Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Μελέτη συστήματος ηλεκτρονικής ψηφοφορίας μέσω διαδικτύου και κατασκευή αντίστοιχης εφαρμογής** **Study of the electronic voting system through internet (I-Voting System) and development a corresponding application**
Ονοματεπώνυμο Φοιτητή	**Μπακτής Νικόλαος**
Πατρώνυμο	**Αριστείδης**
Αριθμός Μητρώου	**ΜΠΣΠ/ 11057**
Κατεύθυνση	**Δικτυοκεντρικά Πληροφοριακά Συστήματα**
Επιβλέπων	**Δουληγέρης Χρήστος, Καθηγητής**
Συνεργαζόμενος Ερευνητής	**Μακροδημήτρης Γεώργιος, Υποψήφιος Διδάκτωρ**

Ημερομηνία Παράδοσης **Απρίλιος 2015**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)	(υπογραφή)	(υπογραφή)
Δουληγέρης Χρήστος Καθηγητής	Κοτζανικολάου Παναγιώτης Λέκτορας	Πατσάκης Κωνσταντίνος Λέκτορας

Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε στο πλαίσιο του Προγράμματος Μεταπτυχιακών Σπουδών 'Προηγμένα Συστήματα Πληροφορικής' του τμήματος Πληροφορικής του Πανεπιστημίου Πειραιώς.

Πρωτίστως θα ήθελα να ευχαριστήσω θερμά το επιβλέποντα καθηγητή της εργασίας μου, Καθηγητή κύριο Χρήστο Δουληγέρη, για την εμπιστοσύνη που μου έδειξε αναθέτοντας μου την παρούσα μεταπτυχιακή διατριβή, για την επίβλεψη και την καθοδήγηση καθ' όλη την διάρκεια της εκπόνησης της, για τις σημαντικές επισημάνσεις του και για την απεριόριστη υπομονή που επέδειξε.

Ομοίως θα ήθελα να ευχαριστήσω ιδιαίτερα τον υποψήφιο Διδάκτορα κύριο Γεώργιο Μακροδημήτρη για την συνεργασία, την συνεχή βοήθεια και την αμεσότητα στην επικοινωνία μας. Οι εύστοχες επισημάνσεις του, η συνεχής στήριξή του σε όλα τα επίπεδα και η αμεσότητα του συνέβαλαν καθοριστικά στην επίλυση των διάφορων προβλημάτων που ανέκυψαν κατά διάρκεια της εκπόνησης αλλά και στην ολοκλήρωση της παρούσας εργασίας.

Στο σημείο αυτό δεν θα μπορούσα να μην ευχαριστήσω το διδακτικό προσωπικό του Προγράμματος Μεταπτυχιακών Σπουδών 'Προηγμένα Συστήματα Πληροφορικής' για την γνώση που μου μετέδωσε κατά την παρακολούθηση των μαθημάτων και των διαλέξεων.

Τέλος οφείλω τις θερμότερες ευχαριστίες σε όλους τους δικούς μου ανθρώπους που όλα αυτά τα χρόνια των σπουδών μου βρίσκονται δίπλα μου, με στηρίζουν και δείχνουν πραγματικό ενδιαφέρον για την προσπάθεια μου.

Περίληψη

Σκοπός του I-VOTING SYSTEM είναι η ανάπτυξη μιας πλατφόρμας για τη διεξαγωγή δημοψηφίσματος μέσω διαδικτύου σε μια χώρα επιτυγχάνοντας την ασφαλή συμμετοχή των πολιτών της.

Το σύστημα υποστηρίζει την εγγραφή νέων ψηφοφόρων και τους παρέχει την δυνατότητα συμμετοχής στο δημοψήφισμα μετά την επιβεβαίωση των στοιχείων τους. Επίσης παρέχει την δυνατότητα διαχείρισης των λειτουργιών του από διαπιστευμένους χρήστες.

Για την ανάπτυξη της εφαρμογής γίνεται χρήση προγραμμάτων και εργαλείων ανοιχτού κώδικα (Apache Tomcat, MySQL, Java). Αυτό έχει ως αποτέλεσμα την μείωση του κόστους ανάπτυξης χωρίς όμως να γίνονται εκπτώσεις στο ζήτημα της αξιοπιστίας και της ασφάλειας. Επίσης, η αρχιτεκτονική MVC που χρησιμοποιείται ενισχύει την εφαρμογή σε ζητήματα ασφάλειας καθώς προσδιορίζει πολύ συγκεκριμένη ροή δεδομένων και επομένως μπορεί να ελέγξει το επίπεδο πρόσβασης των χρηστών. Τέλος όλες οι λειτουργίες που εκτελούν οι χρήστες εκτελούνται με ασφαλείς κρυπτογραφημένες συνεδρίες ανάμεσα στους Η/Υ τους και τον εξυπηρετητή (server) του συστήματος.

Abstract

The purpose of I-VOTING SYSTEM is to develop a platform that allows the organization of a referendum through the Internet in a country providing a safe participation to its citizens.
The system supports the registration of new voters and gives them the opportunity to participate in the referendum after the confirmation of their data. It also provides an administrative area where users with the appropriate authority can manage the system's functions.

We used open source tools and software for application development like Apache Tomcat, MySQL and Java. This results in the reduction of the development costs but without making deductions to the issues of reliability and safety. Also, the MVC architecture which is used enhances the application in terms of safety as well it identifies very specific data streams and, therefore, it can control the level of user access. Finally, all the functions that the users perform are performed by secure encrypted sessions between the user's web-client and the application's server.

Εισαγωγή

Σκοπός της συγκεκριμένης μεταπτυχιακής διατριβής είναι η ανάπτυξη ενός διαδικτυακού συστήματος ψηφοφορίας. Στην παρούσα εργασία θα μελετηθεί το διαδικτυακό σύστημα ηλεκτρονικής ψηφοφορίας σε θέματα ασφαλείας, εμπιστευτικότητας, ιδιωτικότητας και εφαρμογής της κείμενης νομοθεσίας των εκλογικών διαδικασιών. Θα δημιουργηθεί μια αντίστοιχη κατανεμημένη υπηρεσιοστρεφής εφαρμογή μέσω της οποίας θα μπορούν να διεξάγονται διαδικτυακές ψηφοφορίες. Χρησιμοποιείται ο μηχανισμός JSP/Servlets και η επικοινωνία των clients με τον server γίνεται με ασφαλή τρόπο. Χρησιμοποιείται κρυπτογράφηση, αυθεντικοποίηση και εξουσιοδότηση των χρηστών της εφαρμογής και έλεγχος των δεδομένων που εισέρχονται στο σύστημα. Τέλος, ο server επικοινωνεί με μια βάση δεδομένων μέσω JDBC για πρόσβαση στα δεδομένα και για την καταχώριση των ψήφων.

Στο πρώτο κεφάλαιο της εργασίας γίνεται ιστορική αναδρομή στα συστήματα ψηφοφορίας που έχουν χρησιμοποιηθεί σε διάφορες χώρες και στις επιτυχημένες ή μη προσπάθειες χρησιμοποίησης ηλεκτρονικών συστημάτων ψηφοφορίας. Αναλύονται οι κατηγορίες των συστημάτων αυτών ανάλογα με τα μέσα τα οποία χρησιμοποιούν για την διεξαγωγή ψηφοφορίας. Τέλος, επισημαίνονται τα οφέλη που προκύπτουν από ένα σύστημα διαδικτυακής ψηφοφορίας και μελετώνται τα κριτήρια που πρέπει να πληροί ένα ασφαλές και αξιόπιστο σύστημα ψηφοφορίας καθώς και το νομικό πλαίσιο το οποίο πρέπει να το διέπει.

Στο δεύτερο κεφάλαιο γίνεται η ανάλυση του συστήματος. Αρχικά περιγράφονται οι γενικές αρχές ασφαλείας ενός πληροφοριακού συστήματος και γίνεται αντιστοίχιση αυτών με τις αρχές και τα κριτήρια που πρέπει να τηρεί ένα σύστημα ψηφοφορίας. Με βάση αυτές γίνεται η ανάλυση απαιτήσεων και ασφαλείας του συστήματος που θα υλοποιήσουμε. Για την επίτευξη της ασφάλειας μελετώνται οι μηχανισμοί και τα πρωτόκολλα ασφαλείας που μπορεί να χρησιμοποιηθούν σε ένα διαδικτυακό σύστημα και επισημαίνονται οι πιο γνωστές επιθέσεις που μπορεί να δεχθούν. Πέρα από το κομμάτι της ασφάλειας επισημαίνονται οι τεχνολογίες υλοποίησης διαδικτυακών εφαρμογών και η αρχιτεκτονική MVC που θα χρησιμοποιηθεί. Τέλος, αναλύονται οι ρόλοι του συστήματος και οι διαδικασίες που μπορούν να εκτελέσουν.

Στο τρίτο κεφάλαιο γίνεται ο σχεδιασμός της εφαρμογής διαδικτυακής ψηφοφορίας. Παρουσιάζεται το σχήμα της Β.Δ. και η χρησιμότητα του κάθε πίνακα. Σε αντιστοιχία με τη Β.Δ. περιγράφεται το model κομμάτι της εφαρμογής το οποίο και είναι υπεύθυνο για την δημιουργία των αντικειμένων που θα επεξεργάζονται τα servlets.Στη συνέχεια του κεφαλαίου σχεδιάζουμε τους μηχανισμούς ασφαλείας της εφαρμογής και περιγράφουμε τον τρόπο λειτουργίας τους. Με βάση την ανάλυση των ρόλων και των διαδικασιών που έγινε στο προηγούμενο κεφάλαιο σχεδιάζουμε την ροή δεδομένων και διαδικασιών που

μπορεί να εκτελέσει ο κάθε χρήστης κατά την αλληλεπίδρασή του με το σύστημα. Κλείνοντας, αναφέρουμε τα εργαλεία που θα χρησιμοποιήσουμε κατά την υλοποίηση της εφαρμογής.

Στο τέταρτο κεφάλαιο αρχικά παρουσιάζονται κάποια μέρη της υλοποίησης της εφαρμογής. Περιγράφεται ο τρόπος εγκατάστασης του συστήματος και κάποια τμήματα κώδικα που σχετίζονται με την ασφάλεια του συστήματος. Στο τέλος του κεφαλαίου γίνεται συνοπτική παρουσίαση της εφαρμογής.

Στο πέμπτο κεφάλαιο παραθέτουμε συμπεράσματα από την μελέτη και την υλοποίηση του συστήματος και γίνεται σύγκριση του συστήματος διαδικτυακής ψηφοφορίας με το σύστημα ψηφοφορίας που χρησιμοποιείται σήμερα στην χώρα μας. Αναλύονται τα πλεονεκτήματα και τα μειονεκτήματα και παρατίθενται μελλοντικές βελτιώσεις που θα ήταν χρήσιμο να γίνουν στο σύστημα που υλοποιήθηκε.

Πίνακας περιεχομένων

1 Ηλεκτρονική Ψηφοφορία

1.1 Η Χρησιμότητα ενός πληροφοριακού συστήματος

Η έννοια της χρησιμότητας στην οικονομία παρέχει ένα τρόπο σύγκρισης της ικανοποίησης που λαμβάνει ένα άτομο από ένα αγαθό ή μια υπηρεσία.

Η χρησιμότητα μιας υπηρεσίας πηγάζει από τη δυνατότητα που έχει να ικανοποιήσει μια ανθρώπινη ανάγκη, δηλαδή να προσφέρει ικανοποίηση στο χρήστη της.

Η χρησιμότητα οφείλεται σε κάποια πραγματικά χαρακτηριστικά της υπηρεσίας (π.χ. παροχή αποτελεσμάτων ιατρικών εξετάσεων) αλλά σε μεγάλο βαθμό επηρεάζεται από υποκειμενικούς παράγοντες, όπως είναι οι προτιμήσεις του εκάστοτε καταναλωτή, τα στοιχεία της προσωπικότητάς του, η καθημερινότητά του και η δυνατότητά του να χρησιμοποιήσει την εκάστοτε υπηρεσία ή αγαθό.

1.2 Χρησιμότητα I-Voting System

Αν προσπαθήσουμε να μεταφέρουμε τον όρο της χρησιμότητας στην πληροφορική και ειδικότερα σε ένα πληροφοριακό σύστημα θα μπορούσαμε να πούμε πως είναι η δυνατότητα το συγκεκριμένο σύστημα ως υπηρεσία να ικανοποιήσει κάποιες ανάγκες των πολιτών

Στόχος μας λοιπόν είναι να καταφέρουμε το πληροφοριακό σύστημα διαδικτυακής ψηφοφορίας να ικανοποιεί τους πολίτες περισσότερο από ότι οι έως τώρα υπάρχουσες υπηρεσίες .Να είναι δηλαδή μια χρήσιμη υπηρεσία.

Ένα σύστημα διαδικτυακής ψηφοφορίας θα πρέπει να παρέχει στους πολίτες την δυνατότητα να μπορούν να ψηφίσουν σε ένα δημοψήφισμα ή στις καθιερωμένες εκλογικές διαδικασίες μέσω των ηλεκτρονικών υπολογιστών τους αρκεί να έχουν σύνδεση στο διαδίκτυο. Με αυτόν τον τρόπο θα πετύχουμε την χρησιμότητα του, δηλαδή την μεγαλύτερη ικανοποίηση των πολιτών, σε σχέση με άλλα συστήματα ψηφοφορίας που χρησιμοποιούνται ως τώρα.

Αρχικά καλό θα ήταν να γίνει μια παρουσίαση των υπόλοιπων εκλογικών συστημάτων και να μελετήσουμε τα πλεονεκτήματα και τα μειονεκτήματα τους.

1.3 Τυπική ψηφοφορία

Η πλειοψηφία των κρατών διεξάγει εκλογές όπως και η Ελλάδα με την χρήση ψηφοδελτίων και κάλπης σε εκλογικά τμήματα σε ολόκληρη την χώρα. Οι εκλογείς πρέπει να παραστούν στα εκλογικά τμήματα και αφού παραλάβουν το ψηφοδέλτιο να κάνουν την επιλογή τους και να ψηφίσουν στην κάλπη. Έπειτα από το πέρας της διορίας ανοίγουν οι κάλπες και πραγματοποιείται καταμέτρηση των ψήφων για να εξαχθούν τα αποτελέσματα των

εκλογών. Με αυτόν τον τρόπο εξασφαλίζεται η πλήρης μυστικότητα της απόφασης του κάθε εκλογέα. Εκτός όμως από αυτό τον πολύ σημαντικό παράγοντα υπάρχουν πολλά αρνητικά σε αυτή την διαδικασία.

- Χρειάζεται μεγάλη προετοιμασία στο να διοργανωθούν οι εκλογές.
- Χρειάζονται πολλά έξοδα για την διαμόρφωση των εκλογικών τμημάτων και την παραγωγή των ψηφοδελτίων.
- Χρειάζεται μεγάλος αριθμός ανθρώπων ώστε να επιβλέπουν την διεξαγωγή της εκλογικής διαδικασίας και να εξάγουν τα αποτελέσματα.
- Χρειάζεται οι πολίτες να μεταβούν στο εκλογικό κέντρο και να σπαταλήσουν κάποιο χρόνο ανάλογα και με την ουρά αναμονής ώστε να ψηφίσουν.
- Τέλος χρειάζεται αρκετός χρόνος μέχρι να συγκεντρωθούν και να εξαχθούν τα τελικά αποτελέσματα.

Για τους παραπάνω λόγους σε διάφορες χώρες της Ευρώπης, της Αμερικής της Αυστραλίας και της Ασίας έχουν γίνει προσπάθειες να χρησιμοποιηθούν εναλλακτικά συστήματα ψηφοφορίας βασισμένα σε μηχανικά και ηλεκτρονικά μέσα με την βοήθεια των οποίων οι πολίτες μπορούν να ψηφίσουν.

1.4 Ηλεκτρονικοί τρόποι ψηφοφορίας .

Οι πρώτες προσπάθειες εναλλακτικής ψηφοφορίας έχουν ήδη γίνει από τις αρχές τις δεκαετίας του 1960 με την χρησιμοποίηση των διάτρητων καρτών. Οι κάρτες αυτές χρησιμοποιήθηκαν σε δύο μορφές,τις Vomatic και Datavote. Μετά όμως από πρόβλημα που παρατηρήθηκε στις προεδρικές εκλογές της Florida το 2000 σταμάτησε η χρησιμοποίηση τους. Παράλληλα με τις διάτρητες κάρτες χρησιμοποιήθηκαν και τα οπτικά ψηφοδέλτια τα οποία μπορούσαν με την ολοκλήρωση της ψηφοφορίας να σαρωθούν. Τέλος, πριν από τις παραπάνω προσπάθειες ένας εναλλακτικός τρόπος ψηφοφορίας χρησιμοποιούσε μια μηχανολογική εγκατάσταση με μοχλό (mechanical lever machine). Και αυτός όμως παρουσίαζε ανακρίβειες και δεν ήταν ικανός να κρατήσει ιστορικότητα οπότε και εγκαταλείφθηκε.[1]

Με την ανάπτυξη της τεχνολογίας όμως από τα τέλη της δεκαετίας του 60 είχε έρθει ο καιρός να βρεθούν νέοι τρόποι ψηφοφορίας βασιζόμενοι σε ηλεκτρονικά μέσα.

Λόγω του ότι η έννοια ηλεκτρονική ψηφοφορία περικλείει πολλές διαφορετικές μορφές ψηφοφορίας υποβοηθούμενης από ηλεκτρονικά μέσα καλό θα ήταν, σε αυτό το σημείο, να κάνουμε μια κατηγοριοποίηση των μορφών ηλεκτρονικής ψηφοφορίας.

Με τον όρο ηλεκτρονική ψηφοφορία e-voting (electronic voting) εννοούμε τη δυνατότητα που παρέχεται στους πολίτες να ασκήσουν το εκλογικό τους δικαίωμα με τη χρήση ηλεκτρονικών μέσων. Αυτή η δυνατότητα μπορεί να παρέχεται είτε μέσω μηχανημάτων που λειτουργούν σε διάφορα εκλογικά κέντρα, είτε με την χρήση κινητών τηλεφώνων χρησιμοποιώντας το δίκτυο κινητής τηλεφωνίας, είτε με την χρήση προσωπικών ηλεκτρονικών συσκευών (pc,laptop,tablet) και την βοήθεια του διαδικτύου. Αντίστοιχα

λοιπόν με τα παραπάνω έχουν δημιουργηθεί όλα αυτά τα χρόνια συστήματα e-voting, m-voting και i-voting.

1.4.1 Σύστημα ηλεκτρονικής ψηφοφορίας

Ένα σύστημα e-voting χρησιμοποιεί ηλεκτρονικές συσκευές ψηφοφορίας οι οποίες αντικαθιστούν το παραβάν, τις κάλπες και τα ψηφοδέλτια και αυτοματοποιούν την μέθοδο της οργάνωσης και διεξαγωγής της διαδικασίας. Όπως είναι λογικό αυτές οι συσκευές έχουν δημιουργηθεί για αυτόν το σκοπό οπότε είναι εγκατεστημένες σε διάφορα σημεία της κάθε χώρας και οι ψηφοφόροι πρέπει να έχουν αυτοπρόσωπη παρουσία στα σημεία αυτά. Σε πολλές περιπτώσεις οι συσκευές δεν χρησιμοποιούν καθόλου δίκτυο καθώς τα αποτελέσματα εξάγονται από την κάθε συσκευή ξεχωριστά ενώ αν είναι συνδεδεμένες σε κάποιο δίκτυο ώστε να συγκεντρώνονται τα αποτελέσματα, αυτό είναι ιδιωτικό.

Οι πρώτες τέτοιες μηχανές ήταν οι DRE μηχανές ψηφοφορίας (Direct Recording Electronic) οι οποίες χρησιμοποιήθηκαν σε πραγματικές εκλογές το 1975 στο Streamwood και Woodstock του Illinois. [1]

Μια μηχανή ψηφοφορίας DRE επιτρέπει στους ψηφοφόρους να επιλέξουν την ψήφο τους μέσω μιας οθόνης και την καταγράφει. Μετά την εκλογή παράγει τα αποτελέσματα τα οποία αποθηκεύονται σε εξωτερική μνήμη και εκτυπώνονται. Η συγκέντρωση και η εξαγωγή των συνολικών αποτελεσμάτων αρχικά γινόταν χειροκίνητα. Μετέπειτα οι μηχανές αυτές εξελίχθηκαν και ονομάστηκαν public network DRE systems. Μπορούσαν πλέον να συνδεθούν σε ένα κοινό δίκτυο και να εξαχθούν αυτόματα τα τελικά αποτελέσματα.

Η Βραζιλία το 2000 ήταν η πρώτη χώρα που χρησιμοποίησε σύστημα ψηφοφορίας πλήρως βασισμένο σε ηλεκτρονικά μηχανήματα για όλη την επικράτεια και το 2010 τα αποτελέσματα κατάφεραν να εξαχθούν σε 75 λεπτά. Η πολιτεία Georgia των ΗΠΑ ήταν επίσης η πρώτη πολιτεία στις ΗΠΑ που υιοθέτησε εξολοκλήρου μια ενιαία τεχνολογία DRE για ψηφοφορία το 2002. Μάλιστα σε σύγκριση με τις εκλογές της Βραζιλίας του 2010 οπότε και χρησιμοποιήθηκαν ενιαία για πρώτη φορά τα ηλεκτρονικά μηχανήματα το ποσοστό των ψηφοδελτίων όπου δεν είχαν συμπληρωθεί όλα τα απαιτούμενα στοιχεία μειώθηκε από 4,4% σε λιγότερο από 1%, κάτι που αποδεικνύει την ακεραιότητα ενός εκλογικού συστήματος [2] [3] [4] [5].

Μετά από αυτές τις προσπάθειες είναι αξιοσημείωτο ένα πιλοτικό πρόγραμμα στο Ηνωμένο Βασίλειο τα έτη 2002-2006 όπου έγινε προσπάθεια να αντικατασταθεί η παραδοσιακή μέθοδος ψηφοφορίας από ηλεκτρονικές τεχνολογίες. Ήταν η αρχή για να χρησιμοποιηθούν και οι τεχνολογίες του telephone-mobile voting, του SMS text message voting και του Internet voting, παράλληλα με τις DRE.Ο συντονισμός είχε γίνει από κοινού από την κεντρική διοίκηση της χώρας και τις τοπικές αρχές. Οι διάφορες ανεπάρκειες του συστήματος παρουσιάστηκαν σε θέματα οργάνωσης, μικρής ευελιξίας, δυνατότητας ελέγχου αλλά και εξάρτησης από τους εμπορικούς προμηθευτές του εξοπλισμού. Υπήρχε διαθέσιμη και η δυνατότητα των χάρτινων ψηφοδελτίων σαν backup διαδικασία και αποδείχτηκε πως χωρίς αυτό θα είχε προκληθεί πρόβλημα στις διεξαγωγή των εκλογών.[6]

[7]. Εκτός από την Μ. Βρετανία μηχανήματα DRE χρησιμοποιούσαν και άλλες χώρες όπως η Νορβηγία, η Ιρλανδία, η Ολλανδία, η Ινδία, η Ελβετία, ο Καναδάς και η Νέα Ζηλανδία[10].

Τα επόμενα χρόνια και κυρίως το 2009-2010, όπως άρχισε να φαίνεται και από το πιλοτικό πρόγραμμα στην Μ. Βρετανία έγιναν προσπάθειες από διάφορες χώρες σε όλο τον πλανήτη να χρησιμοποιηθούν καινούρια συστήματα ηλεκτρονικής ψηφοφορίας καθώς τα συστήματα DRE κρίθηκαν ανεπαρκή και αρκετά πολυέξοδα για την διεξαγωγή εκλογών από τις εκάστοτε κυβερνήσεις των παραπάνω χωρών. Το μεγάλο στοίχημα πλέον ήταν να επιτευχθεί αυτό που δεν μπορούσαν να πετύχουν τα μέχρι τότε ηλεκτρονικά μηχανήματα. Οι πολίτες να μπορούν να ψηφίζουν από απόσταση χωρίς να χρειάζεται να μετακινούνται στα εκλογικά κέντρα και έτσι να μειωθεί σηματικά το κόστος διεξαγωγής εκλογών.

Μετά την εξέλιξη της τεχνολογίας των κινητών τηλεφώνων αλλά και την ραγδαία ανάπτυξη του διαδικτύου ξεκίνησαν οι προσπάθειες ώστε να δημιουργηθούν ηλεκτρονικά συστήματα ψηφοφορίας κατά την χρησιμοποίηση των οποίων οι ψηφοφόροι δεν θα ήταν απαραίτητο να έχουν αυτοπρόσωπη παρουσία στο εκλογικό κέντρο αλλά θα μπορούν να ψηφίσουν απομακρυσμένα.

1.4.2 Συστήματα κινητής ψηφοφορίας

Μια νέα μορφή ηλεκτρονικής ψηφοφορίας που εμφανίστηκε μετά την ραγδαία ανάπτυξη της κινητής τηλεφωνίας και την γρήγορη εξοικείωση των πολιτών στην χρήση τους είναι το ηλεκτρονικό σύστημα ψηφοφορίας μέσω κινητής συσκευής (mobile voting) .Κάθε ιδιοκτήτης κινητού τηλεφώνου μπορεί να εγγραφεί στην κατάλληλη υπηρεσία που έχει δημιουργηθεί από τους κρατικούς φορείς και να κάνει την επιλογή της ψήφου του μέσω του δικτύου κινητής τηλεφωνίας. Το m-voting παρέχει το πλεονέκτημα της απομακρυσμένης ψηφοφορίας το οποίο μπορεί να προσφέρει αύξηση στην συμμετοχή των πολιτών στις ψηφοφορίες. Επίσης, τα αποτελέσματα μπορεί να εξαχθούν ταχύτερα καθώς συγκεντρώνονται σε έναν διακομιστή. Τέλος είναι σίγουρα φθηνότερα από τις παλαιότερες μορφές ψηφοφορίας καθώς δεν χρειάζονται εκλογικά κέντρα και εκλογικό υλικό [8].

Έχουν γίνει διάφορες απόπειρες χρησιμοποίησης του m-voting σε διάφορες χώρες όπως στην Γαλλία το 2005 για τους πολίτες της που διαμένουν εκτός της χώρας με συμμετοχή πάνω από 600.000 ψηφοφόρων, στην πόλη Ssy-les-Moulineaux της Γαλλίας και πάλι, κατά τις περιφερειακές εκλογές στην πόλη της Βρέμης στην Γερμανία και στην πόλη Kista της Σουηδίας [9].

Ενώ το m-voting παρέχει αρκετά πλεονεκτήματα κατά την διάρκεια την ψηφοφορίας αλλά και κατά την εξαγωγή αποτελεσμάτων σε σχέση με παλιότερα συστήματα ψηφοφορίας παρουσιάζει και κάποια σημαντικά μειονεκτήματα. Μια συσκευή κινητής τηλεφωνίας δεν μπορεί να έχει τις αποδόσεις αλλά και να εκτελέσει τις λειτουργίες που μπορεί να εκτελέσει ένας Η/Υ. Επίσης το ασύρματο δίκτυο κινητής τηλεφωνίας και οι πάροχοί του εξαρχής δεν έχουν εστιάσει στην ασφάλεια των δεδομένων που αποστέλλονται και δεν μπορούν να κερδίσουν εύκολα την εμπιστοσύνη των πολιτών ως προς την τήρηση της ιδιωτικότητάς τους κατά την ψηφοφορία. Τέλος, λόγω του ότι τα περισσότερα κινητά

τηλέφωνα πλέον διαθέτουν πρόσβαση στο διαδίκτυο μπορούν να χρησιμοποιηθούν ως τερματικά (Η/Υ) στην ηλεκτρονική ψηφοφορία i-voting η οποία όπως θα δούμε παρακάτω χρησιμοποιεί το διαδίκτυο ως μέσω αποστολής των ψήφων.

1.4.3 Συστήματα ψηφοφορίας μέσω διαδικτύου

Από το 2000 και μετά συγχρόνως με τις παραπάνω προσπάθειες ξεκίνησαν σε πολλές χώρες του κόσμου ενέργειες ώστε οι ψηφοφόροι να έχουν την δυνατότητα να ψηφίσουν από απόσταση μέσω του διαδικτύου χρησιμοποιώντας μια προσωπική τους υπολογιστική μηχανή, έναν Η/Υ ένα Laptop ή πιο πρόσφατα ένα tablet ή το κινητό τους τηλέφωνο. Αυτή η μορφή ηλεκτρονικής ψηφοφορίας έχει ονομαστεί i-voting (internet voting). Η συγκεκριμένη μεταπτυχιακή διατριβή θα εστιάσει στην δημιουργία μιας ασφαλούς τέτοιας εφαρμογής στην οποία οι ψηφοφόροι θα μπορούν να κάνουν την επιλογή τους αφού πρώτα έχουν πιστοποιηθεί και οι κρατικοί φορείς θα μπορούν να εξάγουν τα αποτελέσματα της ψηφοφορίας αυτόματα.

Έχουν γίνει αρκετές προσπάθειες σε διάφορες χώρες να χρησιμοποιηθούν τέτοια συστήματα. Πολλές ήταν αποτυχημένες αλλά υπήρξαν και απόπειρες που τελικά κρίθηκαν επιτυχημένες.

