαγωγή

Λογισμικό (Software) είναι τα προγράμματα που συντονίζουν και κατευθύνουν τη λειτουργία του υπολογιστή αλλά και επεξεργάζονται τα δεδομένα. Η τεχνολογία λογισμικού (software engineering) είναι η εφαρμογή καλά θεμελιωμένων μεθόδων και η χρήση αντίστοιχων εργαλείων για την ανάπτυξη και υποστήριξη της λειτουργίας του λογισμικού [47].

Κάθε εφαρμογή λογισμικού, από τη σύλληψη μέχρι την απόσυρσή του, διέρχεται από διάφορες φάσεις σε κάθε μία εκ των οποίων πρέπει να γίνονται ορισμένες εργασίες ώστε να επιτυγχάνεται το επιθυμητό αποτέλεσμα. Σε μακροσκοπικό επίπεδο οι πολύ γενικές φάσεις είναι: σύλληψη, κατασκευή, χρήση/συντήρηση και απόσυρση και, όπως είναι εύκολα αντιληπτό, λαμβάνουν χώρα με τη σειρά αυτή [47].

Μια δραστηριότητα ή διαδικασία ανάπτυξης λογισμικού (software process) καθορίζει ποιες ενέργειες πρέπει να γίνουν για να επιτευχθεί το επιθυμητό αποτέλεσμα σε κάποια από τις φάσεις του κύκλου ζωής. Μια δραστηριότητα μπορεί να αναλύεται σε περισσότερες από μία επιμέρους φάσεις [47].



Εικόνα 3: Γενικές φάσεις κύκλου ζωής λογισμικού

4.2 Στάδια Ανάπτυξης Λογισμικού

Τα στάδια ανάπτυξης λογισμικού είναι τα εξής:

- Ανάλυση απαιτήσεων και καθορισμός προδιαγραφών.
- Σχεδίαση συστήματος.
- Σχεδίαση προγραμμάτων.
- Συγγραφή προγραμμάτων.

- Έλεγχος μονάδων.
- Έλεγχος ολοκλήρωσης.
- Έλεγχος συστήματος.
- Παράδοση συστήματος.
- Συντήρηση.

Υπάρχουν διάφορα μοντέλα ανάπτυξης λογισμικού, που δημιουργήθηκαν για την κοινή κατανόηση των δραστηριοτήτων μέσα σε μία ομάδα ή και μεταξύ διαφορετικών ομάδων ανάπτυξης τα οποία είναι: μοντέλο καταρράκτη, μοντέλο V, μοντέλο προτυποποίησης, μοντέλο λειτουργικής επαύξησης, σπειροειδές μοντέλο, μοντέλο πίδακα και άλλα σύγχρονα μοντέλα.

Είναι σημαντικό να δημιουργείται ένα μοντέλο ανάπτυξης στην αρχή μιας διαδικασίας. Με αυτόν τον τρόπο εντοπίζονται και διορθώνονται ασυνέπειες και λάθη που μπορεί να γίνουν και με αυτόν τρόπο γίνεται πιο αποτελεσματική η λειτουργικότητα του τελικού προϊόντος. Η δημιουργία ενός τέτοιου μοντέλου βοηθάει στην υψηλή ποιότητα και στην ελαχιστοποίηση των ελαττωμάτων πολύ γρήγορα.

Στόχος του παρόντος κεφαλαίου είναι να γίνει ανάλυση των απαιτήσεων του συστήματος μελετώντας όλες τις μεθόδους ανάπτυξης λογισμικού με σκοπό την επιλογή ενός ή τον συνδυασμό περισσοτέρων και την εφαρμογή τους στην παρούσα εργασία. Η προσέγγιση αυτή περιγράφει θεωρητικά τα βήματα της μεθοδολογίας, εστιάζοντας στην καταγραφή των απαιτήσεων, την ανάλυση, τη σχεδίαση αλλά και την υλοποίηση της εφαρμογής. Συγκεκριμένα, στην παρούσα εργασία υιοθετείται ένα σύγχρονο μοντέλο ανάπτυξης το οποίο αποτελείται από τέσσερα στάδια: ανάλυση απαιτήσεων, σχεδιασμός, υλοποίηση, έλεγχος σφαλμάτων, τα οποία και θα περιγραφούν αναλυτικά στα επόμενα κεφάλαια. Η διαδικασία αυτή ακολουθεί γραμμική σειρά χωρίς αυστηρούς περιορισμούς μετάβασης και τροποποίησης σε οποιασδήποτε φάση, δίνοντας ευελιξία στους σχεδιαστές να πραγματοποιούν αλλαγές, αφαιρώντας ή προθέτοντας πληροφορίες, σε ζητήματα που μπορεί να προκύψουν σε περεταίρω στάδια.

Το εργαλείο με το οποίο πραγματοποιείται ο σχεδιασμός των λειτουργιών του συστήματος στα επόμενα κεφάλαια (όπως προκύπτει από την ανάλυση απαιτήσεων) ορίζεται

από τα αντίστοιχα διαγράμματα της Ενοποιημένης Γλώσσας Μοντελοποίησης (UML). Στη συνέχεια εξετάζεται η αξιολόγηση της ποιότητας του παραγόμενου σχεδίου, θέμα που απασχολεί την ομάδα ανάπτυξης λογισμικού, και για την ολοκλήρωση του συστήματος πραγματοποιείται μια συνοπτική επισκόπηση του συνολικού αποτελέσματος.

4.3 Απαιτήσεις Λογισμικού

Μια απαίτηση από το λογισμικό είναι μια λειτουργία που αυτό θα πρέπει να επιτελεί ή μια συνθήκη που θα πρέπει να ικανοποιεί όταν θα έχει ολοκληρωθεί η κατασκευή του. Οι απαιτήσεις από το λογισμικό διακρίνονται σε δύο κατηγορίες. Στις λειτουργικές και στις μη-λειτουργικές [47].

Οι **λειτουργικές απαιτήσεις** περιγράφουν τις εργασίες (λειτουργίες) που θα πρέπει να εκτελεί το λογισμικό. Καθορίζουν πλήρως τη συμπεριφορά του συστήματος, δηλαδή τα επιθυμητά αποτελέσματα που αυτό πρέπει να παράγει ή γενικά την απόκριση που πρέπει να εμφανίζει στο περιβάλλον του όταν ισχύουν συγκεκριμένες συνθήκες [47].

Οι **μη-λειτουργικές απαιτήσεις** περιγράφουν χαρακτηριστικά που πρέπει να έχει το λογισμικό, τα οποία δεν αφορούν την εκτέλεση κάποιας λειτουργίας από αυτό. Καθορίζουν ιδιώματα εμφάνισης, περιβάλλοντος λειτουργίας, επιδόσεων και άλλα, τα οποία γενικά χαρακτηρίζουν το λογισμικό χωρίς όμως να μπορούν να ιδωθούν ως λειτουργίες που αυτό επιτελεί. Ως μη-λειτουργικές, επίσης, χαρακτηρίζονται και οι απαιτήσεις που αφορούν κάποια από τις επόμενες φάσεις του κύκλου ζωής του λογισμικού [47].

Κατά την ανάλυση των απαιτήσεων εντοπίζονται για πρώτη φορά οι απαιτήσεις από το λογισμικό και ακολουθούν έναν κύκλο ταξινόμησης, ιεράρχησης και επαλήθευσης. Αποτέλεσμα των εργασιών που εκτελούνται στη φάση αυτή είναι ένα σύνολο απαιτήσεων από το λογισμικό οι οποίες περιγράφονται με μορφή διαγραμμάτων. Η περιγραφή αυτή αποτελεί την είσοδο στο επόμενο βήμα, αυτό της διάκρισης και προδιαγραφής των απαιτήσεων από το λογισμικό [47].

Κατά τη κατανόηση του προβλήματος, ο αναλυτής υπεισέρχεται ο ίδιος στην ουσία του προβλήματος, όσο περισσότερο γίνεται. Η έκθεση των αναγκών που τίθενται δεν είναι συνήθως επαρκής για την αντίληψη της ουσίας σχετικά με το αντικείμενο που αναπτύσσεται το λογισμικό και χωρίς εξοικείωση του κατασκευαστή με την ουσία του προβλήματος, δεν

είναι δυνατή η επιτυχής ανάπτυξη του λογισμικού. Η εξοικείωση αυτή θα πρέπει να γίνει στο αρχικό στάδιο ανάπτυξης και θα σημαδέψει ακολούθως την διαδικασία ανάπτυξης του λογισμικού. Πρόκειται για μια συναρπαστική, ιδιαίτερα δημιουργική και συνάμα δύσκολη εργασία που καθιστά το ρόλο του δημιουργού λογισμικού περισσότερο ενδιαφέρον και λιγότερο μονότονο [47].

Ακολούθως συλλέγονται οι απαιτήσεις των εμπλεκομένων με το λογισμικό και γίνεται μια αρχική καταγραφή τους σε λίστα. Η συλλογή αυτή γίνεται με τη βοήθεια συνεντεύξεων, ερωτηματολογίων, συζητήσεων με ειδικούς, ή με άλλους κατά περίπτωση πρόσφορους τρόπους. Η λίστα που δημιουργείται, αρχικά περιέχει τις απαιτήσεις από το λογισμικό "ατάκτως ειρημένες" και στη συνέχεια γίνεται μια πρώτη ταξινόμηση των απαιτήσεων σε ομάδες, ανάλογα με το υποσύνολο του προβλήματος που αφορούν ή με άλλο κατά περίπτωση πρόσφορο τρόπο [47].

Στο σημείο αυτό ενδεχομένως να εντοπιστούν ασυνέπειες, δηλαδή δύο ή περισσότερες απαιτήσεις, η ικανοποίηση των οποίων δε μπορεί να γίνει ταυτόχρονα, οπότε είναι αναγκαία η επίλυση «συγκρούσεων» η οποία μπορεί να επαναφέρει στο προσκήνιο τις επαφές των εμπλεκομένων με το λογισμικό. Όταν έχει ολοκληρωθεί η επίλυση των «συγκρούσεων», οι απαιτήσεις τοποθετούνται σε μια σειρά προτεραιότητας ως προς τη σειρά ικανοποίησής τους η οποία θα καθορίσει όχι μόνο τη χρονική αλληλουχία με την οποία ενσωματώνονται στο λογισμικό λειτουργίες που ικανοποιούν τις απαιτήσεις, αλλά και το ποιες από αυτές δεν θα ικανοποιηθούν καθόλου αν κάτι τέτοιο επιβληθεί από εξωτερικούς παράγοντες (όπως το κόστος) [47].

Τέλος, η διαδικασία ολοκληρώνεται με την επαλήθευση των απαιτήσεων, όπως έχουν διαμορφωθεί και ιεραρχηθεί. Στη καλή των περιπτώσεων οι απαιτήσεις ικανοποιούν τους εμπλεκόμενους και μπορεί να ξεκινήσει η κατασκευή των μοντέλων περιγραφής λογισμικού [47].

Στις ενότητες που ακολουθούν εμβαθύνουν στα οφέλη και περιγράφουν τομείς όπου η ανάλυση απαιτήσεων βρίσκει εφαρμογή. Συγκεκριμένα, θα αναλυθούν:

- Ο Σκοπός της εφαρμογής
- Οι εμπλεκόμενοι ρόλοι στην εφαρμογή
- Το τεχνολογικό περιβάλλον της εφαρμογής

- Οι Λειτουργίες των χρηστών ανάλογα με το ρόλο που τους έχει τεθεί και οι λειτουργίες του συστήματος.
- Οι μη λειτουργικές απαιτήσεις που προκύπτουν από την ανάλυση απαιτήσεων.

4.4 Κατηγορίες Χρηστών – Ρόλοι

Οι κατηγορίες χρηστών είναι δύο και κατατάσσονται βάσει των δικαιωμάτων που τους έχουν δοθεί.

Χρήστες/Users: Οι χρήστες επισκέπτονται καθημερινά την ιστοσελίδα και τους δίνεται η δυνατότητα προβολής, ανάγνωσης, αναζήτησης και καταχώρησης κάποιας λειτουργίας (batch). Η συμβολή τους είναι σημαντική καθώς χάρη σε αυτούς η εφαρμογή έχει πόρους και δεδομένα για να εξυπηρετήσει τον σκοπό της.

Γενικός Διαχειριστής (Administrator): Ο Administrator είναι αυτός που μπορεί να δώσει δικαίωμα σε κάποιον user να έχει πρόσβαση στην εφαρμογή, να τον διαγράψει και να επεξεργαστεί το προφίλ του. Ακόμα, έχει την δυνατότητα να δημιουργήσει νέους τύπους χρηστών και είναι υπεύθυνος για την σωστή λειτουργία της εφαρμογής.

4.4.1 Λειτουργίες Χρήστη:

- Καταχώρηση batch file
  - Επιλογή αλυσίδας καταχώρησης
  - Εισαγωγή ώρας εκκίνησης διαδικασίας
  - Εισαγωγή ώρας ολοκλήρωσης διαδικασίας
  - Επιλογή κουμπιού για τον υπολογισμό διάρκειας ολοκλήρωσης διαδικασίας
  - Εισαγωγή counter
  - Εισαγωγή throughput

- Εισαγωγή σχολίου
- Επιλογή κουμπιού καταχώρησης batch

- Αναζήτηση λειτουργίας batch file
  - Επιλογή αλυσίδας
  - Επιλογή κατηγορίας ημερομηνίας
  - Εισαγωγή ημερομηνίας
- Προβολή ερωτήσεων / απαντήσεων λειτουργίας της ιστοσελίδας
- Επικοινωνία με το τμήμα εξυπηρέτησης προσωπικού

4.4.2 Λειτουργίες Γενικού Διαχειριστή (Administrator):

- Όλες τις λειτουργίες χρήστη
- Όλα τα δικαιώματα τροποποίησης / μορφοποίησης της ιστοσελίδας
  - Τεχνικό μέρος (κώδικας και άλλες εφαρμογές)
  - Λειτουργικό μέρος (μενού, δομή, εμφάνιση και άλλα)
- Προσθήκη νέου χρήστη
  - Εισαγωγή ονόματος υπαλλήλου
  - Εισαγωγή επίθετο υπαλλήλου
  - Εισαγωγή τηλέφωνο υπαλλήλου
  - Εισαγωγή email υπαλλήλου
  - Εισαγωγή αριθμό μητρώου υπαλλήλου
  - Εισαγωγή όνομα χρήστη υπαλλήλου
  - Εισαγωγή κωδικό υπαλλήλου
  - Επιλογή τύπου λογαριασμού
- Επεξεργασία χρηστών
  - Επεξεργασία προφίλ χρηστών

- Διαγραφή χρηστών
- Αναζήτηση χρηστών
- Export δεδομένων σε CVS, Excel
- Επιλογή Copy και Print των δεδομένων

- Επεξεργασία τύπου χρηστών
  - Δημιουργία τύπου χρήστη
  - Επεξεργασία τύπου χρήστη
  - Διαγραφή τύπου χρήστη
- Ορισμός δικαιωμάτων χρηστών

4.5 Λειτουργικές Απαιτήσεις

Α / Α	Απαίτηση	Περιγραφή
1	Καταχώρηση λειτουργίας	Ο Χρήστης καταχωρεί τα στοιχεία μιας αλυσίδας.
2	Διαχείριση χρηστών	Ο Διαχειριστής επεξεργάζεται όλες λειτουργίες που αφορούν τους χρήστες του συστήματος.

Πίνακας 2: Λειτουργικές απαιτήσεις συστήματος

4.5.1 Λειτουργία 1: «Καταχώρηση Λειτουργίας»

Επισκεπτόμενος την εφαρμογή ένας χρήστης, είναι σε θέση να καταχωρήσει μια λειτουργία μιας αλυσίδας που απαιτείται για την εξασφάλιση της ομαλής λειτουργίας της επιχείρησης. Αυτό επιτυγχάνεται συμπληρώνοντας μια φόρμα που εμπεριέχει όλες τις απαραίτητες πληροφορίες προς αποθήκευση ενός batch.

Τα στοιχεία που καλείται να συμπληρώσει ο χρήστης είναι τα εξής:

Α / Α	Πεδίο	Περιγραφή
1	Όνομα αλυσίδας	Ο χρήστης επιλέγει μια αλυσίδα από την προκαθορισμένη λίστα.
2	Ώρα εκκίνησης διαδικασίας	Ο χρήστης επιλέγει την ώρα που ξεκίνησε η εκτέλεση του batch.

3	Ημερομηνία εκκίνησης διαδικασίας	Ο χρήστης επιλέγει την ημερομηνία που ξεκίνησε η εκτέλεση του batch.
4	Ώρα ολοκλήρωσης διαδικασίας	Ο χρήστης επιλέγει την ώρα που ολοκληρώθηκε η εκτέλεση του batch.
5	Ημερομηνία ολοκλήρωσης διαδικασίας	Ο χρήστης επιλέγει την ημερομηνία που ολοκληρώθηκε η εκτέλεση του batch.
6	Υπολογισμός ολοκλήρωσης	Ο χρήστης πατάει το κουμπί «Υπολογισμός ολοκλήρωσης». Υπολογίζεται αυτόματα η συνολική διάρκεια εκτέλεσης της αλυσίδας.
7	Εισαγωγή Counter	Ο χρήστης εισάγει τον αριθμό μέτρησης του batch.
8	Εισαγωγή Throughput	Ο χρήστης εισάγει τον αριθμό Throughput.
9	Εισαγωγή σχολίου	Ο χρήστης έχει την δυνατότητα εισαγωγής σχολίου σχετικά με την διαδικασία εκτέλεσης της αλυσίδας.
10	Αποθήκευση καταχώρησης	Ο χρήστης αποθηκεύει τα δεδομένα στην βάση.

Πίνακας 3: Πεδία φόρμας υποβολής στοιχείων

4.5.2 Λειτουργία 3: «Αναζήτηση Καταχωρημένων Λειτουργιών»

Έπειτα από τη διαδικασία υποβολής, το σύστημα δίνει κάποιες επιπλέον δυνατότητες στους χρήστες ώστε να συντείνουν στην βελτίωση της απόδοσής του. Η Αναζήτηση αποτελεί μια λειτουργία κατά την οποία οι χρήστες μπορούν να ανακτήσουν μια αποθηκευμένη καταχώρηση κατόπιν εισαγωγής μερικών κριτηρίων όπως:

Α / Α	Κριτήριο Αναζήτησης	Περιγραφή
1	Επιλογή Batch	Δυνατότητα αναζήτησης με βάση την αλυσίδα.
2	Κατηγορία ημερομηνίας	Δυνατότητα αναζήτησης με βάση την κατηγορία ημερομηνίας.
3	Ημερομηνία αναζήτησης	Δυνατότητα αναζήτησης με βάση την ημερομηνία.

Πίνακας 4: Κριτήρια αναζήτησης καταχωρημένων εγγραφών

4.5.3 Λειτουργία 5: «Εισαγωγή νέου χρήστη»

Ο Διαχειριστής του συστήματος έχει την ικανότητα να διαχειρίζεται όλες τις λειτουργίες του συστήματος. Έχει αυξημένες δυνατότητες διαχείρισης όπως να εισάγει, να αφαιρεί και να επεξεργάζεται τις λειτουργίες των χρηστών του συστήματος.

Α / Α	Δημιουργία χρήστη	Περιγραφή
1	Όνομα υπάλληλου	Εισαγωγή ονόματος υπάλληλου.
2	Επίθετο υπάλληλου	Εισαγωγή επίθετο υπάλληλου.
3	Τηλέφωνο υπάλληλου	Εισαγωγή εταιρικού τηλέφωνου υπάλληλου.
4	Αριθμός μητρώου υπάλληλου	Εισαγωγή Αριθμού Μητρώου υπάλληλου.
5	Όνομα χρήστη	Εισαγωγή username υπάλληλου για την είσοδο στο σύστημα.
6	Κωδικός πρόσβασης	Εισαγωγή password υπάλληλου για την είσοδό του στο σύστημα.
7	Τύπος λογαριασμού	Επιλογή τύπου λογαριασμού user ή administrator.

Πίνακας 5: Περιγραφή στοιχείων δημιουργίας νέου χρήστη

4.6 Μη λειτουργικές απαιτήσεις

Στην ενότητα αυτή καταγράφονται οι μη-λειτουργικές απαιτήσεις του συστήματος, όπως αυτές προέκυψαν από την ανάλυση των απαιτήσεων των χρηστών.

Περιγράφουν χαρακτηριστικά που πρέπει να έχει το λογισμικό και τα οποία δεν αφορούν την εκτέλεση κάποιας λειτουργίας από αυτό αλλά καθορίζουν ιδιότητες εμφάνισης (αισθητική, επικοινωνία με το χρήστη), επιδόσεων (αξιοπιστία, χρόνος εκτέλεσης, χρήση πόρων), υλοποίησης και άλλα.

4.6.1 Απαιτήσεις επιδόσεων

- Άμεση απόκριση του συστήματος στις εντολές του χρήστη: < 2sec.
- Μικρός χρόνος φόρτωσης σελίδων εφαρμογής μέσω PSTN σύνδεσης.

4.6.2 Απαιτήσεις διεπαφής χρήστη - Χρηστικότητα

- Αυτό-επεξηγηματικά μενού λειτουργιών.
- Λιτός και καθαρός σχεδιασμός σελίδων εφαρμογής.
- Περιορισμένα πεδία για συμπλήρωση.

4.6.3 Απαιτήσεις υλοποίησης

- Επεξηγηματικά σχόλια μέσα στον κώδικα.
- Επεξηγηματικά, ευανάγνωστα αναγνωριστικά μεταβλητών με χρήση αγγλικής γλώσσας πχ «user_name» αντί για «onoma_xristi», πάντα με χρήση πεζών γραμμάτων.
- Χρήση εσοχών στον κώδικα της εφαρμογής.

4.6.4 Απαιτήσεις τεκμηρίωσης

- Οδηγός εγκατάστασης και χρήσης.
- Τεκμηρίωση κώδικα.
- Η εκμάθηση του συστήματος απαιτεί πάνω από 4 ώρες εκπαίδευση του προσωπικού

4.6.5 Απαιτήσεις ασφάλειας

- Αυθεντικοποίηση διαχειριστών και χρηστών.
- Διαβάθμιση των δικαιωμάτων ανάλογα με τον τύπο χρήστη.
- Κρυπτογράφηση των κωδικών πρόσβασης.
- Κρυπτογράφηση προσωπικών ευαίσθητων δεδομένων.

4.7 Χαρακτηριστικά Λογισμικού

- **Δυνατότητα Συντήρησης** (Maintainability): Εύκολη εξέλιξη του συστήματος σε περίπτωση αλλαγής των απαιτήσεων.
- **Επαληθευσιμότητα** (Verifiability): Εύκολη επαλήθευση της ορθής λειτουργίας του συστήματος.
- **Δυνατότητα Επαναχρησιμοποίησης** (Reusability): Δυνατότητα χρήσης του κώδικα της εφαρμογής για την ανάπτυξη άλλων εφαρμογών.
- **Φορητότητα** (Portability): Δυνατότητα εκτέλεσης του προγράμματος σε διαφορετικά περιβάλλοντα (λειτουργικά συστήματα, βάσεις δεδομένων).

ΚΕΦΑΛΑΙΟ 5 - ΣΧΕΔΙΑΣΜΟΣ

5.1 Εισαγωγή

Το πρόβλημα που αντιμετωπίζεται κατά την φάση της προδιαγραφής των απαιτήσεων είναι το τι θα κάνει το λογισμικό καθώς και το ποια θα είναι τα χαρακτηριστικά του ιδιώματα. Το αποτέλεσμα της φάσης αυτής είναι το έγγραφο προδιαγραφής των απαιτήσεων από το λογισμικό, καθώς και ένα σύνολο μοντέλων παράστασης λογισμικού σε μορφή διαγραμμάτων ροής δεδομένων, οντοτήτων- συσχετίσεων και μετάβασης καταστάσεων.

Στην επόμενη φάση, αυτή της σχεδίασης, θεωρείται γνωστό το τι θα κάνει το λογισμικό και αντιμετωπίζεται το πρόβλημα του πώς θα το κάνει. Όλα τα προϊόντα που έχουν παραχθεί κατά τη φάση προδιαγραφής των απαιτήσεων από το λογισμικό αποτελούν την είσοδο στη φάση της σχεδίασης, δηλαδή το υλικό με το οποίο ο σχεδιαστής λογισμικού θα χτίσει το κατασκεύασμά του.

5.2 Σκοπός της σχεδίασης

Στην ενότητα αυτή θα οριστεί το αντικείμενο της σχεδίασης του λογισμικού και θα οριοθετηθούν οι στόχοι για τη σχεδίαση. Αυτό που ζητείται από τη σχεδίαση, είναι ένας τρόπος περιγραφής της κατασκευής του λογισμικού έτσι ώστε αυτό να ικανοποιεί τις προδιαγραφές που έχουν τεθεί, δηλαδή να μπορεί να εκτελεί τις επιθυμητές λειτουργίες και να έχει τα επιθυμητά χαρακτηριστικά. Η ύπαρξη των προδιαγραφών είναι αναγκαία για να ξεκινήσει η σχεδίαση και επιβάλλεται από τις αρχές της Τεχνολογίας Λογισμικού. Η περιγραφή αυτή, που παράγετε κατά τη σχεδίαση, ονομάζεται σχέδιο του λογισμικού.

5.3 Σχέδιο λογισμικού

Σχέδιο λογισμικού είναι η περιγραφή των μονάδων που αποτελούν το λογισμικό, των συσχετίσεων μεταξύ τους, της διάταξής τους, καθώς και της εσωτερικής τους λεπτομέρειας. Η παραγωγή του σχεδίου του λογισμικού είναι αναγκαία προκειμένου να γίνει δυνατή η κατασκευή του, όπως άλλωστε ισχύει για κάθε τεχνικό έργο. Η λύση στο πρόβλημα της σχεδίασης λογισμικού δεν είναι καθόλου εύκολη. Για κάθε προδιαγραφή μπορεί να κατασκευαστούν περισσότερα του ενός σχέδια, δηλαδή να μπορεί να υλοποιηθεί με

περισσότερους από ένα τρόπους. Μερικές από τις σημαντικότερες πλευρές του προβλήματος της σχεδίασης είναι οι ακόλουθες:

- Με ποια στρατηγική πρέπει να αντιμετωπιστεί η μετάβαση από τις προδιαγραφές στη σχεδίαση έτσι ώστε η εργασία να είναι αποτελεσματική.
- Ποιος από τους τρόπους που επιλέγεται για την υλοποίηση μιας προδιαγραφής είναι ο καλύτερος και πως τεκμηριώνεται αυτό.
- Σε ποιο βαθμό δεσμεύεται η σχεδίαση από το περιβάλλον (γλώσσα προγραμματισμού, εργαλεία, λειτουργικό σύστημα) στο οποίο θα γίνει η κατασκευή του προγράμματος.
- Ποια είναι η περισσότερο επαρκής, δηλαδή κατάλληλη χωρίς να είναι ούτε ελλιπής ούτε φλύαρη, περιγραφή του σχεδίου του λογισμικού.
- Πως εμφανίζεται η ποιότητα του παραγόμενου λογισμικού μέσα από τις εργασίες που λαμβάνουν χώρα κατά τη σχεδίαση.

Σκοπός της σχεδίασης είναι να δώσει την καλύτερη δυνατή – στις εκάστοτε συνθήκες – απάντηση στα παραπάνω ερωτήματα. Ας σημειωθεί, ότι δεν υπάρχει η καλύτερη κατ' απόλυτη έννοια, λύση, παρά μόνον η καλύτερη δυνατή στις εκάστοτε συνθήκες. Η εποχή που η επιδίωξη των μηχανικών λογισμικού ήταν η εύρεση της απόλυτα καλύτερης και γενικής λύσης στο πρόβλημα της σχεδίασης, έχει δώσει τη θέσει της στο ρεαλισμό της επιδίωξης της βέλτιστης λύσης μέσα σε κάθε συγκεκριμένο περιβάλλον κατασκευής λογισμικού.

Παρά τον υποκειμενισμό που γενικά διέπει το χαρακτηρισμό ενός σχεδίου λογισμικού, είναι χρήσιμο να συμφωνούνται κριτήρια ποιότητας, στα οποία να αποδίδεται η προσήκουσα κατά περίπτωση βαρύτητα. Τέσσερα τέτοια κριτήρια είναι τα ακόλουθα:

- Το σχέδιο πρέπει να ικανοποιεί όλες τις προδιαγραφές των απαιτήσεων από το λογισμικό (λειτουργικές και μη).
- Το σχέδιο πρέπει να περιγράφει πλήρως όλες τις πλευρές του λογισμικού: δεδομένα, λειτουργίες και συμπεριφορά, όπως αυτές θεωρούνται από τη πλευρά του κατασκευαστή.
- Το σχέδιο πρέπει να είναι εύκολα κατανοητό από αυτούς που θα συγγράψουν τον πηγαίο κώδικα, δηλαδή τους προγραμματιστές.

- Το σχέδιο δε πρέπει να περιέχει σφάλματα.

Τα παραπάνω, εκτός από κριτήρια, μπορούν να ακουστούν εξίσου εύκολα και ως ευχές. Μια πρακτική σχεδίασης πρέπει να στοχεύει στη καθοδήγηση του κατασκευαστή στη παραγωγή σχεδίου λογισμικού το οποίο να πληροί στο μέγιστο δυνατό βαθμό από τα παραπάνω κριτήρια. Αυτό δεν είναι πάντα εύκολο, καθότι ενδέχεται να υπάρχουν και εσωτερικές αντιφάσεις μέσα στα κριτήρια (λόγω χάρη, απαγορευτικό από τον προϋπολογισμό κόστος πλήρους λεπτομερούς περιγραφής του σχεδίου). Η Τεχνολογία Λογισμικού παρέχει μεθοδολογίες σχεδίασης οι οποίες αποτελούν κατευθυντήριες γραμμές για τον σχεδιαστή. Συνδυάζοντας τη γνώση, την εμπειρία και τον αυτοσχεδιασμό, αλλά και με τη χρήση των κατάλληλων εργαλείων ανάπτυξης λογισμικού, ο σχεδιαστής μπορεί να αντιμετωπίσει τη πρόκληση ικανοποίησης των κριτηρίων αυτών.

5.4 Τεχνοτροπίες σχεδίασης

Το πρόβλημα της σχεδίασης λογισμικού μπορεί να αντιμετωπιστεί με διάφορες στρατηγικές προσεγγίσεις. Οι διάφορες μεθοδολογίες που έχουν παρουσιαστεί μπορούν να ενταχθούν σε δύο μεγάλες κατηγορίες: τις προσανατολισμένες-στις-διαδικασίες (function oriented) και τις προσανατολισμένες-στα-αντικείμενα (object oriented). Παρακάτω δίνεται μια σύντομη αναφορά.

5.4.1 Δομημένη σχεδίαση

Οι μεθοδολογίες που ακολουθούν αυτή τη προσέγγιση προτείνουν τρόπους αποσύνθεσης του συστήματος από πάνω προς τα κάτω (top-down) σε μια ιεραρχία διαδικασιών, συναρτήσεων και άλλων ενεργών μονάδων λογισμικού. Όσο κατεβαίνει κανείς στην ιεραρχία αυτή, τόσο μεγαλύτερη λεπτομέρεια συναντά, μέχρις ότου φτάσει στις απλές δομικές μονάδες, δηλαδή τις εντολές της γλώσσας προγραμματισμού.

Γνωστές μεθοδολογίες που ανήκουν στην οικογένεια αυτή έχουν προταθεί από πολλούς συγγραφείς. Οι περισσότερες των προσεγγίσεων αυτών επικεντρώνουν τη προσοχή τους στις διαδικασίες πρώτα και μετά στα δεδομένα. Οι πιο σύγχρονες καθορίζουν τη δομή των διαδικασιών που επιδρούν πάνω στα δεδομένα με βάση τη δομή των δεδομένων αυτών

και για τον λόγο αυτό χαρακτηρίζονται ως «βασισμένες στα δεδομένα» και συγγενεύουν εν μέρει με τις προσανατολισμένες στα αντικείμενα τεχνοτροπίες.

5.4.2 Αντικειμενοστραφής σχεδίαση

Η αντικειμενοστραφής (object-oriented) προσέγγιση ακολουθεί ένα διαφορετικό δρόμο: αντί το σύστημα να θεωρείται ως μια ιεραρχία διαδικασιών, ανεξαρτήτων από τα δεδομένα, θεωρείται ως μια συλλογή οντοτήτων, καθεμία εκ των οποίων περικλείει και διαδικασίες και δεδομένα. Η προσέγγιση βασίζεται στην ιδέα ότι στον πραγματικό κόσμο δεδομένα και διαδικασίες μπορούν να ιδωθούν ενιαία με βάση το πεδίο ευθύνης κάποιων οντοτήτων που ονομάζονται αντικείμενα. Κάθε αντικείμενο παρέχει στο περιβάλλον του ένα σύνολο υπηρεσιών της ευθύνης του. Η συνεργασία του συνόλου των αντικειμένων του πεδίου μιας εφαρμογής λογισμικού, παράγει το επιθυμητό αποτέλεσμα [48].

Μερικές από τις γνωστές προσεγγίσεις που ανήκουν στην κατηγορία αυτή προτάθηκαν από τους Meyer (1988), Booch (1994), Jacobson (1993), Rumbaugh (1992). Για μεγάλο χρονικό διάστημα επικρατούσε μια σύγχυση σε επίπεδο ορολογίας και συμβολισμών στην οικογένεια της αντικειμενοστραφούς ανάλυσης και σχεδίασης. Τα τελευταία χρόνια η σύγχυση αυτή έχει περιοριστεί με την εμφάνιση «συγχωνευμένων» μεθοδολογιών και ενοποιημένων συμβολισμών.

5.5 Περιπτώσεις χρήσης

Στις παρακάτω ενότητες παρουσιάζονται οι περιπτώσεων χρήσης της εφαρμογής όπως αυτές καθορίστηκαν από το στάδιο της ανάλυσης απαιτήσεων. Οι περιπτώσεις χρήσης οργανώνονται σε συγκεκριμένες κατηγορίες. Οι ρόλοι των χρηστών όπως καθορίστηκαν στο προηγηθέν στάδιο ανάλυσης απαιτήσεων είναι δύο: ο χρήστης και ο διαχειριστής. Ως μια από τις περιπτώσεις χρήσης θα εξετασθεί και το σύστημα καθώς εκτελεί αυτόματες ενέργειες που επεκτείνουν άλλες περιπτώσεις χρήσης ή περιλαμβάνονται σε αυτές.

5.5.1 Χρήστης

- Είσοδος στο σύστημα

- Καταχώρηση Batch file.
- Αναζήτηση Batch file.

5.5.2 Γενικός διαχειριστής

- Είσοδος στο σύστημα.
- Ορισμός δικαιωμάτων - ρόλων χρηστών.
- Προσθήκη νέων χρηστών
- Δημιουργία νέων τύπων χρηστών
- Τροποποίηση ρυθμίσεων συστήματος.

5.5.3 Σύστημα

- Αποστολή στοιχείων αιτήματος.

5.6 Ανάλυση περιπτώσεων χρήσης

Παραπάνω οι περιπτώσεις χρήσης αναφέρονται επιγραμματικά. Για να σχηματιστεί μία πιο αναλυτική περιγραφή των λειτουργιών χρειάζεται να αναλύσουμε τις περιπτώσεις χρήσης. Μια λεπτομερέστερη ανάλυση των περιπτώσεων χρήσης συμβάλλει στην αναθεώρηση του προσχεδίου του περιβάλλοντος διεπαφής και αποσαφηνίζει την ροή λειτουργιών του συστήματος. Η ανάλυση ακολουθεί την ίδια σειρά με την οποία οι περιπτώσεις χρήσης καταγράφηκαν στο κατάλογο. Στην ανάλυση περιπτώσεων χρήσης χρησιμοποιούνται φόρμες δανεισμένες από την μεθοδολογία ανάπτυξης συστημάτων ICONIX, με σκοπό να κάνουν την διαδικασία πιο δομημένη και παραστατική.

5.6.1 Δημιουργία και Αποθήκευση λειτουργίας – Χρήστης, Γενικός Διαχειριστής

Για να καταχωρήσει κάποιος μια αναφορά όταν βρίσκετε στην σελίδα, πρέπει να επιλέξει από το κεντρικό μενού την επιλογή «Καταχώρηση Συμβάντος». Όταν μεταβεί στην καρτέλα αυτή, καλείται να συμπληρώσει μια φόρμα με τα προσωπικά του στοιχεία και περιγραφές σχετικά με την τοποθεσία του προβλήματος. Στη συνέχεια επιλέγει «Αποθήκευση» για να καταχωρήσει το συμβάν. Το σύστημα εμφανίζει μήνυμα: «Η υποβολή σας είναι στην ουρά για επιμέλεια από την διαχείριση του ιστότοπου και θα δημοσιευθεί αφότου εγκριθεί».

Περιγραφή	Αποθήκευση Batch
Βασικός ρόλος	Χρήστης
Προαπαιτούμενα	Σύνδεση στην εφαρμογή
Περίπτωση εσφαλμένης εκτέλεσης	Εμφάνιση μηνύματος σφάλματος
Αποτέλεσμα σωστής εκτέλεσης	Εμφάνιση μηνύματος επιβεβαίωσης και ύστερα καταχώρηση λειτουργίας
Γεγονός εκκίνησης	Επιλογή «Καταχώρηση Batch»
Σενάριο καλής εκτέλεσης	1. Επιλογή Καταχώρηση Συμβάντος. 2. Συμπλήρωση απαραίτητων πεδίων. 3. Αποθήκευση συμβάντος
Παραλλαγές σεναρίου	
Παρατηρήσεις	Εμφάνιση μηνύματος επιβεβαίωσης από το σύστημα

Πίνακας 6: Δημιουργία και αποθήκευση αλυσίδας

5.6.2 Αναζήτηση Batch

Όταν κάποιος χρήστης θέλει να κάνει αναζήτηση μιας καταχώρησης μεταβαίνει αριστερά στο μενού «Λειτουργίες» και ύστερα επιλέγει «Αναζήτηση». Εκεί έχει τη δυνατότητα να κάνει αναζήτηση με βάση την αλυσίδα, το τύπο της ημερομηνίας[4] και την ημερομηνία που επιθυμεί. Ύστερα επιλέγει «Αναζήτηση batch file». Στη περίπτωση που

[4] Τύποι ημερομηνίας: 1) Start date, 2) End date και 3)Ημερομηνία καταχώρησης

υπάρχουν υποβολές, σύμφωνα με τα κριτήρια αναζήτησης, εμφανίζονται στην οθόνη. Αν δεν υπάρχουν συμβάντα με τα συγκεκριμένα κριτήρια τότε εμφανίζεται ο πίνακας με το μήνυμα «No data available in table».

Περιγραφή	Αναζήτηση καταχώρησης
Βασικός ρόλος	Χρήστης
Προαπαιτούμενα	Σύνδεση στην εφαρμογή
Περίπτωση εσφαλμένης εκτέλεσης	Εμφάνιση: Δεν υπάρχουν καταχωρήσεις με τα κριτήρια που επιλέξατε
Αποτέλεσμα σωστής εκτέλεσης	Προβολή καταχώρησης
Γεγονός εκκίνησης	Επιλογή «Αναζήτηση Batch file»
Σενάριο καλής εκτέλεσης	1. Επιλογή Καταχώρηση Batch 2. Συμπλήρωση απαραίτητων πεδίων 3. Αναζήτηση Batch
Παραλλαγές σεναρίου	
Παρατηρήσεις	

Πίνακας 7: Αναζήτηση καταχώρησης

5.6.3 Είσοδος Διαχειριστή στο σύστημα

Ο Διαχειριστής του συστήματος είναι αυτός ο οποίος μπορεί να διαχειριστεί όλες τις λειτουργίες στην εφαρμογή. Για να το κάνει αυτό, αρχικά, πρέπει να συνδεθεί στην εφαρμογή ως administrator. Η είσοδος του διαχειριστή στο σύστημα πραγματοποιείται από την κεντρική σελίδα της εφαρμογής. Ο διαχειριστής του συστήματος εισάγει το συνθηματικό και τον προσωπικό κωδικό πρόσβασης όπως καθορίστηκαν κατά την εγκατάσταση.

Περιγραφή	Είσοδος διαχειριστή στο σύστημα

Βασικός ρόλος	Διαχειριστής
Προαπαιτούμενα	Είσοδος στο σύστημα ως διαχειριστής
Περίπτωση εσφαλμένης εκτέλεσης	Ενημερωτικό μήνυμα για εσφαλμένη εισαγωγή στοιχείων εισόδου
Αποτέλεσμα σωστής εκτέλεσης	Εμφάνιση μενού διαχειριστή
Γεγονός εκκίνησης	Ταυτοποίηση προσωπικών δεομένων διαχείρισης
Σενάριο καλής εκτέλεσης	1. Ο γενικός διαχειριστής εισάγει τα προσωπικά στοιχεία πρόσβασης. 2. Ο γενικός διαχειριστής αποκτά πρόσβαση στην κεντρική σελίδα διαχειριστή.
Παραλλαγές σεναρίου	Αν ο κωδικός πρόσβασης του διαχειριστή περιεχομένου δεν είναι σωστός, δεν αποκτά πρόσβαση στην εφαρμογή
Παρατηρήσεις	

Πίνακας 8: Είσοδος διαχειριστή στο σύστημα

5.6.4 Ορισμός δικαιωμάτων - ρόλων χρηστών

Ο Διαχειριστής ορίζει τα δικαιώματα στα φυσικά πρόσωπα του συστήματος κατά την εισαγωγή τους. Έχει το δικαίωμα να επέμβει σε κάποιον χρήστη και από απλό user να τον κάνει administrator ή και το αντίστροφο.

Περιγραφή	Εισαγωγή νέου χρήστη και ορισμός δικαιωμάτων
Βασικός ρόλος	Διαχειριστής
Προαπαιτούμενα	1. Είσοδος στην εφαρμογή 2. Επιλογή Δημιουργία νέου χρήστη
Περίπτωση εσφαλμένης εκτέλεσης	Εμφάνιση μηνύματος σφάλματος
Αποτέλεσμα σωστής εκτέλεσης	Προσθήκη χρήστη και επιλογή τύπου χρήστη

Γεγονός εκκίνησης	Μενού → Προσθήκη χρήστη → Αποθήκευση
Σενάριο καλής εκτέλεσης	Εισαγωγή νέου χρήστη
Παραλλαγές σεναρίου	Πρόσθεση ή αφαίρεση δικαιωμάτων
Παρατηρήσεις	

Πίνακας 9: Προσθήκη νέου χρήστη και ορισμός δικαιωμάτων

5.6.5 Επεξεργασία τύπου χρηστών από Διαχειριστή

Ο Διαχειριστής είναι υπεύθυνος για τον έλεγχο όλου του συστήματος και είναι αυτός που μπορεί να δημιουργήσει νέους τύπους χρηστών εάν αυτό απαιτηθεί. Ανάλογα με την εξέλιξη της εφαρμογής πιθανά να υπάρξουν νέες ανάγκες τις οποίες μόνο αυτός έχει την ευχέρεια να υλοποιήσει σε συνεργασία με τον wed developer.

Περιγραφή	Δημιουργία και επεξεργασία τύπων χρηστών διαχείρισης της εφαρμογής
Βασικός ρόλος	Διαχειριστής
Προαπαιτούμενα	Εισαγωγή του διαχειριστή στο σύστημα
Περίπτωση εσφαλμένης εκτέλεσης	Εμφάνιση μηνύματος σφάλματος
Αποτέλεσμα σωστής εκτέλεσης	Δημιουργία νέου τύπου χρήστη
Γεγονός εκκίνησης	Ο διαχειριστής εισέρχεται στη βάση του συστήματος
Σενάριο καλής εκτέλεσης	1. Ο διαχειριστής εισέρχεται στο μενού «επεξεργασία τύπου χρήστη» 2. Εισάγει νέο τύπο χρήστη 3. Αποθηκεύετε στην βάση επιπλέον τύπος χρήστη 4. Διαγραφή τύπου χρήστη 5. Ανανέωση σελίδας

Παραλλαγές σεναρίου	Διαγραφή administrator. Αν διαγραφτεί αυτός ο τύπος δεν θα μπορεί πλέον κάποιος να διαχειρίζεται την ασφάλεια του συστήματος
Παρατηρήσεις	

Πίνακας 10: Δημιουργία και επεξεργασία τύπου χρηστών

5.6.6 Επεξεργασία πληροφοριών χρηστών από Διαχειριστή

Σε αυτό το στάδιο, ο διαχειριστής της εφαρμογής μπορεί να επέμβει και να τροποποιήσει κάποια από τα στοιχεία των εγγεγραμμένων χρηστών. Αξίζει να σημειωθεί πως ενώ μπορεί να μπει και να δει τις πληροφορίες των χρηστών τα ευαίσθητα δεδομένα όπως ο κωδικός πρόσβασης είναι κρυπτογραφημένος ούτος ώστε να μην υπάρχει τρόπος παραβίασης του λογαριασμού του.