Ένα τέτοιο πείραμα είναι το πιλοτικό Internet-based σύστημα ψηφοφορίας SERVE (Secure Electronic Registration and Voting Experiment), που αναπτύχθηκε για το FVAP (Federal Voting Assistance Program) του Υπουργείου Άμυνας των ΗΠΑ. Σχεδιάστηκε με σκοπό να εφαρμοστεί στις εκλογές του 2004, ώστε να επιτρέψει στους εν δυνάμει ψηφοφόρους αρχικά να εγγραφούν στους εκλογικούς καταλόγους της περιοχής τους και στη συνέχεια να ψηφίσουν ηλεκτρονικά μέσω Internet απ' οπουδήποτε στον κόσμο. Στο πείραμα θα μπορούσαν να συμμετάσχουν εγγεγραμμένοι ψηφοφόροι που βρίσκονταν εκτός συνόρων, οι στρατιωτικοί και οι ψηφοφόροι 50 επαρχιών σε 7 πολιτείες. Η εφαρμογή αυτού του προγράμματος θα χρησίμευε σαν πρότυπο για τη γενικευμένη χρήση ηλεκτρονικής ψηφοφορίας στο μέλλον. Οι ερευνητές, βέβαια, προειδοποίησαν για ενδεχόμενα κενά του συστήματος και τελικά δεν εφαρμόστηκε με την αρχική του μορφή στις εκλογές του 2004. Αλλά και η εφαρμογή του συστήματος ηλεκτρονικής ψηφοφορίας του 2000 δεν έφερε καλύτερα αποτελέσματα. Αξιοσημείωτο είναι το γεγονός ότι στη Florida εμφανίστηκαν ακόμα μεγαλύτερα προβλήματα από τις εκλογές του 2000. Για παράδειγμα σε κάποιες περιπτώσεις το σύστημα κατέρρευσε και είτε χάθηκαν χιλιάδες ψήφων, είτε χρειάστηκε προσφυγή στα κλασικά ψηφοδέλτια. Εκτός από την Αμερική όπως αναφέραμε παραπάνω ένα πιλοτικό πρόγραμμα στο οποίο χρησιμοποιήθηκε και το i-voting παράλληλα με άλλες μορφές ψηφοφορίας διεξήχθη στην Μεγάλη Βρετανία το 2006 όπου και εκεί υπήρξαν σημαντικά προβλήματα [11] [12] [13].

Στην Ελβετία και συγκεκριμένα στην πόλη της Γενεύης ξεκίνησαν οι προσπάθειες από το 2004 και τελικά αφού ξεπεράστηκαν αρκετά διαδικαστικά και νομικά προβλήματα το 2012 διεξήχθησαν εκλογές και μέσω διαδικτύου το αποτέλεσμα των οποίων κρίθηκε επιτυχημένο. Κατά την ίδια χρονική περίοδο δοκιμές γίνονταν και στην Νορβηγία με αποτέλεσμα να διεξαχθούν δοκιμαστικές τοπικές εκλογές μέσω διαδικτυακής ψηφοφορίας το 2011 και εθνικές τον Σεπτέμβρη του 2013 όπου εμφανίστηκε αύξηση στην συμμετοχή των πολιτών σε σχέση με τις εκλογές του 2011 από 64,5% σε 78,2% [14].Στην Σκωτία στην

πόλη Truno αλλά και στην άλλη πλευρά του Ατλαντικού στον Καναδά επίσης τις χρονιές 2011 και 2012 διεξήχθησαν τοπικές εκλογές σε πόλεις με μικρό πληθυσμό όπως στις Cape Breton, Halifax Regional Municipality και City of Markham. Τα αποτελέσματα και εκεί ήταν ενθαρρυντικά καθώς δεν προέκυψαν μεγάλα προβλήματα αλλά δεν μπορούμε να παραβλέψουμε πως ήταν μικρός ο αριθμός των ψηφοφόρων που συμμετείχαν. [15]

Χώρα	Ταξινόμηση	Είδος εκλογών i-voting
Αυστραλία	Σε μερικές πόλεις	Τοπικές εκλογές - New South Wales
Καναδάς	Σε μερικές πόλεις	Τοπικές εκλογές
Εσθονία	Σε όλη την επικράτεια	Τοπικές, Βουλευτικές, Προεδρικές, Ευρωεκλογές
Γαλλία	Σε μερικές πόλεις	Εκλογές της συνέλευσης των κατοίκων εξωτερικού
Ινδία	Γίνονται πιλοτικές δοκιμές	Τοπικές εκλογές
Ολλανδία	Απορρίφτηκε προς το παρών	Βουλευτικές Εκλογές - ετεροδημότες
Νορβηγία	Γίνονται πιλοτικές δοκιμές	Τοπικά στις βουλευτικές εκλογές
Ισπανία	Απορρίφτηκε προς το παρών	Τοπικές εκλογές της Βαρκελώνης
Ελβετία	Σε μερικές πόλεις	Τοπικές εκλογές
Μ. Βρετανία	Έγιναν πιλοτικές δοκιμές και απορρίφτηκε προς το παρών.	Τοπικές εκλογές
Η.Π.Α.	Γίνονται πιλοτικές δοκιμές	Γενικές εκλογές – Κάτοικοι εξωτερικού, στρατιωτικοί

Πίνακας 1. Η χρησιμοποίηση του i-voting παγκοσμίως [17]

Η χώρα στην οποία η ηλεκτρονική ψηφοφορία μέσω διαδικτύου (i-voting) παρουσίασε την μεγαλύτερη επιτυχία και δίνει ενθαρρυντικά αποτελέσματα για το μέλλον της ηλεκτρονικής ψηφοφορίας σε όλο τον κόσμο είναι η Εσθονία. Το 2005 έγινε η πρώτη δοκιμαστική λειτουργία στην οποία είχαν δικαίωμα συμμετοχής οι ψηφοφόροι ολόκληρης της χώρας. Ήταν μάλιστα η πρώτη φορά παγκοσμίως που συνέβη αυτό καθώς όλες οι υπόλοιπες δοκιμές γίνονταν σε μικρό πληθυσμό. Ο αριθμός συμμετοχής της πρώτης εκείνης εκλογικής διαδικασίας μέσω διαδικτύου ήταν αρκετά μικρός οπότε δεν μπορούσαν να εξαχθούν ασφαλή συμπεράσματα για την λειτουργία της ψηφοφορίας. Η διαδικασία όμως κρίθηκε γενικότερα επιτυχής και έτσι στις επόμενες βουλευτικές εκλογές το 2007 χρησιμοποιήθηκε ξανά [18]. Σε εκείνες τις εκλογές το ποσοστό συμμετοχής αυξήθηκε αρκετά και η ψηφοφορία ήταν επιτυχημένη. Από τότε έως και σήμερα η Εσθονία

χρησιμοποιεί την διαδικτυακή ψηφοφορία σε κάθε εκλογική διαδικασία και μάλιστα ο αριθμός των ψηφοφόρων αυξάνεται συνεχώς [16] .

	Τοπικές Εκλογές 2005	Βουλευτικές Εκλογές 2007	Ευρωεκλογές 2009	Τοπικές Εκλογές 2009	Βουλευτικές Εκλογές 2011	Τοπικές Εκλογές 2013	Ευρωεκλογές 2014	Βουλευτικές Εκλογές 2015
Εκλογικό σώμα	1 059 292	897 243	909 628	1 094 317	913 346	1 086 935	902 873	899 793
Συνολική συμμετοχή	502 504	555 463	399 181	662 813	580 264	630 050	329 766	577 910
Ποσοστό συμμετοχής	47,4%	61,9%	43,9%	60,6%	63,5%	58,0%	36,5%	64,2%
I-voters	9 317	30 275	58 669	104 413	140 846	133 808	103 151	176 491
I-votes	9 287	30 243	58 614	104 313	140 764	133 662	103 105	176 329
Ακυρ/νες I-votes (ψήφος με ψηφοδέλτια)	30	32	55	100	82	146	46	16

Πίνακας 2. Στατιστικά i-voting στην Εσθονία [18]

Το παράδειγμα της Εσθονίας μας δείχνει πως οι ψηφοφόροι είναι έτοιμοι να δεχτούν ένα διαδικτυακό σύστημα ηλεκτρονικής ψηφοφορίας αρκεί αυτό να τους παρέχει την ασφάλεια και την ιδιωτικότητα η οποία είναι απαραίτητη στην εκλογική διαδικασία. Επίσης, παρατηρούμε ότι τα ποσοστά συμμετοχής στις εκλογές μέσω αυτού του συστήματος αυξάνονται καθώς οι ψηφοφόροι μπορούν να ψηφίσουν μέσα σε λίγα λεπτά από οποιοδήποτε σημείο βρίσκονται.

Ίσως λοιπόν θα ήταν καλύτερο αυτά τα συστήματα για χώρες όπως η Ελλάδα των οποίων οι πολίτες δεν είναι εξοικειωμένοι με αυτού του είδους το εκλογικό σύστημα να χρησιμοποιήσουν αρχικά την ηλεκτρονική ψηφοφορία σε εκλογές όπου υπάρχει μεγάλη αποχή αλλά και δεν είναι τόσο σημαντικές από πλευράς ασφάλειας όσο οι εθνικές ή ένα εθνικό δημοψήφισμα. Τέτοιες εκλογικές διαδικασίες ή δημοψηφίσματα θα μπορούσαν να είναι οι σχολικές, οι φοιτητικές, οι κλαδικές ή ακόμα και οι τοπικές δημοτικές.

1.4.4 Σύγκριση συστημάτων

Από την ανάλυση των παραπάνω συστημάτων ηλεκτρονικής ψηφοφορίας βλέπουμε πως η παγκόσμια κοινότητα έχει απορρίψει τα συστήματα e-voting λόγω του μεγάλου κόστους διεξαγωγής εκλογών μέσω αυτών αλλά και του ότι οι ψηφοφόροι χρειάζεται να παραστούν στα εκλογικά κέντρα οπότε και δεν κερδίζουν τίποτα σε σχέση με την τυπική, μη ηλεκτρονική μέθοδο ψηφοφορίας που χρησιμοποιούσαμε μέχρι τώρα Τα συστήματα m-voting από την άλλη αν και παρέχουν στους πολίτες την απομακρυσμένη ψηφοφορία δεν

μπορούν τουλάχιστον προς το παρόν να εγγυηθούν την ασφάλεια της διαδικασίας. Αυτό συμβαίνει καθώς δεν μπορούν να κρυπτογραφηθούν τα δεδομένα που αποστέλλουμε και το δίκτυο της κινητής τηλεφωνίας αν και είναι πιο κλειστό από ό,τι το διαδίκτυο δεν παύει και πάλι να είναι ένα δίκτυο στο οποίο έχουν πρόσβαση και τρίτοι. Είναι λογικό λοιπόν όλα τα κράτη τα οποία θέλουν να προχωρήσουν στη διεξαγωγή εκλογών και δημοψηφισμάτων με την βοήθεια ηλεκτρονικών μέσων να προσπαθούν να εξελίξουν τα συστήματα i-voting και την ασφάλειά τους.

Τα συστήματα i-voting πέρα από την απομακρυσμένη ψηφοφορία μας προσφέρουν το πλεονέκτημα της άμεσης καταγραφής ψήφου και γρήγορης εξαγωγής των αποτελεσμάτων. Όλα αυτά βέβαια δεν θα έχουν καμιά σημασία αν το σύστημα είναι ακατάλληλο από πλευράς προσβασιμότητας, ασφάλειας, αξιοπιστίας και εμπιστευτικότητας. Αυτά είναι μειονεκτήματα τα οποία προκύπτουν λόγω του ότι χρησιμοποιείται το διαδίκτυο ως μέσω αποστολής των ψήφων. Τα παραπάνω θα πρέπει να μελετηθούν και πριν ξεκινήσει η οποιαδήποτε εκλογική διαδικασία μέσω ενός τέτοιου συστήματος να είμαστε σίγουροι πως έχουν εξαλειφθεί.

1.5 Κοινωνική αποδοχή διαδικτυακής ψηφοφορίας

1.5.1 Οφέλη από ένα ασφαλές σύστημα διαδικτυακής ψηφοφορίας

Η συμμετοχή στις εκλογές και τα δημοψηφίσματα σε μια δημοκρατία είναι αναφαίρετο δικαίωμα κάθε πολίτη και ακρογωνιαίος λίθος του πολιτικού αυτού συστήματος. Σε μια δημοκρατική κοινωνία δεν θα πρέπει να αφήνεται το παραμικρό περιθώριο στους πολίτες να αμφιβάλλουν για τα αποτελέσματα οποιασδήποτε εκλογικής διαδικασίας καθώς με βάση αυτά λαμβάνονται οι αποφάσεις της πολιτικής πορείας του κάθε κράτους. Πέρα από την διαδικασία εκλογής αντιπροσώπων στο κοινοβούλιο της κάθε χώρας ή το ευρωπαϊκό κοινοβούλιο τα δημοψηφίσματα είναι η μέγιστη μορφή άμεσης δημοκρατίας. Με την χρήση της διαδικτυακής ηλεκτρονικής ψηφοφορίας τα δημοψηφίσματα μπορούν να πραγματοποιούνται πιο συχνά, καθώς οι πολίτες μπορούν να ψηφίζουν άμεσα και το κόστος διεξαγωγής τους είναι πολύ μικρότερο.

Τα οφέλη λοιπόν που μπορούμε να αποκομίσουμε από την διαδικτυακή ηλεκτρονική ψηφοφορία είναι κοινωνικά, οικονομικά και πολιτικά.

Όπως παρατηρείται τα τελευταία χρόνια μεγάλο ποσοστό του πληθυσμού και ειδικότερα οι πολίτες μικρότερων ηλικιών απέχουν από τις εκλογικές διαδικασίες αφενός διότι έχει χαθεί η εμπιστοσύνη στις πολιτικές δυνάμεις του τόπου και αφετέρου διότι θεωρούν την εκλογική διαδικασία ως μια χρονοβόρα και κουραστική υποχρέωση μέσα στην πολυάσχολη καθημερινότητά τους. Με την διαδικτυακή ψηφοφορία μπορεί να γίνει πιο συχνή η συμμετοχή των πολιτών, μέσω δημοψηφισμάτων, σε σημαντικές αποφάσεις για το μέλλον της χώρας ή της πόλης τους. Οι εκλογές δεν θα πραγματοποιούνται, λόγω της μεγάλης προετοιμασίας που χρειάζεται σήμερα, κάθε τέσσερα χρόνια, ώστε να αναδειχθούν αντιπρόσωποι οι οποίοι θα αποφασίζουν για όλα τα ζητήματα που προκύπτουν ανάμεσα στις εκλογικές διαδικασίες. Έτσι θα ενισχυθεί η εμπιστοσύνη των

πολιτών στο πολιτικό σύστημα και θα αυξηθεί η συμμετοχή τους στις αποφάσεις του κράτους. Επίσης, με την απλούστευση και την αύξηση ευελιξίας της, η εκλογική διαδικασία θα γίνει πιο προσιτή στους ψηφοφόρους καθώς θα μπορούν να ψηφίσουν μέσα σε λίγα λεπτά. Ετεροδημότες ή άλλοι πολίτες που δεν έχουν την δυνατότητα να μεταβούν στον τόπο ψηφοφορίας θα μπορούν ψηφίσουν από οποιοδήποτε σημείο, οποιαδήποτε στιγμή επιθυμούν. Τέλος, ως προς το οικονομικό κομμάτι μακροπρόθεσμα, γιατί αρχικά θα χρειαστεί να δαπανηθούν κάποια χρήματα, οι ψηφοφορίες θα διεξάγονται με πολύ λιγότερη σπατάλη δημόσιου χρήματος αλλά και ανθρώπινων πόρων από ό,τι συμβαίνει τώρα. Με λίγα λόγια το κάθε κράτος θα κερδίσει χρόνο κατά την εξαγωγή των αποτελεσμάτων και τη λήψη των αποφάσεων, μείωση των εξόδων κατά την διεξαγωγή των εκλογών και μείωση ανθρώπινων πόρων για την επιτήρηση της ψηφοφορίας.

Αδιαμφισβήτητα ο ασφαλέστερος, ως προς την ιδιωτικότητα και την δημοκρατικότητα, τρόπος διεξαγωγής μιας ψηφοφορίας είναι ο τρόπος που χρησιμοποιείται σήμερα με την χρήση της κάλπης και των παραβάν. Τα μειονεκτήματα μιας διαδικτυακής ψηφοφορίας είναι πως είναι πολύ δύσκολο να πειστούν οι πολίτες μιας χώρας πως η ψήφος τους θα καταμετρηθεί σίγουρα, θα μείνει κρυφή και πως τα αποτελέσματα θα παραμείνουν ανεπηρέαστα μέχρι την εξαγωγή τους. Έτσι αν θέλει ένα κράτος να κερδίσει τα παραπάνω οφέλη θα πρέπει να έχει κερδίσει πρώτα την κοινωνική αποδοχή για την διεξαγωγή διαδικτυακής ηλεκτρονικής ψηφοφορίας.

1.5.2 Νομικό Πλαίσιο

Στην συνέχεια λοιπόν και πριν προχωρήσουμε στον σχεδιασμό και την ανάπτυξη ενός τέτοιου συστήματος θα ήταν καλό να αναφερθούμε στους νομικούς κανόνες (νομικό πλαίσιο) που θα πρέπει να τηρεί μια ηλεκτρονική ψηφοφορία. Πρέπει να παρατηρήσουμε ότι το νομικό πλαίσιο αφενός δεν έχει αναπτυχθεί πλήρως ακόμα και αφετέρου διαφέρει από χώρα σε χώρα όπως συμβαίνει και με τον παραδοσιακό τρόπο ψηφοφορίας. Παρ' όλα αυτά μπορούμε να ξεχωρίσουμε κάποιες βασικές δημοκρατικές εκλογικές διαδικασίες που η εφαρμογή τους αποτελεί προϋπόθεση για την δημοκρατική νομιμοποίηση και των ηλεκτρονικών ψηφοφοριών. Οι βασικές αυτές αρχές είναι:

Αρχή της δημοκρατίας: Η διαδικτυακή ψηφοφορία θα πρέπει να είναι μια διαφανής διαδικασία και να εξασφαλίζεται ότι καμία ψήφος δεν θα καταμετράται δεύτερη φορά

Αρχή της ελευθερίας: Θα πρέπει να παρέχεται η δυνατότητα και της λευκής ψήφου πέρα από τις άλλες επιλογές των εκλογών και να αποφεύγονται οι υπεράριθμες επιλογές.

Αρχή της ισότητας: Όλοι οι ψηφοφόροι θα πρέπει να έχουν το δικαίωμα μιας και μόνο μιας ψήφου.

Αρχή της γενικότητας: Θα πρέπει να διασφαλίζεται από την εφαρμογή η ισότιμη πρόσβαση και η αποφυγή της πλαστοπροσωπίας.

Αρχή της μυστικότητας: Κατά την διάρκεια της ψηφοφορίας δεν θα πρέπει να εμφανίζονται τα αποτελέσματα της τρέχουσας ψηφοφορίας και η ψήφος δεν θα πρέπει να σχετίζεται με τον κάθε ψηφοφόρο.

Αρχή της ευθύτητας: Η καταμέτρηση των ψήφων θα πρέπει να είναι διαφανής και σωστή.

Από το 1990 εκδόθηκαν από την Federal Election Commission (FEC) των ΗΠΑ η οποία μετά ενσωματώθηκε στην Election Assistance Commision (EAC) οι πρώτες προδιαγραφές για τα συστήματα ηλεκτρονικής ψηφοφορίας VSS (Voting System Standards). Τα συγκεκριμένα πρότυπα αναθεωρήθηκαν το 2002 (2002 VSS) και τον Δεκέμβρη του 2005. Από τον Δεκέμβρη του 2007 είναι πλέον σε ισχύ τα VSS 2005 που αυξάνουν σημαντικά την ασφάλεια και την προσβασιμότητα, προσφέροντας περισσότερες δυνατότητες σε άτομα με ποικίλες ιδιαιτερότητες. Σύμφωνα με αυτά τα standards η National Association of State Elections Directions (NASED) των ΗΠΑ πιστοποιεί τα διάφορα συστήματα ψηφοφορίας ως προς το υλικό και λογισμικό [19] [20] .

1.5.3 Κριτήρια για ασφαλές και δίκαιο διαδικτυακό σύστημα ψηφοφορίας

Σύμφωνα λοιπόν με τα παραπάνω μπορούμε να θέσουμε κάποια κριτήρια τα οποία θα πρέπει να ικανοποιούνται σε ένα ασφαλές και δίκαιο σύστημα ηλεκτρονικής ψηφοφορίας ώστε να πετύχουμε την μέγιστη κοινωνική αποδοχή αλλά και την καλύτερη αποτελεσματικότητα της λειτουργίας του. [21]

Πληρότητα: Θα πρέπει να μετριούνται σωστά όλες οι έγκυρες ψήφοι.

Ανθεκτικότητα: Η εκλογική διαδικασία δεν μπορεί να αλλοιωθεί από κανέναν εξωτερικό παράγοντα.

Ισότητα και επιλεκτικότητα: Θα πρέπει να εξασφαλίζεται μια και μοναδική ψήφος για τον κάθε πολίτη και να τηρούνται έγκυροι εκλογικοί κατάλογοι.

Μη-διπλασιασμός: Να μην μπορεί κάποιος τρίτος να αναπαράγει την ψήφο ενός ψηφοφόρου.

Μυστικότητα και ακεραιότητα: Θα πρέπει η κάθε ψήφος να παραμένει μυστική και να μην αλλάζει από την στιγμή που την έχει υποβάλλει ο ψηφοφόρος. Επίσης από την στιγμή που έχει υποβληθεί δεν θα πρέπει να μπορεί να ξεχωρίσει από τις άλλες ψήφους.

Μη δυνατότητα κατασκευής απόδειξης: Θα πρέπει να μην υπάρχει δυνατότητα έκδοσης απόδειξης από κανέναν ψηφοφόρο σχετικά με το τι ψήφισε ή γενικότερα να καταγραφεί η ψήφος του στον Η/Υ ώστε να αποτραπεί η εξαγορά ή ο εξαναγκασμός ψήφου.

Επαληθευσιμότητα: Το αποτέλεσμα της ψηφοφορίας να μπορεί να επαληθευτεί από οποιονδήποτε.

Δημόσια συμμετοχή: Θα πρέπει όλοι να μπορούν να δουν τα αποτελέσματα και τον αριθμό των συμμετεχόντων στην ψηφοφορία.

Ως συμπέρασμα από τα παραπάνω για να έχει ένα διαδικτυακό σύστημα ηλεκτρονικής ψηφοφορίας την μέγιστη κοινωνική αποδοχή θα πρέπει να τηρηθούν οι βασικές αρχές ασφαλείας των πληροφοριακών συστημάτων προσαρμοσμένες στα ιδιαίτερα κριτήρια ασφάλειας και δικαιοσύνης που πρέπει διέπουν ένα i-voting system. Στο επόμενο κεφάλαιο αφού παρουσιαστούν οι βασικές αρχές ασφάλειας ενός πληροφοριακού συστήματος θα γίνει μια αντιστοίχιση με τα συγκεκριμένα κριτήρια και έπειτα θα προχωρήσουμε στην ανάλυση και τον σχεδιασμό του δικού μας συστήματος διαδικτυακής ψηφοφορίας.

2 Ανάλυση Συστήματος

2.1 Αρχές Ασφάλειας Πληροφοριακού Συστήματος

Οι τρεις βασικές αρχές ασφαλείας των πληροφοριακών συστημάτων είναι η εμπιστευτικότητα, η ακεραιότητα και η διαθεσιμότητα. Εκτός από αυτές ορίζονται και άλλες τρεις επιμέρους αρχές ασφαλείας. Η αυθεντικοποίηση, η εξουσιοδότηση και η μη άρνηση ευθύνης. Παρακάτω θα δοθεί ο ορισμός και οι τρόποι τήρησής τους σε ένα οποιοδήποτε πληροφοριακό σύστημα. [22]

Εμπιστευτικότητα: Σύμφωνα με αυτήν το σύστημα πρέπει να προφυλάσσεται από μη εξουσιοδοτημένη αποκάλυψη ευαίσθητων δεδομένων.

Βασικός τρόπος επίτευξής της σε μια διαδικτυακή εφαρμογή όπου χρησιμοποιείται το διαδίκτυο ως κανάλι επικοινωνίας είναι η κρυπτογράφηση των δεδομένων. Θα πρέπει επίσης να παρέχεται αυστηρός έλεγχος πρόσβασης στην Β.Δ. όπου φυλάσσονται τα δεδομένα του συστήματος.

Ακεραιότητα: Η ακεραιότητα ορίζει πως τα δεδομένα δεν μπορούν να τροποποιηθούν από κάποια μη εξουσιοδοτημένη οντότητα. Τροποποίηση δεδομένων θεωρείται η εγγραφή νέων, η αναβάθμιση των υπαρχόντων και η διαγραφή μέρους ή όλων των δεδομένων.

Μπορούμε να την τηρήσουμε με έλεγχο πρόσβασης στο σύστημα, με μηχανισμούς αυθεντικοποίησης και με ψηφιακές υπογραφές.

Διαθεσιμότητα: Σύμφωνα με αυτήν την αρχή διασφαλίζουμε πως τα δεδομένα και οι λειτουργίες του συστήματος είναι διαθέσιμα στους εξουσιοδοτημένους χρήστες και μόνο σε αυτούς όποτε αυτό απαιτείται.

Για να επιτύχουμε την τήρηση της συγκεκριμένης αρχής ασφαλείας θα πρέπει να διατηρούμε αντίγραφα ασφαλείας, να συντηρούμε και να επιβλέπουμε το υλικό του συστήματος και να μην επιτρέπουμε σε κακόβουλους εξωτερικούς χρήστες να δημιουργήσουν πρόβλημα στην λειτουργία του συστήματος.

Αυθεντικοποίηση: Η αυθεντικοποίηση ορίζει πως δεν θα πρέπει να επιτρέπεται η είσοδος και η χρησιμοποίηση του συστήματος από μη εξουσιοδοτημένους χρήστες.

Για να επιτευχθεί η συγκεκριμένη αρχή θα πρέπει οι χρήστες του συστήματος να ταυτοποιούνται πριν την είσοδό τους και να οδηγούνται σύμφωνα με τον ρόλο τους στην συγκεκριμένη λειτουργία για την οποία έχουν εξουσιοδότηση

Εξουσιοδότηση: Η εξουσιοδότηση αποτελεί την συνέχεια της αυθεντικοποίησης στην ροή λειτουργίας του συστήματος. Ορίζει πώς ο κάθε ταυτοποιημένος χρήστης θα πρέπει να μπορεί να περιηγηθεί και να εκτελέσει λειτουργίες μόνο στα συγκεκριμένα σημεία του συστήματος για τα οποία έχει εξουσιοδότηση.

Για την επίτευξή της το σύστημα θα πρέπει να είναι σχεδιασμένο και βασισμένο στους ρόλους των οντοτήτων οι οποίες λαμβάνουν μέρος στην λειτουργία του. Οι χρήστες θα

πρέπει να έχουν πρόσβαση μόνο στο τμήμα του συστήματος το οποίο έχει σχεδιαστεί για αυτούς.

Μη-Άρνηση της Ευθύνης: Η τήρηση αυτής της αρχής ασφαλείας παρέχει στους διαχειριστές του συστήματος αδιάψευστες αποδείξεις σε περιπτώσεις διαφωνίας με κάποιο χρήστη σχετικά με κάποια του ενέργεια.

Για την επίτευξή της θα πρέπει να χρησιμοποιείται μηχανισμός καταγραφής των κρίσιμων ενεργειών των χρηστών.

Σε ένα διαδικτυακό σύστημα ψηφοφορίας το οποίο θέλουμε να παρέχει στους πολίτες ασφάλεια και να κερδίζει την εμπιστοσύνη τους θα πρέπει να τηρούνται όλες οι παραπάνω γενικές αρχές ασφαλείας πληροφοριακών συστημάτων.

2.2 Αντιστοίχιση βασικών αρχών ασφαλείας με τις απαιτήσεις του Συστήματος Διαδικτυακής Ψηφοφορίας

Αναλογιζόμενοι λοιπόν από την μια τις γενικές αρχές ασφάλειας ενός πληροφοριακού συστήματος και τους τρόπους επίτευξής της και από την άλλη το νομικό πλαίσιο και τα κριτήρια τα οποία θα πρέπει να τηρεί ένα σύστημα διαδικτυακής ψηφοφορίας θα ήταν καλό να γίνει μια αντιστοίχιση τους ώστε να προχωρήσουμε στην ανάλυση απαιτήσεων και τον σχεδιασμό του συστήματος.

Αρχές Ασφαλείας	Νομικό Πλαίσιο – Κριτήρια ασφαλούς και δίκαιου συστήματος i-voting
Εμπιστευτικότητα	Αρχή της μυστικότητας Μυστικότητα και ακεραιότητα
Ακεραιότητα	Αρχή της δημοκρατίας Αρχή της ισότητας Αρχή της ευθύτητας Πληρότητα Ισότητα και επιλεκτικότητα Μη-διπλασιασμός Μυστικότητα και ακεραιότητα
Διαθεσιμότητα	Ανθεκτικότητα
Αυθεντικοποίηση	Αρχή της γενικότητας
Εξουσιοδότηση	
Μη-Άρνηση της Ευθύνης	Επαληθευσιμότητα Δημόσια συμμετοχή

Πίνακας 2 Αντιστοίχιση αρχών ασφαλείας Π.Σ. με το νομικό πλαίσιο και τα κριτήρια ασφαλούς και δίκαιου συστήματος i-voting

2.3 Ανάλυση απαιτήσεων Συστήματος διαδικτυακής Ψηφοφορίας

Σύμφωνα με την παραπάνω αντιστοίχιση παρατηρούμε πως αν λάβουμε μέτρα για όλες τις αρχές ασφαλείας ενός πληροφοριακού συστήματος θα έχουμε καταφέρει να

εξασφαλίσουμε την ασφάλεια του διαδικτυακού συστήματος πληροφορίας αλλά και να τηρήσουμε τα κριτήρια που απαιτούνται για να δημιουργήσουμε ένα ασφαλές και δίκαιο σύστημα i-voting. Θα πρέπει εδώ να τονίσουμε πως η αρχή της εξουσιοδότησης θα πρέπει να επιτευχθεί επειδή το σύστημα θα πρέπει να χρησιμοποιείται και από άλλες οντότητες εκτός από τους ψηφοφόρους. Θα πρέπει να το χειρίζονται οι διαχειριστές του, οι χρήστες των αρχών πιστοποίησης του κράτους και χρήστες οι οποίοι θα έχουν πρόσβαση στα αποτελέσματα και τα στατιστικά. Τέλος, ένα ιδιαίτερο κριτήριο είναι η μη δυνατότητα κατασκευής απόδειξης σύμφωνα με το οποίο, όπως αναφέραμε, δεν θα πρέπει να μπορεί κανένας ψηφοφόρος ή διαχειριστής να εξάγει απόδειξη για την ψηφοφορία κάποιου πολίτη. Αυτό το κριτήριο δεν μπορεί να ενταχθεί στον παραπάνω πίνακα αλλά είναι βασικό για την δημοκρατική λειτουργία της ψηφοφορίας και θα πρέπει να ληφθεί υπόψιν.