Περιγραφή	Προβολή, έλεγχος, αναζήτηση και επεξεργασία των εγγεγραμμένων υπαλλήλων
Βασικός ρόλος	Γενικός διαχειριστής
Προαπαιτούμενα	Εισαγωγή του γενικού διαχειριστή στο σύστημα
Περίπτωση εσφαλμένης εκτέλεσης	Εμφάνιση μηνύματος σφάλματος
Αποτέλεσμα σωστής εκτέλεσης	Εμφάνιση αλλαγών στην οθόνη
Γεγονός εκκίνησης	Ο διαχειριστής μεταβαίνει στην «επεξεργασία χρηστών»
Σενάριο καλής εκτέλεσης	1. Ο διαχειριστής εισέρχεται στην οθόνη του συστήματος για να επεξεργαστεί τις πληροφορίες των χρηστών 2. Επιλογή edit ή delete χρήστη 3. Επεξεργασία στοιχείων 4. Αποθήκευση αλλαγών στην βάση
Παραλλαγές σεναρίου	Αποθήκευση των στοιχείων του πίνακα σε excel και cvs, επιλογή εκτύπωσης και αντιγραφής του περιεχομένου
Παρατηρήσεις	Δεν μπορεί να γίνει edit στο ID και τον Αριθμό Μητρώου

Πίνακας 11: Προβολή και επεξεργασία εγγεγραμμένων χρηστών

5.6.7 Τροποποίηση ρυθμίσεων συστήματος

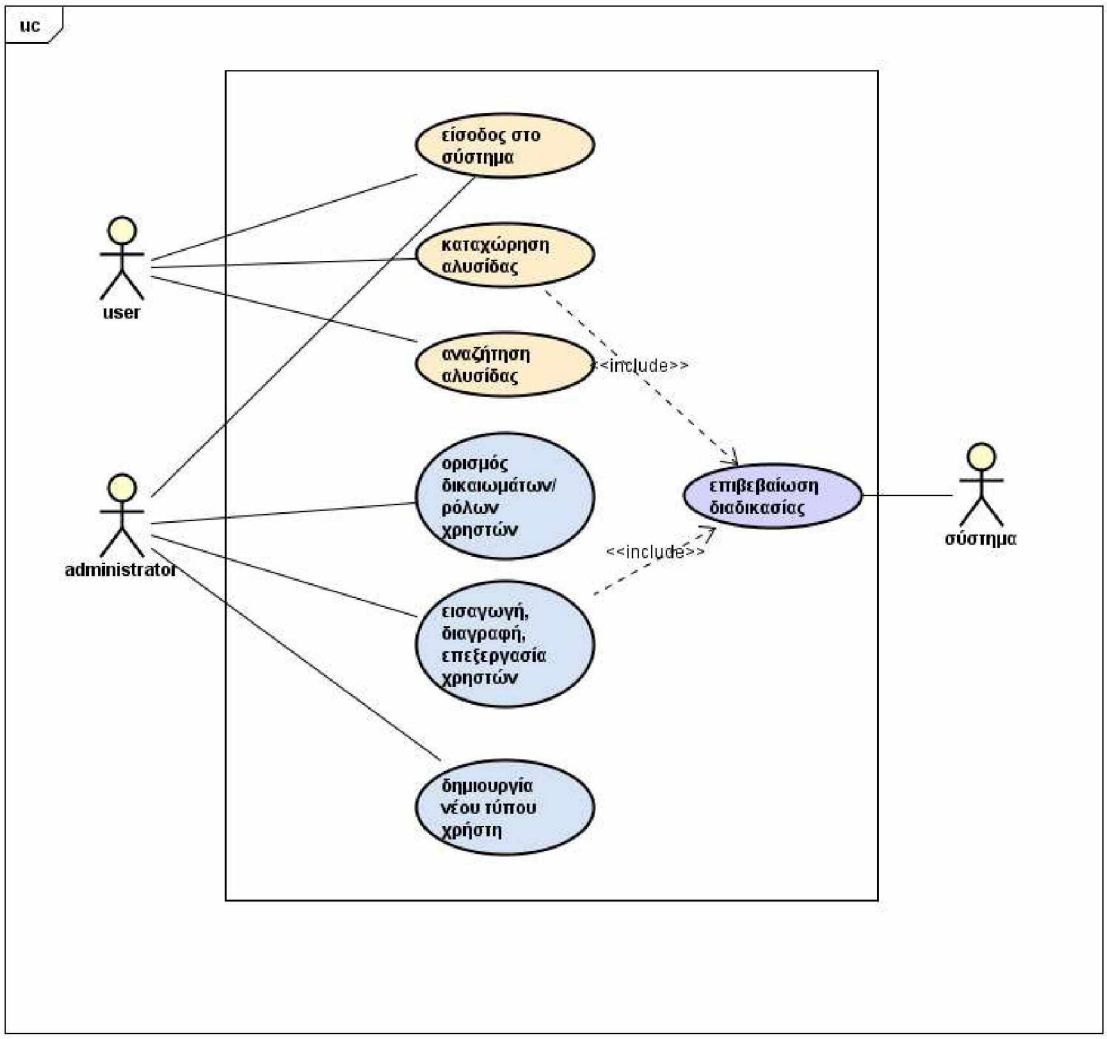
Ο μόνος που μπορεί να πραγματικά να τροποποιήσει ή ακόμα καλύτερα να ορίζει τις ρυθμίσεις του συστήματος είναι ο προγραμματιστής του συστήματος. Με εντολή του βασικού διαχειριστή (προϊστάμενου) έχει τη δυνατότητα να αλλάξει, να προσθέσει, να διαγράψει ή να τροποποιήσει οποιαδήποτε λειτουργία μέσα στη σελίδα, δια μέσω του πηγαίου κώδικα. Μπορεί να αφορά είτε την εμφάνιση είτε μια λειτουργία που έχει το σύστημα όπως για παράδειγμα ο τρόπος σύνδεσης των υπαλλήλων ή τα δικαιώματα που τους έχουν δοθεί. Αυτό γίνετε προγραμματιστικά και αποσκοπεί στην καλύτερη λειτουργικότητα του συστήματος.

Περιγραφή	Τροποποίηση ρυθμίσεων συστήματος
Βασικός ρόλος	Διαχειριστής και web developer
Προαπαιτούμενα	Εισαγωγή του προγραμματιστή στο σύστημα ανάπτυξης λογισμικού
Περίπτωση εσφαλμένης εκτέλεσης	Errors στην εκτέλεση της εφαρμογής
Αποτέλεσμα σωστής εκτέλεσης	Ανάπτυξη νέων λειτουργιών
Γεγονός εκκίνησης	Εντολή διαχειριστή για την εξέλιξη της εφαρμογής
Σενάριο καλής εκτέλεσης	1. Ο wed developer τροποποιεί ή προσθέτει νέες λειτουργίες στην εφαρμογή 2. Αποθήκευση κώδικα και ανέβασμα στην ανάπτυξη
Παραλλαγές σεναρίου	Ο χρόνος υλοποίησης να είναι μεγαλύτερος από τον προκαθορισμένο.
Παρατηρήσεις	

Πίνακας 12: Τροποποίηση ρυθμίσεων συστήματος

5.7 Διάγραμμα περιπτώσεων χρήσης

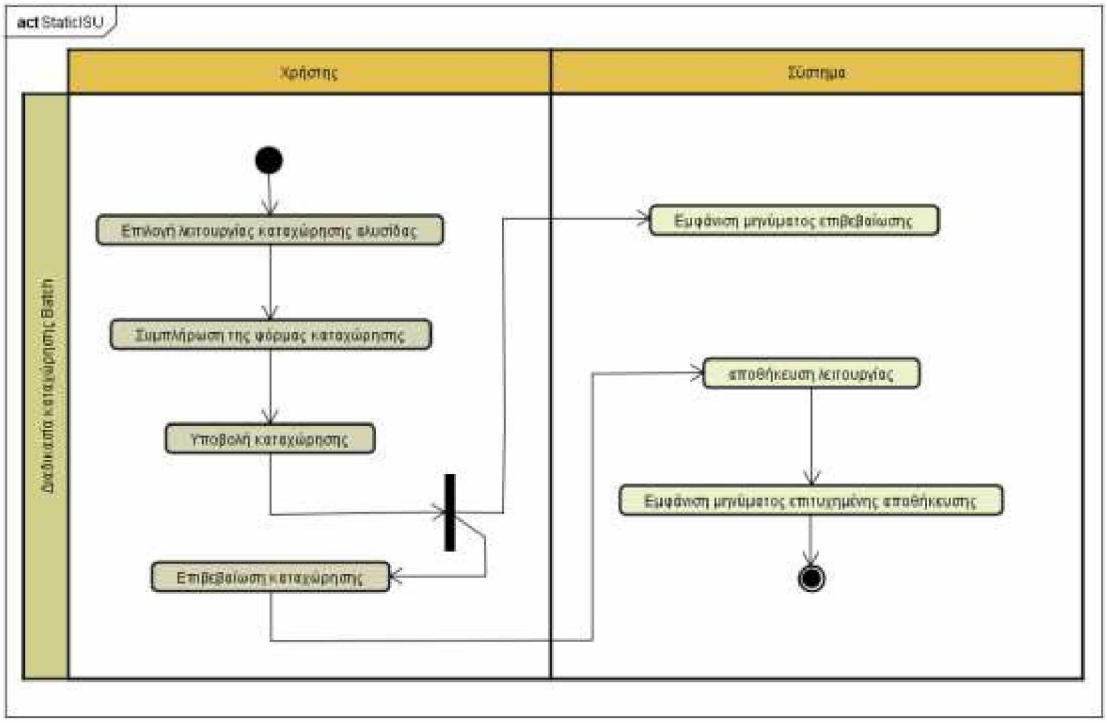
Στο συνολικό διάγραμμα περιπτώσεων χρήσης της Εικόνας 4 αποτυπώνονται καθαρά οι όψεις του συστήματος για όλες τις κατηγορίες χρηστών. Διακρίνονται και οι δύο κατηγορίες χρηστών και κατά συνέπεια τα δύο επίπεδα δικαιωμάτων πάνω στη χρήση του συστήματος.



Εικόνα 4: Διάγραμμα UML Ολική απεικόνιση συστήματος

5.8 Διαγράμματα δραστηριοτήτων

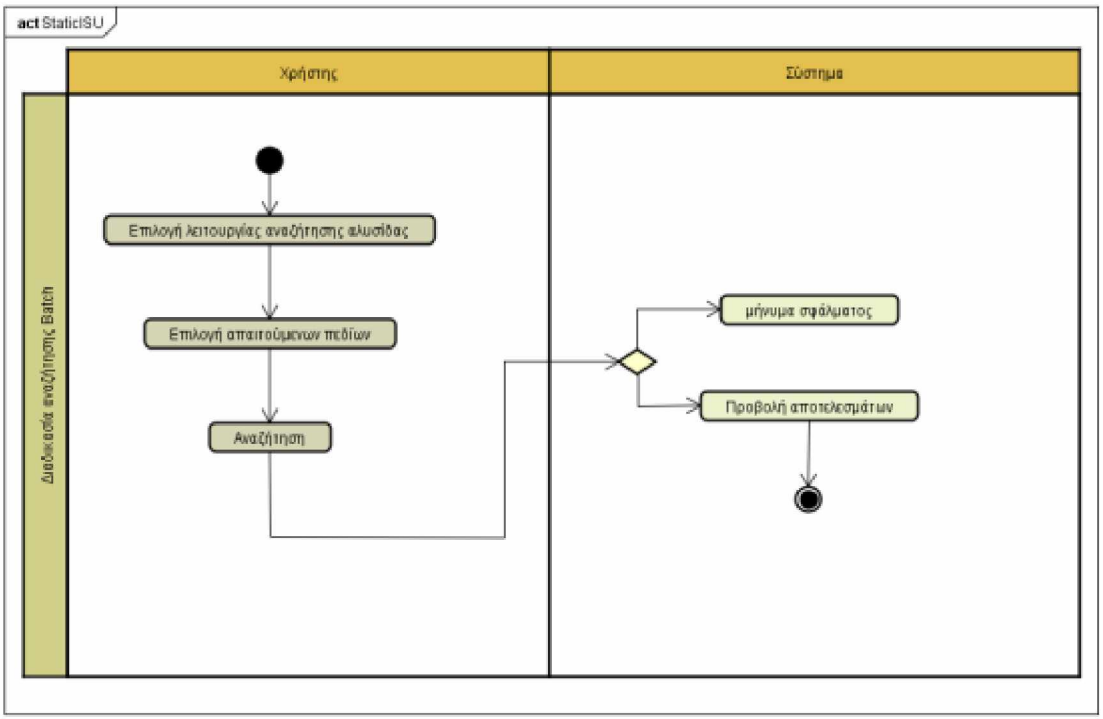
Για την καλύτερη περιγραφή των περιπτώσεων χρήσης και της ροής εργασιών του συστήματος χρησιμοποιούνται διαγράμματα δραστηριοτήτων (Activity Diagrams). Ένα διάγραμμα δραστηριότητας μοντελοποιεί τη ροή της εργασίας, αναπαριστώντας τις διάφορες καταστάσεις εκτέλεσης ενός υπολογισμού (Bohm & Jacopini). Ο ρόλος των διαγραμμάτων δραστηριοτήτων είναι η γραφική αποτύπωση αυτών των Workflows. Προβάλουν γραφικά την ροή εργασίας αποτυπώνοντας τη δυναμική συμπεριφορά μιας διαδικασίας. Η δρομολόγηση γίνεται σύμφωνα με κάποια κριτήρια απόφασης που λαμβάνονται σε κόμβους της ροής, προκειμένου να γίνει εύκολα κατανοητή η πρόοδος μιας διαδικασίας. Παρακάτω παρουσιάζονται σε διαγράμματα δραστηριοτήτων οι κυριότερες διαδικασίες του συστήματος.



Εικόνα 5: Διαδικασία αποθήκευσης Batch/Λειτουργίας

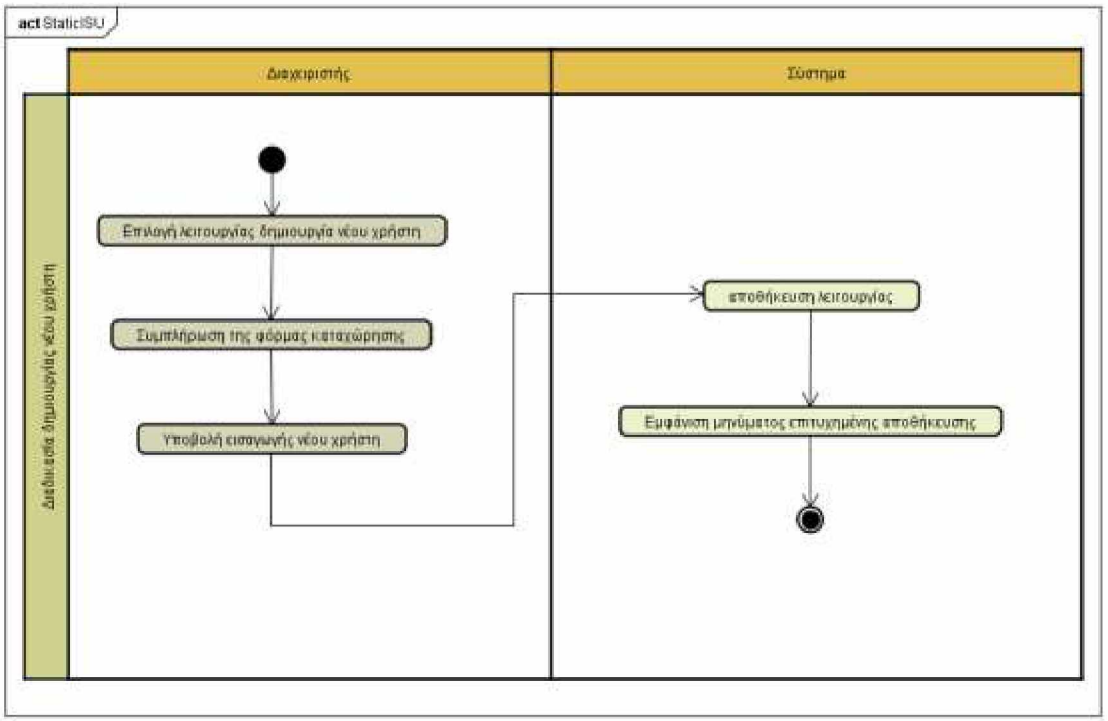
Στο παραπάνω σχήμα αναπαριστάται η διαδικασία υποβολής μίας αλυσίδας στο σύστημα. Το διάγραμμα δραστηριότητας, βοηθάει να προσδιοριστούν οι λειτουργίες του

συστήματος και η ακολουθία τους κατά την διαδικασία υποβολής. Στη διαδικασία υποβολής, πριν την τελική αποθήκευση, θα εμφανίζεται ενημερωτικό μήνυμα με τα εισαχθέντα στοιχεία της φόρμας. Ο χρήστης κάνει επαλήθευση των στοιχείων και μετέπειτα επιλέγει *ακύρωση*, αν κάποιο στοιχείο είναι λάθος ή *επιβεβαίωση* για την οριστική καταχώρηση των δεδομένων στην βάση. Ύστερα, εμφανίζεται μήνυμα πως η αποθήκευση ήταν επιτυχής.



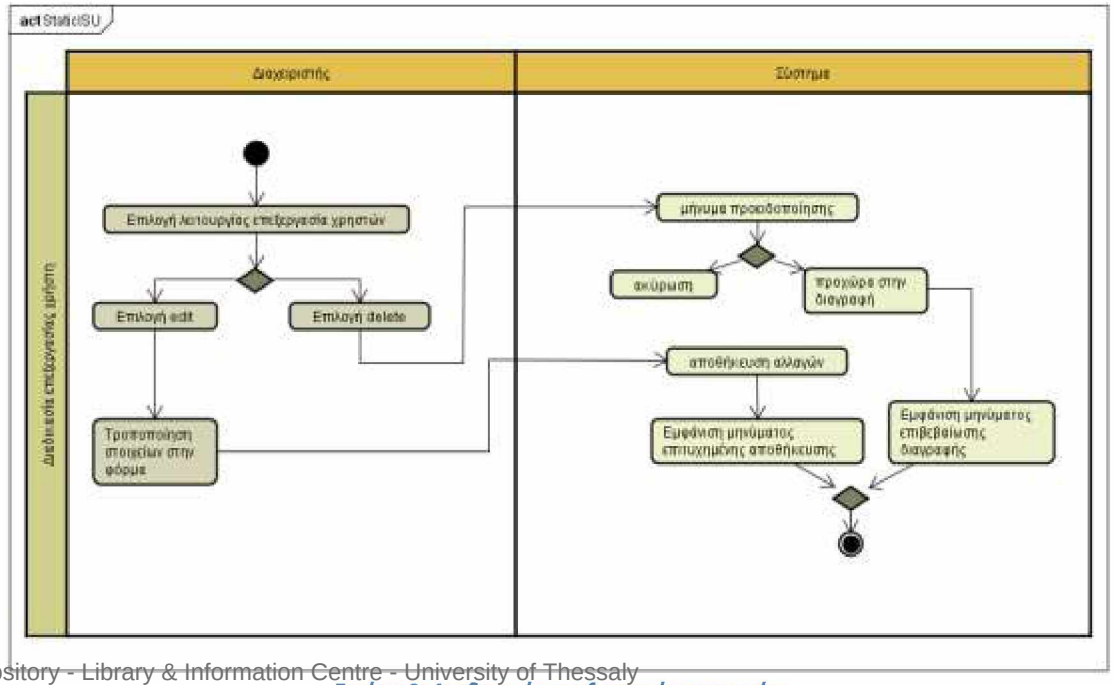
Εικόνα 6: Διαδικασία αναζήτησης λειτουργίας

Στην Εικόνα 6 φαίνεται το διάγραμμα δραστηριότητας για την αναζήτηση μιας αλυσίδας από τον χρήστη. Παρατηρείτε πως ο χρήστης κάνει επιλογή των κριτηρίων σχετικά με την αλυσίδα ή τις αλυσίδες που θέλει να βρει και το σύστημα του επιστρέφει τα αποτελέσματα της αναζήτησης ή σφάλμα δηλαδή μηδέν αποτελέσματα. Συγκεκριμένα γίνεται έλεγχος των δεδομένων στη βάση δεδομένων και επιστροφή αυτών.



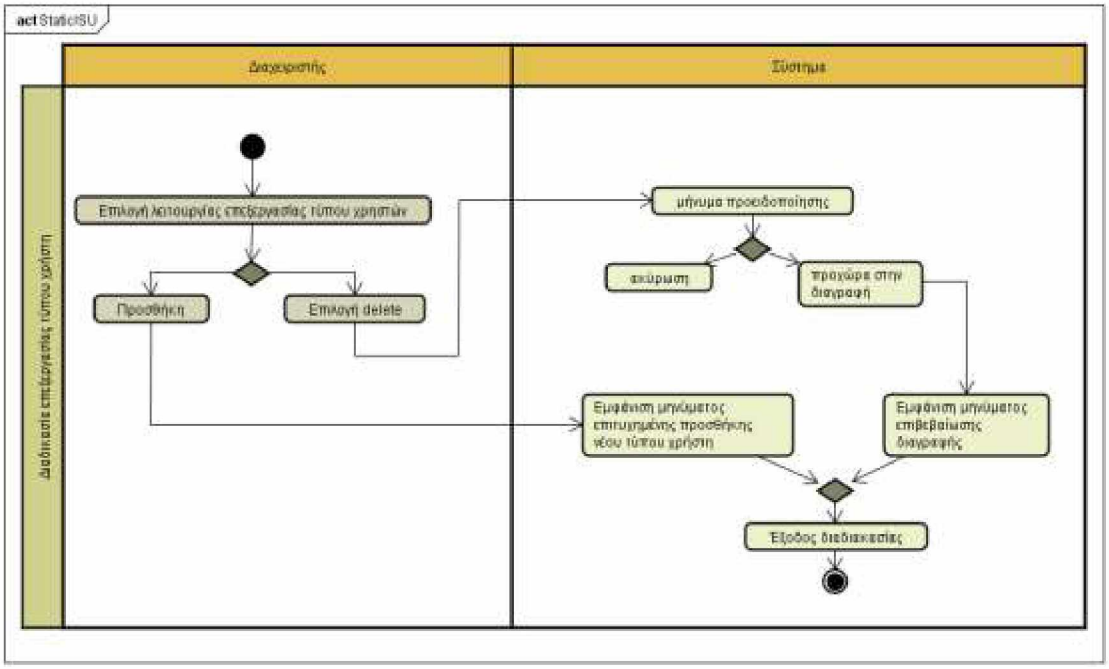
Εικόνα 7: Διαδικασία εισαγωγή νέου χρήστη

Η παραπάνω διαδικασία αναφέρεται στην δημιουργία ενός καινούριου υπάλληλου στο σύστημα. Για την εισαγωγή του στο σύστημα, απατούνται κάποιες βασικές πληροφορίες όπως όνομα, επίθετο, τηλέφωνο, email, usernamekαι password. Το βασικότερο όμως όλων είναι ο τύπος χρήστη που θα του δοθεί. Ανάλογα λοιπόν με τα καθήκοντά του και την αρμοδιότητα του ως προς την εφαρμογή θα μπορεί να είναι απλός χρήστης ή διαχειριστής του συστήματος. Δεν υπάρχει περιορισμός φυσικών προσώπων και για τις δύο κατηγορίες χρηστών.



Εικόνα 8: Διαδικασία επεξεργασίας χρηστών

Στο διάγραμμα δραστηριότητας της Εικόνας 8 αναπαριστάται η ακολουθία λειτουργιών της διαδικασίας επεξεργασίας χρήστη από τον διαχειριστή. Ο διαχειριστής έχει το δικαίωμα να βλέπει όλους τους εγγεγραμμένους χρήστες της εφαρμογής καθώς και το δικαίωμα να μπορεί να επεξεργάζεται βασικές πληροφορίες για αυτούς. Είναι αυτός που μπορεί να τους αλλάξει τα δικαιώματα χρήσης ή ακόμα και να τους αφαιρέσει την πρόσβαση στο σύστημα.



Εικόνα 9: Διαδικασία επεξεργασίας τύπου χρήστη

Στο διάγραμμα αυτό περιγράφεται η διαδικασία εισαγωγής νέων τύπων χρηστών καθώς και αφαίρεσή τους από το σύστημα. Ο διαχειριστή έχει το δικαίωμα να προσθέτει και να αφαιρεί σε αυτή την οθόνη τύπους χρηστών. Πριν από κάθε λειτουργία, εμφανίζεται ένα μήνυμα προειδοποίησης, ένα alert, για να μην πραγματοποιηθεί άθελα του διαχειριστή κάποια απρόσμενη ενέργεια. Εδώ είναι σημαντικό να σημειωθεί πως ο τύπος χρήστη «admin» δεν μπορεί να διαγραφτεί από την βάση για λόγους ασφάλειας.

Χρησιμοποιήθηκαν τα διαγράμματα δραστηριότητας στις περιπτώσεις χρήσης ώστε να αναπαρασταθεί η εσωτερική ολοκλήρωση των παραγόμενων δράσεων δηλαδή, η διαδικαστική ροής ελέγχου. Γίνεται καλύτερη η κατανόηση του συστήματος όσο αφορά τις εσωτερικές λειτουργίες.

5.9 Μοντέλο αλληλεπίδρασης με τον χρήστη

Η αλληλεπίδραση του συστήματος με το χρήση είναι μια από τις σημαντικότερες παραμέτρους που αφορά τη λειτουργικότητα του συστήματος. Σύμφωνα με τον Hassenzahl (2003), κάθε προϊόν έχει ορισμένα χαρακτηριστικά γνωρίσματα (περιεχόμενο, λειτουργία, διαδραστικό ύφος) που επιλέγονται και συνδυάζονται από τον σχεδιαστή προκειμένου να μεταβιβάσουν ένα συγκεκριμένο χαρακτήρα των προϊόντων (Monö, 1997 όπως αναφέρεται στον Hassenzahl, 2003). Ο σκοπός του χαρακτήρα είναι να μειώσει τη γνωστική πολυπλοκότητα και να δημιουργήσει στο χρήστη συγκεκριμένες στρατηγικές σχετικά με τον χειρισμό του συστήματος.

Σύμφωνα με το μοντέλο αλληλεπίδρασης η ανάπτυξη της εφαρμογής βοηθάει από νωρίς να γίνει πιο κατανοητή η λειτουργία του συστήματος. Ακόμα, με αυτό το μοντέλο, τα σημεία που χρειάζεται να δοθεί περισσότερη βάση, οι τεχνικές δυσκολίες που θα προκύψουν και γενικά η πορεία της σχεδίασης αναπτύσσονται και βελτιώνονται από την αρχή της σχεδίασης. Το αποτέλεσμα είναι να υπάρχουν λιγότερα σφάλματα στον σχεδιασμό και βελτιστοποίηση του συστήματος με τις εκάστοτε συνθήκες. Γι' αυτό είναι απαραίτητο ένα αρχικό μοντέλο αλληλεπίδρασης του χρήστη με το σύστημα.