Κατά την ανάλυση απαιτήσεων εκτός από την ασφάλεια η οποία είναι σημαντικός παράγοντας, θα πρέπει κατά το σχεδιασμό και την υλοποίηση του συστήματος να δοθεί ιδιαίτερη βάση στην ευχρηστία του και την φιλικότητά του προς τους χρήστες. Εάν το σύστημα είναι περίπλοκο και κουραστικό θα έχει ως αποτέλεσμα οι χρήστες να μην το χρησιμοποιήσουν. Επίσης, το σύστημα θα πρέπει να είναι σχεδιασμένο ώστε οι χρήστες να έχουν πρόσβαση σε αυτό από όλες τις πιθανές συσκευές και εφαρμογές που έχουν πρόσβαση στο διαδίκτυο χωρίς να δημιουργούνται δυσλειτουργίες ή κενά ασφαλείας. Βασικά χαρακτηριστικά της ευχρηστίας ενός πληροφοριακού συστήματος είναι τα παρακάτω:

- Ευκολία στην μάθηση
- Αποδοτικότητα στη χρήση
- Ευκολία στην απομνημόνευση
- Χαμηλή συχνότητα λαθών
- Υποκειμενική ικανοποίηση
- Συμβατό με όλες τις πιθανές υπολογιστικές μηχανές.

Συνοψίζοντας, ένα διαδικτυακό σύστημα ψηφοφορίας θα να πρέπει καλύπτει τις παρακάτω λειτουργικές απαιτήσεις.

Υψηλή διαθεσιμότητα (High Availability) Κάλυψη των απαιτήσεων των χρηστών

Χρηστικότητα (Usability) Ευκολία στο τρόπο εκμετάλλευσης των λειτουργιών του από τους χρήστες.

Διαλειτουργικότητα (Interoperability) Επικοινωνία με συστήματα διαφορετικών τεχνολογιών.

Σταθερότητα (Stability) Δυνατότητα διαχείρισης μεγάλου αριθμού χρηστών με αξιοπιστία και αποτελεσματικότητα.

Ασφάλεια (Security) Ισχυρή αυθεντικοποίηση και ταυτοποίηση χρηστών, εξουσιοδότηση, εμπιστευτικότητα, ακεραιότητα δεδομένων, μη άρνηση ευθύνης και αδιάλειπτη διαθεσιμότητα υπηρεσιών.

Στις επόμενες ενότητες θα γίνει ανάλυση των παραπάνω απαιτήσεων αρχίζοντας από την ασφάλεια. Οι απαιτήσεις της σταθερότητας και της υψηλής διαθεσιμότητας εξαρτώνται πέρα από το λογισμικό του συστήματος και από το υλικό και την δικτύωση τα οποία θα

χρησιμοποιούνται ώστε να λειτουργεί. Στο σημείο αυτό και για την συγκεκριμένη εργασία θεωρούμε πως το υλικό και η δικτύωση του συστήματος μας είναι τέτοια ώστε να μην υπάρχει κίνδυνος αποτυχίας λειτουργίας του. Αρχικά λοιπόν θα ορίσουμε τις τελικές απαιτήσεις ασφαλείας του συστήματος και διερευνώντας τα ευάλωτα σημεία των διαδικτυακών συστημάτων και τους τρόπους αντιμετώπισης αυτών των ευπαθειών θα ορίσουμε κατά τον σχεδιασμό τις τεχνολογίες που θα χρησιμοποιήσουμε. Λόγω του ότι το σύστημα που θα αναπτύξουμε είναι διαδικτυακό έχουμε την πολυτέλεια να χρησιμοποιήσουμε διάφορα εργαλεία τα οποία με τον κατάλληλο σχεδιασμό και υλοποίηση μπορούν να καλύψουν και την απαίτηση της διαλειτουργικότητας και της χρηστικότητας.

2.4 Απαιτήσεις ασφάλειας συστήματος

Για την επίτευξη της μέγιστης ασφάλειας το σύστημα θα πρέπει να καλύπτει επάξια όλες τις αρχές ασφαλείας. Λόγω του ότι είναι διαδικτυακό οι κίνδυνοι αποτυχίας ασφαλείας αυξάνονται.

Κάθε κακόβουλος χρήστης θα μπορεί να συνδεθεί και να εξαπολύσει επιθέσεις από οποιοδήποτε Η/Υ σε οποιοδήποτε σημείο αρκεί να έχει πρόσβαση στο διαδίκτυο. Το σύστημα λοιπόν θα πρέπει να προβλέψει κάθε πιθανή επίθεση η οποία είναι γνωστή ως τώρα και να την περιορίσει χωρίς να τεθεί σε κίνδυνο η λειτουργία του ή η ορθή συγκομιδή των αποτελεσμάτων. Επίσης λόγω του ότι και οι μη κακόβουλοι χρήστες θα μπορούν να ψηφίσουν από δίκτυα τα οποία δεν γνωρίζουμε αν είναι ασφαλή καθώς και από τους Η/Υ τους οι οποίοι και πάλι δεν γνωρίζουμε αν είναι ασφαλείς θα πρέπει να εξασφαλίσουμε πως όλα τα δεδομένα που αποστέλλουν είναι προστατευμένα ως προς την ανάγνωσή τους από τρίτους.

Για την επίτευξη της εμπιστευτικότητας θα πρέπει το σύστημά μας να χρησιμοποιεί τεχνολογίες κρυπτογραφίας των δεδομένων που στέλνει και λαμβάνει. Κάθε χρήστης από την στιγμή που εισέρχεται στο σύστημα θα πρέπει να ξεκινάει με αυτό μια ασφαλή συνεδρία μέσα στην οποία ό,τι αποστέλλεται να είναι κρυπτογραφημένο. Πέρα όμως από την κρυπτογραφία η οποία μας διασφαλίζει πως τα δεδομένα δεν θα είναι αναγνώσιμα από τρίτους θα πρέπει να χρησιμοποιηθούν μηχανισμοί αυθεντικοποίησης και εξουσιοδότησης ώστε να διασφαλίζεται η ακεραιότητα των προσωπικών δεδομένων των χρηστών, των αποφάσεων τους και των τελικών αποτελεσμάτων των ψηφοφοριών. Με την χρησιμοποίηση αυτών των μηχανισμών αλλά και με τη σωστή σχεδίαση και υλοποίηση της ροής λειτουργίας του συστήματος θα πρέπει να διασφαλίζεται πως οι χρήστες δεν έχουν δικαίωμα να ψηφίσουν δεύτερη φορά σε κάθε ψηφοφορία και πως κανείς από τους άλλους χρήστες του συστήματος δεν θα μπορεί από την στιγμή που οι ψηφοφόροι έχουν εγγραφεί σε αυτό να δει ή να τροποποιήσει τα στοιχεία ή τις αποφάσεις τους.

Για πρώτη φορά κάνουμε αναφορά και σε άλλους χρήστες πέρα από τους ψηφοφόρους οι οποίοι θα αλληλεπιδρούν με το σύστημα. Σε επόμενη ενότητα της ανάλυσης θα γίνει μεγαλύτερη αναφορά στις διαδικασίες τις οποίες θα πρέπει να μπορούν να εκτελούν όταν

αλληλεπιδρούν με το διαδικτυακό σύστημα ψηφοφορίας. Οι χρήστες αυτοί θα έχουν διαχειριστικό ρόλο στην λειτουργία του συστήματος· οπότε, όπως είναι λογικό, επιβάλλεται να χρησιμοποιηθούν οι παραπάνω μηχανισμοί ασφαλείας και κατά τις δικές τους συνεδρίες με την εφαρμογή. Επίσης με τους μηχανισμούς εξουσιοδότησης και αυθεντικοποίησης θα πρέπει να διασφαλίσουμε πως κανένας χρήστης του συστήματος δεν θα πρέπει να μπορεί να συμμετέχει σε διαδικασίες ή να διαχειρίζεται δεδομένα για τα οποία είναι εξουσιοδοτημένη άλλη ομάδα χρηστών.

2.5 Τύποι επιθέσεων σε διαδικτυακά συστήματα

Όπως είναι προφανές από όλα τα παραπάνω το μεγαλύτερο πρόβλημα το οποίο έχουμε να αντιμετωπίσουμε κατά την ανάπτυξη ενός διαδικτυακού συστήματος ψηφοφορίας είναι οι επιθέσεις τις οποίες μπορεί αυτό να δεχτεί κατά την διάρκεια της διεξαγωγής μιας ψηφοφορίας ή κατά την διάρκεια στην οποία είναι αδρανές. Δυστυχώς οι επιθέσεις αυτές είναι ένα πρόβλημα το οποίο δεν έχει πλήρως λυθεί σε καμία διαδικτυακή εφαρμογή και όπως όλα φαίνονται δεν θα λυθεί εύκολα. Ο λόγος που συμβαίνει αυτό είναι πως όσο εξελίσσεται η άμυνα στις διάφορες επιθέσεις τόσο βρίσκονται νέοι τρόποι ώστε να διασπάται.

Ο μόνος τρόπος με τον οποίο μπορούμε να καλυφτούμε επαρκώς από τέτοιες επιθέσεις είναι το σύστημά μας να είναι έτοιμο να αντιμετωπίσει όσες από αυτές είναι γνωστές έως τώρα και να παρακολουθείται και να ενημερώνεται συνεχώς ως προς την άμυνα σε νέες που γίνονται γνωστές. Παρακάτω θα κάνουμε μια αναφορά σε κάποιες από τις πιο γνωστές επιθέσεις οι οποίες μπορεί να γίνουν σε ένα οποιοδήποτε διαδικτυακό σύστημα στο διαδίκτυο.

Κοινωνική Μηχανική (Social Engineering)

Μια μορφή επίθεσης είναι η κοινωνική μηχανική (Social Engineering). Την χαρακτηρίζουμε μορφή επίθεσης γιατί ουσιαστικά δεν είναι ακριβής επίθεση στο σύστημά μας. Ουσιαστικά πραγματοποιείται όταν κάποιος προσπαθεί με προφορικούς ή πρακτικούς δόλιους τρόπους να αποσπάσει πληροφορίες από τους χρήστες. Για κάποιον έμπειρο χρήστη Η/Υ αυτή η μορφή επίθεσης ίσως είναι ακίνδυνη. Αλλά για την πλειοψηφία των χρηστών μπορεί να αποδειχθεί επικίνδυνη. Βασίζεται είτε στην περιέργεια είτε στην άγνοια των χρηστών. Πολλοί χρήστες δεν προσέχουν που δίνουν τα στοιχεία σύνδεσης τους ή άλλα προσωπικά δεδομένα τα οποία μπορεί να χρησιμοποιηθούν από κάποιους ώστε τελικά να υποκλαπούν οι κωδικοί τους. Επίσης δεν δίνουν σημασία στα αντιϊκά προγράμματα και μπορεί στον Η/Υ τους να εκτελείται μια εφαρμογή που μπορεί να υποκλέψει δεδομένα. Ένας ακόμα τρόπος social engineering είναι το λεγόμενο «ψάρεμα» όπου μέσα από μια ψεύτικη ιστοσελίδα κάποιος μπορεί να ξεγελάσει έναν χρήστη και να τον κάνει να δώσει τα στοιχεία σύνδεσης του για κάποια άλλη εφαρμογή. Λόγω του ότι η διαδικτυακή ψηφοφορία θα απευθύνεται και σε πολλούς χρηστές που δεν έχουν αρκετές γνώσεις προφύλαξης από τα παραπάνω θα πρέπει να δοθεί ιδιαίτερο βάρος και σε αυτήν την μορφή επίθεσης [23][24].

Άρνηση Υπηρεσιών (Denial of Service)

Η επίθεση άρνησης υπηρεσιών στοχεύει στο να προκαλέσει στο σύστημα τέτοια ζημιά ώστε να μην μπορέσει να εξυπηρετήσει τους χρήστες και να χρειαστεί, στην απλούστερη περίπτωση επανεκκίνηση. Τέτοιες επιθέσεις μπορούν γίνουν στέλνοντας στο σύστημα μεγάλο αριθμό αιτήσεων ή πακέτων. Κάποιες από τις γνωστές αυτές επιθέσεις είναι οι syn flood, ping flood, teardrop (κεφαλίδες ip), ping of death, smurf (ping ICMP), και fraggle (UDP). Είναι κυρίως επιθέσεις οι οποίες μπορούν να καταπολεμηθούν κυρίως από έλεγχο στα χαμηλότερα επίπεδα του δικτύου και όχι στο επίπεδο εφαρμογής. Παρόμοια επίθεση, στο επίπεδο της εφαρμογής μπορεί να πραγματοποιηθεί όμως στέλνοντας συνεχόμενες αιτήσεις ή μεγάλο αριθμό δεδομένων ζητήματα τα οποία θα πρέπει το σύστημα να τα αποκλείει πριν του δημιουργήσουν πρόβλημα [25][26].

Υποκλοπή Συνεδρίας (Session Hijacking)

Μια ακόμα εξίσου σημαντική μορφή επίθεσης είναι η υποκλοπή της συνεδρίας. Με τον όρο συνεδρία εννοούμε έναν μοναδικό αριθμό τον οποίο έχει ο κάθε χρήστης καθώς συνδέεται στον εξυπηρετητή του συστήματος. Για να καταφέρει κάποιος να υποκλέψει αυτόν τον αριθμό θα πρέπει είτε να έχει ήδη πρόσβαση στον Η/Υ του χρήστη και να κλέψει το αρχείο στο οποίο είναι αποθηκευμένος, είτε να έχει πρόσβαση στο δίκτυο από το οποίο συνδέεται ο χρήστης και να διαβάζει τα δεδομένα που κυκλοφορούν είτε ξεγελώντας τους χρήστες κάνοντας τους να συνδεθούν σε μια ιστοσελίδα που έχει παρόμοιο κώδικα με της εφαρμογής και να καταφέρει να ζητήσει και να πάρει το κλειδί της συνεδρίας. Θα πρέπει λοιπόν σίγουρα η εφαρμογή μας να δέχεται και να στέλνει πίσω αν χρειάζεται κρυπτογραφημένο το session key ώστε να αποφύγουμε την υποκλοπή μέσω διαδικτύου και να γίνεται συνεχώς σε κάθε λειτουργία του χρήστη έλεγχος της ταυτότητάς του. Τέλος, όπως και στο social engineering οι χρήστες θα πρέπει να έχουν ενημερωθεί και να χρησιμοποιούν ένα αναβαθμισμένο αντιϊκο πρόγραμμα.

Οι παραπάνω επιθέσεις όπως είδαμε έχουν να κάνουν κυρίως με επιθέσεις στους χρήστες ή στο δίκτυο και όχι στην εφαρμογή. Οι βασικότερες επιθέσεις που μπορούν γίνουν στην εφαρμογή μας είναι η επίθεση Cross-Site Scripting (XSS) , η επίθεση sql-injection και η επίθεση buffer overflow.[27]

Cross-Site Scripting (XSS)

Η επίθεση cross-site scripting πραγματοποιείται όταν κάποιος καταφέρει να εκμεταλλευτεί κάποιες ευπάθειες του συστήματος ώστε να εκτελέσει δικό του κώδικα με σκοπό να υποκλέψει στοιχεία χρηστών ή να εκτελέσει μια μη επιτρεπτή για αυτόν λειτουργία. Ένας κακόβουλος χρήστης μπορεί να επιτύχει κλοπή των κωδικών των χρηστών, κλοπή προσωπικών δεδομένων, αλλαγή στις ρυθμίσεις της εφαρμογής, κλοπή των cookies ή ψεύτικη διαφήμιση προς όφελός του. Η συγκεκριμένη επίθεση χωρίζεται σε τρείς κατηγορίες. Τη μη μόνιμη όπου όταν τα δεδομένα δίνονται στον πελάτη ιστού χρησιμοποιούνται από κάποιο σενάριο (script) το οποίο εμφανίζει αναληθή αποτελέσματα στον πελάτη χωρίς να έχουν ελεγχτεί από τον εξυπηρετητή. Τη μόνιμη, κατά την οποία ο κακόβουλος χρήστης καταφέρνει να αποστείλει δεδομένα στον εξυπηρετητή και αυτός τα

εμφανίζει πλέον στις ιστοσελίδες. Τέλος, οι βασισμένες σε Document Object Model (DOM) τρόπο με τον οποίο εμφανίζονται τα HTML και XML αντικείμενα, ευπάθειες οι οποίες και εμφανίζονται σε web 2.0 εφαρμογές. Αυτές οι επιθέσεις συμβαίνουν όταν ο χρήστης έχει δεχθεί τα δεδομένα από τον εξυπηρετητή και ο πελάτης ιστού που χρησιμοποιεί ετοιμάζει το αποτέλεσμα.

Ο καλύτερος τρόπος αντιμετώπισης της συγκεκριμένης ευπάθειας είναι να αναζητηθούν τα σημεία του κώδικα της εφαρμογής τα οποία χρησιμοποιούν δεδομένα τα οποία έχει εισαγάγει ο χρήστης μέσω κάποιας φόρμας. Αυτά τα δεδομένα θα πρέπει να φιλτράρονται και να είμαστε σίγουροι πως δεν επιτρέπουμε να εισαχθούν στο σύστημα χαρακτήρες πέρα από αυτούς του οποίους περιμένει η εφαρμογή [28][29].

Έγχυση SQL (SQL Injection)

Η συγκεκριμένη επίθεση έχει να κάνει πάλι με την ευχέρεια ενός κακόβουλου χρήστη να βρει τις ευπάθειες του συστήματος και να εισαγάγει σε αυτό κώδικα ο οποίος μπορεί να εκτελέσει μια διεργασία επιβλαβή για το σύστημα ή τα δεδομένα του. Στη συγκεκριμένη επίθεση χρησιμοποιείται κώδικας sql ο οποίος αν βρεθεί το κατάλληλο κενό ασφαλείας μπορεί να προκαλέσει ζημιά στα δεδομένα της βάσης δεδομένων ή να τα αποκαλύψει σε χρήστες που δεν θα έπρεπε να έχουν πρόσβαση σε αυτά. Και σε αυτήν την επίθεση χρησιμοποιούνται τα σημεία στα οποία οι χρήστες εισάγουν δεδομένα στο σύστημα. Εκεί με την κατάλληλη ακολουθία χαρακτήρων μπορούν να σχηματίσουν ένα ερώτημα (query) το οποίο αν τελικά εκτελεστεί στον εξυπηρετητή μπορεί να υποκλέψει ή να διαβάλει τα δεδομένα της εφαρμογής.

Για την αντιμετώπιση αυτής της επίθεσης θα πρέπει να γίνονται έλεγχοι στους χαρακτήρες που εισάγονται στο σύστημα. Εκτός από αυτό, τα ερωτήματα που εκτελούνται και χρησιμοποιούν δεδομένα που έχει εισαγάγει ο χρήστης θα πρέπει να τα δέχονται ως μεταβλητές και όχι κατευθείαν. Με αυτόν τον τρόπο στο σχηματισμό του κάθε ερωτήματος θα μπορεί να ελέγχεται και πάλι η ορθότητα των χαρακτήρων που περιέχουν οι μεταβλητές [30][31][32].

Υπερχείλιση μνήμης (Buffer Overflows)

Η συγκεκριμένη επίθεση όπως λέει και το όνομα της προκαλεί υπερχείλιση της προσωρινής μνήμης τους συστήματος. Ένας κακόβουλος χρήστης μπορεί να περάσει και πάλι ως παράμετρο ένα μεγάλο κομμάτι κώδικα. Ο κώδικας αυτός δεν είναι ανάγκη να εκτελεστεί. Αν το μέγεθος που εισάγει είναι μεγαλύτερο από το μέγεθος της μνήμης που έχει οριστεί για την συγκεκριμένη παράμετρο θα προκληθεί υπερχείλιση. Και πάλι το σύστημα θα πρέπει να είναι σε θέση να ελέγχει τις παραμέτρους που εισάγει ο χρήστης και να μην επιτρέπει να εισαχθούν δεδομένα που θα προκαλέσουν υπερχείλιση [33].

Επιθέσεις ωμής βίας (Brute-force attacks)

Κατά την επίθεση brute-force γίνεται εξαντλητική δοκιμή κλειδιών είτε για να βρεθούν τα στοιχεία πρόσβασης ενός χρήστη είτε για να παραχθεί ένα κρυπτογράφημα μέσω του οποίου μπορεί να αναλυθεί το αρχικό μήνυμα που έχει κρυπτογραφηθεί. Συνήθως χρησιμοποιούνται ώστε να «σπάσουν» τους αλγόριθμους κρυπτογράφησης. Γι' αυτό οι

αλγόριθμοι που θα χρησιμοποιήσουμε θα πρέπει να έχουν μεγάλο σε αριθμό χαρακτήρων κλειδί κρυπτογράφησης ώστε να είναι απίθανο από πλευράς χρόνου να μπορεί να κρυπταναλυθεί ο αλγόριθμος με αυτήν την επίθεση [34].

Εκτός λοιπόν από τις επιθέσεις στα συστήματα των χρηστών, στο δίκτυο ή στο διαδικτυακό σύστημα μπορούν να γίνουν επιθέσεις και στην κρυπτογραφία του συστήματος.

Όπως έχουμε τονίσει και παραπάνω είναι απαραίτητο να χρησιμοποιήσουμε αλγόριθμους κρυπτογράφησης για τα δεδομένα που διαχειρίζεται η εφαρμογή. Για να επιτευχθεί αυτό θα πρέπει ο χρήστης με το σύστημα να δημιουργούν μια ασφαλή κρυπτογραφημένη συνεδρία. Για έναν κακόβουλο χρήστη το καλύτερο που θα μπορούσε να πετύχει θα ήταν να καταφέρει να αποσπάσει τα δεδομένα από αυτήν την συνεδρία. Υπάρχουν διάφοροι τρόποι με τους οποίους προσπαθούν και έχουν καταφέρει κακόβουλοι χρήστες να «σπάσουν» την κρυπτογράφηση που χρησιμοποιείται. Οι επιθέσεις κατά τις οποίες γίνονται τέτοιες απόπειρες άλλοτε επιτυχημένες και άλλοτε αποτυχημένες ονομάζονται SSL/TLS Attacks.

Beast attack

Τον Σεπτέμβριο του 2011 οι ερευνητές της Ταϊλάνδης Duong και Juliano Rizzo απέδειξαν μια ευπάθεια στον cipher block chaning (CBC) του TLS1.0. Δηλαδή στον αλγόριθμο κρυπτογράφησης του TLS 1.0[35].Τα ευχάριστα νέα είναι πως η άμυνα στην επίθεση αυτή έχει επιτευχθεί από την έκδοση του πρωτόκολλο TLS 1.1 και μετά. Ο RC4 αρχικά είχε θεωρηθεί πως έχει ανοσία στην επίθεση beast αλλά τελικά από το 2013 και μετά ανακαλύφθηκε πως έχει και αυτός ευπάθειες και η χρήση του θα πρέπει να αποφεύγεται[36]. Οι διάφορες εταιρίες των πελατών ιστού έχουν βελτιώσει τις εφαρμογές τους ώστε να ανταπεξέρχονται στην επίθεση αλλά για να συμβεί αυτό θα πρέπει να υποστηρίζει αρχικά ο εξυπηρετητής της εφαρμογής το πρωτόκολλο TLS1.1 (ή νεότερο) αλλιώς αναγκαστικά οι web clients θα κάνουν μετάπτωση στην προηγούμενη έκδοση και οι συνεδρίες θα είναι ευάλωτες σε επιθέσεις beast [37][38].

Crime attack – Breach attack

Από του δημιουργούς της beast attack δημιουργήθηκε και η crime attack. Σε αυτήν ο κακόβουλος χρήστης μπορεί να πετύχει πρόσβαση στα κρυπτογραφημένα δεδομένα αν χρησιμοποιείται συμπίεση του TLS [39][40]. Οι καινούριες εκδόσεις των web-clients σύμφωνα με τους δημιουργούς τους περιορίζουν τις συγκεκριμένες επιθέσεις αρκεί να είναι ενημερωμένες και να χρησιμοποιείται από τον εξυπηρετητή το πιο πρόσφατο πρωτόκολλο TLS1.2. Το 2013 δημιουργήθηκε μια νέα έκδοση της crime η Breach attack με την οποία, σύμφωνα με τους δημιουργούς της, μπορούμε να αποκτήσουμε πρόσβαση σε ευαίσθητα κρυπτογραφημένα δεδομένα σε λιγότερο από 30 δευτερόλεπτα. Η συγκεκριμένη επίθεση βασίζεται στην συμπίεση του HTTP οπότε και είναι δύσκολο να αποφευχθεί αν απλά απενεργοποιήσουμε την συμπίεση του TLS [41][42]. Δυστυχώς μέχρι και σήμερα ο μόνος τρόπος αποφυγής αυτών των επιθέσεων αν και δεν είναι και απόλυτα σίγουρο πως θα πετύχει, είναι να απενεργοποιηθεί η συμπίεση των δεδομένων πράγμα όμως το οποίο θα προξενήσει μεγαλύτερες καθυστερήσεις στην εκτέλεση των διαδικασιών

της εφαρμογής [43]. Αυτό δυστυχώς είναι ένα μειονέκτημα του TLS και η μόνη λύση είναι να προσπαθούμε να εγκαθιστούμε συνεδρίες χωρίς συμπίεση και ανάμεσα σε εξυπηρετητές και πελάτες οι οποίοι είναι απόλυτα ενημερωμένοι.

Heart-bleed attacks

Το σφάλμα Heartbleed είναι μια σοβαρή ευπάθεια στην βιβλιοθήκη λογισμικού OpenSSL. Οι εκδόσεις που επηρεάζουν είναι οι 1.0.0 για η 1.0.1. Το σφάλμα Heartbleed επιτρέπει σε οποιονδήποτε στο Internet για να διαβάσει τη μνήμη των συστημάτων τα οποία έχει αποφασιστεί να προστατεύονται από τις ευπαθείς εκδόσεις του OpenSSL. Αυτό όπως καταλαβαίνουμε ελλοχεύει τον κίνδυνο να υποκλαπούν τα μυστικά κλειδιά ή ακόμα και οι κωδικοί σύνδεσης ή άλλα ευαίσθητα δεδομένα του συστήματος. Με τις νεότερες εκδόσεις του OpenSSL ευτυχώς το πρόβλημα αυτό διορθώθηκε [44][45].

Lucky 13

Η επίθεση lucky 13 είναι επίθεση ενάντια στο πρωτόκολλο TLS η οποία αναφέρθηκε πρώτη φορά τον Φεβρουάριο του 2013 [46]. Είναι ουσιαστικά επίθεση στον έλεγχο του message authentication code (MAC) του πρωτόκολλου TLS. Λίγο καιρό μετά την δημοσίευση της με αναβαθμίσεις στα λογισμικά κρυπτογράφησης ο κίνδυνος εξαλείφθηκε [47].

Poodle

Η επίθεση poodle είναι μια Man-in-the-middle επίθεση η οποία ουσιαστικά προκαλεί μια μετάπτωση του πρωτόκολλου κρυπτογράφησης σε παλιότερη έκδοση η οποία είναι τρωτή. Η επίθεση αυτή αναφέρθηκε τον Οκτώβριο του 2014. Για την αποφυγή της θα πρέπει ο εξυπηρετητής να μην επιτρέπει την έναρξη συνεδρίας μεταξύ αυτού και του χρήστη με πρωτόκολλο κρυπτογράφησης SSL 3.0 ή του TLS 1.0 αλλά μόνο με τις νεότερες εκδόσεις. Με αυτόν τον τρόπο ακόμα και αν ο χρήστης άθελά του προσπαθήσει να εγκαταστήσει ασφαλή επικοινωνία με τρωτό πρωτόκολλο κρυπτογράφησης αυτό δεν θα είναι εφικτό [48][49].

SSL Strip

Η συγκεκριμένη επίθεση λειτουργεί και αυτή με τον τρόπο man-in-the-middle και παρουσιάστηκε το 2009 [50]. Θα πρέπει ο κακόβουλος χρήστης να έχει πρόσβαση σε κοινό δίκτυο με το θύμα και χρησιμοποιώντας κάποιο εργαλείο arp-spoofing να παρέμβει στην επικοινωνία του με τον εξυπηρετητή. Κατά αυτήν την επίθεση ο επιτιθέμενος παρουσιάζει στο θύμα σελίδες μη κρυπτογραφημένες ως κρυπτογραφημένες με γνωστό στον επιτιθέμενο κλειδί κρυπτογράφησης με σκοπό την εξαπάτησή του και την υποκλοπή στοιχείων. Εκτός από αυτό ο κακόβουλος χρήστης μπορεί να προωθήσει αίτηση από μια κρυπτογραφημένη σελίδα σε μια άλλη μη κρυπτογραφημένη με σκοπό και πάλι την αρπαγή δεδομένων [51].