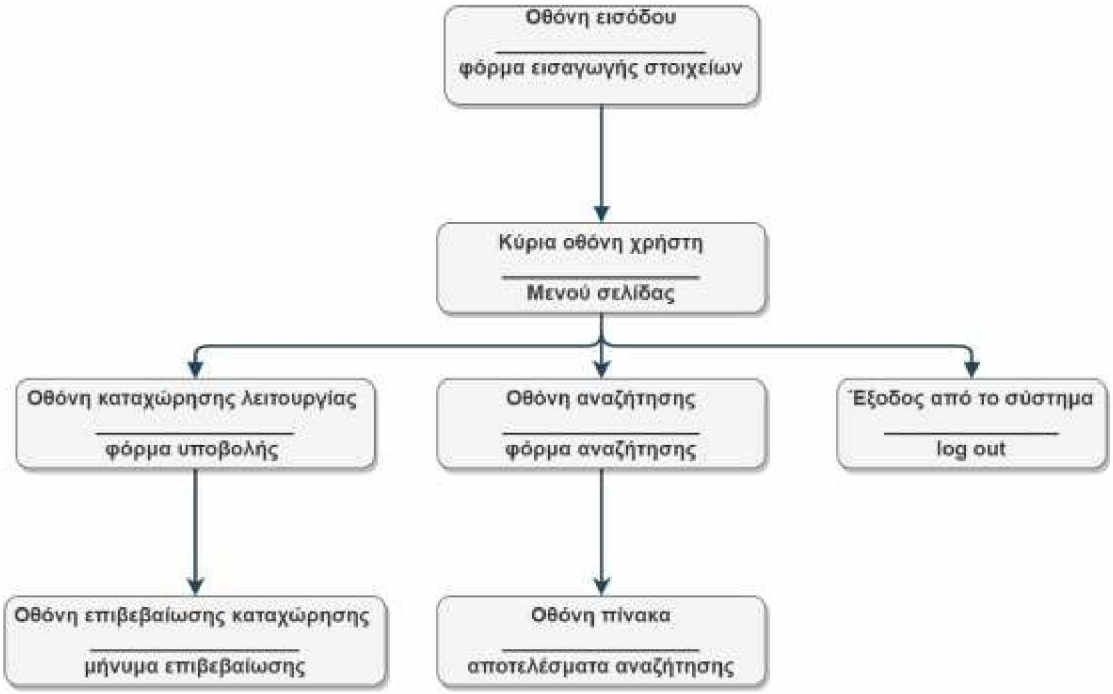
5.9.1 Διαγράμματα ροής συστήματος

Τα Διαγράμματα Ροής Δεδομένων ή Συστήματος βοηθούν στη κατανόηση της λογικής του συστήματος, μέσω της γραφικής απεικόνισης των διαδικασιών και της ροής των πληροφοριών σε ένα πληροφοριακό σύστημα. Στην παρούσα φάση του σχεδιασμού απεικονίζεται μία πιο ολοκληρωμένη εικόνα για το σύστημα. Το επόμενο βήμα είναι να σχεδιαστεί το τελικό μοντέλο αλληλεπίδρασης του χρήστη με το σύστημα.

Στα διαγράμματα που ακολουθούν το βασικό δομικό στοιχείο είναι η οθόνη (ή σελίδα) του συστήματος και συμβολίζεται με ένα ορθογώνιο χωρισμένο σε δύο τμήματα. Στο επάνω τμήμα αναγράφεται ο τίτλος της οθόνης ενώ στο κάτω τα βασικά συστατικά της οθόνης. Τα βέλη υποδεικνύουν την αναμενόμενη μετάβαση από την τρέχουσα-στην επόμενη σελίδα, κατά την χρήση του συστήματος. Δεν έχουν δεσμευτικό χαρακτήρα καθώς η πλοήγηση σε μία διαδικτυακή εφαρμογή πρέπει να θέτει όσο το δυνατόν λιγότερους περιορισμούς στη συμπεριφορά του χρήστη.

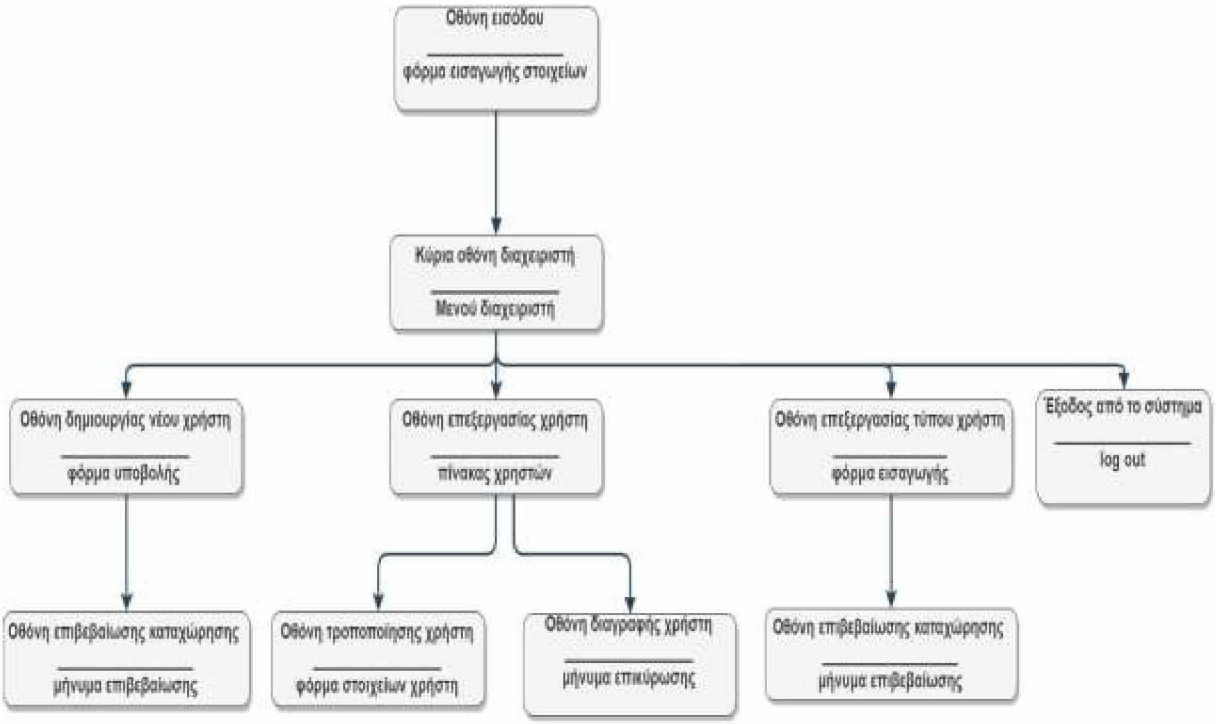
Με τα διαγράμματα ροής συστήματος που παρουσιάζονται εδώ, καταγράφονται οι περιπτώσεις χρήσης του συστήματος σε συνάρτηση με το επίπεδο παρουσίασης. Πρέπει να σημειωθεί ότι αυτός ο τύπος διαγράμματος είναι μία προσωπική προσαρμογή του διαγράμματος χάρτη πλοήγησης που χρησιμοποιείται κατά τον σχεδιασμό συστημάτων με χρήση draw, εφαρμογή σχεδίασης διαγραμμάτων του google drive. Τα διαγράμματα ροής συστήματος όπως παρουσιάζονται εδώ αποτελούν το ενδιάμεσο βήμα μεταξύ της ανάλυσης των περιπτώσεων χρήσης και του τελικού σχεδιασμού της διασύνδεσης χρήστη.

Παρακάτω απεικονίζονται τα διάγραμμα ροής του συστήματος όπως αυτά αλληλεπιδρούν με τον κάθε ρόλο ξεχωριστά. Αναλυτικά περιγράφεται πως κινείται ο συγκεκριμένος ρόλος στο σύστημα και το τι βλέπει κάθε φορά στην οθόνη του σύμφωνα με τη λειτουργία που θα επιτελέσει.



Εικόνα 10: Διάγραμμα ροής συστήματος από πλευρά χρήστη

Στην Εικόνα 10 παρουσιάζεται ένα διάγραμμα ροής το οποίο περιγράφει την εικόνα που έχει ο χρήστης όταν εισέρχεται στο σύστημα. Σε πρώτη φάση βλέπει το μενού το οποίο ορίζεται από καρτέλες που αποτελούν και τις βασικές λειτουργίες του χρήστη.



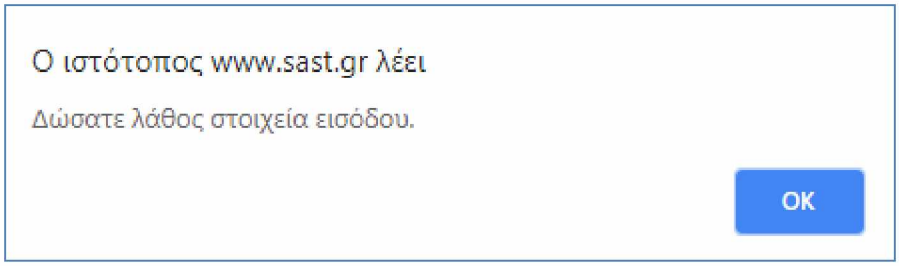
Εικόνα 11: Διάγραμμα ροής συστήματος από την πλευρά του διαχειριστή

Το σχήμα στην Εικόνα 11 αναφέρεται στον Γενικό Διαχειριστή και την αλληλεπίδραση που έχει με το σύστημα.

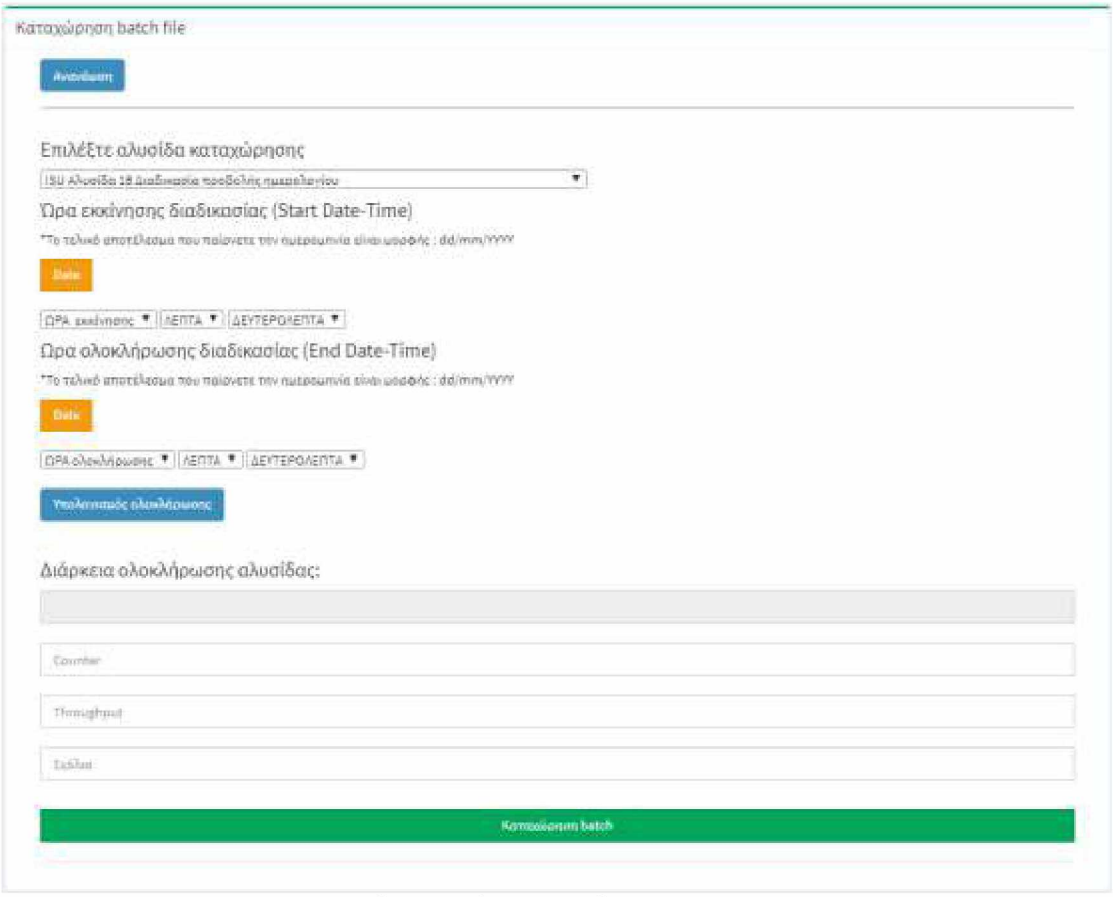
5.9.2 Σχεδίαση διεπαφής χρήστη

Σε αυτό το σημείο του σχεδιασμού θα κατασκευαστεί το τελικό μοντέλο διεπαφής με το χρήστη. Μέχρι τώρα έχει καταγραφεί ο τρόπος που θα αλληλεπιδρά το σύστημα με το χρήστη το οποίο έχει αποτυπωθεί μέσα από διαγράμματα. Για την λεπτομερέστερη και κατανοητότερη συμπεριφορά του μοντέλου, στα επόμενα σχήματα θα παρουσιαστούν επακριβώς πως φαίνεται ο σχεδιασμός στην οθόνη του υπολογιστή.

Εικόνα 12: Οθόνη εισόδου στο StaticISU

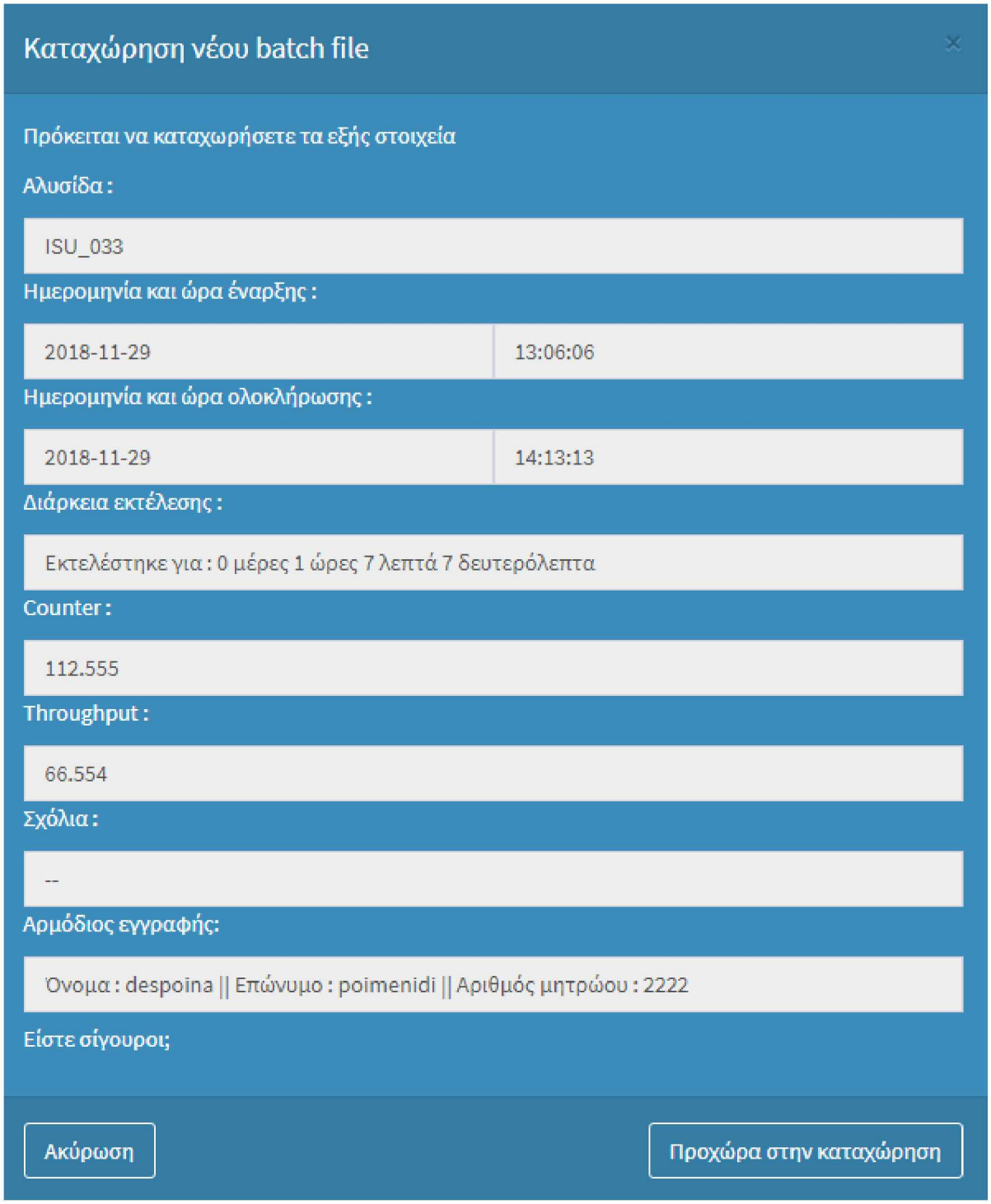


Εικόνα 13: Μήνυμα λάθους στοιχείων εισόδου



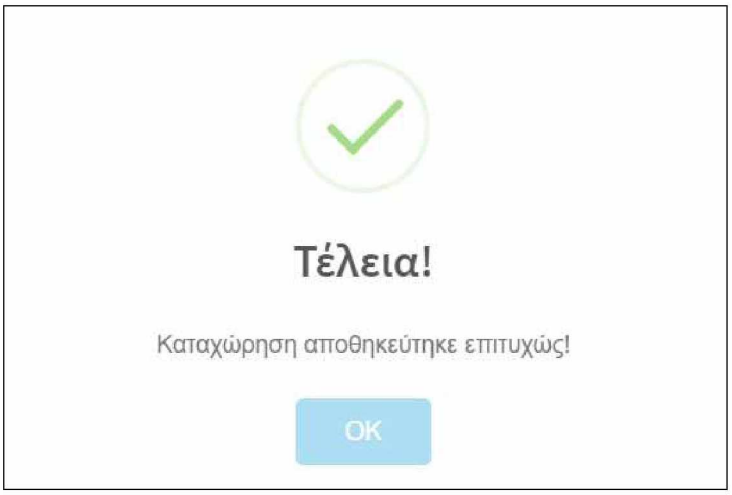
Εικόνα 14: Φόρμα καταχώρησης batch file

Αφού ο χρήστης έχει μπει στην εφαρμογή StaticISU επιλέγει την καρτέλα «Καταχώρηση». Εκεί του εμφανίζεται μια φόρμα εισαγωγής με διάφορα στοιχεία που πρέπει να εισάγει για να πραγματοποιήσει την υποβολή. Ένα τέτοιο παράδειγμα παρουσιάζεται στην Εικόνα 14. Όταν συμπληρώσει τα στοιχεία της φόρμας επιλέγει «Καταχώρηση batch» ώστε να ολοκληρώσει την υποβολή του. Μετά από αυτή την ενέργεια, εμφανίζεται ένα ειδοποιητικό μήνυμα στην οθόνη με όλες τις πληροφορίες που εισήγαγε ο χρήστης.



Εικόνα 15: Μήνυμα επιβεβαίωσης καταχώρησης λειτουργίας

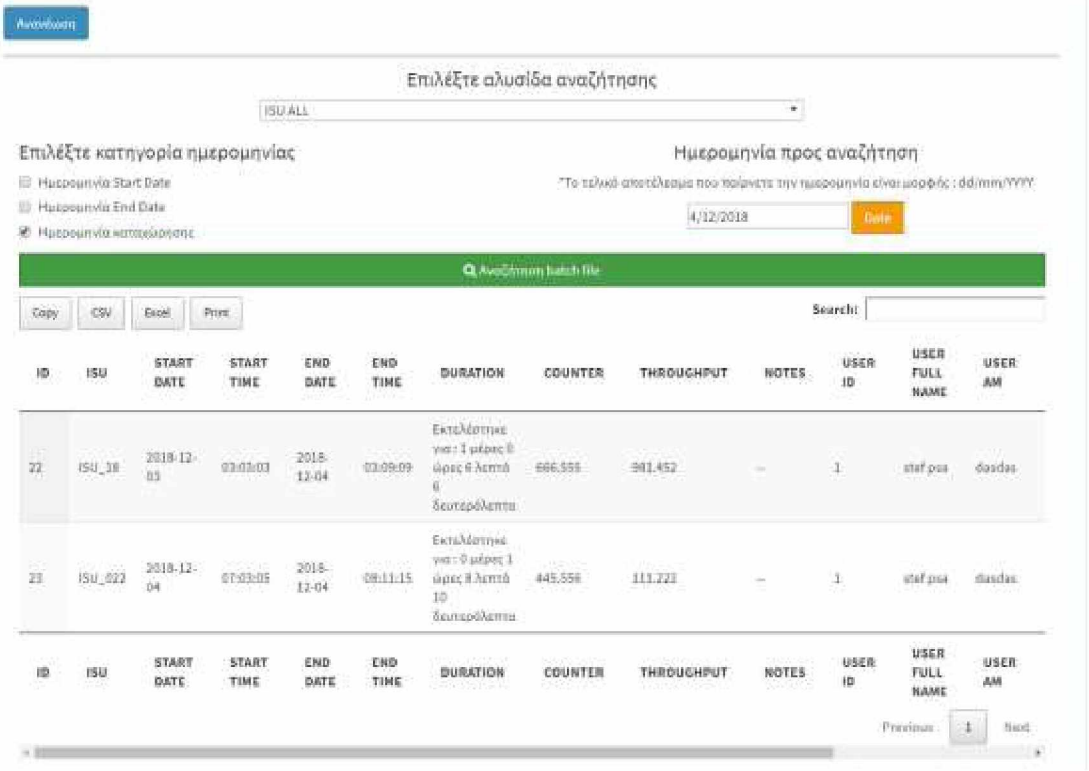
Αφού, επιβεβαιώσει τα στοιχεία αυτά επιλέγει το κουμπί «Προχώρα στην καταχώρηση». Αυτόματα του εμφανίζεται το παρακάτω μήνυμα στην οθόνη.



Εικόνα 16: Ανταπόκριση συστήματος σε υποβολή καταχώρησης

Η καταχώρηση πραγματοποιήθηκε με επιτυχία.

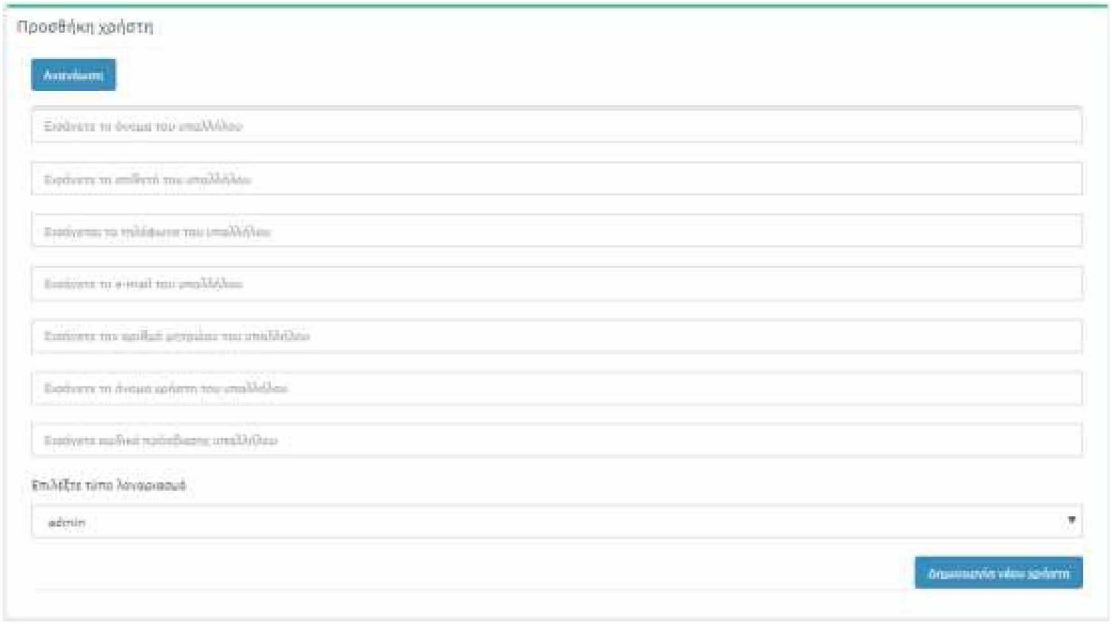
Τώρα ο χρήστης μπορεί να ψάξει την καταχώρηση αυτή στην καρτέλα «Αναζήτηση» από το μενού. Συμπληρώνοντας τα πεδία και πατώντας το κουμπί «Αναζήτηση Batch file» θα εμφανιστεί ένας datatable με όλες τις καταχωρήσεις που υπάρχουν με αυτά τα στοιχεία.