Για την αποφυγή αυτής της επίθεσης θα πρέπει να χρησιμοποιούμε ένα πιστοποιημένο από μια αρχή πιστοποίησης πιστοποιητικό και να ενημερώσουμε τους χρήστες πως αν δεν βλέπουν το συγκεκριμένο πιστοποιητικό στη συνεδρία τους με την σελίδα δεν θα πρέπει

να προχωρούν σε καμιά διεργασία με την εφαρμογή καθώς κατά πάσα πιθανότητα δεν θα είναι η αληθινή. Επίσης θα πρέπει να διασφαλίζουμε πως οι χρήστες αλληλεπιδρούν με τον διακομιστή μόνο μέσα από μια κρυπτογραφημένη διαδικασία και σε καμία περίπτωση μέσα από μια μη κρυπτογραφημένη. Ένας τρόπος για να το πετύχουμε αυτό είναι να είναι ανοιχτή μόνο η πόρτα SSH στην οποία ακούει ο εξυπηρετητής τις αιτήσεις https και ένας άλλος η χρησιμοποίηση της πολιτικής HSTS.

HSTS (HTTP Strict Transport Security)

Σύμφωνα με αυτόν τον μηχανισμό μέσα από μια σύνδεση HTTPS, με την χρήση μια κεφαλίδας, ο εξυπηρετητής παρέχει μια πολιτική, για κάποιο ορισμένο χρόνο, η οποία δεν επιτρέπει στον χρήστη να συνδεθεί με μη ασφαλή σύνδεση με τον εξυπηρετητή. Συγχρόνως, αν δεν μπορεί να διασφαλιστεί η ασφαλής σύνδεση (π.χ. μη έγκυρο πιστοποιητικό), εμφανίζεται μήνυμα και δεν του επιτρέπεται η εγκαθίδρυση ασφαλούς επικοινωνίας άρα και η αλληλεπίδραση με την εφαρμογή [52][53] .

Δυστυχώς έχουν παρατηρηθεί και σε αυτήν την πολιτική επιθέσεις.

Unauthenticated NTP

Με αυτήν την επίθεση ο κακόβουλος χρήστης καταφέρνει να αλλάξει την ώρα στην κεφαλίδα μηνυμάτων NTP[54] που στέλνονται μέσω πακέτων UDP. Έτσι μπορεί εύκολα να προσπελαστεί το χρονικό όριο της πολιτικής HSTS και να προκληθεί δυσλειτουργία στην κρυπτογραφημένη επικοινωνία.

Legit certificates

Όσο παράλογο και αν ακούγεται μπορεί ένα πλαστό πιστοποιητικό να έχει θεωρηθεί από μια αρχή πιστοποίησης ως έγκυρο και μ' αυτόν τον τρόπο οι χρήστες να μην δουν κανέναν μήνυμα επιβεβαίωσης λανθασμένου πιστοποιητικού από τον web client τους αν κάποιος έχει παρεμβληθεί ανάμεσα σε αυτούς και τον εξυπηρετητή. Είναι μια εξεζητημένη περίπτωση αλλά έχει συμβεί καθώς κακόβουλοι χρήστες έχουν πετύχει επίθεση σε αρχή πιστοποίησης και για κάποιο χρονικό διάστημα είχαν καταφέρει να διανέμουν έγκυρα-πλαστά πιστοποιητικά.

TOFU

Το πρόβλημα (Trust on first Use) TOFU έγκειται στο ότι οι χρήστες πρέπει να επιβεβαιώσουν, πριν προχωρήσουν, πως αποδέχονται το πιστοποιητικό του εξυπηρετητή. Αυτή την επιλογή επιβεβαίωσης την προβάλλει ο web-client κάθε φορά που παρατηρεί κάποια αλλαγή σχετικά με τον εξυπηρετητή ή το πιστοποιητικό του. Σε μια τέτοια ένδειξη, την πρώτη φορά δηλαδή που γίνεται η σύνδεση μετά από κάποια αλλαγή, μπορεί κάποιος να παρεμβληθεί και να αλλάξει την κεφαλίδα του HSTS σε απλή HTTP και έτσι η συνεδρία να εγκατασταθεί με τον κακόβουλο χρήστη ενδιάμεσο χωρίς καμιά ουσιαστική κρυπτογράφηση [55].

Οι παραπάνω επιθέσεις στο HTST δεν έχουν ακόμα αποφευχθεί πλήρως αλλά έχουν βρεθεί κάποιοι τρόποι που μας παρέχουν μιας μορφής προστασία.

Ο ένας είναι να καταχωρήσουμε την εφαρμογή μας σε μια προφορτωμένη λίστα (**preload list)** που χρησιμοποιεί η Google [56] και την οποία χρησιμοποιούν και άλλοι πάροχοι εφαρμογών web client ώστε οι clients να γνωρίζουν πως η εφαρμογή υποχρεωτικά χρησιμοποιεί HTTPS και HSTS. Σύμφωνα με την Google οι προϋποθέσεις για να συμβεί αυτό είναι να έχουμε ένα έγκυρο πιστοποιητικό, να γίνεται προώθηση όλων των σελίδων μας από http σε https και να χρησιμοποιείται η κατάλληλη κεφαλίδα HTST πριν εγκαθιδρυθεί η συνεδρία.

Ένας άλλος τρόπος είναι ο λεγόμενος **certificate pinning.** Και πάλι ο client έχει μια λίστα με έγκυρα domains και πιστοποιητικά και ουσιαστικά πιστοποιεί και αυτός την εγκυρότητα της εφαρμογής και του πιστοποιητικού [57].

Τέλος, υπάρχουν και περεταίρω προεκτάσεις κάποιων client οι οποίες αναγκάζουν τις εφαρμογές να εγκαθιδρύσουν μόνο ασφαλείς συνεδρίες.[58]

2.6 Μηχανισμοί ασφάλειας διαδικτυακού πληροφοριακού συστήματος

Μετά την ανάλυση των παραπάνω επιθέσεων είμαστε πλέον σε θέση να προσδιορίσουμε ποιους μηχανισμούς θα πρέπει να χρησιμοποιήσουμε ώστε να πετύχουμε την μέγιστη ασφάλεια στο διαδικτυακό σύστημα ψηφοφορίας.

Θα πρέπει λοιπόν αρχικά κατά την πρώτη επικοινωνία του χρήστη με το σύστημα να εγκαθίσταται ασφαλής συνεδρία με ένα πρωτόκολλο ασφαλούς επικοινωνίας. Το πρωτόκολλο αυτό θα πρέπει να είναι όσο πιο αναβαθμισμένο και σύγχρονο είναι δυνατόν ώστε να καλύπτει όλα τα πιθανά κενά ασφαλείας. Για την πιστοποίηση όμως της σωστής και ασφαλούς επικοινωνίας και ώστε να έχουν και οι χρήστες γνώση σχετικά με αυτήν θα πρέπει να έχουμε ένα έγκυρο πιστοποιητικό ασφαλείας από κάποια γνωστή και αναγνωρισμένη αρχή πιστοποίησης. Μόνο τότε οι χρήστες θα είναι σίγουροι πως ανταλλάσσουν δεδομένα αποκλειστικά με τον εξυπηρετητή μας μέσα σε ένα κέλυφος ασφάλειας. Από την στιγμή που θα έχουμε επιτύχει την κρυπτογραφημένη ανταλλαγή δεδομένων θα πρέπει να διαφυλάξουμε πως κανείς δεν είναι σε θέση να μπει ενδιάμεσα και να παραπλανήσει τους χρήστες. Αυτό όπως είπαμε και παραπάνω δεν εξαρτάται αποκλειστικά από εμάς αλλά και από το δίκτυο του χρήστη. Οι χρήστες θα πρέπει να έχουν ενημερωθεί πως θα είναι σωστό να συνδέονται στο σύστημα μέσω ενός ασφαλούς ιδιωτικού δικτύου και όχι μέσω ενός δημοσίου. Επίσης εμείς από πλευράς μας θα πρέπει να χρησιμοποιούμε τον μηχανισμό HSTS ο οποίος αποκλείει κάθε πιθανότατα μη κρυπτογραφημένης σύνδεσης με τον εξυπηρετητή.

Πέρα από το θέμα της κρυπτογράφησης το σύστημα θα πρέπει να είναι σε θέση να ελέγχει τα δεδομένα που δέχεται και να μην επιτρέπει να εισαχθούν σε αυτό μη ασφαλείς χαρακτήρες που μπορεί να προκαλέσουν είτε την δυσλειτουργία του, είτε την υποκλοπή είτε την παραποίηση των δεδομένων. Σε αυτό το σημείο οι χρήστες και πάλι θα πρέπει να είναι ενημερωμένοι ώστε πριν συνδεθούν στο σύστημα να έχουν αναβαθμίσει τους web clients και τα αντιϊκά προγράμματα που χρησιμοποιούν ώστε να μην υπάρχει κίνδυνος επίθεσης στο δικό τους Η/Υ.

Από την στιγμή που έχουμε εξασφαλίσει και την εισαγωγή των δεδομένων στο διαδικτυακό σύστημα θα πρέπει να εξασφαλίσουμε και την ορθή ροή των διαδικασιών που εκτελούνται σε αυτό αλλά και την σωστή εξουσιοδότηση που δίνεται στην κάθε ομάδα χρηστών. Οι φόρμες σύνδεσης σε περίπτωση λάθους δεν θα πρέπει να εμφανίζουν στοιχεία σχετικά με το πιο πεδίο (όνομα χρήστη ή κωδικός) είναι λάθος ώστε να αποκλείσουμε την εύρεση από κακόβουλο χρήστη ενός εκ των παραπάνω δεδομένων.

Τα sessionid που παράγονται αφού έχουν συνδεθεί οι χρήστες θα πρέπει να αποτελούνται από πολλούς χαρακτήρες ώστε να είναι δύσκολο να βρεθούν και να αποθηκεύονται σε μη προσπελάσιμο σημείο του συστήματος ώστε να μην είναι εύκολο να υποκλαπούν. Τα sessionid επίσης θα πρέπει να λήγουν μετά από κάποιο χρονικό διάστημα και να διαγράφονται την στιγμή που ο χρήστης αποσυνδέεται από το σύστημα.

Κάθε χρήστης που εκτελεί οποιαδήποτε διεργασία στο σύστημα θα πρέπει να ελέγχεται ως προς το δικαίωμα αυτό πριν την εκτελέσει και αν δεν έχει πρόσβαση να αποσυνδέεται από το σύστημα. Υποχρεωτική αποσύνδεση από το σύστημα θα πρέπει να γίνεται και εάν ο χρήστης προσπαθήσει να εισαγάγει δεδομένα που έχει προβλεφτεί πως μπορεί να το βλάψουν.

Οι ψηφοφόροι και τα προσωπικά τους στοιχεία θα πρέπει να έχουν πιστοποιηθεί από μια αρχή πιστοποίησης πριν μπορέσουν να εγγραφούν στο διαδικτυακό σύστημα. Τα στοιχεία σύνδεσής τους αφού έχουν παραχθεί αυτόματα θα αποστέλλονται μέσω email στον προσωπικό τους λογαριασμό τον κωδικό σύνδεσης του οποίου γνωρίζουν μόνο αυτοί . Από την στιγμή της πρώτης τους σύνδεσης θα πρέπει να μπορούν να αλλάξουν τον συγκεκριμένο κωδικό ώστε πλέον να μην τον γνωρίζει κανείς.

Οι χρήστες που ανήκουν στην ομάδα της αρχής πιστοποίησης δεν θα μπορούν να επεξεργαστούν τα στοιχεία των ψηφοφόρων από την στιγμή που αυτοί έχουν κάνει την εγγραφή τους στο σύστημα.

Με βάση τους παραπάνω μηχανισμούς ασφάλειας στο επόμενο κεφάλαιο, το κεφάλαιο σχεδιασμού, θα εξηγήσουμε αναλυτικότερα το πώς σχεδιάζουμε αυτούς τους μηχανισμούς για να χρησιμοποιηθούν στο διαδικτυακό σύστημα ψηφοφορίας. Πριν από αυτό μιας και όπως έχουμε δει παραπάνω είναι πολύ σημαντική η κρυπτογραφία των δεδομένων του συστήματος θα γίνει μια γενική αναφορά στα σύγχρονα πρωτόκολλα ασφαλείας, τα πιστοποιητικά και την κρυπτογραφία που μπορούμε να χρησιμοποιήσουμε.

2.6.1 Πρωτόκολλα ασφαλείας

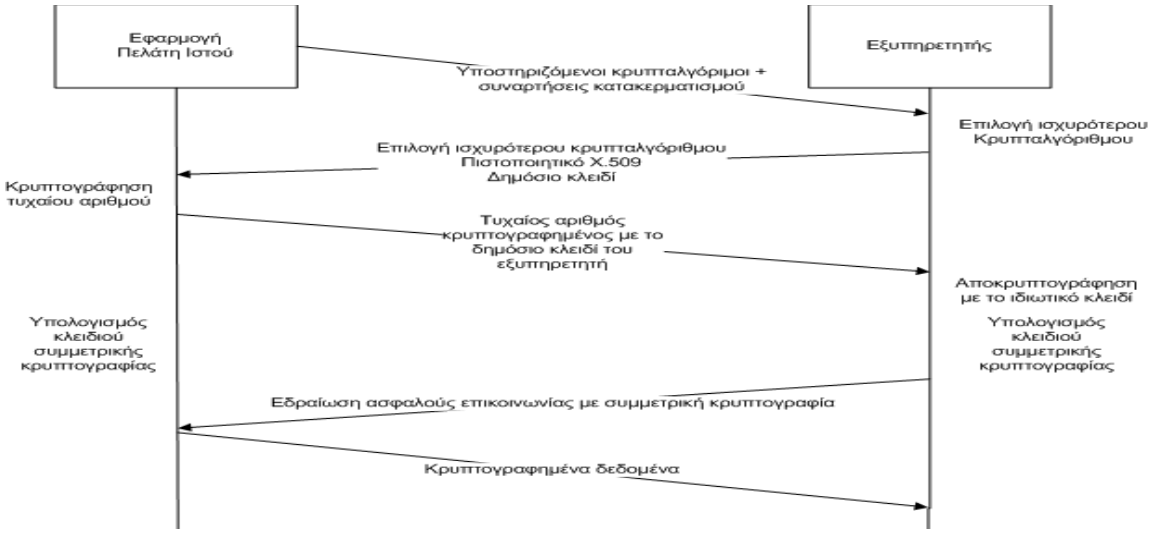
Δύο βασικά πρωτόκολλα ασφαλείας είναι το TSL και ο πρόγονός του το SSL. Το SSL πλέον έχει κριθεί ανασφαλές και έτσι χρησιμοποιείται ευρέως το TLS.

Το TLS είναι ένα κρυπτογραφικό πρωτόκολλο που έχει σαν στόχους την διαφύλαξη της ιδιωτικότητας και την ακεραιότητας των δεδομένων κατά την επικοινωνία ανάμεσα σε έναν χρήστη και σε μια εφαρμογή εξυπηρετητή. Το TLS χρησιμοποιείται ως ενδιάμεσο πρωτόκολλο μεταξύ του επιπέδου εφαρμογών και του επιπέδου μεταφοράς. Χρησιμοποιεί ασύμμετρη κρυπτογραφία για ανταλλαγή κλειδιών, συμμετρική κρυπτογραφία για ιδιωτικότητα και κώδικες επαλήθευσης αυθεντικοποίησης μηνυμάτων (Message

Authentication Codes) για επαλήθευση της ακεραιότητας των δεδομένων. Το TLS είναι πρότυπο του IETF, με τελευταία ανανέωση στο RFC5246 και όπως προαναφέραμε βασίζεται στις προδιαγραφές του παλαιότερου πρωτοκόλλου SSL που αναπτύχτηκε από τη Netscape Communications.

Ο εξυπηρετητής πρέπει να δηλώσει αν υποστηρίζει το TLS. Αυτό μπορεί να γίνει με δύο τρόπους. Ο πρώτος είναι να διατεθεί μια ξεχωριστή πόρτα επικοινωνίας που θα υποστηρίζει χρήση του TLS. Για παράδειγμα το HTTPS που είναι HTTP πάνω από TLS, συνήθως χρησιμοποιεί την πόρτα 443 αντί για την πόρτα 80 που χρησιμοποιεί παραδοσιακά το απλό HTTP.Ο δεύτερος τρόπος είναι να χρησιμοποιηθεί η ίδια πόρτα με την υπηρεσία χωρίς TLS αλλά να δοθεί η δυνατότητα στην web client εφαρμογή του χρήστη να μετατρέψει τη σύνδεση σε TLS στέλνοντας μια ειδική εντολή στα πλαίσια του αρχικού πρωτοκόλλου.

Όταν ο χρήστης και ο εξυπηρετητής αποφασίσουν να ξεκινήσουν μια σύνδεση TLS, καταρχήν διαπραγματεύονται τις παραμέτρους της επικοινωνίας μέσα από μια διαδικασία χειραψίας (TLS Handshake) [60].



Σχήμα 1 .Χειραψία SSL/TLS

1. Η χειραψία ξεκινά με τον χρήστη να ζητάει μια ασφαλή σύνδεση, στέλνοντας τη λίστα με τις σουίτες κρυπταλγορίθμων (cipher suite = κρυπταλγόριθμος + συνάρτηση κατακερματισμού) που υποστηρίζει.
2. Ο εξυπηρετητής επιλέγει από τη λίστα αυτή το ισχυρότερο από τα cipher suites που υποστηρίζει ο ίδιος και ενημερώνει την εφαρμογή του χρήστη για την απόφαση.
3. Ο εξυπηρετητής στέλνει την ταυτότητά του στη μορφή ενός ψηφιακού πιστοποιητικού X.509. Το πιστοποιητικό περιέχει το όνομα του εξυπηρετητή, την αρχή πιστοποίησης και το δημόσιο κλειδί του εξυπηρετητή.
4. Ο χρήστης επαληθεύει την εγκυρότητα του πιστοποιητικού, ενδεχομένως επικοινωνώντας με την αρχή πιστοποίησης.
5. Προκειμένου να παραχθούν τα κλειδιά της συνεδρίας (session keys), η εφαρμογή του χρήστη κρυπτογραφεί έναν τυχαίο αριθμό με το δημόσιο κλειδί του εξυπηρετητή και στέλνει το αποτέλεσμα. Ο εξυπηρετητής είναι ο

μόνος που μπορεί να αποκρυπτογραφήσει το μήνυμα, με τη βοήθεια του ιδιωτικού κλειδιού του.

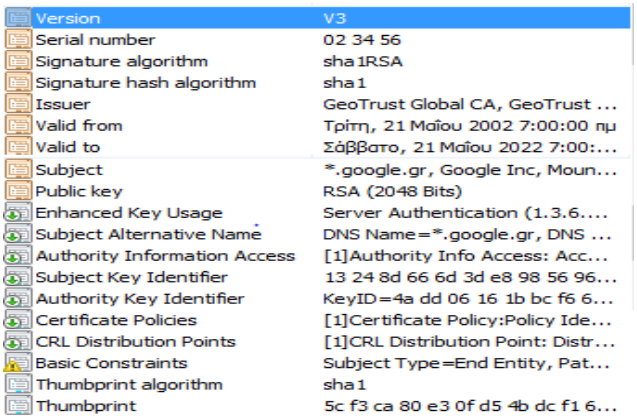
6. Με βάση τον αριθμό αυτό, τόσο η εφαρμογή του χρήστη όσο και ο εξυπηρετητής παράγουν τα κλειδιά συνεδρίας για την κρυπτογράφηση και την αποκρυπτογράφηση του περιεχομένου.
7. Μετά από αυτά τα βήματα ξεκινά η ασφαλής σύνδεση, τα περιεχόμενα της οποίας κρυπτογραφούνται μέχρι το τέλος της.

Αν αποτύχει κάποιο από τα παραπάνω βήματα η χειραψία αποτυγχάνει και δεν ξεκινά ασφαλής σύνδεση.

Το SSL είχε τρεις εκδόσεις τις 1.0, 2.0,και 3,0. Από την έκδοση αυτή και μετά, τον Ιανουάριο του 2009, ορίστηκε το πρωτόκολλο TLS με την πρώτη έκδοση την 1.0. Εκείνη η έκδοση υποστήριζε υποβάθμιση πρωτοκόλλου επικοινωνίας σε SSL3.0. Μετά από αλλαγές στο TLS1.0 ορίστηκε τον Απρίλη του 2006 το TLS 1.1 και τον Αύγουστο του 2008 το TLS 2.2 με σημαντικές αλλαγές στην ασφάλεια όπως π.χ. ότι ο αλγόριθμος MD5-SHA-1 που χρησιμοποιούνταν και για τον οποίο είχαν αναφερθεί προβλήματα ευπάθειας αντικαταστάθηκε από τον SHA-256 ο οποίος θεωρείται ασφαλέστερος. Άλλη μια βασική αλλαγή η οποία προστέθηκε το 2011 στο TLS 1.2 είναι πως πλέον δεν υποστηρίζει συμβατότητα υποβάθμισης σε SSL2.0 το οποίο και έχει κριθεί ανασφαλές [59].

2.6.2 Πιστοποιητικά

Το SSL/TLS κάνει εκτεταμένη χρήση των πιστοποιητικών δημοσίου κλειδιού για την απόδειξη γνησιότητας τόσο του client όσο και του server στις SSL/TLS συνεδρίες. Το SSL/TLS κάνει χρήση των πιστοποιητικών X.509 v3 για τον έλεγχο του ζεύγους κλειδιών RSA, και ένα τροποποιημένο X.509 πιστοποιητικό για τον έλεγχο δημόσιων κλειδιών που χρησιμοποιούνται από το πρωτόκολλο ανταλλαγής κλειδιών U.S. Department of Defense Forteza/DMS. Το X.509 v3 πιστοποιητικό είναι ένα δημοφιλές πρότυπο για τα πιστοποιητικά δημόσιου κλειδιού. Το κάθε πιστοποιητικό X.509 περιλαμβάνει έναν αριθμό έκδοσης, έναν σειριακό αριθμό, πληροφορίες ταυτότητας, πληροφορίες σχετικές με τον αλγόριθμο και την υπογραφή της αρχής που το εκδίδει καθώς και πληροφορίες σχετικά με την ημερομηνία έκδοσης και λήξης του [61].



Εικόνα 1 .Το πιστοποιητικό της google.

Τα πιστοποιητικά τα δημιουργεί ο διαχειριστής ενός ιστότοπου ο οποίος θέλει να χρησιμοποιεί ασφαλή επικοινωνία και τα πιστοποιεί μια εγκεκριμένη αρχή πιστοποίησης η οποία και αναλαμβάνει την ευθύνη για την εγκυρότητα του ιστότοπου και των περιεχομένων του. Μέσω αυτής της πιστοποίησης μια web client εφαρμογή αλλά και οι χρήστες της μπορούν να γνωρίζουν πως επικοινωνούν πραγματικά με τον εξυπηρετητή της εφαρμογής για την οποία έχουν κάνει αίτηση.

Όπως παρατηρούμε στην εικόνα όπου και παρουσιάζεται το πιστοποιητικό της google φαίνονται ο σειριακός κωδικός του , ο αλγόριθμος κρυπτογράφησης που χρησιμοποιεί, ο αλγόριθμος κατακερματισμού που χρησιμοποιεί, η αρχή πιστοποίησης, οι ημερομηνίες έκδοσης και λήξης ,το δημόσιο κλειδί και το μέγεθός του και πληροφορίες σχετικά με το dns του ιστοτόπου αλλά και σχετικά με την αρχή πιστοποίησης.

2.6.3 Κρυπτογραφία

Κρυπτογράφηση είναι η διεργασία με την οποία ένα μήνυμα (που ονομάζεται plaintext) μετατρέπεται σε ένα άλλο μήνυμα (που ονομάζεται ciphertext) χρησιμοποιώντας μια μαθηματική συνάρτηση (αλγόριθμος κρυπτογράφησης) και ένα ειδικό κωδικό κρυπτογράφησης, που ονομάζεται κλειδί.

Αποκρυπτογράφηση είναι η αντίστροφη διεργασία: το ciphertext μετατρέπεται στο αρχικό κείμενο (plaintext) χρησιμοποιώντας μια άλλη μαθηματική συνάρτηση και ένα άλλο κλειδί.

Υπάρχουν δύο βασικά είδη κρυπτογραφίας σε χρήση μέχρι σήμερα· η συμμετρική και η ασύμμετρη.

Συμμετρική

Κατά τη συμμετρική κρυπτογραφία διανέμεται με μυστικό τρόπο το ίδιο κλειδί σε όλες τις οντότητες που πρόκειται να συμμετάσχουν στην διαδικασία κρυπτογράφησης και αποκρυπτογράφησης. Με αυτόν τον τρόπο η πρώτη οντότητα κρυπτογραφεί με ένα κλειδί που γνωρίζουν και οι δύο τα δεδομένα που θέλει , τα αποστέλλει και η δεύτερη οντότητα αφού τα λάβει τα αποκρυπτογραφεί με το ίδιο κλειδί.

Σχήμα 2. Συμμετρική κρυπτογραφία.

Κάποιοι από τους πιο γνωστούς αλγόριθμους συμμετρικής κρυπτογραφίας είναι ο DES (μη ασφαλής πλέον), ο AES ο οποίος αντικαθιστά τον DES, ο IDEA στον οποίο απαιτούνται κατά την κρυπτογράφηση 8 σύνθετες επαναλήψεις, χρησιμοποιεί κλειδιά 128 bits αλλά χρησιμοποιείται κυρίως σε hardware και οι RC2,RC4,RC5. Ο RC2 είναι 3-5 φορές πιο γρήγορος από τον DES και το κλειδί του έχει μεταβλητό μήκος. Ο RC4 είναι ο αλγόριθμος που χρησιμοποιείται στο πρωτόκολλο SSL/TLS έχει και έχει και αυτός μεταβλητό μήκος κλειδιού. Τέλος ο RC5 μπορεί να χρησιμοποιήσει έως και 255 επαναλήψεις έχει μέγεθος Block ίσο με 32/64/128 bits και το κλειδί του έχει μεταβλητό μήκος [62][63].

Ασύμμετρη

Η ασύμμετρη κρυπτογραφία ονομάζεται και κρυπτογραφία δημοσίου κλειδιού. Σε αυτήν την μορφή κρυπτογραφίας έχουμε δύο διαφορετικά κλειδιά τα οποία όμως είναι μαθηματικά συσχετιζόμενα μεταξύ τους. Αν χρησιμοποιηθεί το ένα κλειδί δεν μπορεί να γίνει γνωστό το άλλο. Για τον λόγο θεωρείται πολύ ισχυρή μορφή κρυπτογραφίας και είναι αυτή που χρησιμοποιείται στο διαδίκτυο ώστε να εγκαθιδρυθεί κάθε νέα συνεδρία. Ουσιαστικά η κάθε οντότητα μπορεί να κρυπτογραφεί ένα κείμενο με το δημόσιο κλειδί της άλλης και ενώ όλοι μπορούν να γνωρίζουν αυτό το κλειδί να μπορεί μόνο η οντότητα που έχει και το ιδιωτικό κλειδί να το αποκρυπτογραφήσει. Από την άλλη, αν κάποιος κρυπτογραφήσει δεδομένα με το ιδιωτικό του κλειδί μπορεί κάποιος με το δημόσιο να τα αποκρυπτογραφήσει. Για τον λόγο αυτό η κρυπτογράφηση των ευαίσθητων δεδομένων πρέπει να γίνεται με το δημόσιο και η αποκρυπτογράφηση με το ιδιωτικό.

Σχήμα 3. Ασύμμετρη κρυπτογραφία

Κάποιοι από τους πιο γνωστούς αλγόριθμους ασύμμετρης κρυπτογραφίας είναι ο RSA και ο DSA. Ο DSA είναι πιο γρήγορος στην παραγωγή των υπογραφών από τον RSA αλλά υστερεί διότι παρουσιάζει έλλειψη ευελιξίας, αργή επαλήθευση κωδικών και αδυναμία συνεργασίας με άλλα πρωτόκολλα πιστοποίησης ταυτότητας. Ο RSA από την άλλη είναι πιο ευέλικτος πιο γρήγορος στην επαλήθευση υπογραφών και μπορεί να συνεργαστεί και με άλλα πρωτόκολλα πιστοποίησης ταυτότητας. Το πρωτόκολλο TLS για την αποστολή του κλειδιού συμμετρικής κρυπτογραφίας κατά την εγκαθίδρυση της ασφαλούς επικοινωνίας χρησιμοποιεί τον αλγόριθμο RSA.