Εικόνα 17: Εμφάνιση αποτελεσμάτων αναζήτησης

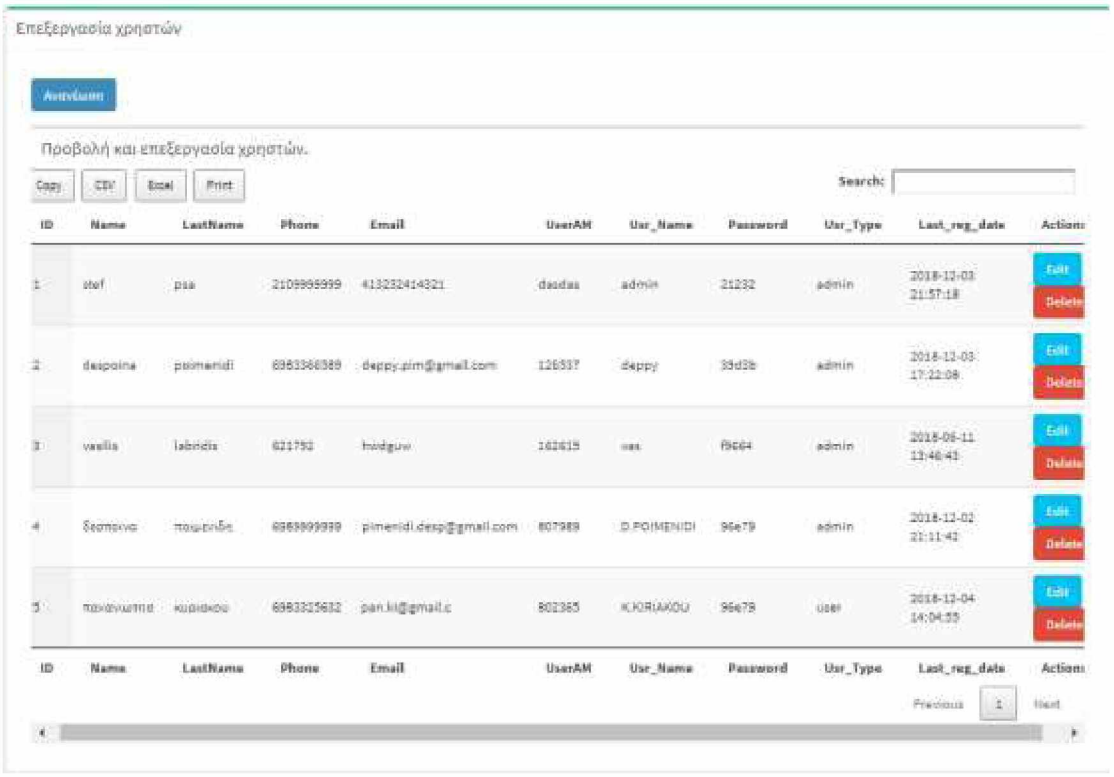
5.9.3 Σχεδίαση διεπαφής διαχειριστή

Όταν ο Διαχειριστής κάνει είσοδο στο σύστημα θα του εμφανιστούν επιπλέον λειτουργίες στο μενού, που μόνο αυτός έχει πρόσβαση. Αρχικά, μπορεί να εισάγει νέο χρήστη όπως φαίνεται και στην Εικόνα 18.



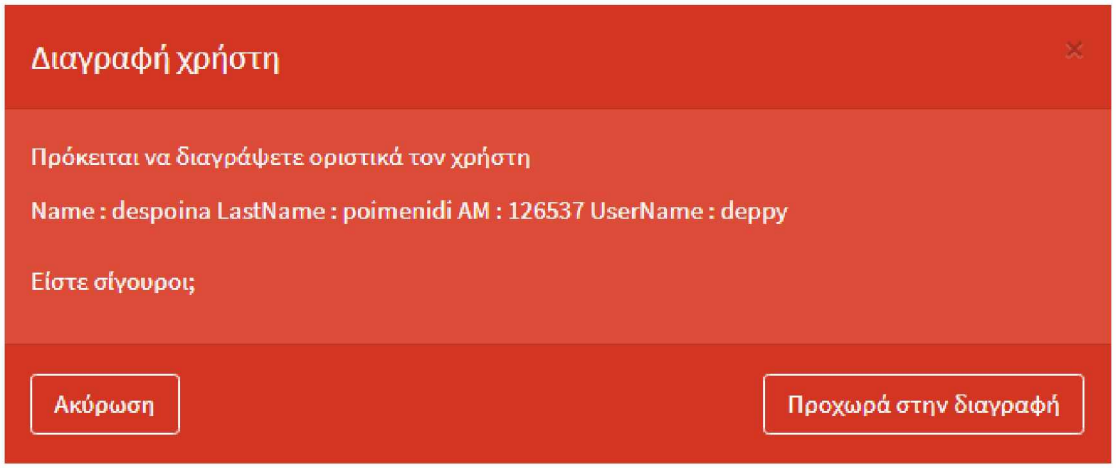
Εικόνα 18: Φόρμα προσθήκης νέου χρήστη

Η δεύτερη καρτέλα που μπορεί να δράξει είναι η «επεξεργασία χρηστών». Όπως έχει αναφερθεί, εδώ ο διαχειριστής μπορεί να είτε να επεξεργαστεί τα στοιχεία των υπαλλήλων είτε να τους διαγράψει από την βάση.



Εικόνα 19: Πίνακας επεξεργασίας χρηστών συστήματος

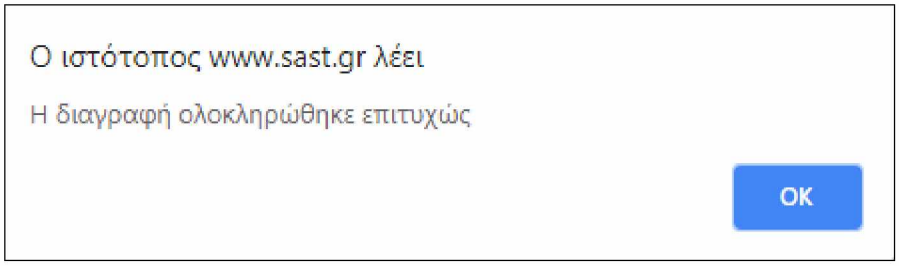
Στην περίπτωση της διαγραφής, θα εμφανιστεί προειδοποιητικό μήνυμα στην βασική οθόνη που θα ρωτάει τον διαχειριστεί αν είναι σίγουρος για αυτή την ενέργεια.



Εικόνα 20: Προειδοποιητικό μήνυμα διαγραφής χρήστη

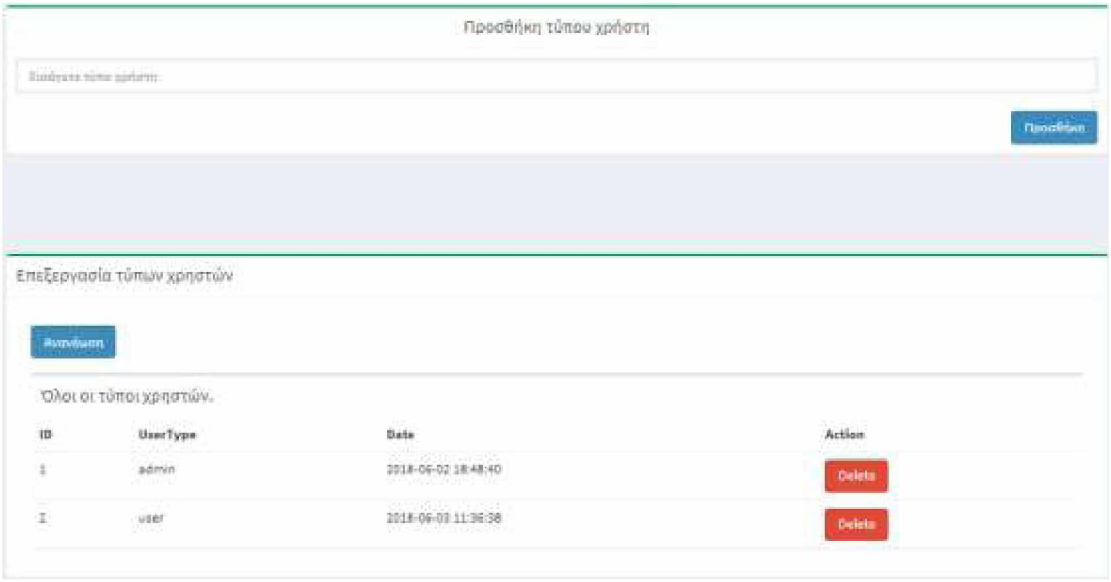
Εάν ο διαχειριστής επιλέξει ακύρωση, επιστρέφει στην οθόνη «επεξεργασίας χρηστών» χωρίς να έχει εκτελεστεί κάποια ενέργεια. Στην περίπτωση που επιλέξει «Προχώρα

στην διαγραφή», εμφανίζεται μήνυμα επιβεβαίωσης διαγραφής και πλέον ο χρήστης δεν υπάρχει πλέον στην βάση του συστήματος.



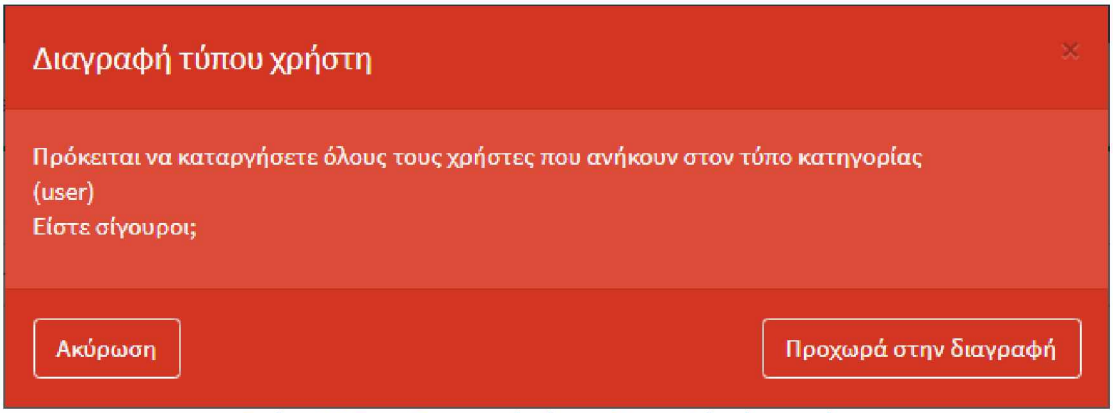
Εικόνα 21: Μήνυμα ανταπόκρισης συστήματος στην διαγραφή χρήστη

Στην τρίτη οθόνη ο διαχειριστής μπορεί να προσθέσει έναν νέο τύπο χρήστη ή να διαγράψει κάποιον τύπο. Επιπλέον, δεν επιτρέπει το σύστημα να δημιουργηθεί τύπος χρήστη με το ίδιο όνομα δύο φορές.



Εικόνα 22: Εισαγωγή τύπου χρήστη και επεξεργασία τύπων χρηστών

Στην περίπτωση διαγραφής, εμφανίζεται προειδοποιητικό μήνυμα όπως προηγουμένως. Αν ο διαχειριστής προχωρήσει στην διαγραφή τύπου χρήστη θα διαγραφούν από την βάση και όλοι οι υπάλληλοι που ανήκουν σε αυτόν τον τύπο.



Εικόνα 23: Προειδοποιητικό μήνυμα διαγραφής τύπου χρήστη

5.10 Προγράμματα που χρησιμοποιήθηκαν στο Σχεδιασμό

Για τον δημιουργία της σελίδας χρησιμοποιήθηκαν κάποια επιπλέον προγράμματα στο στάδιο του σχεδιασμού. Το Astah και το draw.io βοήθησαν σημαντικά για την διαμόρφωση του site.

Το Astah Community. Ένα add-on εργαλείο για τη UML. Η προσέγγιση μοντελοποίησης του Astah ενισχύει και επιταχύνει την UML. Με αυτό το πρόγραμμα μοντελοποίησης σχεδιάστηκαν τα UseCase Diagrams (Διαγράμματα Περιπτώσεων Χρήσης) και Activity Diagrams (Διαγράμματα Δραστηριοτήτων) κατά τη διάρκεια του σχεδιασμού.

Το draw.io είναι ένα πρόγραμμα που βοηθάει στην δημιουργία διαγραμμάτων. Στην παρούσα εργασία χρησιμοποιήθηκε για τη δημιουργία διαγραμμάτων ροής του συστήματος.

ΚΕΦΑΛΑΙΟ 6 – ΥΛΟΠΟΙΗΣΗ

6.1 Εισαγωγή

Σε αυτή τη φάση της υλοποίησης θα αναλυθεί η δημιουργία του συστήματος, σύμφωνα με τις λειτουργικές απαιτήσεις που ορίστηκαν σε προηγούμενο κεφάλαιο. Ειδικότερα, σε αυτό το στάδιο θα αναπτυχθεί ο πηγαίος κώδικας της εφαρμογής. Το κεφάλαιο αυτό αποτελεί έναν καλό οδηγό για όσους θέλουν να μάθουν σε βάθος τη λειτουργία του συστήματος, αλλά και ένα καλό σημείο αναφοράς για όσους ενδιαφέρονται να αναπτύξουν κώδικα για παρόμοιες εφαρμογές.

Όπως έχει αναφερθεί σε προηγούμενα κεφάλαια για την υλοποίηση της εφαρμογής χρησιμοποιήθηκαν διάφορες τεχνολογίες λογισμικού. Αρχικά, χρησιμοποιήθηκε ένα theme που βοήθησε στην δομή και την εμφάνιση της εφαρμογής. Συγκεκριμένα, χρησιμοποιήθηκε το πρότυπο διαχείρισης του Bootstrap το οποίο είναι δωρεάν προς όλους. Αυτό το πρότυπο χρησιμοποιεί τα προεπιλεγμένα στυλ Bootstrap 3 μαζί με μια ποικιλία ισχυρών προσθηκών για τη δημιουργία ενός ισχυρού πλαισίου για τη δημιουργία πινάκων διαχειριστή, εφαρμογών ιστού ή πινάκων ελέγχου από πίσω. Η AdminLTE[5] παρέχει μια σειρά προσαρμοστικών, επαναχρησιμοποιήσιμων και κοινώς χρησιμοποιούμενων στοιχείων για την καλή διαχείριση του κώδικα.

Τα βασικά αρχεία της εφαρμογής είναι πέντε:

- Το index.php που περιέχει τον κώδικα για το login στην εφαρμογή.
- Το dashboard.php που περιέχει τον html κώδικα που είναι ο υπεύθυνος για την εμφάνιση του περιβάλλοντος (Theme) προς τους χειριστές.
- Το tlbx.php που περιλαμβάνει όλο τον κώδικα php της σελίδας. Εδώ δηλώνεται όλη η συμπεριφορά του συστήματος και η συνεργασία του με την database.
- Το dash_actions.php που είναι το αρχείο που συμπεριλαμβάνει την συνδεσμολογία με την βάση.
- Και το lout.php που περιέχει την αποσύνδεση με το σύστημα.

[5] https://adminlte.io/

Εν συνεχεία θα αναφερθούν κάποια παραδείγματα μέσα από τον κώδικα για να κατανοηθεί ο τρόπος λειτουργίας της εφαρμογής. Θα δοθεί έμφαση σε παραδείγματα που εμπεριέχουν κώδικα για την προστασία του συστήματος με κάποιο είδος κρυπτογράφησης.

6.2 Δημιουργία βάσης δεδομένων

Πρωτίστως, πρέπει να σημειωθεί πως για την υλοποίηση της εφαρμογής δημιουργήθηκαν οι πίνακες με όλα τα πεδία που θα χρειαστούν για να λειτουργήσει αρμονικά η εφαρμογή. Τα πεδία που θα εμπεριέχουν οι πίνακες καθορίζονται από την ανάλυση απαιτήσεων, σύμφωνα με τις ανάγκες της επιχείρησης. Το πρόγραμμα που χειρίζεται τα δεδομένα είναι το HeidiSQL, το οποίο είναι ελεύθερο λογισμικό.

Υπάρχουν τρεις πίνακες οι οποίοι επικοινωνούν με το StaticISU:

- Users
- User_insert
- User_types

6.2.1 Δημιουργία πίνακα users

```
CREATETABLE`users`(
`id`INT(11)NOTNULLAUTO_INCREMENT,
`Name`TEXTNOTNULLDEFAULT'0',
`LastName`TEXTNOTNULLDEFAULT'0',
`Phone`TEXTNOTNULLDEFAULT'0',
`Email`TEXTNOTNULLDEFAULT'0',
`UserAM`TEXTNOTNULLDEFAULT'0',
`Usr_Name`TEXTNULLDEFAULTNULL,
`Password`TEXTNOTNULLDEFAULT'0',
`Usr_Type`TEXTNULLDEFAULTNULL,
`ftc`INT(1)NULLDEFAULTNULL,
`Last_reg_date`TIMESTAMPNULLDEFAULTCURRENT_TIMESTAMP,
PRIMARYKEY(`id`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=4
;
```

Στον πίνακα users αποθηκεύονται όλοι οι χρήστες που μπορούν να έχουν πρόσβαση στην εφαρμογή, ανεξάρτητα από τον τύπο χρήστη. Είναι σημαντικό να τονιστεί πως και στη βάση όλα τα πεδία έρχονται κρυπτογραφημένα. Παρακάτω εμφανίζεται ο πίνακας Users με τρεις καταχωρήσεις χρηστών. Ο τρόπος κρυπτογράφησης να αναλυθεί στη συνέχεια.

id	Name	LastName	Phone	Email	UserAM	Usr_Name	Password	Usr_Type
1	dNTmCxTQw+4ENp2DI3...	lQqxsuIzWh/BKqINd/BJww...	hOINgZIrSNI7M/O4...	Kgco4EIhqJihSc+L/z...	DzaoC3k9p69eSFg/qFiO...	bhp0QadMrSyfTftW...	21232f297a57a5a743...	bhp0QadMrSyfTftW
2	Xjwqev1xnGUSHNRbqH...	Xjwqev1xnGUSHNRbqHhD...	Xjwqev1xnGUSHNR...	Xjwqev1xnGUSHNR...	iJY/ycomNB07j/gg6pDtv...	Xjwqev1xnGUSHNR...	e00cf25ad42683b3df...	aS7TxSzCaGg+y5+
3	AxJUpJPcLyHwVxDosrX...	8ScYqhHjHpnD053KNFDh+...	aWNYHMfPC4zZLGk3...	P/OGf0Yxq2Srkm9d...	jMowYX0UgDpP0ZL93h6...	RRHx7k6/BKq6J7m...	ca6d8e04d27bd46c58...	aS7TxSzCaGg+y5+

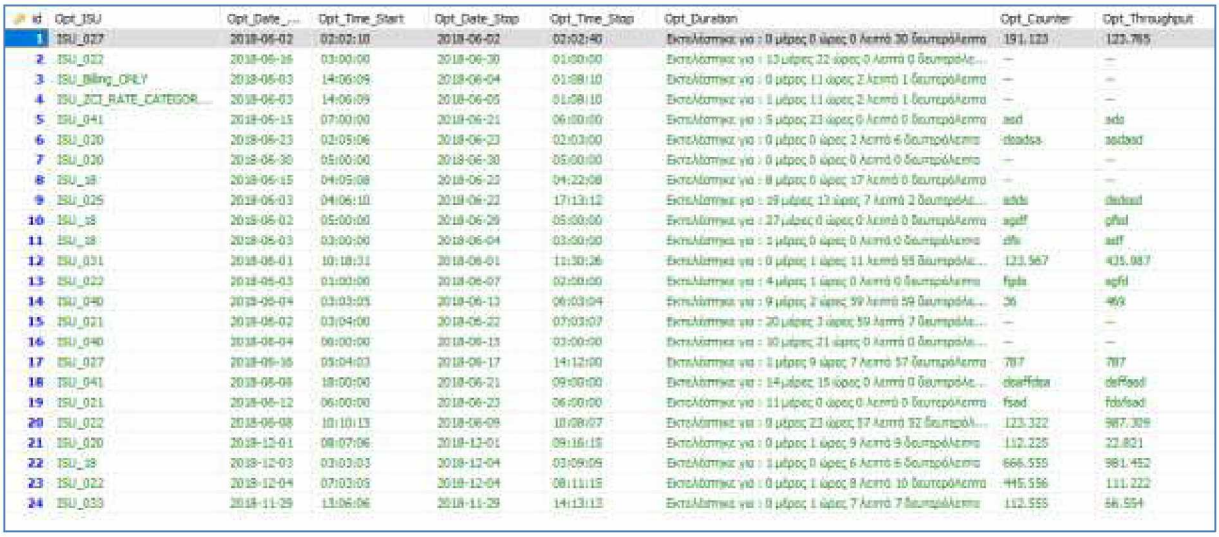
UserAM	Usr_Name	Password	Usr_Type	ftc	Last_reg_date
DzaoC3k9p69eSFg/qFiO0w==	bhp0QadMrSyfTftWMFJ4IA==	21232f297a57a5a743894a0e4a801fc3	bhp0QadMrSyfTftWMFJ4IA==	0	2018-12-08 21:15:32
iJY/ycomNB07j/gg6pDtvg==	Xjwqev1xnGUSHNRbqHhD7Q==	e00cf25ad42683b3df678c61f42c6bda	aS7TxSzCaGg+y5+rws9sJw==	0	2018-12-08 21:24:23
jMowYX0UgDpP0ZL93h6LdQ==	RRHx7k6/BKq6J7mBoyVirw==	ca6d8e04d27bd46c5885602f1b0430f5	aS7TxSzCaGg+y5+rws9sJw==	0	2018-12-10 00:37:57

Εικόνα 24: Πίνακας χρηστών στη βάση δεδομένων με κρυπτογραφημένα πεδία

6.2.2 Δημιουργία πίνακα user_insert

```
CREATETABLE`user_insert`(
`id`INT(11)NOTNULLAUTO_INCREMENT,
`Opt_ISU`TEXTNOTNULL,
`Opt_Date_Start`TEXTNOTNULL,
`Opt_Time_Start`TEXTNOTNULL,
`Opt_Date_Stop`TEXTNOTNULL,
`Opt_Time_Stop`TEXTNOTNULL,
`Opt_Duration`TEXTNOTNULL,
`Opt_Counter`TEXTNOTNULL,
`Opt_Throughput`TEXTNOTNULL,
`Opt_Note`TEXTNOTNULL,
`Usr_DB_Id`TEXTNOTNULL,
`Usr_N_LN`TEXTNOTNULL,
`Usr_AM`TEXTNOTNULL,
`Usr_Email`TEXTNOTNULL,
`Usr_Phone`TEXTNOTNULL,
`reg_date`TIMESTAMPNOTNULLDEFAULTCURRENT_TIMESTAMP,
PRIMARYKEY(`id`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=25
;
```

Στον πίνακα user_insert αποθηκεύονται όλες οι καταχωρήσεις των χρηστών από την οθόνη «καταχώρηση». Επομένως, κάθε φορά που ένας χρήστης κάνει αναζήτηση με βάση την ημερομηνία για να του εμφανιστούν οι αλυσίδες που καταχωρήθηκαν, η εφαρμογή αντλεί τις πληροφορίες από τον πίνακα user_insert.