Έστω ότι έχουμε m το μήνυμα που θέλουμε να κρυπτογραφήσουμε, c το κρυπτογραφημένο μήνυμα n το δημόσιο κλειδί και d το ιδιωτικό κλειδί. Αν χρησιμοποιήσουμε τις παρακάτω εξισώσεις μπορούμε να κρυπτογραφήσουμε και να αποκρυπτογραφήσουμε ένα μήνυμα:

$$m = c^d mod\, n \qquad c = m^e mod\, n$$

Με αυτόν τον τρόπο λειτουργεί ο αλγόριθμος RSA και με αυτόν τον τρόπο ο εξυπηρετητής εγκαθιστά ασφαλή επικοινωνία με τους χρήστες κατά την διαδικασία TLS. Στέλνει αρχικά το δημόσιο κλειδί του το οποίο βρίσκεται μέσα στο εγκεκριμένο από την αρχή πιστοποίησης πιστοποιητικό του. Ο χρήστης πλέον έχει το δημόσιο κλειδί και με αυτό κρυπτογραφεί έναν τυχαίο αριθμό και τον στέλνει πίσω στον εξυπηρετητή. Αυτός είναι ο μόνος που μπορεί να αποκρυπτογραφήσει το μήνυμα αυτό με το ιδιωτικό του κλειδί. Πλέον, μόνο ο χρήστης και εξυπηρετητής γνωρίζουν αυτόν τον αριθμό και βάσει αυτού

δημιουργούν το κλειδί συμμετρικής κρυπτογραφίας με την οποία πλέον εκτελείται η ασφαλής συνεδρία καθώς είναι πολύ πιο γρήγορη από την ασύμμετρη [64][65].

2.7 Τεχνολογίες υλοποίησης διαδικτυακών εφαρμογών

Σε αυτήν την ενότητα θα παρουσιάσουμε τις τεχνολογίες διαδικτυακών εφαρμογών τις οποίες κρίνουμε κατάλληλο να χρησιμοποιηθούν. Η γλώσσα Java, η οποία είναι η γλώσσα προγραμματισμού, που θα χρησιμοποιηθεί έχει αποδείξει ότι μπορεί να παρέχει στους προγραμματιστές ενός πληροφοριακού συστήματος μεγάλη ευελιξία, προσαρμοστικότητα, ταχύτητα στον προγραμματισμό αλλά και στην εκτέλεση της εφαρμογής καθώς και μέγιστο επίπεδο ασφάλειας. Όπως θα αναλυθεί και παρακάτω η αρχιτεκτονική που θεωρούμε καλύτερο να χρησιμοποιηθεί είναι η MVC (model-view-controller) καθώς καλύπτει όλες τις παραπάνω απαιτήσεις του συστήματος και από πλευράς ασφάλειας αλλά και ευελιξίας και χρηστικότητας.

Για την επίτευξη της συγκεκριμένης αρχιτεκτονικής θα χρειαστεί να χρησιμοποιήσουμε Java Servlets, Javabeans, JSP, Κώδικα Javascript, κώδικα css και την τεχνολογία Ajax.

2.7.1 JavaBeans

Τα JavaBeans είναι επαναχρησιμοποιήσιμα στοιχεία λογισμικού της γλώσσας Java. Πρόκειται για κλάσεις που έχουν συνταχθεί στη γλώσσα προγραμματισμού Java σύμφωνα με μια συγκεκριμένη σύμβαση. Χρησιμοποιούνται για τη συμπύκνωση πολλών αντικειμένων σε ένα μεμονωμένο αντικείμενο, ώστε να μπορούν να περαστούν ως ένα αντικείμενο (bean) αντί για πολλά μεμονωμένα αντικείμενα. Ένα JavaBean είναι ένα αντικείμενο Java που μπορεί να σειριοποιηθεί, έχει ένα nullary constructor, και επιτρέπει πρόσβαση σε ιδιότητες, μεθόδους getter και μεθόδους setter [66].

2.7.2 Servlets

Τα Servlets είναι αντικείμενα της γλώσσας προγραμματισμού Java που δυναμικά επεξεργάζονται αιτήσεις και δημιουργούν απαντήσεις. Το Servlet Java API επιτρέπει στον προγραμματιστή λογισμικού να προσθέσει δυναμικό περιεχόμενο σε ένα διακομιστή Web χρησιμοποιώντας την πλατφόρμα Java. Το περιεχόμενο που δημιουργείται είναι συνήθως HTML, αλλά μπορεί να είναι και σε άλλες μορφές, όπως XML. Τα Servlets είναι το αντίστοιχο των μη-Java τεχνολογιών δυναμικού περιεχομένου Web όπως ASP.NET, PHP, CGI. Τα Servlets μπορούν να διατηρούν την κατάσταση σε πολλές συναλλαγές διακομιστή, χρησιμοποιώντας HTTP cookies, μεταβλητές περιόδου λειτουργίας (session variables) ή επανεγγραφή URL (URL rewriting).

Το servlet API, που περιέχεται ιεραρχικά στο Java πακέτο javax.servlet, ορίζει τις αναμενόμενες επικοινωνίες ενός Web container και ενός servlet. Το Web container είναι ουσιαστικά το στοιχείο ενός διακομιστή Web που αλληλεπιδρά με τα servlets. Το Web container είναι υπεύθυνο για τη διαχείριση του κύκλου ζωής των servlets, την αντιστοίχηση μιας διεύθυνσης URL σε ένα συγκεκριμένο servlet και του ότι αυτός που ζητά το URL έχει τα σωστά δικαιώματα πρόσβασης.

Ένα Servlet είναι ένα αντικείμενο που λαμβάνει μια αίτηση και δημιουργεί μια απόκριση που βασίζεται σε αυτό το αίτημα. Το βασικό πακέτο servlet ορίζει αντικείμενα Java για την αναπαράσταση αιτήσεων και αποκρίσεων servlet. Επίσης ορίζει τα αντικείμενα να αντικατοπτρίζουν τις παραμέτρους του servlet στο περιβάλλον εκτέλεσης. Το πακέτο javax.servlet.http ορίζει εξειδικευμένες υποκλάσεις για το HTTP από τα στοιχεία γενικής χρήσης servlet, συμπεριλαμβανομένων των αντικειμένων διαχείρισης περιόδου λειτουργίας που παρακολουθούν πολλαπλές αιτήσεις και αποκρίσεις μεταξύ του διακομιστή Web και έναν υπολογιστή-πελάτη. Τα Servlets μπορούν να συσκευαστούν σε ένα αρχείο WAR ως μια εφαρμογή Web.

Τα Servlets μπορούν να δημιουργηθούν αυτόματα από το μεταγλωττιστή των JavaServer Pages (JSP), ή, εναλλακτικά, με γεννήτριες προτύπων όπως με την WebMacro. Τα Servlets συχνά χρησιμοποιούνται σε συνδυασμό με τα JSPs σε ένα μοτίβο που ονομάζεται "Μοντέλο 2" (Model 2), δηλαδή μια γεύση της αρχιτεκτονικής MVC [67].

2.7.3 JSP

Οι JavaServer Pages (JSP) είναι μια τεχνολογία Java που επιτρέπει στους προγραμματιστές λογισμικού να δημιουργούν δυναμικά ιστοσελίδες, σε HTML, XML ή άλλους τύπους εγγράφων, απαντώντας σε μια διαδικτυακή αίτηση υπολογιστή-πελάτη. Οι σελίδες JSP είναι φορτωμένες στο διακομιστή και λειτουργούν από ένα δομημένο ειδικά εγκατεστημένο πακέτο διακομιστή Java που ονομάζεται J2EE εφαρμογής Web και συχνά είναι συσκευασμένο ως ένα αρχείο, το αρχείο .war ή .ear. Η τεχνολογία JSP επιτρέπει την ενσωμάτωση κώδικα Java και ορισμένων προκαθορισμένων ενεργειών σε μια στατική σελίδα περιεχομένου, η οποία μεταγλωττίζεται στο διακομιστή κατά το χρόνο εκτέλεσης [68].

2.7.4 JavaScript

Η JavaScript είναι μια αντικειμενοστρεφής γλώσσα σεναρίων η οποία χρησιμοποιείται ώστε να έχουμε πρόσβαση σε αντικείμενα της εφαρμογής που εκτελείται στον πελάτη (client). Εφαρμόζεται ως μια ολοκληρωμένη συνιστώσα του browser, επιτρέποντας την ανάπτυξη ενισχυμένων διεπαφών χρήστη και δυναμικών ιστοσελίδων. Στην εφαρμογή θα μπορούσαμε να χρησιμοποιήσουμε Javascript κυρίως για τον έλεγχο των πεδίων των φορμών εισαγωγής στοιχείων αλλά και για την χρησιμοποίηση της τεχνολογίας Ajax [69].

2.7.5 Cascading Style Sheets (CSS)

Τα CSS χρησιμοποιούνται για να περιγράψουν την εμφάνιση και τη μορφοποίηση ενός εγγράφου που έχει συνταχθεί σε μια γλώσσα σήμανσης (markup language). Η πιο κοινή εφαρμογή είναι σε στυλ σελίδων γραμμένων σε HTML και XHTML, αλλά η γλώσσα μπορεί να εφαρμοστεί σε κάθε είδους εγγράφο XML, περιλαμβανομένων των SVG και XUL.

Τα CSS έχουν σχεδιαστεί κυρίως για να επιτρέψουν το διαχωρισμό του περιεχομένου του εγγράφου (γραμμένο σε HTML ή σε μια παρόμοια γλώσσα σήμανσης) από την παρουσίαση του εγγράφου, συμπεριλαμβάνοντας τη διάταξη, τα χρώματα και τις γραμματοσειρές. Ο διαχωρισμός αυτός μπορεί να βελτιώσει την προσβασιμότητα του περιεχομένου, παρέχει

μεγαλύτερη ευελιξία και έλεγχο των χαρακτηριστικών της παρουσίασης, επιτρέπει να γίνεται κοινή χρήση της μορφοποίησης σε πολλές σελίδες και μειώνει την πολυπλοκότητα και την επανάληψη στο διαρθρωτικό περιεχόμενο (επιτρέποντας σχεδίαση στον ιστό χωρίς πίνακες). Τα CSS μπορούν επίσης να επιτρέψουν στην ίδια σελίδα να παρουσιάζεται με διαφορετικά στυλ για διαφορετικές μεθόδους απόδοσης, όπως στην οθόνη, στην εκτύπωση, με φωνή (κατά την ανάγνωση από ένα πρόγραμμα περιήγησης που βασίζεται σε ομιλία ή αναγνώστη οθόνης) και στις συσκευές που βασίζονται σε σύστημα Braille. Ενώ ο συντάκτης ενός εγγράφου συνδέει συνήθως αυτό το έγγραφο σε ένα φύλλο στυλ CSS, οι αναγνώστες μπορούν να χρησιμοποιήσουν ένα διαφορετικό φύλλο στυλ και να παρακάμψουν αυτό που έχει ορίσει ο συντάκτης.

Τα CSS καθορίζουν ένα συνδυασμό προτεραιότητας για να προσδιοριστούν ποιοι κανόνες στυλ ισχύουν σε ένα συγκεκριμένο στοιχείο στις περιπτώσεις όπου αντιστοιχούν περισσότεροι του ενός κανόνα σε αυτό. Προτεραιότητες ή βάρη υπολογίζονται και εκχωρούνται σε κανόνες, έτσι ώστε τα αποτελέσματα να είναι προβλέψιμα. Οι προδιαγραφές CSS διατηρούνται από το World Wide Web Consortium (W3C) [70].

2.7.6 AJAX

Τα αρχικά AJAX σημαίνουν Asynchronous JavaScript And XML. Η τεχνολογία AJAX δεν είναι μια νέα γλώσσα προγραμματισμού, αλλά μία νέα τεχνική για τη δημιουργία πιο γρήγορων και πιο φιλικών για το χρήστη διαδικτυακών εφαρμογών. Η τεχνολογία AJAX χρησιμοποιεί JavaScript για την αποστολή και λήψη δεδομένων μεταξύ ενός φυλλομετρητή (web browser) και τον διακομιστή (web server). Η τεχνική AJAX κάνει της ιστοσελίδες πιο διαδραστικές επιτρέποντας την αποστολή δεδομένων στο παρασκήνιο χωρίς να χρειάζεται η μεταφόρτωση της ιστοσελίδας κάθε φορά που ο χρήστης αλληλεπιδρά με την σελίδα.

Η τεχνολογία AJAX χρησιμοποιεί ασύγχρονη μεταφορά δεδομένων (HTTP κλήσεις) μεταξύ του φυλλομετρητή και του κεντρικού διακομιστή, επιτρέποντας στις σελίδες web να ζητούν μικρές πληροφορίες από τον εξυπηρετητή αντί για πλήρεις σελίδες. Η τεχνολογία AJAX είναι κυρίως τεχνολογία που υποστηρίζεται στο φυλλομετρητή (web browser) και όχι τεχνολογία του εξυπηρετητή (web server). Η τεχνολογία AJAX βασίζεται σε ανοικτά πρότυπα και αυτό την κάνει πολύ εύχρηστη και ελκυστική για εταιρείες ανάπτυξης λογισμικού. Μερικές από τις τεχνολογίες που ενσωματώνει ο AJAX είναι οι JavaScript, XML, HTML και CSS.

Τα πρότυπα που χρησιμοποιούνται από τη τεχνολογία AJAX είναι πλήρως καθορισμένα, και υποστηρίζονται πλήρως από τους πιο γνωστούς φυλλομετρητές. Οι εφαρμογές AJAX είναι επίσης γνωστές ως Cross-Platform και Cross-Browser δηλαδή τεχνολογίες που τρέχουν σε όλες τις πλατφόρμες (λειτουργικά συστήματα) και σε όλους τους φυλλομετρητές. Τέλος, η ανάκτηση δεδομένων γίνεται συνήθως χρησιμοποιώντας το αντικείμενο XMLHttpRequest. Παρά το όνομα, η χρήση JavaScript και XML δεν απαιτείται, ούτε οι αιτήσεις πρέπει να είναι ασύγχρονες [69].

2.8 Αρχιτεκτονική MVC

Η χρήση των τεχνολογιών JavaBean, Servlet και JSP είναι ορθότερο να χρησιμοποιηθούν με βάση την αρχιτεκτονική MVC (Model – View – Controller). Η αντιστοιχία τους είναι η εξής: [71]

Model	→	JavaBean
Controller	→	Servlet
View	→	JSP

Η MVC αρχιτεκτονική συνδυάζει τη χρήση των servlets και JSP. Εδώ, το servlet έχει το ρόλο του controller που είναι επιφορτισμένος με την επεξεργασία των αιτήσεων και τη δημιουργία όλων των JavaBeans που χρησιμοποιούνται από τη σελίδα JSP, καθώς και για την επιλογή της κατάλληλης σελίδας. Στην ίδια την σελίδα JSP δεν γίνεται κάποια επεξεργασία. Η μόνη υπευθυνότητά της είναι να ανακτήσει τα στοιχεία από το JavaBeans που έχει δημιουργήσει το servlet.

Model: αντιπροσωπεύει τα στοιχεία (data) της εφαρμογής. Συνήθως το model είναι ένα ή περισσότερα JavaBeans

View: είναι υπεύθυνη για την παρουσίαση στοιχείων. Συνήθως είναι μια ή περισσότερες σελίδες JSP.

Controller: υπεύθυνος για τη μετάφραση των ενεργειών του χρήστη, που πραγματοποιούνται στη συνιστώσα View, σε μεταβολές της συνιστώσας Model. Συνήθως είναι ένα ή περισσότερα servlets.

2.8.1 Πλεονεκτήματα της MVC

Τα κυριότερα πλεονεκτήματα της MVC είναι τα παρακάτω:

Ευκολότερη συντήρηση των sites / web applications: η υλοποίηση μίας αλλαγής καθώς και ο εντοπισμός ενός προβλήματος γίνονται πιο εύκολα/γρήγορα.

Διευκόλυνση της ομαδικής δουλειάς: με το διαχωρισμό των περιοχών μπορούν να δουλέψουν παράλληλα περισσότεροι του ενός προγραμματιστές πάνω στο ίδιο έργο.

Καλύτερη ποιότητα των sites / web applications: με τη χρήση του MVC μπορεί να γίνει πιο εύκολα το λεγόμενο Test-Driven Development (TDD), μια και οι προγραμματιστές μπορούν να ενσωματώσουν tests νωρίτερα κατά την ανάπτυξη βελτιώνοντας έτσι το τελικό αποτέλεσμα.

Μεγαλύτερη ασφάλεια: καθώς οι μεταβλητές που προωθούνται ανάμεσα στα διάφορα μέρη της αρχιτεκτονικής μπορούν ελεγχθούν καλύτερα και αναλόγως να απορρίψουν ή όχι μια αίτηση. Λόγω της τμηματοποίησης του κώδικα στα τρία παραπάνω μέρη επιτυγχάνουμε ασφάλεια στην εξουσιοδότηση των χρηστών καθώς μπορούμε να ελέγχουμε συνεχώς αν πρέπει να έχουν πρόσβαση σε συγκεκριμένα δεδομένα, διεργασίες ή σελίδες.

2.9 Χρήστες του συστήματος

Ένα σύστημα διαδικτυακής ψηφοφορίας προσομοιώνει την διαδικασία της τυπικής ψηφοφορίας όπως γίνεται έως τώρα στα εκλογικά κέντρα με την βοήθεια των Η/Υ και του διαδικτύου. Οι ψηφοφόροι πιστοποιούνται με βάση τους εκλογικούς καταλόγους και μετά σε μια προσωπική συνεδρία με την εφαρμογή θα μπορούν ψηφίσουν. Υπάρχουν χρήστες που μπορούν να διαχειριστούν τις ψηφοφορίες και το υπόλοιπο σύστημα, χρήστες που βοηθούν στην πιστοποίηση των ψηφοφόρων και χρήστες που έχουν πρόσβαση στα αποτελέσματα και τα στατιστικά. Στην ενότητα αυτή θα παρουσιαστούν οι διάφοροι χρήστες και οι ρόλοι τους στο σύστημα.

Ψηφοφόρος (ivoter)

Ο βασικός χρήστης του συστήματος είναι ο ψηφοφόρος. Όλοι οι πολίτες που είναι εγγεγραμμένοι στους εκλογικούς καταλόγους και έχουν δικαίωμα ψήφου μπορούν να εγγραφούν στο σύστημα. Μπορούν επίσης να συνδεθούν στο σύστημα και να ψηφίσουν στις ενεργές ψηφοφορίες. Μπορούν να αλλάξουν τον προσωπικό τους κωδικό και να δουν τα αποτελέσματα των ανενεργών ψηφοφοριών. Οι λειτουργίες που μπορεί να εκτελέσει ένας ηλεκτρονικός ψηφοφόρος είναι οι παρακάτω:

- Εγγραφή στο σύστημα. Εφόσον είναι εγγεγραμμένος ως voter.
- Σύνδεση / Αποσύνδεση από το σύστημα
- Μπορεί να ψηφίσει ενεργές ψηφοφορίες
- Μπορεί να δει τα αποτελέσματα των ανενεργών ψηφοφοριών

Αρχή πιστοποίησης (cerauthor)

Οι χρήστες που ανήκουν σε αυτήν την κατηγορία είναι ουσιαστικά οι εργαζόμενοι στις διάφορες αρχές πιστοποίησης οι οποίοι αφού πιστοποιήσουν τους ψηφοφόρους κάνουν την αρχική τους εγγραφή στο σύστημα. Έχουν δικαίωμα να εγγράψουν τους ψηφοφόρους και να τροποποιήσουν τα στοιχεία πιστοποίησης τους αν οι ψηφοφόροι δεν έχουν κάνει την τελική εγγραφή στο σύστημα. Οι λειτουργίες που μπορεί να εκτελέσει ένας υπεύθυνος αρχής πιστοποίησης είναι οι παρακάτω:

- Σύνδεση / Αποσύνδεση από το σύστημα
- Δημιουργία voter
- Αλλαγή στοιχείων ψηφοφόρου που δεν έχει εγγραφεί ακόμα στο σύστημα

Ελεγκτής – Θεατής (auditor)

Οι χρήστες αυτής της κατηγορίας μπορούν να δουν στατιστικά στοιχεία των ψηφοφοριών και τα αποτελέσματα. Δεν έχουν δικαίωμα να κάνουν καμία αλλαγή στο σύστημα. Οι λειτουργίες που μπορεί να εκτελέσει ένας ελεγκτής του συστήματος είναι οι παρακάτω:

- Σύνδεση / Αποσύνδεση από το σύστημα
- Θέαση στατιστικών
- Θέαση αποτελεσμάτων

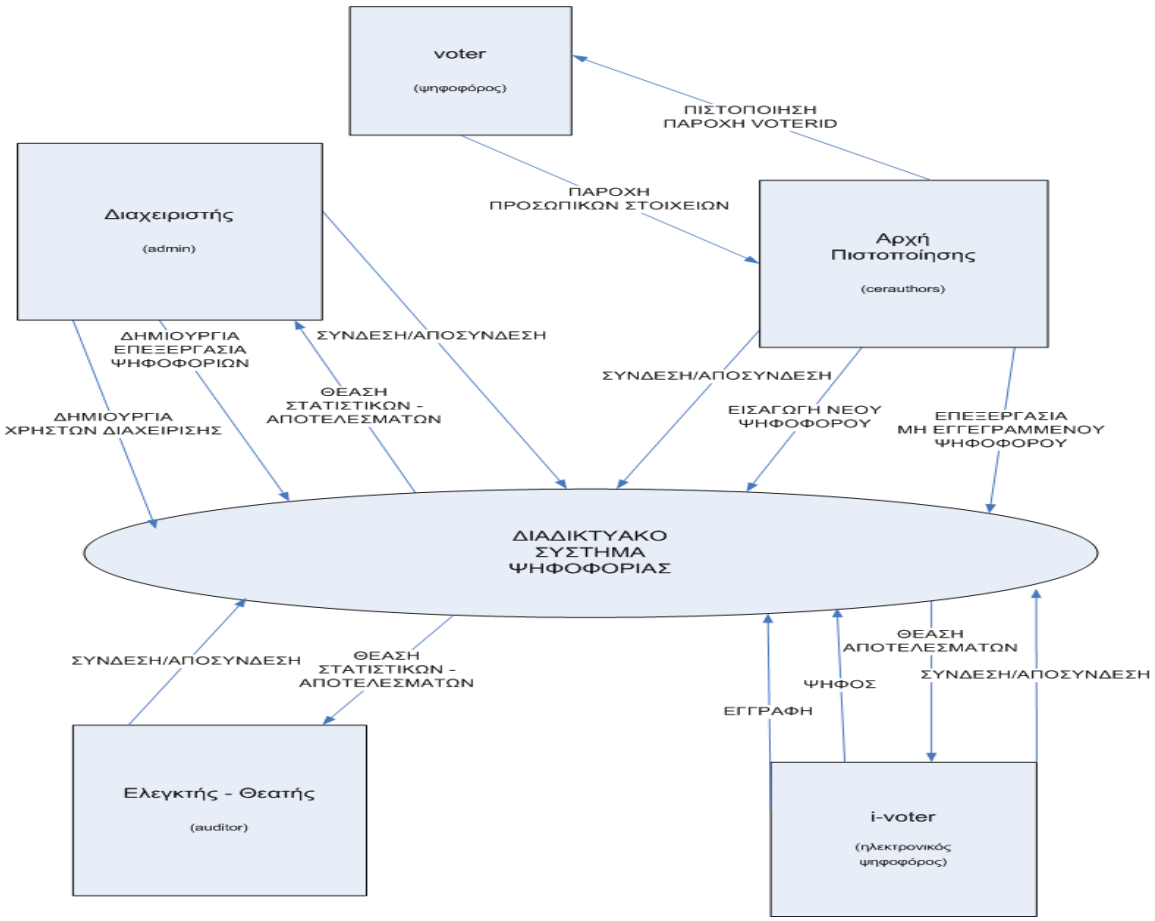
Διαχειριστής (admin)

Οι χρήστες αυτής της κατηγορίας είναι οι διαχειριστές του συστήματος. Μπορούν να δημιουργήσουν cerauthors, auditors και άλλους admins. Μπορούν αν ξεκινήσουν νέα ψηφοφορία και να κλείσουν μια ενεργή. Δεν μπορούν να εγγράψουν νέους ψηφοφόρους ούτε να δουν τα στοιχεία τους. Τέλος μπορούν να βλέπουν τα στατιστικά του συστήματος και τα αποτελέσματα των ψηφοφοριών. Οι λειτουργίες που μπορεί να εκτελέσει ένας διαχειριστής του συστήματος είναι οι παρακάτω:

- Σύνδεση / Αποσύνδεση από το σύστημα
- Δημιουργία νέων ψηφοφοριών
- Κλείσιμο ενεργών ψηφοφοριών
- Δημιουργία νέων χρηστών
- Θέαση στατιστικών
- Θέαση αποτελεσμάτων

Voter

Στο σύστημα χρησιμοποιείται και μια εσωτερική οντότητα οι voters. Είναι οι ivoters αφού έχουν πιστοποιηθεί αλλά πριν εγγραφούν στο σύστημα. Αν ένας ψηφοφόρος δεν έχει γίνει voter δεν μπορεί να γραφτεί σαν ivoter. Πρακτικά ο voter είναι ο πολίτης πριν γραφτεί στο σύστημα διαδικτυακής ψηφοφορίας. Εφόσον υπάρχει στην Β.Δ. μπορεί μετά την εγγραφή (register) του ψηφοφόρου να μετατραπεί σε ivoter.



Σχήμα 4. Αλληλεπίδραση Χρηστών με το Πληροφοριακό Σύστημα

2.10 Διαδικασίες συστήματος

2.10.1 Σύνδεση στο σύστημα

Οι χρήστες τους συστήματος αλληλεπιδρούν με το σύστημα, ξεχωριστά, ο καθένας στην δική του περιοχή. Έχουν κοινές διεργασίες όπως είναι αυτή της σύνδεσης στο σύστημα κατά την οποία γίνεται και η αυθεντικοποίση. Από την στιγμή της σύνδεσης όμως μεταφέρεται ο καθένας στην περιοχή του συστήματος για την οποία έχει εξουσιοδότηση.

Η αυθεντικοποίηση των χρηστών γίνεται με την χρήση username και password τα οποία θα πρέπει να είναι δύσκολο να ανιχνευθούν από κακόβουλους χρήστες.

Επίσης θα πρέπει να διασφαλιστεί πως όλα τα στοιχεία που αποστέλλονται από τον Η/Υ των χρηστών στο σύστημα είναι κρυπτογραφημένα ώστε να μην μπορούν να υποκλαπούν. Αυτό επιτυγχάνεται με την εδραίωση μιας κρυπτογραφημένης συνεδρίας κατά την είσοδο του χρήστη στην σελίδα της εφαρμογής.

Η διαδικασία της εδραίωσης της συγκεκριμένης συνεδρίας έχει ως εξής.

- Το πιστοποιητικό του συστήματος αποστέλλεται στον Η/Υ του χρήστη.
- Αυθεντικοποιείται από την εφαρμογή που χρησιμοποιεί ο χρήστης.
- Δημιουργείται ένα προσωρινό κλειδί το οποίο κρυπτογραφείται μαζί με το δημόσιο κλειδί που έχει αποσταλεί από το i-voting system.
- Το κρυπτογραφημένο πλέον κλειδί αποστέλλεται στο σύστημα το οποίο με την σειρά του το αποκρυπτογραφεί με τη χρήση του ιδιωτικού του κλειδιού.
- Το σύστημα και ο χρήστης δημιουργούν ένα νέο κλειδί το οποίο κρυπτογραφούν και αποστέλλουν μεταξύ τους ένα μήνυμα επιβεβαίωσης.
- Από την στιγμή αυτή έχει εδραιωθεί η κρυπτογραφημένη συνεδρία και ό,τι δεδομένο αποστέλλεται μεταξύ του χρήστη και του Η/Υ είναι κρυπτογραφημένο.

2.10.2 Θέαση αποτελεσμάτων

Άλλη μια κοινή διεργασία είναι η θέαση των αποτελεσμάτων. Και πάλι εδώ όμως ο κάθε χρήστης μπορεί να δει τα αποτελέσματα για τα οποία είναι εξουσιοδοτημένος. Το σύστημα είναι υπεύθυνο να μην επιτρέπει την εμφάνιση αποτελεσμάτων τα οποία δεν πρέπει να δουν συγκεκριμένοι χρήστες.

- Οι ψηφοφόροι μπορούν να δουν τα αποτελέσματα μόνο των ανενεργών ψηφοφοριών. Δηλαδή των ψηφοφοριών οι οποίες έχουν ολοκληρωθεί.
- Οι ελεγκτές και οι διαχειριστές μπορούν να βλέπουν τα αποτελέσματα των ενεργών ψηφοφοριών καθώς και τα στατιστικά τους για να έχουν καλύτερη επίβλεψη του συστήματος.
- Οι χρήστες της αρχής πιστοποίησης δεν μπορούν να δουν στην δική τους περιοχή τα αποτελέσματα ούτε τα στατιστικά των ψηφοφοριών.

2.10.3 Πιστοποίηση νέων ψηφοφόρων

Ο μόνος υπεύθυνος για την πιστοποίηση και την εγγραφή νέων ψηφοφόρων στο σύστημα είναι ο χρήστης της αρχής πιστοποίησης. Κανένας άλλος χρήστης δεν έχει εξουσιοδότηση να εγγράφει χρήστες και να βλέπει τα στοιχεία τους.

Η διαδικασία αυτή έχει ως εξής.

Ο ψηφοφόρος προσέρχεται στην αρμόδια αρχή πιστοποίησης π.χ. σε κάποιο ΚΕΠ με τα προσωπικά του στοιχεία. Ο υπεύθυνος πιστοποίησης ελέγχει τα στοιχεία αυτά σύμφωνα με τους εκλογικούς καταλόγους και αν είναι ορθά ξεκινάει την διαδικασία εγγραφής νέου ψηφοφόρου. Το σύστημα είναι υπεύθυνο να ελέγξει για τυχόν διπλή εγγραφή ψηφοφόρου. Ελέγχεται αν το ΑΦΜ του ψηφοφόρου υπάρχει ήδη στην Β.Δ. Αν δεν υπάρχει η διαδικασία προχωράει.

Σε αυτό το σημείο θα πρέπει να τονίσουμε πως ο υπεύθυνος της αρχής πιστοποίησης θα πρέπει να ελέγχεται νομικά σε περίπτωση παρατυπίας και εισαγωγής ψευδών στοιχείων.

Εάν ολοκληρωθεί η εγγραφή εκτυπώνεται μια καρτέλα με τα στοιχεία του ψηφοφόρου τα οποία και θα πρέπει να συμπληρώσει κατά την εγγραφή του στο σύστημα ως ivoter.Τα στοιχεία που δηλώνει ο ψηφοφόρος και ελέγχονται αν είναι πραγματικά είναι :

- Όνομα
- Επώνυμο
- Όνομα Πατρός
- ΑΦΜ
- Α.Τ.
- Email (χρησιμοποιείται ώστε να αποσταλλούν τα στοιχεία κατά την εγγραφή στο σύστημα)
- Ένας κωδικός ο οποίος θα χρησιμοποιηθεί κατά την εγγραφή στο σύστημα. Δεν θα είναι ο κωδικός σύνδεσης σε αυτό.

Μετά την πιστοποίηση του ψηφοφόρου παράγεται ο Αριθμός Μητρώου ψηφοφόρου ο οποίος θα πρέπει και αυτός να χρησιμοποιηθεί κατά την εγγραφή του στο σύστημα διαδικτυακής ψηφοφορίας.

2.10.4 Αλλαγή στοιχείων ψηφοφόρου

Ο υπεύθυνος πιστοποίησης μπορεί να ανανεώσει τα στοιχεία κάποιου ψηφοφόρου εφόσον ο ψηφοφόρος δεν έχει κάνει εγγραφή-register στο σύστημα. Αν έχει κάνει ο ψηφοφόρος απλώς εμφανίζεται ως ενεργός χρήστης ivoter και απαγορεύεται από την εφαρμογή ο οποιοσδήποτε να αλλάξει τα στοιχεία του.

2.10.5 Εγγραφή - Register στο σύστημα.

Μόνο οι ψηφοφόροι μπορούν να κάνουν εγγραφή στο σύστημα ως ivoters. Για να έχουν αυτό το δικαίωμα θα πρέπει πρώτα να έχουν πιστοποιηθεί από την αρχή πιστοποίησης και να έχουν παραλάβει την καρτέλα ψηφοφόρου με τα στοιχεία τους όπως ακριβώς είναι αποθηκευμένα στην Β.Δ. Τότε και μόνο τότε μπορούν να χρησιμοποιήσουν την διαδικασία εγγραφής.

- Στην καρτέλα εγγραφής θα πρέπει να συμπληρώσουν τα στοιχεία τους και να αποστείλουν μια αίτηση εγγραφής στο σύστημα.
- Για την αποφυγή πολλαπλής αυτοματοποιημένης αποστολής αιτήσεων στο σύστημα με σκοπό την δυσλειτουργία του ή την εγγραφή ψευδών προσώπων ο

ψηφοφόρος θα πρέπει να συμπληρώσει και ένα ακόμα πεδίο στο οποίο επαληθεύει ένα αλφαριθμητικό που βλέπει στην οθόνη του.

- Αφού ολοκληρωθεί αυτή η διαδικασία το σύστημα επαληθεύει τα στοιχεία του.
- Αν τα στοιχεία είναι σωστά δημιουργούνται αυτόματα από την εφαρμογή το username και το password με τα οποία μπορεί να συνδεθεί ο ψηφοφόρος στην περιοχή της ψηφοφορίας.
- Τα στοιχεία αυτά αποστέλλονται αυτόματα στο email που έχει δηλώσει ο ψηφοφόρος στην αρχή πιστοποίησης.
- Ο ψηφοφόρος είναι υπεύθυνος και πρέπει να γνωρίζει μόνο αυτός τα στοιχεία πρόσβασης στο email του. Αφού εισέλθει στον λογαριασμό ηλεκτρονικού ταχυδρομείου του μπορεί να δει τα στοιχεία σύνδεσης και να τα χρησιμοποιήσει για να συνδεθεί στο σύστημα διαδικτυακής ψηφοφορίας.

Με την παραπάνω διαδικασία διασφαλίζεται πως ο μοναδικός που μπορεί να δει τα στοιχεία πρόσβασης στο σύστημα είναι ο ψηφοφόρος. Ακόμα και αν ο υπεύθυνος της αρχής πιστοποίησης προσπαθήσει κακόβουλα να κάνει εγγραφή στο σύστημα με τα στοιχεία κάποιου ψηφοφόρου δεν θα μπορέσει ποτέ να παραλάβει τα στοιχεία σύνδεσης στην εφαρμογή.

2.10.6 Αλλαγή κωδικού πρόσβασης.

Από την στιγμή που έχει συνδεθεί στο σύστημα ο χρήστης για μεγαλύτερη ασφάλεια έχει δικαίωμα να αλλάξει τον κωδικό σύνδεσης του. Το username θα παραμείνει για πάντα το ίδιο αλλά ο κωδικός πλέον θα είναι γνωστός μόνο σε αυτόν ακόμα και αν υποθέσουμε ότι κάποιος από τον πάροχο ηλεκτρονικής αλληλογραφίας προσπαθήσει να χρησιμοποιήσει αυτόν που είχε αρχικά αποσταλεί στο email.

2.10.7 Ψηφοφορία

Μετά την είσοδο του χρήστη στο διαδικτυακό σύστημα ψηφοφορίας εμφανίζονται στην οθόνη του όλες οι ψηφοφορίες· ενεργές στις οποίες δεν έχει ψηφίσει, ενεργές στις οποίες έχει ήδη ψηφίσει και ανενεργές.

Μπορεί να επιλέξει μια ενεργή στην οποία έχει ψηφίσει και να προχωρήσει στα βήματα της ψηφοφορίας.

- Στο πρώτο βήμα διαβάζει την ερώτηση του δημοψηφίσματος και κάνει την επιλογή του ανάμεσα στο ΝΑΙ , ΟΧΙ και το ΛΕΥΚΟ.
- Στο δεύτερο βήμα ερωτάται από το σύστημα να επαληθεύσει ξανά την επιλογή του. Αν έχει επιλέξει ΝΑΙ ή ΟΧΙ στο πρώτο βήμα η επιλογή ΛΕΥΚΟ δεν εμφανίζεται. Αν παραμείνει σταθερός στην επιλογή του προχωράει στο τρίτο βήμα. Αλλιώς επιστρέφει ξανά στο πρώτο για να επιλέξει από τη αρχή.
- Στο τρίτο βήμα ζητείται από τον ψηφοφόρο να επιβεβαιώσει την απόφασή του και να προχωρήσει στην ψηφοφορία. Αφού επιβεβαιώσει ακολουθεί μια τελευταία ερώτηση για το αν είναι σίγουρος και αν απαντήσει θετικά η ψήφος προστίθεται στις υπόλοιπες της επιλογής του.

Αυτόματα δηλώνεται πως έχει ψηφίσει και όταν επανέλθει στην αρχική οθόνη η συγκεκριμένη ψηφοφορία εμφανίζεται στις ενεργές στις οποίες όμως έχει ψηφίσει.

Στην συγκεκριμένη κατηγορία ο χρήστης δεν έχει δικαίωμα να εκτελέσει κάποια λειτουργία καθώς σύμφωνα με την αρχή της μυστικότητας δεν θα πρέπει κατά την διάρκεια της ψηφοφορίας οι ψηφοφόροι να βλέπουν τα αποτελέσματα.

Αποτελέσματα μπορεί να δει για τις ψηφοφορίες οι οποίες έχουν ολοκληρωθεί και είναι ανενεργές.

Τέλος να σημειωθεί πως δεν πρέπει να κρατείται κανένα αντίγραφο στην Β.Δ. το οποίο να συσχετίζει τα στοιχεία του ψηφοφόρου με την ψήφο του σύμφωνα με την αρχή της μη δυνατότητας κατασκευής απόδειξης.

Η μόνη απόδειξη που είναι αρκετή για τους διαχειριστές για να καλυφθεί η αρχή ασφαλείας της μη άρνησης ευθυνών είναι η καταγραφή πως ο συγκεκριμένος ψηφοφόρος έχει λάβει μέρος στην συγκεκριμένη εκλογική διαδικασία.

2.10.8 Δημιουργία νέων χρηστών

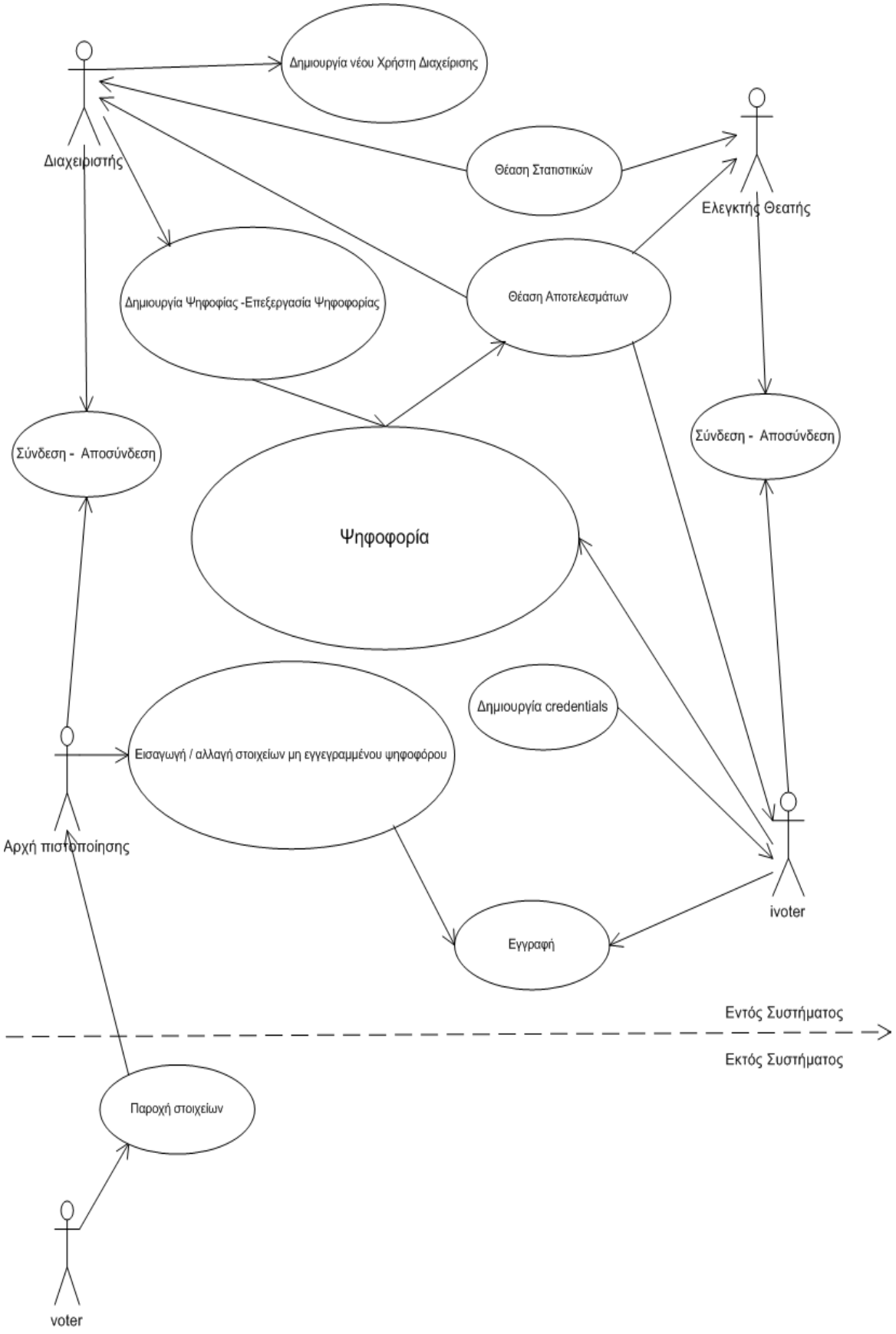
Την συγκεκριμένη διεργασία μπορεί να την εκτελεί μόνο ο διαχειριστής συστήματος από την στιγμή που έχει επαληθευτεί η ταυτότητα του και έχει εδραιώσει κρυπτογραφημένη επικοινωνία. Μόνο αυτοί οι χρήστες έχουν εξουσιοδότηση να βρίσκονται σε αυτή την περιοχή της εφαρμογής. Ο διαχειριστής μπορεί να δημιουργήσει άλλον διαχειριστή, χρήστη αρχής πιστοποίησης και ελεγκτή.

2.10.9 Δημιουργία νέας ψηφοφορίας

Οι διαχειριστές και μόνο αυτοί έχουν εξουσιοδότηση να δημιουργούν νέα δημοψηφίσματα. Μπορούν αν εισάγουν την ερώτηση και να ορίσουν αν το δημοψήφισμα είναι ενεργό.

2.10.10 Απενεργοποίηση ψηφοφορίας

Οι διαχειριστές και μόνο αυτοί μπορούν να απενεργοποιήσουν μια ψηφοφορία όταν αυτή έχει ολοκληρωθεί. Από την στιγμή που η ψηφοφορία έχει απενεργοποιηθεί ο ψηφοφόρος δεν θα μπορέσει να ψηφίσει ακόμα και αν βρίσκεται στο $2^{ο}$ ή στο $3^{ο}$ βήμα της ψηφοφορίας.

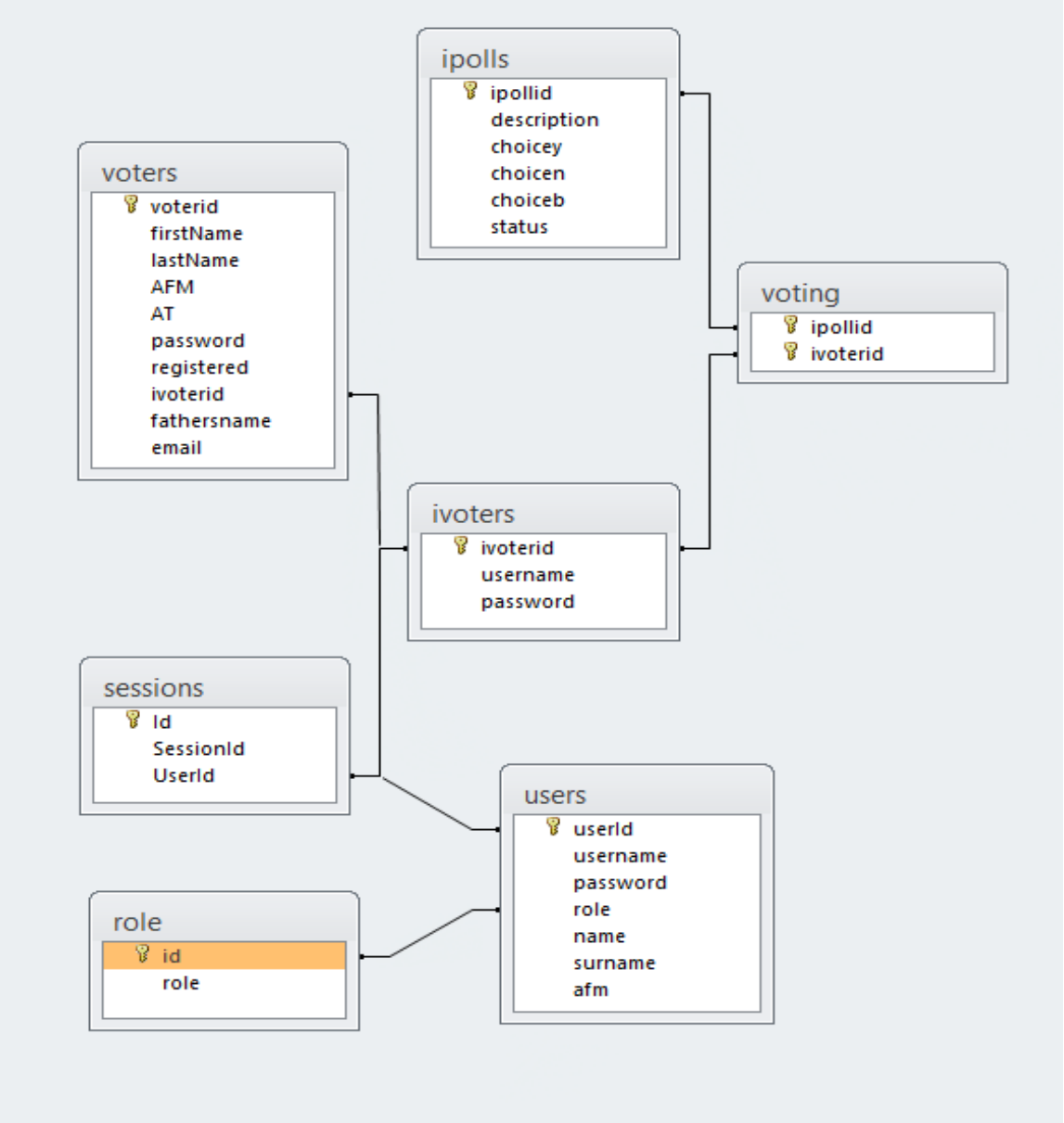


Σχήμα 5. Διάγραμμα παρουσίασης διαδικασιών συστήματος

3 Σχεδιασμός Συστήματος

3.1 Περιγραφή Β.Δ.

Όπως περιγράψαμε στο προηγούμενο κεφάλαιο για την ανάπτυξη του συστήματος θα χρησιμοποιηθεί η αρχιτεκτονική MVC. Στην συγκεκριμένη αρχιτεκτονική το ένα τμήμα του κώδικα το model αποτελείται από κλάσεις java οι οποίες αντιπροσωπεύουν την Βάση Δεδομένων του συστήματος. Αρχικά λοιπόν θα σχεδιάσουμε με βάση τους ρόλους και τις διαδικασίες που απαιτούνται για το σύστημα, τη Βάση δεδομένων.



Σχήμα 6. Σχήμα Βάσης Δεδομένων συστήματος

Στον σχήμα 6 φαίνεται πως χρησιμοποιήσαμε 7 πίνακες. Τα πεδία του εκάστοτε πίνακα θα χρησιμοποιηθούν ως μεταβλητές στις κλάσεις του model και η πρόσβαση και η επεξεργασία των στα δεδομένων τους θα γίνεται μέσω των ρουτινών των κλάσεων.

Αυτό συμβαίνει διότι για να επεξεργαστούμε ή απλά για να δούμε τα στοιχεία ενός πίνακα θα πρέπει να έχουν πακεταριστεί σε αντικείμενα ή σε λίστες αντικειμένων. Τα αντικείμενα αυτά λοιπόν ορίζονται από τις κλάσεις του model της εφαρμογής μας.

Παρακάτω θα περιγράψουμε τον κάθε πίνακα ξεχωριστά και θα αναφέρουμε ποιες ενέργειες χρειάζεται να κάνει το σύστημα σε αυτόν. Έπειτα, σύμφωνα με αυτά που είπαμε παραπάνω, θα περιγράψουμε τις κλάσεις του Model του συστήματος δηλαδή τις μεταβλητές τους και τις ρουτίνες τις οποίες χρησιμοποιούν τα αντικείμενά τους.

Voters

Ο πίνακας voters περιέχει τα στοιχεία του κάθε ψηφοφόρου ως πολίτη. Περιέχει δηλαδή το όνομά του, το επώνυμό του, το όνομα πατρός, τον Αριθμό Φορολογικού του Μητρώου,τον Αριθμό της Αστυνομικής του Ταυτότητας και ένα email του. Τα πεδία registered και Voterid χρησιμοποιούνται από το σύστημα ώστε να γνωρίζουμε αν ο χρήστης έχει εγγραφεί και ώστε να κρατάμε το id του ως ηλεκτρονικού ψηφοφόρου αντίστοιχα. Κύριο κλειδί του πίνακα είναι το voterid το οποίο και θα πρέπει να είναι μοναδικό για τον κάθε πολίτη.

Users

Ο πίνακας users περιέχει τα στοιχεία των χρηστών διαχείρισης του συστήματος, των administrators, των υπευθύνων αρχής πιστοποίησης και των ελεγκτών. Περιέχει δηλαδή το όνομα το επώνυμο και το ΑΦΜ των χρηστών καθώς και το όνομα χρήστη τον κωδικό πρόσβασης και ένα αριθμητικό πεδίο το οποίο προσδιορίζει τον ρόλο τους στο σύστημα. Κύριο κλειδί είναι το Userid του κάθε χρήστη.

Role

Ο βοηθητικός πίνακας role περιέχει τους ρόλους του συστήματος. Με βάση αυτόν οι εγγραφές των χρηστών παίρνουν τιμές στο πεδίο role. Ανάλογα με την τιμή του πεδίου αυτού ορίζεται και η εξουσιοδότηση του κάθε χρήστη στο σύστημα.

Sessions

Άλλος ένας βοηθητικός πίνακας είναι ο sessions. Σε αυτόν τον πίνακα αποθηκεύεται το sessionid του χρήστη από την στιγμή που εισέρχεται στο σύστημα. Το sessionid αποθηκεύεται σε αντιστοιχία με το Userid του κάθε χρήστη και με αυτόν τον τρόπο γνωρίζουμε πότε είναι συνδεδεμένος στο σύστημα ώστε να του παραχωρείται πρόσβαση στις διεργασίες που θέλει να εκτελέσει. Όταν ο χρήστης αποσυνδεθεί η εγγραφή του διαγράφεται μέχρι να συνδεθεί ξανά.

Ivoters

Στον πίνακα ivoters αποθηκεύονται τα credentials των ηλεκτρονικών ψηφοφόρων. Κύριο κλειδί αυτού του πίνακα είναι το ivoterid το οποίο είναι μοναδικό για κάθε έναν.

Χρησιμοποιείται όπως και το Userid στον πίνακα sessions ώστε να γνωρίζουμε πότε ένας ψηφοφόρος είναι συνδεδεμένος στο διαδικτυακό σύστημα και να του παρέχουμε την κατάλληλη εξουσιοδότηση.

Ipolls

Ο πίνακας ipolls περιέχει τις ψηφοφορίες – δημοσκοπήσεις του συστήματος. Για τον σκοπό της συγκεκριμένης πτυχιακής είναι σχεδιασμένος ώστε να μπορεί να αποθηκεύσει απαντήσεις σε ένα δημοψήφισμα. Αν θέλαμε να επεκτείνουμε το σύστημα σε σύστημα εκλογών θα μπορούσαμε να χρησιμοποιήσουμε έναν ακόμα πίνακα ο οποίος θα περιείχε ως εγγραφές τις πιθανές εκλογικές επιλογές. Προς το παρόν αυτός ο πίνακας αποθηκεύει για κάθε ψηφοφορία τις απαντήσεις ναι, όχι ,λευκό στα ανάλογα πεδία. Για κάθε ψήφο που επιλέγεται αυξάνεται κατά ένα η τιμή της κατάλληλης εγγραφής. Εκτός από αυτά τα πεδία περιέχει την περιγραφή της ψηφοφορίας και την κατάσταση στην οποία βρίσκεται. Οι πιθανές καταστάσεις μπορεί να είναι ενεργή (active) όταν διεξάγεται μια ψηφοφορία, ανενεργή (inactive) όταν έχει ολοκληρωθεί και εκκρεμής (pending) όταν έχει καταχωρηθεί στο σύστημα άλλα δεν έχει ενεργοποιηθεί ακόμα. Κύριο κλειδί αυτού του πίνακα είναι το pollid το οποίο θα πρέπει να είναι και μοναδικό για την κάθε ψηφοφορία.

Voting

Άλλος ένας βοηθητικός πίνακας είναι ο voting. Στον συγκεκριμένο πίνακα αποθηκεύουμε πληροφορία για το αν κάποιος Ivoter έχει ψηφίσει σε συγκεκριμένη ψηφοφορία. Αν το ζευγάρι ipollid – ivoterid υπάρχει σε αυτόν τον πίνακα σημαίνει πως ο ψηφοφόρος έχει ήδη ψηφίσει.

3.2 Περιγραφή Model εφαρμογής

Το τμήμα του κώδικα που ονομάζουμε model αποτελείται από κλάσεις οι οποίες συνδέονται άμεσα με την βάση δεδομένων. Τα αντικείμενα που μπορούν να δημιουργηθούν από αυτές τις κλάσεις θα περιέχουν τα δεδομένα που θέλουμε είτε να επεξεργαστούμε, είτε να προσθέσουμε, είτε να ανακτήσουμε από τη βάση δεδομένων του συστήματος. Οι λειτουργίες αυτές εκτελούνται με την βοήθεια των ρουτινών των κλάσεων και τα πεδία της Β.Δ. αντικατοπτρίζονται από τις μεταβλητές τους. Για την καλύτερη κατανόηση του σχεδιασμού και των λειτουργιών του συστήματος θα κάνουμε μια σύντομη περιγραφή στις κλάσεις, τις ρουτίνες και τις ιδιότητες τους.

Voter

Μεταβλητές:

int voterId, afm, ivoterId, registered

String name, surname, fathersname, email, at, registeredPass

Ρουτίνες:

getVoter (int voterId)

Χρησιμοποιούμε την ρουτίνα αυτή όταν θέλουμε να εξαγάγουμε τα στοιχεία ενός Voter

με βάση το voterid του. Επιστρέφει ένα αντικείμενο της κλάσης voter.

getVoterAfm (int afm)

Επιστρέφει ένα αντικείμενο voter με τα στοιχεία του ψηφοφόρου με βάση το ΑΦΜ του αν είναι ήδη καταχωρημένος. Αν όχι επιστρέφεται Null.

getAllVoters (int registered)

Επιστρέφει μια λίστα αντικειμένων της κλάσης voters, με όλους τους εγγεγραμμένους ψηφοφόρους. Κάθε αντικείμενο έχει τα στοιχεία του κάθε ψηφοφόρου.

writeVoter (String name, String surname, String fathersname, String email, int afm, String at, String registeredPass, int ivoterId, int registered)

Με την συγκεκριμένη ρουτίνα γίνεται εισαγωγή ενός νέου ψηφοφόρου στο σύστημα και επιστρέφεται το νέο voterid που του δόθηκε.

writeVoter (int voterId , String name, String sumame, String fathersname, String email ,int afm, String at, String registeredPass, int ivoterId, int registered)

Με την συγκεκριμένη ρουτίνα γίνονται update τα στοιχεία ενός ψηφοφόρου με βάση το voterid του.

getVotersSum()

Η ρουτίνα αυτή δεν δέχεται παραμέτρους. Μας επιστρέφει το άθροισμα των εγγεγραμμένων ψηφοφόρων.

Users

Μεταβλητές:

int userId, afm

String username, password, role, name, surname

Ρουτίνες:

getUsers(String username)

Αύτη την ρουτίνα την χρησιμοποιούμε όταν θέλουμε να εξάγουμε τα στοιχεία ενός χρήστη ή να ελέγξουμε αν υπάρχει ο συγκεκριμένος χρήστης με βάση το username του.

getUsers(int userId)

Με αυτήν την ρουτίνα εξάγουμε τα στοιχεία ενός User με βάση το userId του.

getAllVoters (int registered)

Επιστρέφει μια λίστα αντικειμένων της κλάσης voters, με όλους τους εγγεγραμμένους ψηφοφόρους. Κάθε αντικείμενο έχει τα στοιχεία του κάθε ψηφοφόρου.

writeUsers (String username, String password, String role, String name, String sumame,int afm)

Με την συγκεκριμένη ρουτίνα γίνεται εισαγωγή των στοιχείων ενός νέου χρήστη στο σύστημα. Εδώ καθορίζεται και ο ρόλος του χρήστη.

writeUsers (int userId, String username, String password, String role, String name, String surname,int afm)

Με την συγκεκριμένη ρουτίνα γίνονται update τα στοιχεία ενός χρήστη με βάση το

userid του.

getConnectediv()
Η ρουτίνα αυτή δεν δέχεται παραμέτρους. Μας επιστρέφει το σύνολο των συνδεδεμένων στο σύστημα ηλεκτρονικών ψηφοφόρων.

getConnecteda()
Η ρουτίνα αυτή δεν δέχεται παραμέτρους. Μας επιστρέφει το σύνολο των συνδεδεμένων διαχειριστών (admins) .

getConnectedca()
Η ρουτίνα αυτή δεν δέχεται παραμέτρους. Μας επιστρέφει το σύνολο των συνδεδεμένων χρηστών αρχής πιστοποίησης.

getConnectedau()
Η ρουτίνα αυτή δεν δέχεται παραμέτρους. Μας επιστρέφει το σύνολο των συνδεδεμένων ελεγκτών – θεατών.

Sessions

Μεταβλητές:

int id

String sessionId, userId

Ρουτίνες:

Authenticate(String sessionId,String userId)

Με την βοήθεια της συγκεκριμένης ρουτίνας παρέχουμε την εξουσιοδότηση των χρηστών στο σύστημα. Εδώ ελέγχεται αν ο χρήστης έχει συνδεθεί στο σύστημα.

Authenticate(String userId)
Με αυτήν την ρουτίνα παίρνουμε το session ενός συνδεδεμένου χρήστη με βάση του serId του.

login (String sessionId, String userId)

Η συγκεκριμένη ρουτίνα δημιουργεί την εγγραφή με την οποία συνδέουμε το userid ενός χρήστη με το session του. Εκτελείται κατά την σύνδεση του χρήστη στο σύστημα ώστε να γνωρίζουμε από εκείνη την στιγμή πως είναι συνδεδεμένος και να του παρέχουμε την κατάλληλη εξουσιοδότηση.

Update(String sessionId,String userId)

Εάν κάποιος χρήστης δεν αποσυνδεθεί φυσιολογικά ή λήξει το session του όταν ξανασυνδεθεί ανανεώνουμε την εγγραφή του στον πίνακα sessions με το νέο session που χρησιμοποιεί.