id	Opt_ISU	Opt_Date_...	Opt_Time_Start	Opt_Date_Stop	Opt_Time_Stop	Opt_Duration	Opt_Counter	Opt_Throughput
1	ISU_027	2018-06-02	02:02:10	2018-06-02	02:02:40	Εκτελέστηκε για : 0 μέρες 0 ώρες 0 λεπτά 30 δευτερόλεπτα	191.123	123.765
2	ISU_022	2018-06-26	03:00:00	2018-06-30	01:00:00	Εκτελέστηκε για : 10 μέρες 22 ώρες 0 λεπτά 0 δευτερόλε...	--	--
3	ISU_Billing_OPEY	2018-06-03	14:06:09	2018-06-04	01:08:10	Εκτελέστηκε για : 0 μέρες 11 ώρες 2 λεπτά 1 δευτερόλεπτα	--	--
4	ISU_ZCT_RATE_CATEGOR...	2018-06-03	14:06:09	2018-06-05	01:08:10	Εκτελέστηκε για : 1 μέρες 11 ώρες 2 λεπτά 1 δευτερόλεπτα	--	--
5	ISU_041	2018-06-15	07:00:00	2018-06-21	06:00:00	Εκτελέστηκε για : 5 μέρες 23 ώρες 0 λεπτά 0 δευτερόλεπτα	asd	sda
6	ISU_020	2018-06-23	02:05:06	2018-06-23	02:03:00	Εκτελέστηκε για : 0 μέρες 0 ώρες 2 λεπτά 6 δευτερόλεπτα	dsadsa	asdasd
7	ISU_020	2018-06-30	05:00:00	2018-06-30	05:00:00	Εκτελέστηκε για : 0 μέρες 0 ώρες 0 λεπτά 0 δευτερόλεπτα	--	--
8	ISU_18	2018-06-15	04:05:08	2018-06-23	04:22:08	Εκτελέστηκε για : 8 μέρες 0 ώρες 17 λεπτά 0 δευτερόλεπτα	--	--
9	ISU_025	2018-06-03	04:06:10	2018-06-22	17:13:12	Εκτελέστηκε για : 19 μέρες 13 ώρες 7 λεπτά 2 δευτερόλε...	adds	dadasd
10	ISU_18	2018-06-02	05:00:00	2018-06-29	05:00:00	Εκτελέστηκε για : 27 μέρες 0 ώρες 0 λεπτά 0 δευτερόλεπτα	agdf	gfsd
11	ISU_18	2018-06-03	03:00:00	2018-06-04	03:00:00	Εκτελέστηκε για : 1 μέρες 0 ώρες 0 λεπτά 0 δευτερόλεπτα	dfs	asff
12	ISU_031	2018-06-01	10:10:31	2018-06-01	11:30:26	Εκτελέστηκε για : 0 μέρες 1 ώρες 11 λεπτά 55 δευτερόλε...	123.567	435.987
13	ISU_022	2018-06-03	01:00:00	2018-06-07	02:00:00	Εκτελέστηκε για : 4 μέρες 1 ώρες 0 λεπτά 0 δευτερόλεπτα	fgds	agfd
14	ISU_040	2018-06-04	03:03:05	2018-06-13	06:03:04	Εκτελέστηκε για : 9 μέρες 2 ώρες 59 λεπτά 59 δευτερόλε...	36	469
15	ISU_021	2018-06-02	03:04:00	2018-06-22	07:03:07	Εκτελέστηκε για : 20 μέρες 3 ώρες 59 λεπτά 7 δευτερόλε...	--	--
16	ISU_040	2018-06-04	06:00:00	2018-06-13	03:00:00	Εκτελέστηκε για : 10 μέρες 21 ώρες 0 λεπτά 0 δευτερόλε...	--	--
17	ISU_027	2018-06-16	05:04:03	2018-06-17	14:12:00	Εκτελέστηκε για : 1 μέρες 9 ώρες 7 λεπτά 57 δευτερόλεπτα	787	787
18	ISU_041	2018-06-08	18:00:00	2018-06-21	09:00:00	Εκτελέστηκε για : 14 μέρες 15 ώρες 0 λεπτά 0 δευτερόλε...	dasffdas	daffasd
19	ISU_021	2018-06-12	06:00:00	2018-06-23	06:00:00	Εκτελέστηκε για : 11 μέρες 0 ώρες 0 λεπτά 0 δευτερόλεπτα	fsad	fdsfsad
20	ISU_022	2018-06-08	10:10:13	2018-06-09	10:08:07	Εκτελέστηκε για : 0 μέρες 23 ώρες 57 λεπτά 52 δευτερόλ...	123.322	987.309
21	ISU_020	2018-12-01	08:07:06	2018-12-01	09:16:15	Εκτελέστηκε για : 0 μέρες 1 ώρες 9 λεπτά 9 δευτερόλεπτα	112.225	22.821
22	ISU_18	2018-12-03	03:03:03	2018-12-04	03:09:09	Εκτελέστηκε για : 1 μέρες 0 ώρες 6 λεπτά 6 δευτερόλεπτα	666.555	981.452
23	ISU_022	2018-12-04	07:03:05	2018-12-04	08:11:15	Εκτελέστηκε για : 0 μέρες 1 ώρες 8 λεπτά 10 δευτερόλεπτα	445.556	111.222
24	ISU_033	2018-11-29	13:06:06	2018-11-29	14:13:13	Εκτελέστηκε για : 0 μέρες 1 ώρες 7 λεπτά 7 δευτερόλεπτα	112.555	66.554

Εικόνα 25: Πίνακας στη βάση δεδομένων με όλες τις καταχωρήσεις

Τα βασικά πεδία για αυτόν τον πίνακα είναι οι ημερομηνίες καταχώρησης και οι ημερομηνίες που έτρεξε μια αλυσίδα. Επιλέγοντας την αλυσίδα και μία από τις ημερομηνίες ο χρήστης θα έχει τα αποτελέσματα στην οθόνη του, αν φυσικά υπάρχουν στην βάση.

6.2.3 Δημιουργία πίνακα user_types

```
CREATETABLE`user_types`(
`id`INT(11)NOTNULLAUTO_INCREMENT,
`Type_Name`TEXTNULLDEFAULT'0',
`reg_date`TIMESTAMPNULLDEFAULTCURRENT_TIMESTAMP,
PRIMARYKEY(`id`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=6
;
```

Ο πίνακας user_types περιέχει όλους τους τύπους χρηστών που έχουν δικαιώματα στην εφαρμογή. Από την εφαρμογή δίνεται η δυνατότητα στον διαχειριστή να προσθέσει ή και να αφαιρέσει έναν τύπο χρήστη. Τα δικαιώματα θα τα καθορίσει μέσα από τον κώδικα ο προγραμματιστής του συστήματος. Σε συτό το σημείο της ανάπτυξης του συστήματος υπάρχουν δύο τύποι χρηστών. Ο admin και οι users.

id	Type_Name	reg_date
1	admin	2018-06-02 18:48:40
2	user	2018-06-03 11:36:38

Εικόνα 26: Πίνακας στη βάση δεδομένων με τους τύπους χρηστών

6.3 Φιλοσοφία της εφαρμογής

Ο τρόπος λειτουργίας της εφαρμογής θα μπορούσε να διαχωριστεί σε δυο κατηγορίες. Τις ασύγχρονες λειτουργίες και τις σταθερές λειτουργίες.

Ασύγχρονες λειτουργίες

Σε αυτή την κατηγορία ανήκουν όλες οι λειτουργίες που ο χρήστης μπορεί να εκτελεί ασύγχρονα όπως:

- Η Καταχώρηση νέου batch file
- Η δημιουργία νέου χρήστη
- Η δημιουργία νέου τύπου χρήστη
- Η αναζήτηση μια αλυσίδας στη βάση
- Διαγραφή τύπου χρηστών
- Έλεγχος στοιχείων χρήστη κατά την είσοδο

Σταθερές λειτουργίες

Στις σταθερές λειτουργίες εμφανίζονται όλες οι μη δυναμικές λειτουργίες. Είναι αυτές που ο χρήστης δεν θα χρειαστεί να επέμβει με κάποια εντολή για να του εμφανιστούν στην οθόνη. Τέτοιες λειτουργίες είναι:

- Ο πίνακας με τους users. Κάθε φορά που θα κάνει log in ο admin θα βρίσκει τους ίδιους users.
- Όλες οι λειτουργίες που εμφανίζονται κατά την είσοδο στην σελίδα.

Οι ασύγχρονες λειτουργίες περιγράφονται στο αρχείο tlbx.php και οι σταθερές στο dashboard.php. Κάθε φορά που καλείτε μια ασύγχρονη λειτουργία εκτελείτε ένα Ajax Request.

Με την χρήση της τεχνολογίας Ajax όλη η φιλοσοφία της εφαρμογής βασίζεται σε μία σελίδα (single-page application) η οποία θα φορτώνεται μία φορά και θα εκτελούνται όλες οι εργασίες με ασύγχρονη επικοινωνία. Αυτό έχει σαν αποτέλεσμα η εφαρμογή να είναι γρηγορότερη από την δημιουργία πολλών επανασυνδέσεων (reconnection) με την βάση.

Το Ajax request γίνεται από την JavaScript κάθε φορά που κάνει ένα post σε ένα αρχείο php με κάποιες μεταβλητές. Το php αρχείο βάση των μεταβλητών καλεί κάποια λογική και εμφανίζει ένα αποτέλεσμα στο Ajax request που του θέσαμε.

Αρχείο tlbx περιλαμβάνει:

- Ενημέρωση εργαλείου χρήσης
- Στοιχεία σύνδεσης βάσης δεδομένων
- Κλειδιά κρυπτογράφησης
- Φόρτωση κρυπτογράφησης
- Σύνδεση με τη βάση δεδομένων μία φορά
  - Προσθήκη νέου τύπου χρηστών
  - Διαγραφή τύπου χρηστών
  - Προσθήκη νέου χρήστη
  - Προσθήκη νέου χρήστη
  - Αποθήκευση αλλαγών χρήστη
  - Διαγραφή χρήστη
  - Σύνδεση με τη βάση δεδομένων
  - Έλεγχος στοιχείων χρήστη κατά την είσοδο
  - Καταχώρηση νέου batch file
  - Αναζήτηση batch file με ημερομηνία

Αρχείο index.php περιλαμβάνει:

- Όλο τον html κώδικα
- Όλο τον JavaScript και Ajax κώδικα

Για την υλοποίηση της εφαρμογής δημιουργήθηκε μία βάση δεδομένων για την καταχώρηση και τον έλεγχο των πληροφοριών όλου του συστήματος.

6.4 Ανάλυση κώδικα

Σε αυτό το σημείο θα περιγραφούν κομμάτια κώδικα και το πώς συνδέονται μεταξύ τους. Θα δοθεί ιδιαίτερη προσοχή σε σημεία που περιλαμβάνουν ασφάλεια στον κώδικα.

Η σύνδεση είναι κρυπτογραφημένη με τον αλγόριθμο MD5. Η δήλωση έχει γίνει στο αρχείο tlbx.php. Ο κρυπτογράφος MD5 παράγει ένα άθροισμα ελέγχου για τα δύο σύνολα δεδομένων και στη συνέχεια συγκρίνοντας τα αθροίσματα ελέγχου επιβεβαιώνει ότι είναι τα ίδια. Όταν ένας επιτιθέμενος προσπαθήσει να υποκλέψει τους κωδικούς από κάποιον χρήστη της εφαρμογής θα αντιμετωπίσει επίπεδα δυσκολίας. Το αποτέλεσμα που θα πάρει θα είναι κάπως έτσι:

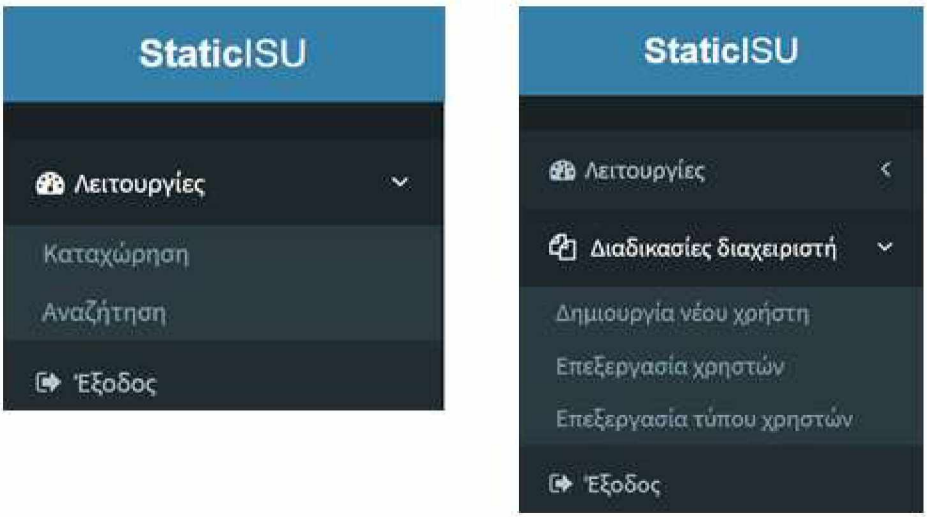
Jsmqw98&*&(^()*)Nou89

Θα χρειαστεί αρκετή προσπάθεια για να καταφέρει να «σπάσει» την κρυπτογραφία για να αποκτήσει πρόσβαση.

Για να εκτελεστούν όλες οι λειτουργίες θα πρέπει να γίνεται σύνδεση με την βάση. Αφού, γίνει αυτό ο εκάστοτε χρήστης για να πάρει κάποιο αποτέλεσμα κάνει ένα ερώτημα στην βάση και αν είναι ορθό πραγματώνεται διαφορετικά γυρίζει μήνυμα σφάλματος. Θα αναλυθεί κάθε μία λειτουργία ξεχωριστά.

6.4.1 Ασφάλεια κατά την αρχική είσοδο του χρήστη στο StaticISU

Αρχικά, ο επισκέπτης της εφαρμογής θα πρέπει να κάνει σύνδεση για να μπορέσει να εισέλθει σε αυτή. Ανάλογα, με τον τύπο του χρήστη που θα κάνει την σύνδεση θα του εμφανιστεί διαφορετικό μενού, συνεπώς θα μπορεί να δράξει και διαφορετικές λειτουργίες.



Εικόνα 27: Μενού StaticISU χρήστη και διαχειριστή

Η κρυπτογράφηση έχει χρησιμοποιηθεί σε διάφορα σημεία μέσα στον κώδικα για την προστασία των δεδομένων. Στην αρχική οθόνη εισαγωγής στην εφαρμογή, έχει χρησιμοποιηθεί κρυπτογράφηση MD5 ενώ τα στοιχεία το username και το password είναι κρυπτογραφημένα με τον αλγόριθμο AES-256-CBC.

Ο κώδικας για την είσοδο στο σύστημα βρίσκετε στο index.php αρχείο. Η δήλωση των κλειδιών κρυπτογράφησης έχουν γίνει στο αρχείο tlbx.php όπως φαίνεται και στον πίνακα 15.

```
<button onclick="Vd_CheckLogin();" type="submit" class="btn btn-primary btn-block btn-flat">Είσοδος</button>
```

Πίνακας 13: Περιγραφή html κώδικα στο index.php

```
<script>
 $(function () {
  $('input').iCheck({
    checkboxClass: 'icheckbox_square-blue',
    radioClass: 'iradio_square-blue',
    increaseArea: '20%' /* optional */
  });
 });

 function Vd_CheckLogin(){
        var username = $('#usr_nm').val();
        var password = $('#usr_ps').val();

        $.post('tlbx.php',{
                checkFor_pst:6,
```

```
                usr_nm_pst:username,
                usr_ps_pst:password},
                function(text)
                {
                        if(text == 0){
                                alert("Δώσατε λάθος στοιχεία εισόδου.");
                        }
                        if(text == 2){

                                window.location.href = "dashboard.php";
                        }


                });
 }
</script>
```

Πίνακας 14: Διάταξη function για τον έλεγχο των στοιχείων εισόδου στο αρχείο index.php

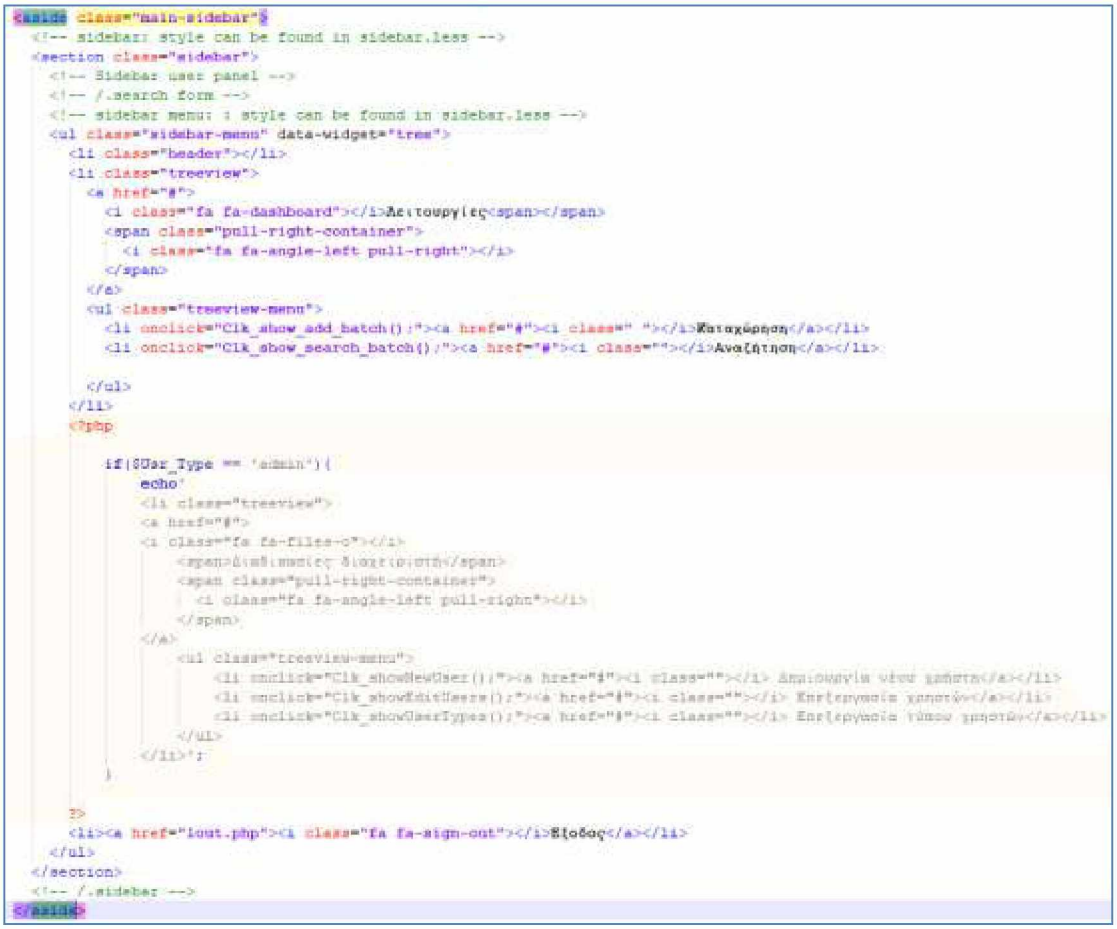
Η function ψάχνει το checkFor_pst:6 στο αρχείο tlbx.php για να ταυτοποιήσει τα στοιχεία που εισήγαγε ο χρήστης. Αφού λοιπόν, τα αποκρυπτογραφήσει, τα ελέγχει αν υπάρχουν στην βάση. Αν ταυτοποιούνται, τότε ξεκινάει η περιήγηση στην εφαρμογή. Σε αυτό το σημείο ανάλογα με τα δεδομένα εισόδου το σύστημα κάνει έλεγχο και για τα δικαιώματα που έχει ο εκάστοτε λογαριασμός. Αυτό σημαίνει πως κάνει διπλό έλεγχο και ανάλογα με τον τύπο χρήστη εκτελούνται διαφορετικά βήματα για την εισαγωγή του. Στον πίνακα 14 υπάρχουν όλα τα βήματα που εκτελούνται. Τώρα, αν δεν ταυτοποιηθούν τα στοιχεία, εμφανίζεται το μήνυμα «Δώσατε λάθος στοιχεία εισόδου» και ο χρήστης θα πρέπει να ξαναβάλει τα στοιχεία του από την αρχή.

```
if($ToolAction == 6){

    function vd_decrypt_usr($val,$usr_iv,$usr_pk)    {

        $outpout             = "";
        $encrypt_method  = 'AES-256-CBC';
        $secret_key          = $usr_pk;
        $secret_iv           = $usr_iv;
        $key                   = hash('sha256',$secret_key);
        $initialization_vector = substr(hash('sha256',$secret_iv),0,16);
        $outpout               =   base64_decode($val);
        $outpout               = openssl_decrypt($val,$encrypt_method,$key,0,$initialization_vector);
        return                 $outpout ;
                }

   function vd_encrypt_usr($val,$usr_iv,$usr_pk)    {

        $outpout             = "";
        $encrypt_method  = 'AES-256-CBC';
        $secret_key          = $usr_pk;
```

```
        $secret_iv          = $usr_iv;
        $key                = hash('sha256',$secret_key);
        $initialization_vector  = substr(hash('sha256',$secret_iv),0,16);
        $outpout            = openssl_encrypt($val,$encrypt_method,$key,0,$initialization_vector);
        return              $outpout;
                           }

        $user_pst = $_POST['usr_nm_pst'];
        $user_pst = mb_strtolower($user_pst);
        $user_pst = trim($user_pst);
        $user_pst = vd_encrypt_usr($user_pst,$sha_secret_iv,$secret_key);
        $pass_pst = $_POST['usr_ps_pst'];
        $pass_pst = md5($pass_pst);
        $sql = "SELECT Name,LastName,Usr_Name,Password,Usr_Type,UserAM,id,Email,Phone
FROM users WHERE Usr_Name = '$user_pst';";
        $result = $conn->query($sql);
        if ($result->num_rows > 0) {
            while($row = $result->fetch_assoc()) {
                if($user_pst == $row["Usr_Name"]){
                    if($pass_pst == $row["Password"]){
                        session_start();

            $_SESSION['Usr_Name'] = vd_decrypt_usr($row["Name"],$sha_secret_iv,$secret_key);
            $_SESSION['Usr_LastName']= vd_decrypt_usr($row["LastName"],
                                        $sha_secret_iv,$secret_key);
           $_SESSION['Usr_AM']=vd_decrypt_usr($row["UserAM"],$sha_secret_iv,$secret_key);
          $_SESSION['Usr_Id'] =  vd_decrypt_usr($row["id"],$sha_secret_iv,$secret_key);
          $_SESSION['Usr_Eml'] =  vd_decrypt_usr($row["Email"],$sha_secret_iv,$secret_key);
          $_SESSION['Usr_Phn'] =  vd_decrypt_usr($row["Phone"],$sha_secret_iv,$secret_key);
           $_SESSION['usr_type']=  vd_decrypt_usr($row["Usr_Type"],$sha_secret_iv,$secret_key);
              echo "2";
                }
        else {
                $c = $row["Password"];
                echo "0";
            }
                } else {
                   echo "0";
                        }
                                }
                        } else {
                                echo "0";
                        }
                        die;
                }
```

Πίνακας 15: Διαδικασία ελέγχου στοιχείων χρηστών κατά τη είσοδο στο αρχείο tlbx.php

Επομένως, το αποτέλεσμα των δύο διαφορετικών μενού παρουσιάζεται στην εικόνα 28. Σε αυτό το σημείο, έχει χρησιμοποιηθεί η γλώσσα προγραμματισμού PHP μέσα σε γλώσσα HTML για να εμφανιστούν οι επιπλέον λειτουργίες στο διαχειριστικό μενού όταν συνδεθεί στην εφαρμογή ο Admin.