Logout(String sessionId)

Η συγκεκριμένη ρουτίνα εκτελείται όταν ένας χρήστης αποσυνδεθεί από το σύστημα. Διαγράφουμε την εγγραφή του από τον πίνακα ώστε να γνωρίζουμε πως πλέον δεν έχει εξουσιοδότηση στις λειτουργίες της εφαρμογής.

Ipolls

Μεταβλητές:

int ipollId, choicey, choicen, choiceb

String description, status

Ρουτίνες:

getIpoll(intipollId)

Η συγκεκριμένη ρουτίνα μας επιστρέφει ένα αντικείμενο της κλάσης Ipolls με βάση το ipollId που της έχουμε ζητήσει. Σε αυτό θα περιέχονται η περιγραφή της ψηφοφορίας, τα αποτελέσματα και η κατάστασή της.

getAllPolls(String status)

Με την χρήση αυτής της ρουτίνας εξάγουμε μια λίστα αντικειμένων της κλάσης ipolls με βάση την κατάστασή τους την οποία χρησιμοποιούμε για να κατασκευάσουμε την λίστα των ψηφοφοριών.

getAllActCanVotePolls(int ivoterId)

Αυτή η ρουτίνα χρησιμοποιείται για να εξάγουμε την λίστα των ενεργών ψηφοφοριών στις οποίες ένας ivoter έχει δικαίωμα να ψηφίσει. Δηλαδή δεν έχει ψηφίσει ήδη. Επιστρέφει μια λίστα αντικειμένων ψηφοφοριών.

getifCanVotePolls(int ipollId,int ivoterId)

Η συγκεκριμένη ρουτίνα χρησιμοποιείται ώστε να γίνει έλεγχος για το εάν ένας ψηφοφόρος επιτρέπεται να ψηφίσει σε μια συγκεκριμένη ψηφοφορία. Ουσιαστικά είναι ένας δεύτερος έλεγχος κατά την διαδικασία της ψηφοφορίας ώστε να είμαστε σίγουροι πως ένας ψηφοφόρος έχει δικαίωμα να ψηφίσει στην συγκεκριμένη ψηφοφορία.

getAllActCantVotePolls(int ivoterId)

Αυτή η ρουτίνα μας επιστρέφει μια λίστα με τις ψηφοφορίες οι οποίες είναι ανενεργές ή έχει ήδη συμμετάσχει συγκεκριμένος ψηφοφόρος. Ουσιαστικά εξάγουμε μια λίστα αντικειμένων με τις ψηφοφορίες στις οποίες ένας ψηφοφόρος δεν επιτρέπεται να συμμετάσχει.

writeIpoll (String description, int choicey, int choicen, int choiceb, String status)

Με την συγκεκριμένη ρουτίνα δημιουργείται μια νέα ψηφοφορία. Οι μεταβλητές των ψήφων γεμίζουν αυτόματα με 0 και στην εγγραφή του πίνακα ουσιαστικά προστίθεται από τον χρήστη μόνο η περιγραφή και η κατάσταση της ψηφοφορίας.

writeIpoll(int ipollId,String description,String status)

Με την χρησιμοποίηση αυτής της ρουτίνας αναβαθμίζουμε τα πεδία της περιγραφής και της κατάστασης μιας ήδη καταχωρημένης ψηφοφορίας. Η αναβάθμιση μπορεί να γίνει μόνο σε εκκρεμείς ψηφοφορίες.

Vote (int ipollId , String choice,int ivoterId)

Με την συγκεκριμένη ρουτίνα καταχωρείται η ψήφος του ivoter για μια συγκεκριμένη ενεργή ψηφοφορία στην οποία έχει δικαίωμα ψήφου στο σύστημα. Ανάλογα με την επιλογή του αυξάνεται κατά ένα ο αριθμός του κατάλληλου πεδίου

της εγγραφής της ψηφοφορίας με το συγκεκριμένο ipollid. Επίσης ενημερώνεται ο πίνακας Voting με το ζεύγος ipollId και ivoterId ώστε να γνωρίζουμε πως ο ηλεκτρονικός ψηφοφόρος δεν έχει δικαίωμα ψήφου ξανά στην συγκεκριμένη ψηφοφορία.

getVoted (int ipollId)

Η συγκεκριμένη ρουτίνα επιστρέφει το άθροισμα των ψηφοφόρων που έχουν ψηφίσει σε μια ψηφοφορία με το ipollid που ζητάμε.

getiVoters()

Με αυτήν την ρουτίνα εξάγουμε το συνολικό αριθμό των ηλεκτρονικών ψηφοφόρων που έχουν εγγραφεί στο σύστημα. Χρησιμοποιείται για την εξαγωγή στατιστικών.

PswGenerator

Μεταβλητές:

Έχει σταθερές μεταβλητές τις τιμές των πιθανών χαρακτήρων που θέλουμε να περιέχει ένας κωδικός.

Ρουτίνες:

generatePswd (int minLen, int maxLen, int noOfCAPSAlpha, int noOfDigits, int noOfSplChars)

Με την συγκεκριμένη ρουτίνα παράγεται ο νέος κωδικός που θα αποσταλεί σε κάθε νέο-εγγεγραμμένο ivoter. Δέχεται ως παραμέτρους το μέγεθος του κωδικού και το σύνολο των χαρακτήρων ανά είδος που θέλουμε να περιέχει ο κωδικός. Σχεδιάζουμε να αποτελείται από 16 χαρακτήρες και να περιέχει και τα 4 είδη· κεφαλαία γράμματα, πεζά γράμματα ,αριθμούς και σύμβολα.

Ivoter

Μεταβλητές:

Int ivoterId

String username, password

Ρουτίνες:

getIvoter(String username)

Με την συγκεκριμένη ρουτίνα ελέγχουμε αν υπάρχει ένας ivoter με το username το οποίο εισάγουμε ως παράμετρο. Αν υπάρχει μας επιστρέφει ένα αντικείμενο της κλάσης ivoter με τα στοιχεία του, ενώ αν δεν υπάρχει επιστρέφει Null.

getIvoter(int ivoterId)

Με αυτήν την ρουτίνα εξάγουμε ένα αντικείμενο της κλάσης ivoter με βάση το ivoterId.

checkIvoter (int voterId, String name, String surname, String fathersname, String email, intafm,Stringat,StringregisteredPass)

Με αυτήν την ρουτίνα γίνεται ο έλεγχος των στοιχείων που έχει εισαγάγει ο υπεύθυνος της αρχής πιστοποίησης στο σύστημα σε σχέση με τα στοιχεία που

εισάγει ο ψηφοφόρος κατά την διαδικασία της εγγραφής του. Αν βρεθεί ένα πεδίο του οποίου όλα τα στοιχεία ταιριάζουν με αυτά που εισάγει ο χρήστης η ρουτίνα μας επιστρέφει το voterId από τον πίνακα voters. Αν δεν βρεθεί επιστρέφει 0.

writeIvoter(int voterId)

Η συγκεκριμένη ρουτίνα χρησιμοποιείται ώστε να γίνει η εγγραφή του νέου ivoter στο σύστημα. Από την στιγμή που τα στοιχεία που εισάγει στην φόρμα εγγραφής είναι έγκυρα εκτελείται η ρουτίνα αυτή και δημιουργείται το όνομα χρήστη και το password και αποστέλλονται μέσω email στο email το οποίο έχει δηλώσει ο ψηφοφόρος.

writeIvoter(int ivoterId,String username,String password)

Με αυτήν την ρουτίνα πραγματοποιείται η αλλαγή κωδικού του Ivoter.

Check

Μεταβλητές:

-

Ρουτίνες:

Check (String parString, int flag, String field)

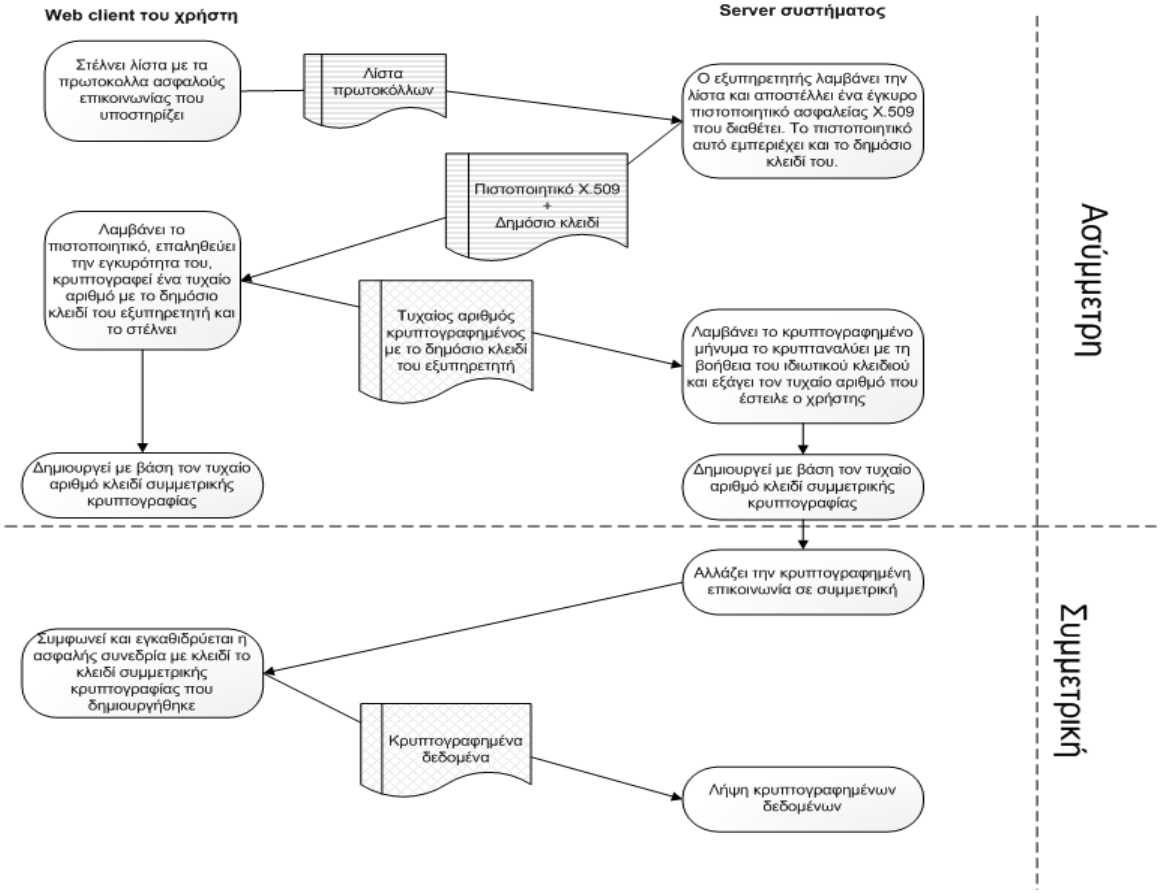
Με την συγκεκριμένη ρουτίνα πραγματοποιούμε όλους τους ελέγχους στα δεδομένα που εισάγουν οι χρήστες από τις διάφορες φόρμες του συστήματος. Λόγω του ότι μπορεί να προκληθούν προβλήματα στην ασφάλεια αν ένας κακόβουλος χρήστης προσπαθήσει να στείλει ακατάλληλα δεδομένα στο σύστημα όσα δεδομένα εισάγονται εξωτερικά ελέγχονται για την καταλληλότητα των χαρακτήρων που περιέχουν μέσω αυτής της ρουτίνας. Δέχεται τρεις παραμέτρους. Την τιμή που εισάγεται, τον τύπο του πεδίου από το οποίο έχει εισαχθεί και μια μεταβλητή την flag την οποία και επιστρέφει με αλλαγμένη τιμή αν βρεθεί πρόβλημα στην τιμή που έχει εισαχθεί

3.3 Περιγραφή ασφάλειας συστήματος

Κατά την ανάλυση του συστήματος διαπιστώθηκε πως πέρα από την καλή χωρίς λάθη ευέλικτη και προσαρμοστική στις εφαρμογές των χρηστών λειτουργία της εφαρμογής θα πρέπει να δοθεί ιδιαίτερη βάση στην ασφάλειά της. Σ’ αυτήν την ενότητα θα γίνει μια παρουσίαση των τρόπων με τους οποίους σχεδιάζουμε να πετύχουμε την μέγιστη ασφάλεια κατά την λειτουργία της εφαρμογής και κατά την καταμέτρηση και εξαγωγή των αποτελεσμάτων.

Τα δεδομένα που εισέρχονται και οι απαντήσεις του διαδικτυακό συστήματος κατά τις διεργασίες των χρηστών θα κρυπτογραφούνται. Θα χρησιμοποιήσουμε ένα πρωτόκολλο που χρησιμοποιεί συμμετρική και ασύμμετρη κρυπτογραφία και το οποίο εγκαθιστά ασφαλή κρυπτογραφημένη συνεδρία ανάμεσα στον εξυπηρετητή και στους χρήστες κατά την πρώτη τους επικοινωνία με την εφαρμογή. Αρχικά θα χρησιμοποιηθεί ασύμμετρη κρυπτογραφία για να ανταλλάξουμε το κλειδί της συμμετρικής κρυπτογραφίας η οποία θα εκτελείται σε όλη την υπόλοιπη διάρκεια της συνεδρίας.

Η εφαρμογή του χρήστη στέλνει τα πρωτόκολλα με τα οποία μπορεί να συνεργαστεί και ο εξυπηρετητής μας απαντά πως συνεργάζεται μόνο με τα δύο πιο σύγχρονα TLS 1.1 και TLS 1.2. Ο εξυπηρετητής μαζί με αυτήν την απάντηση στέλνει και το πιστοποιητικό ασφαλείας X.509 που έχουμε δημιουργήσει για την διαδικτυακή εφαρμογή το οποίο περιέχει και το δημόσιο κλειδί ασύμμετρης κρυπτογραφίας της εφαρμογής. Με τη χρήση του δημόσιου κλειδιού η εφαρμογή του χρήστη κρυπτογραφεί και αποστέλλει έναν τυχαίο μεγάλο αριθμό τον οποίο και θα χρησιμοποιήσουμε για κλειδί συμμετρικής κρυπτογραφίας. Ο εξυπηρετητής αφού λάβει το κλειδί το αποκρυπτογραφεί με το ιδιωτικό του και αποστέλλει πλέον συμμετρικά κρυπτογραφημένο μήνυμα και ξεκινά η ασφαλής επικοινωνία.

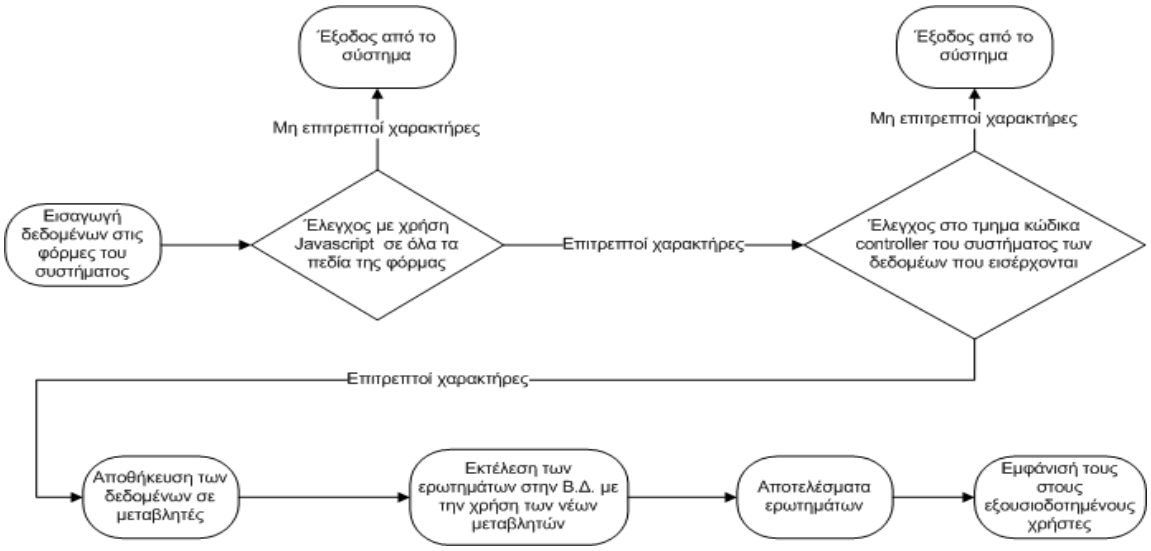


Σχήμα 7. Διάγραμμα εγκαθίδρυσης ασφαλούς επικοινωνίας

Πέρα, όμως, από την κρυπτογραφία των δεδομένων θα πρέπει να διαφυλάξουμε πως τα δεδομένα που εισέρχονται στο σύστημα από τους χρήστες είναι ασφαλή. Για τον λόγο αυτό πραγματοποιείται διπλός έλεγχος αρχικά στην εφαρμογή του χρήστη και έπειτα στον εξυπηρετητή και αν παρουσιαστεί εσφαλμένος ή ακατάλληλος χαρακτήρας απαγορεύεται στον χρήστη η συνέχεια της διεργασίας και τα δεδομένα δεν επεξεργάζονται περαιτέρω από το σύστημα. Αν οι χαρακτήρες που εισέρχονται είναι επιτρεπτοί αποθηκεύονται σε νέες μεταβλητές και εκτελούνται με την χρήση αυτών τα ερωτήματα προς την Β.Δ. Με αυτόν τον τρόπο γνωρίζουμε πως τα δεδομένα που εισέρχονται στην Β.Δ είναι ασφαλή και

πως το σύστημά μας δεν κινδυνεύει από επιθέσεις οι οποίες χρησιμοποιούν την αποστολή πλαστών δεδομένων για να ξεγελάσουν είτε αυτό είτε τους χρήστες του.

Κατά την εγγραφή των χρηστών για να αποφύγουμε επιθέσεις Bruteforce χρησιμοποιούμε έναν μηχανισμό ο οποίος παράγει μια εικόνα με τυχαίους χαρακτήρες. Τότε ζητείται από τους χρήστες να πληκτρολογήσουν τους χαρακτήρες που βλέπουν και αν κάνουν λάθος δεν τους επιτρέπεται να συνεχίσουν την εγγραφή τους στο σύστημα. Ο συγκεκριμένος μηχανισμός χρησιμοποιείται για να αποφύγουμε αυτοματοποιημένες επιθέσεις από μηχανές που ίσως προσπαθήσουν να εγγράψουν πλαστούς χρήστες στο σύστημα καθώς ακόμα και να πετύχουν τον συνδυασμό όλων των δεδομένων δεν θα μπορούν να διαβάσουν τους χαρακτήρες της εικόνας.



Σχήμα 8. Διάγραμμα εισαγωγής δεδομένων στο σύστημα

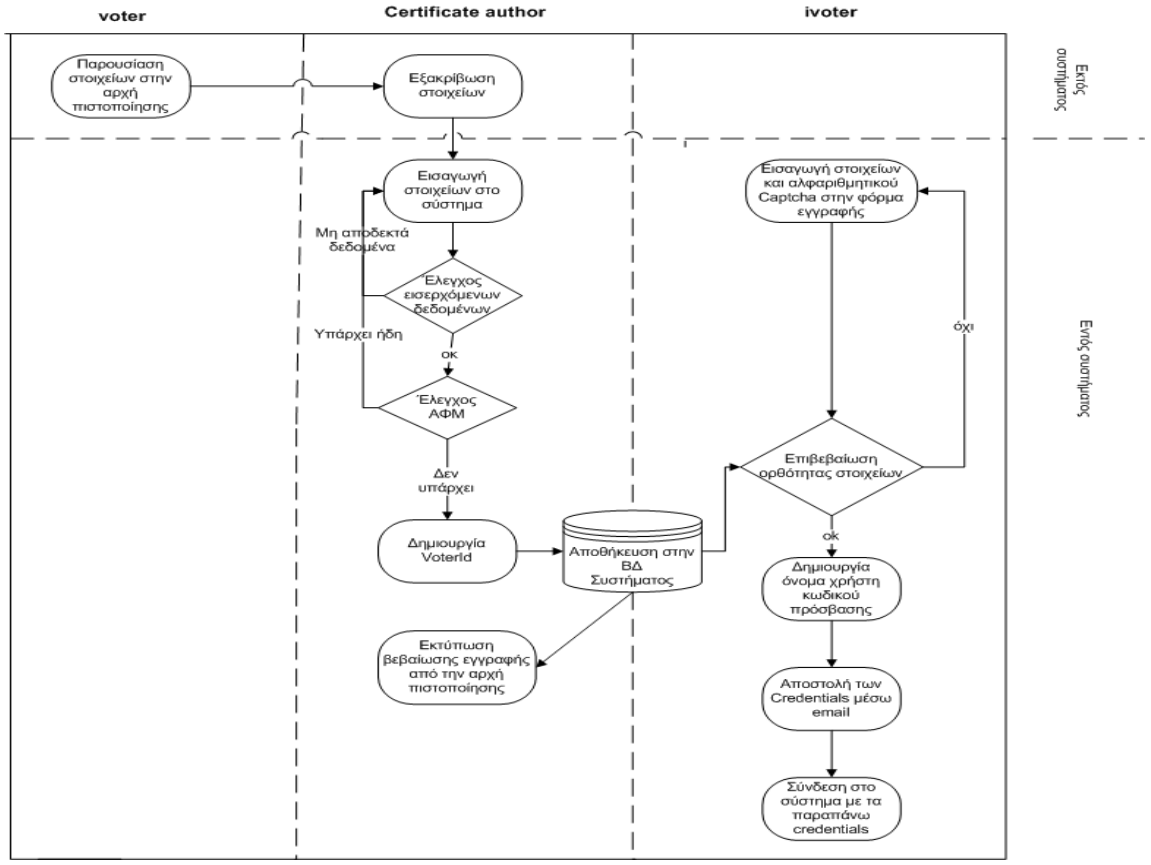
Λόγω του ότι η εγγραφή των χρηστών στο σύστημα είναι από τις σημαντικότερες από πλευράς ασφάλειας λειτουργίες της εφαρμογής καθώς σε καμία περίπτωση δεν θέλουμε να προκύψουν πλαστοί ψηφοφόροι που θα έχουν δικαίωμα ψήφου θα χρησιμοποιήσουμε ακόμα έναν μηχανισμό ασφαλείας.

Ο συγκεκριμένος μηχανισμός είναι ο βασικός μηχανισμός του συστήματος. Είναι η πιστοποίηση των στοιχείων των πολιτών και η εξακρίβωσή τους από το σύστημα ώστε να μπορούν να εγγραφούν ως ηλεκτρονικοί ψηφοφόροι και να έχουν δικαίωμα συμμετοχής στις διαδικτυακές ψηφοφορίες.

Οι ψηφοφόροι την πρώτη φορά που θέλουν να εγγραφούν στο σύστημα θα πρέπει να προσέλθουν σε μια αρχή πιστοποίησης η οποία θα μπορούσε να είναι κάποιο ΚΕΠ. Εκεί αφού αποδείξουν με κάποιο επίσημο έγγραφο τα στοιχεία τους ο υπεύθυνος αρχής πιστοποίησης τα εισάγει στο σύστημα. Τα στοιχεία που εισάγονται είναι το ονοματεπώνυμο, το όνομα πατρός, ο Αριθμός Φορολογικού Μητρώου και ο Αριθμός Αστυνομικής Ταυτότητας. Κατά την εισαγωγή γίνεται αυτόματος έλεγχος από το σύστημα ώστε να διαπιστωθεί πως το ΑΦΜ δεν υπάρχει ξανά στην βάση δεδομένων και να αποφύγουμε διπλοεγγραφές. Πέρα από τα παραπάνω στοιχεία ο πολίτης δηλώνει και έναν προσωπικό κωδικό μεγαλύτερο των 7 χαρακτήρων και το προσωπικό του email ώστε να

χρησιμοποιηθούν από το σύστημα κατά τη διαδικασία της εγγραφής. Όταν ολοκληρωθεί αυτή η διαδικασία παράγεται ο Αριθμός Μητρώου ψηφοφόρου και μαζί με τα υπόλοιπα στοιχεία εκτυπώνονται και παραδίδονται στον πολίτη όπως ακριβώς έχουν εισαχθεί στο σύστημα.

Ο ψηφοφόρος πλέον μπορεί από οποιονδήποτε Η/Υ να συνδεθεί με το διαδικτυακό σύστημα ψηφοφορίας και να εγγραφεί σε αυτό. Στην φόρμα εγγραφής νέου ψηφοφόρου πρέπει να εισάγει τα παραπάνω στοιχεία όπως ακριβώς είναι εκτυπωμένα στο έντυπο που πήρε από την αρχή πιστοποίησης και να πληκτρολογήσει τον τυχαίο αριθμό που εμφανίζεται στην φόρμα. Αν όλα τα στοιχεία εξακριβωθούν πως ανήκουν σε έναν συγκεκριμένο ψηφοφόρο προχωράει η διαδικασία της δημιουργίας του ως ivoter στο σύστημα.



Σχήμα 9. Διάγραμμα πιστοποίησης και εγγραφής ψηφοφόρου

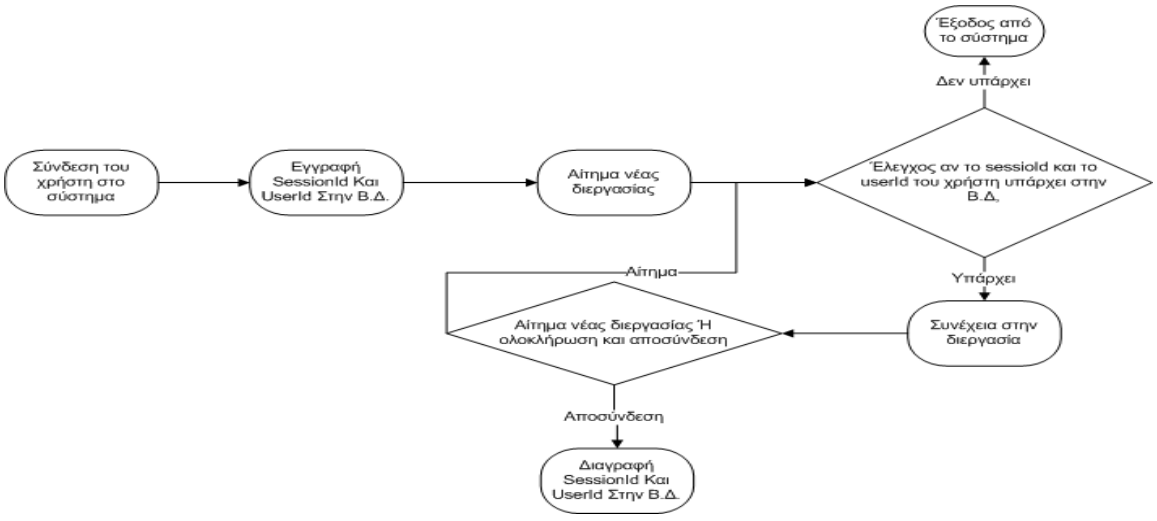
Κατά την δημιουργία του ivoter παράγεται αυτόματα από το σύστημα το όνομα χρήστη και ο κωδικός του. Για να δημιουργήσουμε το όνομα χρήστη χρησιμοποιούμε τους δύο πρώτους χαρακτήρες του επώνυμου του χρήστη, τους δύο πρώτους χαρακτήρες του ονόματός του, μια τελεία, τον τρίτο έως τον έβδομο χαρακτήρα του ΑΦΜ, τον πρώτο χαρακτήρα του ονόματος πατρός του ,μια ακόμα τελεία, τον τρίτο έως τον έκτο χαρακτήρα του Αριθμού ταυτότητας και τον δεύτερο έως τον τέταρτο χαρακτήρα του κωδικού τον οποίο είχε δηλώσει ο χρήστης. Θα έχει δηλαδή την παρακάτω μορφή.

```
surname(0,2)+name(0,2)+"."+afm(2,7)+fathersname(0,1)+"."+a
t(2,6)+registeredPass(1,3)
```

Για την δημιουργία του κωδικού χρησιμοποιούμε μια ρουτίνα η οποία παράγει ένα τυχαίο αλφαριθμητικό 16 χαρακτήρων ο οποίος αποτελείται από κεφαλαία και πεζά γράμματα από αριθμούς και από τα σύμβολα !@#$^*_+- . Δεν χρησιμοποιούμε όλα τα σύμβολα καθώς η εισαγωγή κάποιων από αυτά (όπως /,%,',”) είναι απαγορευμένη στο σύστημά μας για την αποφυγή επιθέσεων.

Όταν έχουν παραχθεί τα credentials του Ivoter αποστέλλονται αυτόματα στο προσωπικό email το οποίο έχει δηλώσει και έτσι γνωρίζει μόνο αυτός τα στοιχεία σύνδεσής του στην εφαρμογή. Με αυτό τον τρόπο αποφεύγουμε την προσπάθεια κάποιου κακόβουλου χρήστη ο οποίος μπορεί έχοντας την εκτύπωση της αρχής πιστοποίησης ή πρόσβαση στα δεδομένα της ίδιας αρχής να κάνει εγγραφή με τα στοιχεία κάποιου χρήστη εν αγνοία του. Μετά την είσοδό του στο σύστημα, ο χρήστης μπορεί να αλλάξει τον κωδικό πρόσβασης που του έχει αποσταλεί ώστε να είναι σίγουρος πως δεν γνωρίζει ούτε κάποιος κακόβουλος διαχειριστής του email παρόχου του τα στοιχεία σύνδεσής του.