Εικόνα 28: Αναδυόμενο μενού ανάλογα με τον τύπο χρήστη

6.4.1 Ασφάλεια των πληροφοριών του χρήστη στο StaticISU

Όλες οι πληροφορίες των χρηστών είναι κρυπτογραφημένες στη βάση όπως έχει αναφερθεί παραπάνω. Σε αυτό το σημείο θα περιγραφεί ο κώδικας ώστε, τα δεδομένα αυτά να είναι επεξεργάσιμα από την μεριά του administrator. Ο διαχειριστής είναι ο μόνος που μπορεί να δει τα προσωπικά δεδομένα των χρηστών εκτός από τον κωδικό πρόσβασης. Χρησιμοποιείται μέθοδος αποκρυπτογράφησης ώστε να μπορούν να φανερώνονται και να είναι επεξεργάσιμα τα δεδομένα των users. Το αποτέλεσμα της διαδικασίας φαίνεται στην οθόνη «επεξεργασίας χρηστών».

```
function Vd_Edit_Usr(id){

        var id_to_db = id.substring(8);
        $.post('tlbx.php',{
        checkFor_pst:3,
        Edit_Id_pst:id_to_db},
```

```
        function(text)
        {
                console.log(text);
                var res = text.split(":");
        //      alert(text);
                $("#id_edt_usr").val(res[0]);
                $("#id_edt_name_new_usr").val(res[1]);
                $("#id_edt_lastname_new_usr").val(res[2]);
                $("#id_edt_phone_new_usr").val(res[3]);
                $("#id_edt_email_new_usr").val(res[4]);
                $("#id_edt_AM_new_usr").val(res[5]);
                $("#id_edt_usrnm_new_usr").val(res[6]);
                $("#id_edt_usrps_new_usr").val(res[7]);
                $("#id_edt_now_type_usr").val(res[8]);
                $("#id_edt_now_type_usr").html(res[8]);


        });

}

function Vd_RefreshPage() {//Ανανέωση σελίδας
        location.reload();
}
```

Πίνακας 16: Επεξεργασία χρήστη - αρχείο dashboard.php

```
if($ToolAction == 3){

    function vd_decrypt_usr($val,$usr_iv,$usr_pk)    {

        $outpout                = "";
        $encrypt_method         = 'AES-256-CBC';
        $secret_key             = $usr_pk;
        $secret_iv              = $usr_iv;
        $key                    = hash('sha256',$secret_key);
        $initialization_vector  = substr(hash('sha256',$secret_iv),0,16);
        $outpout                =   base64_decode($val);
        $outpout            = openssl_decrypt($val,$encrypt_method,$key,0,$initialization_vector);
        return                      $outpout ;
                    }

      $UserUD                   =  $_POST['Edit_Id_pst'];
        $sql = "SELECT id,Name,LastName,Phone,Email,UserAM,Usr_Name,Password,Usr_Type
FROM users WHERE id = '$UserUD';";

        $result = $conn->query($sql);
            if ($result->num_rows > 0) {
                while($row = $result->fetch_assoc()) {
                    echo $row["id"].':'
                .vd_decrypt_usr($row["Name"],$sha_secret_iv,$secret_key).':'
                .vd_decrypt_usr($row["LastName"],$sha_secret_iv,$secret_key). ':'
                .vd_decrypt_usr($row["Phone"],$sha_secret_iv,$secret_key). ':'
                .vd_decrypt_usr($row["Email"],$sha_secret_iv,$secret_key). ':'
```

```
                    .vd_decrypt_usr($row["UserAM"],$sha_secret_iv,$secret_key). ':'
                    .vd_decrypt_usr($row["Usr_Name"],$sha_secret_iv,$secret_key). ':'
                    .$row["Password"]. ':'
                    .vd_decrypt_usr($row["Usr_Type"],$sha_secret_iv,$secret_key) ;

                            }
                    }

            }
```

Πίνακας 17: Εύρεση στοιχείων χρήστη για επεξεργασία - αρχείο tlbx.php

```
<div id="3_show_all_users" class="col-md-12" style="display:none;">
    <section class="content">
        <div class="box box-success ">
            <div class="box-header with-border">
              <h5 class="box-title">Επεξεργασία χρηστών</h5>
            </div><div class="box-body">
                <br><div class=""><div class="col-xs-12">
                        <div class="text-left" style="margin-bottom:15px;">
                            <button type="submit" onclick="Vd_RefreshPage();"
                            class="btn btn-primary">Ανανέωση</button>
                        </div><div class="box">
                        <div class="box-header">
                            <h5 class="box-title">Προβολή και επεξεργασία χρηστών.</h5></div>
                            <!-- /.box-header -->
                            <div class="box-body table-responsive no-padding">
                                <table id="all_usrs"  class="display" style="width:100%">
                                    <thead><tr>
                                            <th>ID</th>
                                            <th>Name</th>
                                            <th>LastName</th>
                                            <th>Phone</th>
                                            <th>Email</th>
                                            <th>UserAM</th>
                                            <th>Usr_Name</th>
                                            <th>Password</th>
                                            <th>Usr_Type</th>
                                            <th>Last_reg_date</th>
                                            <th>Actions</th>
                                        </tr>
                                    </thead>
                                    <tbody>
                                        <?php echo "$Vd_Get_Usrs_Table"; ?>
                                    </tbody>
                                    <tfoot><tr>
                                            <th>ID</th>
                                            <th>Name</th>
                                            <th>LastName</th>
                                            <th>Phone</th>
                                            <th>Email</th>
                                            <th>UserAM</th>
                                            <th>Usr_Name</th>
                                            <th>Password</th>
                                            <th>Usr_Type</th>
                                            <th>Last_reg_date</th>
                                            <th>Actions</th>
                                        </tr></tfoot></table>
                                </div></div></div></div></div></div></section>
</div>
```

Εικόνα 29: Εμφάνιση οθόνης επεξεργασίας χρηστών

Επειδή είναι αδύνατο να περιγραφεί όλος ο κώδικας σε μια εργασία, όποιος θελήσει θα μπορέσει να μελετήσει τα κομμάτια κώδικα μέσα από τα αρχεία της εφαρμογής.

Ασφαλής σύνδεση στο διαδίκτυο

Το HTTPS (Hyper Text Transfer Protocol Secure) είναι μια ασφαλής κρυπτογραφημένη έκδοση του HTTP (Hyper Text Transfer Protocol). Είναι το πρωτόκολλο για την αποστολή των δεδομένων μεταξύ του προγράμματος περιήγησης που χρησιμοποιείτε και του ιστότοπου που χρησιμοποιείτε μόνο για περιήγηση. Το HTTPS αποστέλλει όλα τα ευαίσθητα δεδομένα σε κρυπτογραφημένη μορφή, ενώ το HTTP στέλνει όλα τα δεδομένα σε απλό κείμενο. Το SSL χρησιμοποιεί έναν μαθηματικό αλγόριθμο για να κρύψει την πραγματική σημασία των δεδομένων. Ο αλγόριθμος είναι τόσο περίπλοκος ώστε είναι αδύνατο και απαγορευτικά δύσκολο να σπάσει.

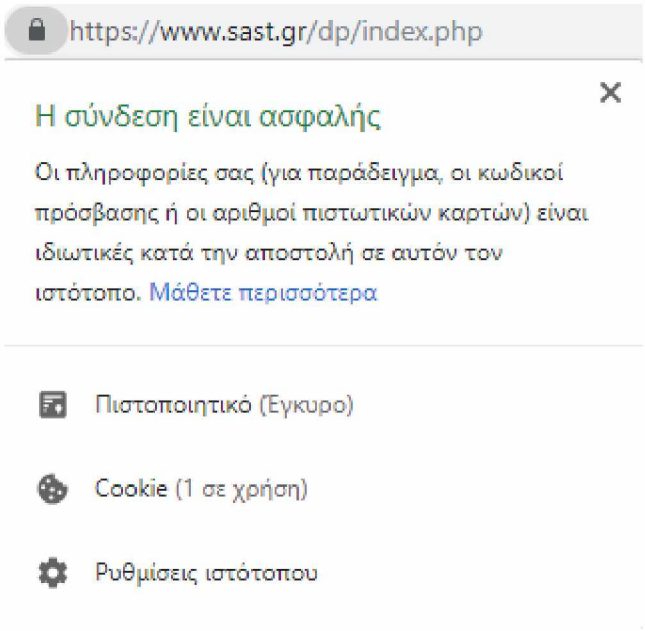
Το HTTPS αποτελεί τη βασική εμπιστοσύνη που μπορεί να δοθεί στους χρήστες.

Για να μετατραπεί από απλή σύνδεση στο δίκτυο σε κρυπτογραφημένη σύνδεση, γίνεται παρέμβαση στον κώδικα της εφαρμογής.

Με την χρήση της PHP γίνεται δήλωση στο πάνω μέρος των αρχείων dashboard.php και index.php ο κώδικας:

```php
<?php
if(empty($_SERVER['HTTPS']) || $_SERVER['HTTPS'] == "off"){
    $redirect = 'https://' . $_SERVER['HTTP_HOST'] . $_SERVER['REQUEST_URI'];
    header('HTTP/1.1 301 Moved Permanently');
    header('Location: ' . $redirect);
    exit();
}
?>
```

Έτσι, μόλις κάποιος χρήστης πληκτρολογήσει στο URL την διεύθυνση της εφαρμογής, πριν ακόμα κάνει login θα προσέξει πως δίπλα αριστερά θα υπάρχει ένα σύμβολο λουκέτου. Αυτό σημαίνει πως η σύνδεση είναι ασφαλής από επιθέσεις και τα δεδομένα κρυπτογραφούνται. Αν επιλεγεί το λουκέτο θα εμφανίσει το παρακάτω μήνυμα που φαίνεται στην εικόνα.



Εικόνα 30: Πιστοποίηση ασφαλούς σύνδεσης στην εφαρμογή

Σύνοψη

Για την ολοκλήρωση του σταδίου της υλοποίησης της εφαρμογής πραγματοποιήθηκαν αρκετοί έλεγχοι για την σωστή λειτουργία της και την αποτελεσματικότητά της με απώτερο σκοπό την επίτευξη των απαιτήσεων που είχαν τεθεί στην αρχή της εργασίας.

Ένας από τους στόχος είναι η χρησιμοποίηση μεθόδων κρυπτογράφησης ώστε να προστατεύονται τα δεδομένα από κακόβουλες επιθέσεις.

Για την κρυπτογράφηση της εφαρμογής χρησιμοποιήθηκε ο αλγόριθμος κρυπτογράφησης MD5 και ο αλγόριθμος AES-256-CBC. Είναι δύο ισχυροί αλγόριθμοι για τους οποίους δεν υπάρχει η υπολογιστική δύναμη για να διαπεράσει κάποιος την ασφάλεια που προσφέρουν.

Σε ένα υποθετικό σενάριο παραβίασης της εφαρμογής, ο επιτιθέμενος αν κατάφερνε να μπει στην εφαρμογή το πιο πιθανό είναι πως δεν θα πετύχαινε να αποκωδικοποιήσει τα δεδομένα της βάσης αφού είναι κρυπτογραφημένα. Η ζημιά που θα μπορούσε ενδεχομένως να επιφέρει στην εφαρμογή, είναι να κάνει insert απεριόριστα batch με αποτέλεσμα να δημιουργήσει πρόβλημα στην βάση. Ακόμα, θα μπορούσε να διαγράψει όλους τους χρήστες από το σύστημα πέρα του admin.

ΚΕΦΑΛΑΙΟ 7 – ΑΝΙΧΝΕΥΣΗ ΚΑΙ ΔΙΟΡΘΩΣΗ ΣΦΑΛΜΑΤΩΝ

7.1 Εισαγωγή

Το τελευταίο στάδιο ανάπτυξης της εφαρμογής περιλαμβάνει την εγκατάσταση του συστήματος σε έναν εξυπηρετητή διαδικτύου με σκοπό την διεξαγωγή δοκιμών λειτουργίας και την διόρθωση σφαλμάτων.

Στη θεωρία της **πληροφορίας** και τη θεωρία **κωδικοποίησης** με εφαρμογές στη επιστήμη των υπολογιστών και τις τηλεπικοινωνίες, η **ανίχνευση** και η **διόρθωση σφαλμάτων** ή ο **έλεγχος σφαλμάτων** είναι τεχνικές που επιτρέπουν την αξιόπιστη παράδοση ψηφιακών δεδομένων μέσω ανακριβών καναλιών επικοινωνίας. Πολλά κανάλια επικοινωνίας υπόκεινται σε **θόρυβο** καναλιού και έτσι μπορούν να εισαχθούν σφάλματα κατά τη μετάδοση από τη πηγή σε ένα δέκτη. Οι τεχνικές ανίχνευσης σφαλμάτων επιτρέπουν την ανίχνευση τέτοιων σφαλμάτων, ενώ η διόρθωση σφαλμάτων επιτρέπει σε πολλές περιπτώσεις την ανασυγκρότηση των αρχικών δεδομένων.

Ο editor που χρησιμοποιήθηκε για την ανάπτυξη του κώδικα είναι το Visual Studio Code. Ένα από τα βασικά χαρακτηριστικά του Visual Studio Code είναι η μεγάλη υποστήριξή του στην εκσφαλμάτωση. Το ενσωματωμένο πρόγραμμα εντοπισμού σφαλμάτων του κώδικα VS βοηθά στην επιτάχυνση του βρόχου επεξεργασίας (edit), μεταγλώττισης (compile) και εντοπισμού σφαλμάτων (debug loop).

Το Debugging είναι η διαδικασία εύρεσης και επίλυσης ελαττωμάτων ή προβλημάτων σε ένα πρόγραμμα υπολογιστή που εμποδίζουν τη σωστή λειτουργία του λογισμικού ή του συστήματος.

Για την εμφάνιση της προβολής εντοπισμού σφαλμάτων, γίνετε κλικ στο εικονίδιο Debug στη γραμμή δραστηριότητας στο πλάι του κώδικα VS. Η προβολή εντοπισμού σφαλμάτων εμφανίζει όλες τις πληροφορίες που σχετίζονται με την εκσφαλμάτωση και έχει μια γραμμή κορυφής (top bar) με εντολές εντοπισμού σφαλμάτων και ρυθμίσεις παραμέτρων.

7.2 Διαδικασία δοκιμών και αποσφαλμάτωσης

Μπορεί να διαχωριστεί η διαδικασία δοκιμών και αποσφαλμάτωσης σε δύο διακριτές φάσεις.

- Η πρώτη φάση διήρκησε καθ' όλη την περίοδο συγγραφής του κώδικα. Κατά την υλοποίηση των συναρτήσεων και των λειτουργικών τμημάτων της εφαρμογής γινόταν συνεχόμενες δοκιμές της λειτουργικότητάς τους με σκοπό να εντοπιστούν τα συντακτικά λάθη, τα λάθη χρόνου εκτέλεσης καθώς και λογικά λάθη. Γι αυτόν τον σκοπό υλοποιήθηκαν βοηθητικά script που προσομοίωναν την είσοδο στην υπό κατασκευή, λειτουργική μονάδα.

- Η δεύτερη φάση δοκιμών ξεκίνησε μετά την μερική ολοκλήρωση του κώδικα του συστήματος και περιλάμβανε δοκιμές βασισμένες σε σενάρια χρήσης. Οι δοκιμές έγιναν στο διαδίκτυο, πάντα βάσει σεναρίων, που περιελάμβαναν ποικίλες περιπτώσεις χρήσης. Κατά την διάρκεια αυτής της φάσης δοκιμών εντοπίστηκαν λογικά λάθη στον κώδικα της εφαρμογής και έγιναν αρκετές βελτιστοποιήσεις. Επίσης έγιναν διορθωτικές παρεμβάσεις στην λειτουργικότητα του συστήματος.

- Κατά την τελική φάση δοκιμών του συστήματος χρησιμοποιήθηκαν τρεις φυλλομετρητές[6] με τους οποίους θα έπρεπε να είναι συμβατό το σύστημα. Δεν προέκυψαν ασυμβατότητες, μόνο μικρές διαφορές στην εμφάνιση του συστήματος που δεν επηρεάζουν την λειτουργικότητά του. Επιπλέον οι διαδικτυακές δοκιμές έδειξαν ότι το σύστημα λειτουργεί άψογα σε περιβάλλον Linux.

Συνολικά το τελικό σύστημα πληροί τις λειτουργικές και μη απαιτήσεις που τέθηκαν στην πρώτη φάση ανάπτυξης. Βέβαια κατά την φάση δοκιμών έγιναν παρεμβάσεις στην λειτουργικότητα του συστήματος αποκλίνοντας από τον αρχικό σχεδιασμό, κάτι που θεωρείται απολύτως φυσιολογικό και αναμενόμενο.

Παραδείγματα αλλαγών που έγιναν στην λογική του συστήματος κατά τις δοκιμές είναι:

- Προσθήκη μιας βοηθητικής σελίδας στο μενού

[6] Chrome, Mozilla και Internet explorer

ΚΕΦΑΛΑΙΟ 8 - ΕΠΙΛΟΓΟΣ

8.1 Συμπέρασμα

Η αυξημένη χρήση του Διαδικτύου μεταξύ των εταιρειών και των ατόμων έχει επηρεάσει τον τρόπο λειτουργίας των επιχειρήσεων. Αυτό, οδήγησε στην ευρεία υιοθέτηση των εφαρμογών ιστού, καθώς οι εταιρείες μετατοπίζονται από παραδοσιακά μοντέλα σε μοντέλα που βασίζονται σε cloud-based και grid models. Οι εφαρμογές web δίνουν στις επιχειρήσεις τη δυνατότητα να εξορθολογήσουν τις λειτουργίες τους, να αυξήσουν την αποδοτικότητά τους και να μειώσουν το κόστος τους.

Αυτές οι εφαρμογές παρέχουν την ίδια λειτουργικότητα με τις εκδόσεις της επιφάνειας εργασίας (desktop versions). Ωστόσο, έχουν ένα επιπλέον πλεονέκτημα, να εργάζονται σε πολλαπλές πλατφόρμες, να έχουν ευρύτερη εμβέλεια και να είναι εύκολα προσβάσιμες από οπουδήποτε.

Μετά την ολοκλήρωση της εφαρμογής οι υπάλληλοι μπορούν να χρησιμοποιούν τις λειτουργίες της εφαρμογής με ασφάλεια και ευχρηστία. Ο αλγόριθμος MD5 που έχει χρησιμοποιηθεί παρέχει μεγάλη ασφάλεια για τους κωδικούς πρόσβασης μιας και δεν έχει αποκρυπτογραφηθεί. Παρόλα αυτά, στην περίπτωση που κάποιος επιτιθέμενος καταφέρει να ξεπεράσει αυτό το στάδιο ασφάλειας, πιθανά μέσω Man In The Middle επίθεση, δεν θα καταφέρει να αποκρυπτογραφήσει τα ευαίσθητα προσωπικά δεδομένα και τις πληροφορίες από την βάση. Αυτό συμβαίνει λόγο της επιπλέον κρυπτογράφησης με τον αλγόριθμο AES-256-CBC όπως έχει αναλυθεί σε προηγούμενο κεφάλαιο. Αυτό σημαίνει πως η εφαρμογή είναι πλήρως αξιόπιστη και έτοιμη προς συστηματική χρήση.

Οι υπάλληλοι θα είναι σε θέση να διαχειρίζονται τον μεγάλο όγκο των δεδομένων που θα αποθηκεύουν καθημερινά στην βάση στου StaticISU.

Στην περίπτωση της περεταίρω ανάπτυξης της εφαρμογής, η λογική που έχει αναπτυχθεί ο κώδικας μέσω PHP βοηθάει στην κατανόησή του και από κάποιον άλλο web developer και όχι μόνο από τον δημιουργό του. Επιπλέον, ο ίδιος κώδικας μπορεί να χρησιμοποιηθεί σαν οδηγός και για άλλες παρόμοιες εφαρμογές.

Η χρήση ασύγχρονων αιτημάτων επιτρέπει στην εφαρμογή να είναι πιο διαδραστική και να ανταποκρίνεται γρηγορότερα στις ενέργειες του χρήστη. Με την χρήση ajax υπάρχει μείωση των απαιτούμενων συνδέσεων προς τον διακομιστή. Ακόμα, η κεντρική σελίδα

(container page) φορτώνεται μία φορά και οι μεταβλητές JavaScript παραμένουν φορτωμένες μέχρι τη λήξη της σύνδεσης.

8.1 Εκτιμώμενη Διάρκεια Έργου

Ένα έργο αποτελείται από μια ακολουθία δραστηριοτήτων όπου με τη χρήση απαραίτητων πόρων ολοκληρώνουν έναν συγκεκριμένο σκοπό σε προκαθορισμένο χρόνο. Κάθε διεργασία (ή δραστηριότητα) έχει ειδικές απαιτήσεις και περιορισμένη διάρκεια. Επομένως η διάρκεια ενός έργου, από την φάση σύλληψης του μέχρι και την ολοκλήρωσή του, μπορεί να εκτιμηθεί με μεγάλη ακρίβεια.

Όπως και στη πραγματική ζωή, έτσι και στη ζωή ενός έργου υπάρχουν διαταρακτικοί παράγοντες (π.χ. καιρικές συνθήκες, ζητήματα υγείας και άλλα) οι οποίοι μπορούν να αποκλίνουν το έργο από τον εκτιμώμενο χρόνο και δεν είναι εφικτό να προβλεφθούν.

Στο παρακάτω διάγραμμα παρουσιάζεται χρόνος δημιουργίας και ολοκλήρωσης της παρούσας εργασίας, ο οποίος και αναλύεται σε διεργασίες ανά εβδομάδα.

	Task	Start	End	Dur	2018
					Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
	StaticISU	10/2/18	28/12/18	223	
1	Μελέτη τεχνολογιών και εργαλείων	10/2/18	25/2/18	9	
2	Έρευνα στον τομέα τεχνολογίας λογισμικού	25/2/18	2/3/18	5	
3	Ανάλυση απαιτήσεων	3/3/18	12/3/18	6	
4	Σχεδιασμός	15/3/18	29/3/18	11	
5	Υλοποίηση	30/3/18	23/8/18	103	
6	Δοκιμές	5/6/18	30/8/18	62	
7	Συγγραφή διπλωματικής εργασίας	5/11/18	28/12/18	38	

Εικόνα 31: Διάγραμμα Gantt για StaticISU

Οι μπλε ράβδοι αναπαριστούν τον πραγματικό χρόνο των. Το εργαλείο που χρησιμοποιείται για την χρονική αναπαράσταση του έργου, είναι το SmartDraw. Το πρόγραμμα αυτό χρησιμοποιεί αυτοματοποιημένα εργαλεία σχεδίασης που καθιστούν γρήγορη και εύκολη τη δημιουργία ενός καλαίσθητου οπτικού αποτελέσματος το οποίο απεικονίζει τις ροές εργασίας, τις λειτουργίες, τα διαγράμματα, καθώς και πολλές άλλες δραστηριότητες ενός έργου.

8.2 Επίδραση Εργασίας στον Αναγνώστη

Μελετώντας κάποιος την παρούσα εργασία διαπιστώνει πως αυτή μπορεί να επηρεάσει θετικά τις γνώσεις του. Ο αναγνώστης ενημερώνεται για τα τεχνικά αλλά και τα λειτουργικά μέρη της εφαρμογής. Συγκεκριμένα τα αποτελέσματα μετά το τέλος της μεταπτυχιακής εργασίας είναι:

- Γνώσεις πάνω σε γλώσσες προγραμματισμού που χρησιμοποιούνται για την ανάπτυξη μιας διαδικτυακής εφαρμογής.
- Μελέτη και παρουσίαση πληροφοριών για διάφορα προγράμματα όπως το Astah και το Microsoft Visio που βοήθησαν στη διαγραμματική απεικόνιση και κατανόηση των λειτουργιών των χρηστών και του συστήματος.
- Μελέτη και προτάσεις βημάτων για την ανάπτυξη ενός λογισμικού. Ο προγραμματισμός και η εφαρμογή ενός πλάνου για την δημιουργία και την ολοκλήρωση ενός λογισμικού θεωρούνται απαραίτητα στοιχεία για την σωστή προσέγγιση μιας αντίστοιχης εφαρμογής.
- Παρουσίαση και εξοικείωση με τον τρόπο λειτουργίας μια διαδικτυακής εφαρμογής που βασίζεται στη συμμετοχή χρηστών και την ανεπτυγμένη ασφάλεια.
- Πρακτική εφαρμογή της κρυπτογράφησης σε μια εφαρμογή. Ενημέρωση για τα προσφερόμενα οφέλη.

8.3 Πιθανές Επεκτάσεις Συστήματος και Προτάσεις για Έρευνα

Η Εφαρμογή που δημιουργήθηκε είναι δυναμική και παρέχει ανεξέλεγκτες δυνατότητες ανάπτυξης και επέκτασης. Παρακάτω καταγράφονται κάποιες από τις επεκτάσεις που μπορούν να γίνουν μελλοντικά για την καλύτερη και αποδοτικότερη χρήση του StaticISU.

- Στα μελλοντικά σχέδια την εφαρμογής προβλέπετε να προστεθεί μια επιπλέον λειτουργία που θα αφορά την εξαγωγή στατιστικών στοιχείων των αποθηκευμένων δεδομένων. Αναλυτικότερα, θα δημιουργηθεί φόρμα από την οποία θα επιλέγει ο χρήστης πληροφορίες και με το πάτημα ενός κουμπιού «εξαγωγή». Μετέπειτα, θα

εμφανίζεται η γραφική παράσταση με τα δεδομένα που τέθηκαν. Για παράδειγμα, ο χρήστης θα επιλέγει την επιθυμητή αλυσίδα και ένα χρονικό διάστημα τριών μηνών (Μάρτιος με Μάιο) του 2017 και του 2018. Το αποτέλεσμα που θα προκύπτει θα είναι μια γραφική παράσταση ώστε να μπορεί να γίνει σύγκριση του χρόνου, των δραστηριοτήτων, των καταχωρήσεων και όποιον άλλων δεδομένων ζητηθούν για την συλλογή των αποτελεσμάτων.

- Ισχυρότερο και ασφαλέστερο λογισμικό από επιθέσεις δημιουργώντας VPN συνδέσεις στην περίπτωση που κάποιος θελήσει να φτιάξει μια άλλη εφαρμογή με γνώμονα την συγκεκριμένη εργασία.
- Σχεδιασμός της εφαρμογής ώστε να είναι εύκολα διαχειρίσιμη από ανθρώπους με δυσκολία όρασης.

Βιβλιογραφία

[1] Jim Conallen, "Web Application," *Communications of the ACM*, 1999.

[2] Amos Ndegwa. (2016, May 31) What is a Web Application Firewall (WAF)? [Online]. https://www.maxcdn.com/one/visual-glossary/web-application/

[3] Jeremy Petersen, "Benefits of Using the N-Tiered Approach for Web Applications," ColdFusion Developer Center, 2007.

[4] Debashis Davison, A. C. Hinkley, D. V. Kushary, "Bootstrap Methods and Their Application," *Technometrics*, 2000, https://doi.org/10.2307/1271471.

[5] Rasmus Tatroe, Kevin MacIntyre, Peter Lerdorf, *Programming PHP*.: OReilly, 2006, https://doi.org/10.1145/1047124.1047480.

[6] Michael K. Le Scouarnec, Y. Naramore, E. Mailer, G. Stolz, J. & Gerner, J Glass, *Beginning PHP, Apache, MySQL Web Development*.: John Wiley & Sons, 2004.

[7] Chanchai Supaartagorn, "PHP Framework for database management based on MVC pattern," *International Journal of Computer Science & Information Technology (IJCSIT)*, pp. 251-258, 2011.

[8] Daniel Bartholomew, *MariaDB cookbook*.: Packt Publishing Ltd, 2014.

[9] Daniel Bartholomew, *Getting Started with MariaDB*.: Packt Publishing Ltd, 2013.

[10] Ansgar Becker. (2015) heidisql. [Online]. https://www.heidisql.com/

[11] Marc Delisle, *Mastering phpMyAdmin 3.1 for effective MySQL management*.: Packt Publishing Ltd, 2009.

[12] Dedi Iskandar and Ratna Juita Inan, "Analysis and design complex and large data base using MySQL workbench," *International Journal of Computer Science & Information Technology*, vol. 3, no. 5, p. 173, 2011.

[13] Tatu, and Chris Lonvick Ylonen, The secure shell (SSH) protocol architecture, 2005.

[14] L. Masinter D. Connolly. (2000, June) RFC 2854. [Online]. https://tools.ietf.org/html/rfc2854

[15] Jon Duckett, *HTML & CSS: design and build websites*. IN: Wiley, 2011, vol. 15.

[16] Press release announcing JavaScript. (1995, December) Netscape and Sun announce JavaScript.

[17] David Flanagan, *JavaScript: the definitive guide*.: O'Reilly Media, Inc, 2006.

[18] Maximiliano Firtman, *jQuery Mobile: Up and Running: Up and Running*.: O'Reilly Media, Inc, 2012.

[19] G., Wood, L., Champion, M., & Byrne, S Nicol, *Document Object Model (DOM) level 3 core specification*.: W3C Working Draft, 2001, vol. 13.

[20] SourceForge. (2018) [Online]. https://sourceforge.net/projects/winscp/

[21] Ars Technica. (2015, November) Visual Studio now supports debugging Linux apps; Code editor now open source. [Online]. https://arstechnica.com/information-technology/2015/11/visual-studio-now-supports-debugging-linux-apps-code-editor-now-open-source/

[22] Microsoft. (2016, August) Microsoft Software License Terms. [Online]. code.visualstudio.com

[23] Stack Exchange. (2018, April) Developer Survey Results 2018.

[24] The Apache Software Foundation. (2015, February) HTTP Server Project. [Online]. https://httpd.apache.org/

[25] Chris, and Lucinda Dykes Ullman, *Beginning Ajax*.: John Wiley & Sons, 2007.

[26] Li, Zong-yuan YANG, and Jin-kui XIE Tan, "Data response optimization of ajax," vol. 021, July 2010.

[27] J., Menezes, A. J., Van Oorschot, P. C., & Vanstone, S. A Katz, *Handbook of applied cryptography*.: CRC press, 1996.

[28] Friedrich L. Bauer, *Cryptanalysis*.: Encyclopedia of Cryptography and Security, 2011.

[29] W. Stallings, *Cryptography and Network Security: Principles amd Practices*, 5th ed.: Network, 2011.

[30] Jawahar, and Nagesh Kumar Thakur, *DES, AES and Blowfish: Symmetric key cryptography algorithms simulation based performance analysis*.: International journal of emerging technology and advanced engineering, 2011, vol. 1.

[31] John E. Canavan, *Fundamentals of Network Security*.: Computer Standards & Interfaces, 2002.

[32] R. Housley Spyrus. (1999, January) RFC 2459.

[33] Harold F., and Micki Krause Nozaki Tipton, *Information security management handbook*.: CRC press, 2007.

[34] J., Ota, K., Dong, M., & Li, C Wu, *A Hierarchical Security Framework for Defending Against Sophisticated Attacks on Wireless Sensor Networks in Smart Cities*.: IEEE Access, 2016, vol. 4.

[35] ISO/IEC, *Information technology -- Security techniques-Information security risk management*,

27005th ed.: ISO/IEC FIDIS, 2008.

[36] H., Bellare, M., & Canetti, R Krawczyk. (1997, February) HMAC: Keyed-hashing for message authentication No. RFC 2104.

[37] B., Govaerts, R., & Vandewalle, J Preneel, *Hash functions based on block ciphers: A synthetic approach*. Heidelberg, Berlin: In Annual International Cryptology Conference, August 1993.

[38] R. Rivest. (1992, April) The MD5 Message-Digest Algorithm No. RFC 1321.

[39] John, Matt Messier, and Pravir Chandra Viega, *Network security with openSSL: cryptography for secure communications*.: O'Reilly Media, Inc, 2002.

[40] OpenSSL Software Foundation. (1999-2018) OpenSSL Cryprography ans SSL/TLS Toolkit. [Online]. https://www.openssl.org/docs/man1.0.2/apps/openssl.html

[41] The PHP Group. (2001-2018) openssl_encrypt. [Online]. http://php.net/manual/en/function.openssl-encrypt.php

[42] R. Glenn, S. Kelly S. Frankel, The AES-CBC Cipher Algorithm and Its Use with IPsec No RFC 3602, September 2003.

[43] National Institute of Standards and Technology. (2018, Octomber) Cryptographic Standards and Guidelines. [Online]. https://csrc.nist.gov/projects/cryptographic-standards-and-guidelines/archived-crypto-projects/aes-development

[44] E. Rescorla T. Dierks, The Transport Layer Security (TLS) Protocol Version 1.2 No RFC 5246, August 2008, Errata, Obsoletes RFC 3268, RFC 4346, RFC 4366, Obsoleted by RFC 8446, Updates RFC 4492, Updated by RFC 5746, RFC 5878, RFC 6176, RFC 7465, RFC 7507, RFC 7568, RFC 7627, RFC 7685, RFC 7905, RFC 7919, RFC 8447.

[45] P. Karlton, P. Kocher A. Freier, The Secure Sockets Layer (SSL) Protocol Version 3.0 No RFC 6101, August 2011.

[46] Marsh, and Steve Dispensa Ray. (2009) Renegotiating tls. [Online]. http://extendedsubset.com/

[47] Εμμανουήλ Σκοσρδαλάκης, *Εισαγωγή στην τεχνολογία λογισμικού*.: Συμμετρία, 1991.

[48] Grady Booch, *Object-Oriented Analysis and Design with Applications*, 3rd ed. Redwood, USA: Addison Wesley Longman Publishing Co., 2004.

ΟΡΟΛΟΓΙΕΣ

Διαδίκτυο: (αγγλ. Internet) είναι παγκόσμιο σύστημα διασυνδεδεμένων δικτύων υπολογιστών, οι οποίοι χρησιμοποιούν καθιερωμένη ομάδα πρωτοκόλλων για να εξυπηρετεί εκατομμύρια χρηστών καθημερινά σε ολόκληρο τον κόσμο. Οι διασυνδεδεμένοι ηλεκτρονικοί υπολογιστές ανά τον κόσμο, οι οποίοι βρίσκονται σε ένα κοινό δίκτυο επικοινωνίας, ανταλλάσσουν μηνύματα (πακέτα) με τη χρήση διαφόρων πρωτοκόλλων (τυποποιημένοι κανόνες επικοινωνίας), τα οποία υλοποιούνται σε επίπεδο υλικού και λογισμικού. Το κοινό αυτό δίκτυο καλείται Διαδίκτυο. Αναφέρεται και σαν Δίκτυο (Net) ή Διαδίκτυο ή Κυβερνοχώρος. (CyberSpace).

Παγκόσμιος ιστός: WWW (World Wide Web) είναι μια υπηρεσία του διαδικτύου και το πιο γρήγορα αναπτυσσόμενο τμήμα του Internet, που περιέχει πληροφορίες οι οποίες παρουσιάζονται με τη μορφή κειμένου, γραφικών, βίντεο και ήχου.

FTP: (File Transfer Protocol-Πρωτόκολλο Μεταφοράς Αρχείων) είναι μία μέθοδος διαχείρισης αρχείων σε κάποιον απομακρυσμένο υπολογιστή. Επιτρέπει την μεταφορά αρχείων από έναν υπολογιστή σε έναν άλλο (upload ή download) αλλά και επιπλέον δυνατότητες όπως η διαγραφή και μετονομασία αρχείων, ανάλογα με τα δικαιώματα χρήσης.

Email: ηλεκτρονικό ταχυδρομείο είναι μια Υπηρεσία του Διαδικτύου, η οποία επιτρέπει τη συγγραφή, αποστολή, λήψη και αποθήκευση μηνυμάτων με χρήση ηλεκτρονικών συστημάτων τηλεπικοινωνιών. Μία από τις πιο διαδεδομένες υπηρεσίες του Διαδικτύου.

Web page: Ιστοσελίδα είναι ένα είδος εγγράφου του παγκόσμιου ιστού (WWW) που περιλαμβάνει πληροφορίες με την μορφή κειμένου, υπερκειμένου, εικόνας, βίντεο και ήχου. Πολλές ιστοσελίδες μαζί συνθέτουν έναν ιστότοπο.

Web Site: (ιστοχώρος, δικτυακός τόπος, ιστότοπος ή Internet site) είναι η συλλογή σελίδων που είναι διαθέσιμες σε κάποιο σημείο του Παγκόσμιου Ιστού και παρουσιάζουν πληροφορίες σταθερές ή δυναμικές.

Home Page (εισαγωγική σελίδα): Γνωστή και ως αρχική σελίδα, αποτελεί την αρχική ιστοσελίδα ενός δικτυακού τόπου, από την οποία ξεκινάει η περιήγηση στον δικτυακό τόπο και έχει συνήθως την ονομασία index.html (που είναι και η συνηθέστερη) ή default.asp ή main.html ή home.html κ.ά.

Web browser: (φυλλομετρητής ιστοσελίδων, πλοηγός Web, πρόγραμμα περιήγησης Web ή περιηγητής Ιστού) είναι ένα λογισμικό που επιτρέπει στον χρήστη να προβάλλει, και να αλληλεπιδρά με κείμενα, εικόνες, βίντεο, μουσική, παιχνίδια και άλλες πληροφορίες συνήθως αναρτημένες σε μια ιστοσελίδα ενός ιστότοπου στον Παγκόσμιο Ιστό ή σε ένα τοπικό δίκτυο.

Hyperlink: (υπερδεσμός-υπερσύνδεσμος) Ο γενικός όρος που χρησιμοποιείται για τον γνωστό σύνδεσμο (link), ένα θερμό σημείο όπου μ" ένα απλό κλικ του ποντικιού σε μια λέξη ή φράση ή εικόνα ή πλήκτρο ή οτιδήποτε, μπορούμε να μεταφερθούμε σ" ένα άλλο σημείο στην ίδια ιστοσελίδα ή σε μια άλλη ιστοσελίδα του ίδιου δικτυακού τόπου ή σ" έναν άλλον δικτυακό τόπο.

Hypertext: (υπερκείμενο) Πρόκειται για τον τρόπο οργάνωσης πληροφοριών που βρίσκονται σε μορφή κειμένου. Επιτρέπει την ελεύθερη πλοήγηση του ανγνώστη η οποία επιτυγχάνεται με την χρήση υπερσυνδέσμων. Σε ένα υπερκείμενο ενσωματώνονται πληροφορίες όπως εικόνα ήχος και βίντεο. Τα δεδομένα στο υπερκείμενο αποθηκεύονται σε κόμβους οι οποίοι συνδέονται μεταξύ τους με συνδέσμους οι οποίοι επιτρέπουν το πέρασμα από το ένα κόμβο στον άλλον .Αποτελεί βασικό τρόπο οργάνωσης των ατελείωτων πληροφοριών που υπάρχουν στο Internet. Μεταφέρεται σύμφωνα με το Http(πρωτόκολλο μεταφοράς ιστοσελίδων).

Hypermedia: (Υπερμέσα) Σύνολο πληροφοριών αποτελούμενο από επιμέρους τμήματα που ονομάζονται κόμβοι και περιλαμβάνουν δεδομένα κάθε μορφής όπως κείμενο, εικόνες, ήχο, βίντεο, κ.λ.π. και τα οποία είναι διασυνδεδεμένα μεταξύ τους με συνδέσμους (Links). Τα υπερμέσα μπορούν να προσπελασθούν με πολλούς τρόπους ανάλογα με την διαδρομή που θα ακολουθήσει ο χρήστης μεταξύ των κόμβων. Όταν οι κόμβοι περιέχουν μόνο κείμενο, τα υπερμέσα ταυτίζονται με το υπερκείμενο.

Multimedia: (πολυμεσικές εφαρμογές, τίτλοι πολυμέσων ή πολυμέσα) είναι ο κλάδος της πληροφορικής τεχνολογίας που ασχολείται με τον συνδυασμό ψηφιακών δεδομένων πολλαπλών μορφών, δηλ. κειμένου, γραφικών, εικόνας, κινούμενης εικόνας (animation), ήχου και βίντεο, για την αναπαράσταση, παρουσίαση, αποθήκευση, μετάδοση και επεξεργασία πληροφοριών.

Πλοήγηση: (Navigation) Η δυνατότητα του χρήστη να κινείται ανάμεσα στα τμήματα της διαθέσιμης πληροφορίας σε ένα σύστημα υπερκειμένου ή υπερμέσων, ή με άλλες λέξεις να κινείται μεταξύ των κόμβων πληροφοριών. Το πιο χαρακτηριστικό παράδειγμα πλοήγησης είναι η χρήση του παγκοσμίου ιστού.

SQL: (Structured Query Language) είναι μία γλώσσα υπολογιστών στις βάσεις δεδομένων, που σχεδιάστηκε για τη διαχείριση δεδομένων, σε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (Relational Database Management System, RDBMS) και η οποία, αρχικά, βασίστηκε στη σχεσιακή άλγεβρα.

W3C: (World Wide Web Consortium) Η Κοινοπραξία του Παγκοσμίου Ιστού είναι μια διεθνής κοινοπραξία όπου οι οργανισμοί μέλη, το εξειδικευμένο προσωπικό πλήρους απασχόλησης και το κοινό συνεργάζονται για να αναπτύξουν πρότυπα του παγκοσμίου ιστού.

ICONIX: Μεθοδολογία ανάπτυξης λογισμικού, εξαιρετικά «ελαφριά» και ευέλικτη. Χρησιμοποιεί τέσσερα διαγράμματα UML για σχεδιασμό της εφαρμογής, περιλαμβάνει τέσσερα διαδοχικά βήματα που καταλήγουν στην παραγωγή λειτουργικού κώδικα από τις περιπτώσεις χρήσης.

UML: (Unified Modeling Language) Πρότυπη γλώσσα μοντελοποίησης στη μηχανική λογισμικού. Χρησιμοποιείται για τη γραφική απεικόνιση, προσδιορισμό, κατασκευή και τεκμηρίωση των στοιχείων ενός συστήματος λογισμικού. Μπορεί να χρησιμοποιηθεί σε διάφορες φάσεις ανάπτυξης, από την ανάλυση απαιτήσεων ως τον έλεγχο ενός ολοκληρωμένου συστήματος.

Activity Diagrams: Διαγράμματα Δραστηριότητας, Ένα διάγραμμα δραστηριότητας μοντελοποιεί τη ροή της εργασίας, αναπαριστώντας τις διάφορες καταστάσεις εκτέλεσης ενός υπολογισμού (Böhm & Jacopini). Τα διαγράμματα δραστηριοτήτων χρησιμοποιούνται για την απεικόνιση διαδικασιών, επιχειρηματικών διεργασιών και ροής εργασίας.

Use Case Diagram: Διάγραμμα περίπτωσης χρήσης, απεικονίζει την αλληλεπίδραση του συστήματος με τους χρήστες του ή και με άλλα συστήματα και περιγράφει με σχηματικό τρόπο τους χρήστες του συστήματος και τον τρόπο με τον οποίο αναμένουν να αλληλεπιδρούν με αυτό.

phpMyAdmin: Εργαλείο ανοιχτού κώδικα γραμμένο σε PHP που επιτρέπει τον χειρισμό της MySQL μέσω διαδικτύου.

Account: είναι ο λογαριασμός πρόσβασης σε ένα Web site και επιτυγχάνεται με τη χρήση ενός ονόματος χρήστη (user name) και ενός μυστικού κωδικού πρόσβασης (password). Ο λογαριασμός πρόσβασης δίνει στο κάτοχό του ορισμένα δικαιώματα.

Apache HTTP: είναι ένας από τους δημοφιλέστερους εξυπηρετητές παγκόσμιου ιστού. Όποτε ένας χρήστης επισκέπτεται ένα ιστότοπο το πρόγραμμα πλοήγησης (browser) επικοινωνεί με έναν διακομιστή (server) μέσω του πρωτοκόλλου HTTP, ο οποίος παράγει τις ιστοσελίδες και τις αποστέλλει στο πρόγραμμα πλοήγησης.

Server: Ο εξυπηρετητής είναι ένας υπολογιστής που έχει τον κεντρικό έλεγχο ενός δικτύου, παρέχοντας βασικές υπηρεσίες στους χρήστες του. Σε ένα δίκτυο ο server ελέγχει την όλη λειτουργεία και για λόγους ασφαλείας επιτρέπει την πρόσβαση μόνο σε εξουσιοδοτημένους χρήστες, παραχωρώντας τους λειτουργίες ανάλογα με τα δικαιώματα που έχουν.

URL: Uniform Resource Locator ή Ενιαίος Εντοπιστής Πόρων πρόκειται για τη πλήρη διεύθυνση μια ιστοσελίδας, που περιλαμβάνει το πρωτόκολλο επικοινωνίας, όπως http:// ή https://.

Web Design: αποκαλείται η διαδικασία της δημιουργίας ενός δικτυακού τόπου (ιστοσελίδας), που περιλαμβάνει τον αρχικό σχεδιασμό της εμφάνισής του έως και την τελική διάταξη (τοποθέτηση) των περιεχομένων του και τη δημιουργία των γραφικών στοιχείων.

Plug-in: αποδίδεται στα ελληνικά ως πρόσθετο και πρόκειται για ειδικά προγράμματα, που είναι μικρά σε μέγεθος και χρησιμοποιούνται για να μπορούν να τρέξουν οι εκτελέσιμες μορφές κάποιων άλλων προγραμμάτων.

jQuery: Η jQuery είναι μια βιβλιοθήκη εντολών Javascript. Χρησιμοποιώντας την μπορούμε να κάνουμε πάρα πολλά πράγματα όπως: επιλογή HTML στοιχείων, διαμόρφωση HTML στοιχείων, διαμόρφωση CSS στοιχείων, διεργασίες HTML γεγονότων, εφέ JavaScript και animations, διαμόρφωση του HTML DOM(Document Object Module), χρήση AJAX, πληθώρα άλλων εφαρμογών.

API: Application Programming Interface αναφέρεται στο σύνολο των λειτουργιών/συναρτήσεων/κλάσεων που προσφέρει συνήθως μια βιβλιοθήκη προγραμματισμού.

GET: Η μέθοδος GET αφορά στην ανάκτηση της οποιασδήποτε πληροφορίας (αντικειμένου) καθορίζεται από το URI της αίτησης (Request URI). Εάν το URI της αίτησης υποδεικνύει μία διαδικασία επεξεργασίας δεδομένων θα πρέπει να επιστραφούν, ως απάντηση, τα δεδομένα όπως αυτά προέκυψαν από την σχετική διαδικασία. URI (Universal Resource Identifier) είναι σειρές χαρακτήρων που προσδιορίζουν τοποθεσίες στο διαδίκτυο. Ένα URL είναι δημοφιλής τύπος του URI.

TLS: (Transport Layer Security) Το TLS (ασφαλές επίπεδο μεταφοράς) είναι ένα πρωτόκολλο που χρησιμοποιείται για να κρυπτογραφήσει την επικοινωνία μεταξύ δύο υποδοχών δισκτύου (network sockets). Διαδέχτηκε το SSL (secure socket layer).

SSL: (Secure Socket Layer) Το SSL (ασφαλές επίπεδο επικοινωνίας) είναι ένα πρωτόκολλο που χρησιμοποιείται για να κρυπτογραφήσει την επικοινωνία μεταξύ δύο υποδοχών δικτύου (network sockets). Το συναντάμε πολλές φορές όταν χρησιμοποιείται για να ασφαλίσει την επικοινωνία ενός φυλλομετητή ιστού και ενός διακομιστή οπότε το πρωτόκολλο στην γραμμή διεύθυνσης του φυλλομετητή γράφει https. Χρησιμοποιεί πιστοποιητικά ασφαλείας για να πιστοποιήσει την αυθεντικότητα του διακομιστή (εάν είναι αυτός που υποστηρίζει πως είναι). Τα πιστοποιητικά διαμοιράζονται από υπηρεσίες πιστοποίησης οι οποίες είναι γνωστές στους φυλλομετρητές. Το SSL διαδέχτηκε το TLS.