Όσον αφορά την εξουσιοδότηση με την χρησιμοποίηση των sessions και του id των χρηστών πριν από κάθε διεργασία που ζητούν να εκτελέσουν γίνεται έλεγχος των στοιχείων τους και αν αποδειχθεί πως δεν έχουν το δικαίωμα οδηγούνται εκτός συστήματος. Ακόμα και αν κάποιος χρήστης έχει υποκλέψει το id και το session ενός χρήστη με άλλα δικαιώματα δεν μπορεί να συνδεθεί με τα δικαιώματα αυτού καθώς τα δύο παραπάνω στοιχεία εγγράφονται στην βάση δεδομένων μόνο κατά την επαλήθευση των στοιχείων κατά τη σύνδεση του χρήστη. Επίσης το session λήγει αυτόματα μετά από ένα μικρό χρονικό διάστημα οπότε ακόμα και αν υποτεθεί ότι ένας χρήστης δεν έχει αποσυνδεθεί σωστά από το σύστημα και τα στοιχεία αυτά έχουν μείνει ως σκουπίδι στη βάση δεδομένων δεν μπορούν να χρησιμοποιηθούν.



Σχήμα 10. Διάγραμμα εξουσιοδότησης χρηστών

Στο κομμάτι της διαχείρισης του συστήματος χρησιμοποιούνται και πάλι usernames και passwords μεγαλύτερα των 6 χαρακτήρων. Αυτά δεν παράγονται αυτόματα αλλά τα δημιουργεί ο διαχειριστής του συστήματος ο οποίος είναι και υπεύθυνος για αυτά. Και πάλι χρησιμοποιείται η ιδία μορφή εξουσιοδότησης ανάμεσα στους ρόλους των

διαχειριστών του συστήματος και δεν επιτρέπεται κανένας να εκτελέσει διεργασία για την οποία δεν έχει δικαίωμα.

Τέλος σύμφωνα με την αρχή της μυστικότητας δεν πρέπει κανείς να γνωρίζει την ψήφο κάποιου ψηφοφόρου και οι ψηφοφόροι δεν θα πρέπει να γνωρίζουν τα αποτελέσματα κατά την διάρκεια της ψηφοφορίας. Το πρώτο διασφαλίζεται καθώς κατά την ψηφοφορία δεν κρατείται από το σύστημα κανένα απολύτως στοιχείο που να σχετίζει το id του ψηφοφόρου με την ψήφο του παρά μόνο σύμφωνα με την επιλογή του αυξάνεται κατά ένα το κατάλληλο πεδίο του πίνακα ψηφοφοριών. Το δεύτερο διασφαλίζεται καθώς δεν επιτρέπεται η επιλογή της θέασης των αποτελεσμάτων των ψηφοφοριών που δεν έχουν ολοκληρωθεί σε κανέναν ψηφοφόρο.

Με τους παραπάνω τρόπους προσπαθούμε να καλύψουμε όλα τα κριτήρια, τα νομικά πλαίσια και τις αρχές ασφαλείας που ορίστηκαν στο πρώτο κεφάλαιο και όπως είχαμε επισημάνει θα πρέπει να διασφαλίζονται από ένα διαδικτυακό σύστημα ψηφοφορίας. Στην επόμενη ενότητα θα περιγραφούν παραστατικά με την βοήθεια σχεδιαγραμμάτων οι διαδικασίες και η ροή δεδομένων κατά την εκτέλεσή τους από τους χρήστες της εφαρμογής.

3.4 Σχεδιαγράμματα ροής δεδομένων και διαδικασιών

Με βάση τα όσα επισημάνθηκαν στην ανάλυση του συστήματος και στις προηγούμενες ενότητες του σχεδιασμού μπορεί πλέον να σχεδιαστεί η ροή δεδομένων ανάμεσα στις διαδικασίες που έχει εξουσιοδότηση ο κάθε χρήστης να εκτελεί. Κατά την υλοποίηση της εφαρμογής θα χρησιμοποιήσουμε τον συγκεκριμένο σχεδιασμό. Σύμφωνα με την αρχιτεκτονική MVC, που όπως προαναφέραμε θα χρησιμοποιηθεί, ο κώδικας θα τμηματοποιηθεί στις παρακάτω κατηγορίες.

Στο τμήμα του κώδικα model του συστήματος η εφαρμογή θα εκτελεί όλες τις διεργασίες που έχουν να κάνουν με την Β.Δ, στο τμήμα controller θα γίνονται όλοι οι απαραίτητοι έλεγχοι και οι ανακατευθύνσεις στις κατάλληλες σελίδες του τμήματος view το οποίο αποτελείται από τις jsp σελίδες της εφαρμογής και είναι υπεύθυνο για την παρουσίαση των δεδομένων στους χρήστες.

Ξεκινώντας από τις διεργασίες των χρηστών διαχείρισης του συστήματος θα περιγραφούν οι λειτουργίες που είναι εξουσιοδοτημένος να εκτελεί ο διαχειριστής του - admin.

Ο διαχειριστής του συστήματος αφού επιλέξει το πεδίο διαχείρισης της εφαρμογής μπορεί χρησιμοποιώντας το όνομα χρήστη και τον κωδικό πρόσβασης που έχει να συνδεθεί στο σύστημα. Αν πληκτρολογήσει σωστά τα στοιχεία του συνδέεται αλλιώς του ζητείται να τα εισάγει ξανά χωρίς να προσδιορίζεται σε ποιο από τα δυο έχει κάνει λάθος. Αφού συνδεθεί οδηγείται στην κεντρική σελίδα διαχείρισης. Εκεί έχει τις εξής επιλογές. Μπορεί να δημιουργήσει έναν νέο χρήστη, μπορεί να δημιουργήσει μια νέα ψηφοφορία, μπορεί να μεταφερθεί στην σελίδα παρακολούθησης του συστήματος ή μπορεί να αποσυνδεθεί από το σύστημα. Επίσης στην κεντρική σελίδα διαχείρισης μπορεί να δει και να επιλέξει να

επεξεργαστεί τις ψηφοφορίες που βρίσκονται σε κατάσταση αναμονής. Δηλαδή δεν είναι ούτε ενεργές, ούτε και έχουν ολοκληρωθεί.

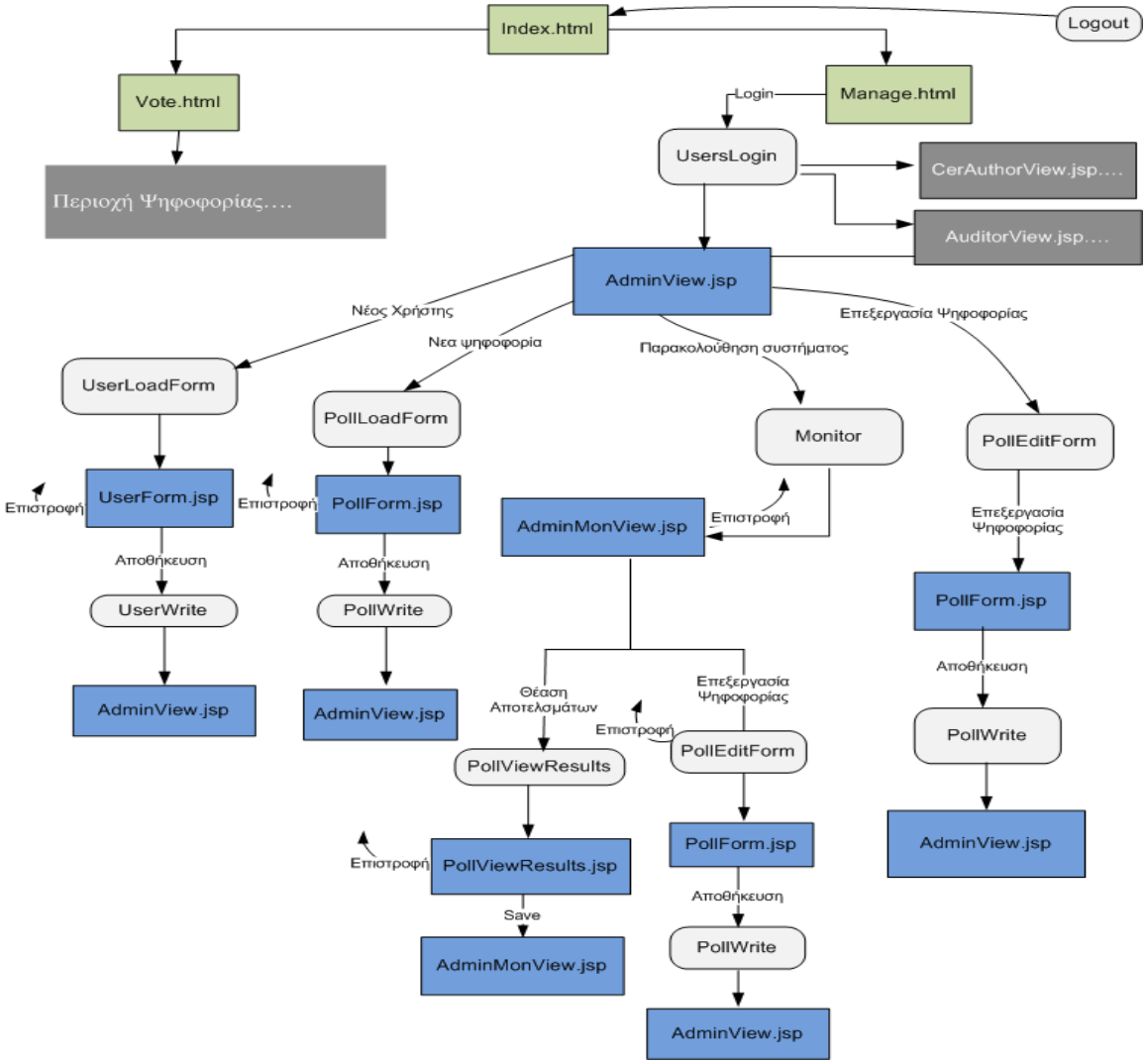
Κατά την δημιουργία ενός νέου χρήστη διαχείρισης θα πρέπει να εισαγάγει ένα όνομα χρήστη το οποίο να περιέχει μόνο γράμματα αριθμούς και το σύμβολο. Επίσης θα πρέπει να εισαγάγει έναν κωδικό πρόσβασης μεγαλύτερο των 5 χαρακτήρων ο οποίος θα μπορεί να αποτελείται από αριθμούς γράμματα και τα σύμβολα @#$^*_+- . Τον κωδικό σύνδεσης θα πρέπει να τον πληκτρολογήσει 2 φορές για επαλήθευση. Μετά από αυτό θα πρέπει να επιλέξει τον ρόλο που θα έχει ο χρήστης που δημιουργεί. Οι πιθανοί ρόλοι μπορεί να είναι admin, certification author και auditor οι οποίοι είναι και οι χρήστες διαχείρισης του συστήματος. Τέλος θα πρέπει να εισαγάγει το όνομα, το επώνυμο και το ΑΦΜ του χρήστη που δημιουργεί. Το ΑΦΜ θα πρέπει να αποτελείται από 9 αριθμούς ενώ το όνομα και το επώνυμο από γράμματα μόνο. Όλα αυτά τα δεδομένα όπως είδαμε στη ενότητα ασφαλείας του συστήματος ελέγχονται δύο φορές και αν είναι έγκυρα δημιουργείται ο νέος χρήστης διαχείρισης.

Κατά την δημιουργία νέας ψηφοφορίας ο admin θα πρέπει να εισαγάγει μια περιγραφή για το νέο δημοψήφισμα η οποία θα πρέπει να αποτελείται μόνο από γράμματα, αριθμούς και τα σύμβολα .? και να μην είναι μεγαλύτερη των 100 χαρακτήρων. Αφού πληκτρολογήσει την περιγραφή θα πρέπει να επιλέξει την κατάσταση στην οποία θα βρίσκεται η ψηφοφορία. Αν επιλέξει ενεργή τότε αυτομάτως θα εμφανίζεται στους ψηφοφόρους και θα μπορούν να ψηφίσουν σε αυτήν. Αν επιλέξει εκκρεμής τότε η ψηφοφορία δεν εμφανίζεται στους ψηφοφόρους καθώς θεωρείται πως ακόμα δεν έχει ξεκινήσει. Η επιλογή ανενεργή σημαίνει πως η ψηφοφορία έχει ολοκληρωθεί και δεν έχει κανείς πλέον δικαίωμα ψήφου σε αυτήν.

Στην οθόνη παρακολούθησης ο admin έχει δικαίωμα να δει όλες τις ψηφοφορίες του συστήματος. Στις ενεργές μπορεί να επιλέξει είτε να τις επεξεργαστεί ώστε να τις μετατρέψει σε ενεργές είτε να δει τα μέχρι στιγμής στατιστικά τους. Τις εκκρεμείς μπορεί να τις μετατρέψει είτε σε ενεργές είτε σε ανενεργές και στις ανενεργές μπορεί να δει μόνο τα στατιστικά τους.

Κατά την θέαση των στατιστικών μεταφέρεται στην σελίδα αποτελεσμάτων της κάθε ψηφοφορίας. Εκεί μπορεί να δει τα αποτελέσματα αλλά και το ποσοστό συμμετοχής των ψηφοφόρων που συμμετείχαν σε σχέση με τους συνολικούς εγγεγραμμένους ivoters. Επίσης, μπορεί να συγκρίνει το άθροισμα των ψήφων σε σχέση με τους ivoters που φαίνεται στο σύστημα πως έχουν συμμετάσχει στην συγκεκριμένη ψηφοφορία ώστε να μπορεί να κάνει μια επαλήθευση της εγκυρότητας των αποτελεσμάτων

Τέλος, στην οθόνη παρακολούθησης μπορεί να δει τον αριθμό των χρηστών οι οποίοι είναι συνδεδεμένοι στο σύστημα και το σύνολο των πιστοποιημένων ψηφοφόρων αλλά και αυτών οι οποίοι έχουν κάνει ήδη εγγραφή σε αυτό και είναι ενεργοί ivoters.



Σχήμα 11. Διάγραμμα ροής δεδομένων διαχειριστή συστήματος

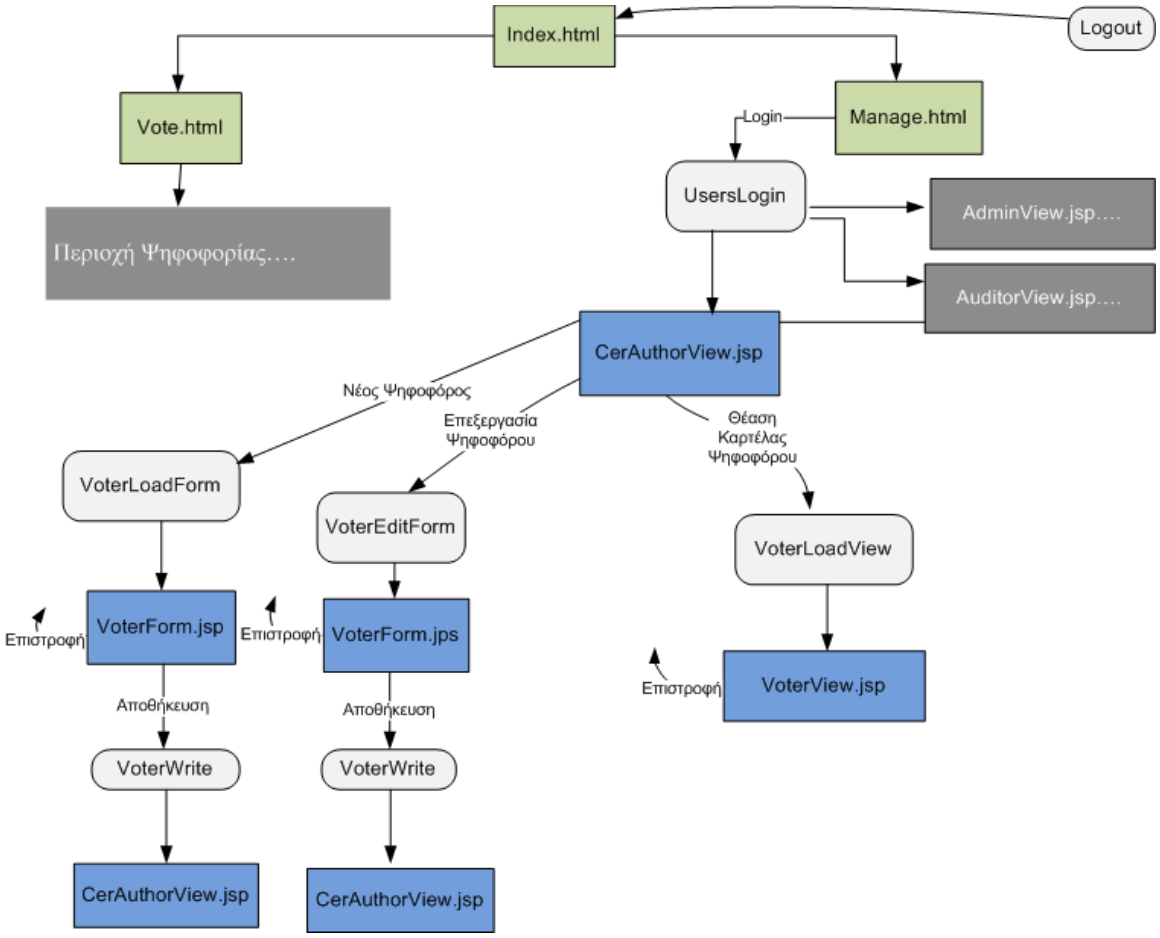
Ο χρήστης με ρόλο certification author (υπεύθυνος αρχής πιστοποίησης) είναι υπεύθυνος για την πιστοποίηση και την εισαγωγή των στοιχείων των πολιτών στο σύστημα. Μπορεί να δημιουργήσει έναν καινούριο ψηφοφόρο ή να επεξεργαστεί τα στοιχεία ενός, ο οποίος ακόμα δεν έχει κάνει εγγραφή στο σύστημα και είναι ανενεργός στις διαδικασίες των ψηφοφοριών.

Αφού μεταφερθεί στο πεδίο διαχείρισης του συστήματος μπορεί να εισαγάγει τα στοιχεία πρόσβασής του και μετά την αυθεντικοποίηση του από το σύστημα μεταφέρεται στην κεντρική οθόνη του υπεύθυνου πιστοποίησης. Σε αυτήν μπορεί να δει όλους τους ψηφοφόρους χωρισμένους σε ανενεργούς και ενεργούς. Ανενεργοί είναι οι ψηφοφόροι των οποίων τα στοιχεία έχουν εισαχθεί στο σύστημα αλλά ακόμα δεν έχουν κάνει εγγραφή σε αυτό ενώ ενεργοί είναι εκείνοι οι οποίοι έχουν προχωρήσει στην εγγραφή τους στο σύστημα, έχουν στοιχεία σύνδεσης και μπορούν να συμμετάσχουν στην διαδικασία της ψηφοφορίας. Επίσης, έχει την επιλογή της προσθήκης ενός νέου ψηφοφόρου.

Κατά την δημιουργία ψηφοφόρου θα πρέπει να εισαγάγει τα προσωπικά του στοιχεία. Το όνομα, το επώνυμο, το όνομα πατρός, το ΑΦΜ, τον αριθμό ταυτότητας, έναν λογαριασμό email του ψηφοφόρου και έναν κωδικό τον οποίο του έχει δώσει ο κάθε ψηφοφόρος για την πρώτη εγγραφή στο σύστημα. Τα στοιχεία που εισάγει στο σύστημα όπως είπαμε και στην ενότητα όπου περιγράφηκε η ασφάλεια του συστήματος ελέγχονται δύο φορές ενώ επίσης ελέγχεται και το ΑΦΜ ώστε να μην υπάρχει ήδη στην Β.Δ. Αν όλα τα στοιχεία περιέχουν επιτρεπόμενους από το σύστημα χαρακτήρες ο ψηφοφόρος δημιουργείται ως ανενεργός.

Κατά την επεξεργασία ενός ανενεργού ψηφοφόρου ο υπεύθυνος της αρχής πιστοποίησης μπορεί να επεξεργαστεί όλα του τα στοιχεία εκτός από τον αριθμό μητρώου ψηφοφόρου ο οποίος εκδίδεται μια φορά κατά την δημιουργία του σύστημα. Αν το ΑΦΜ αλλαχτεί πραγματοποιείται και πάλι έλεγχος ώστε να μην υπάρχει στην Β.Δ. του συστήματος.

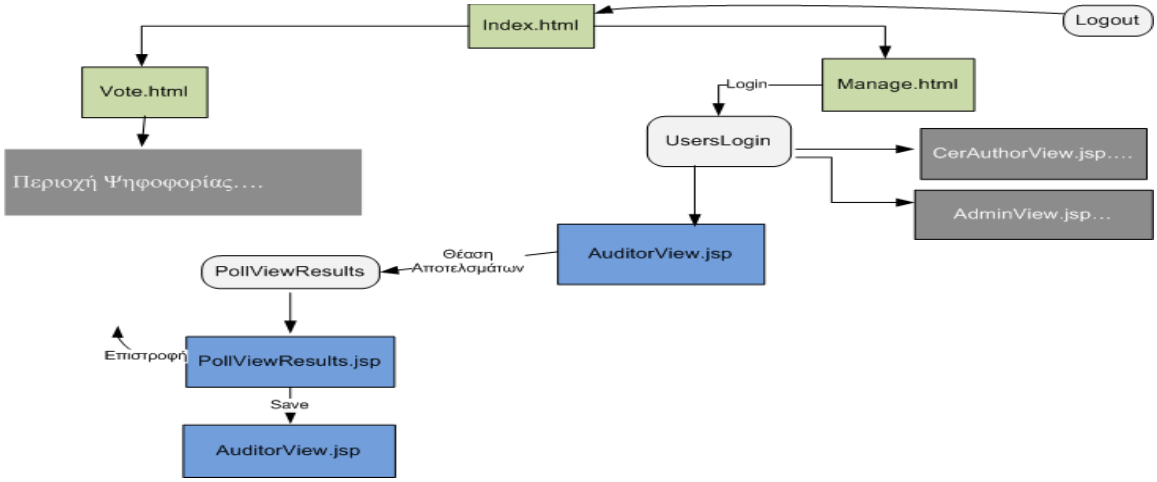
Εκτός από την δημιουργία και την επεξεργασία ενός ανενεργού ψηφοφόρου ο υπεύθυνος της αρχής πιστοποίησης μπορεί να επιλέξει να δει την καρτέλα με τα στοιχεία ενός ανενεργού ψηφοφόρου ώστε να την εκτυπώσει και να την δώσει σε αυτόν.



Σχήμα 12. Διάγραμμα ροής δεδομένων υπεύθυνου αρχής πιστοποίησης

Ο ελεγκτής ή παρατηρητής του συστήματος (auditor) μπορεί να δει τα στατιστικά και τα αποτελέσματα των ψηφοφοριών. Αφού συνδεθεί και αυθεντικοποιηθεί και αυτός στο πεδίο διαχείρισης του συστήματος, όπως και οι προηγούμενοι χρήστες, μεταφέρεται στην

κεντρική σελίδα ελέγχου. Εκεί μπορεί να δει όλες τις ψηφοφορίες και αντίστοιχα με τον διαχειριστή του συστήματος να δει τα αποτελέσματα και τα στατιστικά των ψηφοφοριών. Μπορεί να δει και αυτός πέρα από το αποτέλεσμα τα ποσοστά συμμετοχής στην κάθε ψηφοφορία καθώς και να συγκρίνει τον αριθμό ψήφων με τον αριθμό των ψηφοφόρων που συμμετείχαν σε μια συγκεκριμένη ψηφοφορία. Τέλος, μπορεί να δει τον αριθμό των πιστοποιημένων ψηφοφόρων του συστήματος, τον αριθμό των εγγεγραμμένων σε αυτό και το πλήθος αυτών οι οποίοι είναι συνδεδεμένοι την συγκεκριμένη στιγμή στο σύστημα.



Σχήμα 13. Διάγραμμα ροής δεδομένων παρατηρητή – ελεγκτή

Ο ψηφοφόρος αρχικά θα πρέπει να κάνει εγγραφή στο σύστημα ώστε να αποκτήσει τα στοιχεία πρόσβασης σε αυτό. Αφού επιλέξει το πεδίο ψηφοφορίας του συστήματος θα πρέπει να μεταβεί στην εγγραφή. Στην φόρμα η οποία εμφανίζεται στην συγκεκριμένη σελίδα θα πρέπει να εισαγάγει τα στοιχεία του όπως ακριβώς τα έχει παραλάβει από τον υπεύθυνο πιστοποίησης. Τα πεδία, όπως είναι λογικό, για να γίνει ο απαραίτητος έλεγχος είναι τα ίδια με αυτά που χρησιμοποιεί και ο υπεύθυνος πιστοποίησης. Εκτός από αυτό θα πρέπει να εισάγει σωστά το αλφαριθμητικό που εμφανίζεται στο κάτω μέρος της φόρμας. Στην συγκεκριμένη φόρμα γίνεται έλεγχος με τεχνολογία AJAX κάθε φορά που εισάγει έναν χαρακτήρα στο πεδίο του Αριθμού Μητρώου ώστε να επαληθευτεί αν ο αριθμός που έχει πληκτρολογήσει είναι εισαγμένος στο σύστημα.

Αφού πληκτρολογήσει όλα τα πεδία της φόρμας αυτά ελέγχονται για μη επιτρεπτούς χαρακτήρες και αν έχουν μόνο επιτρεπτούς στέλνονται στο σύστημα για την επιβεβαίωση των στοιχείων. Αν τα στοιχεία επιβεβαιωθούν αποστέλλονται στον ψηφοφόρο μέσω email στην διεύθυνση την οποία έχει δηλώσει τα στοιχεία σύνδεσής του και μεταφέρεται στην σελίδα σύνδεσης. Αν τα στοιχεία που έχει εισαγάγει στην φόρμα ή το αλφαριθμητικό επιβεβαίωσης που πληκτρολόγησε είναι λάθος ενημερώνεται πως θα πρέπει να προσπαθήσει ξανά.

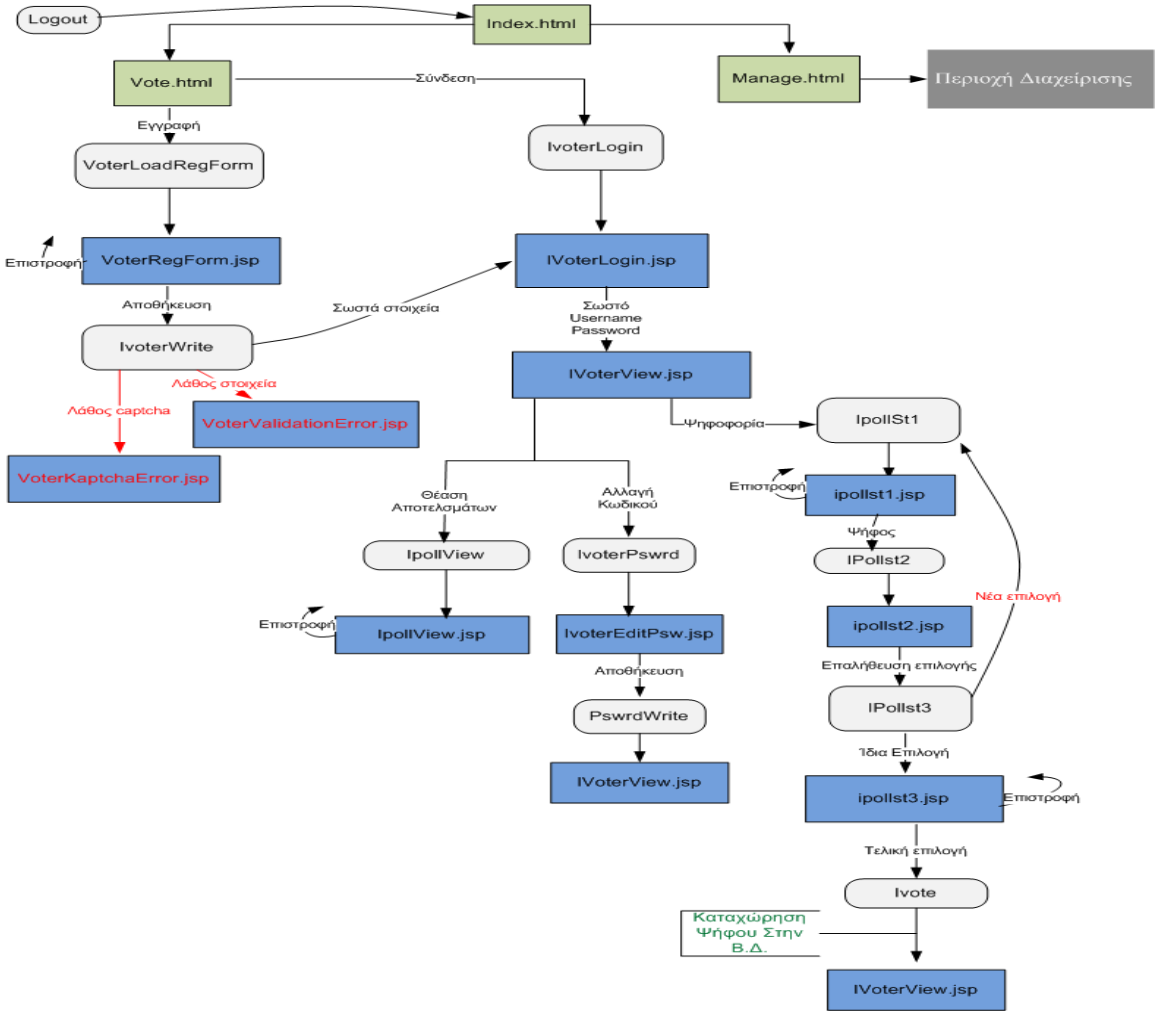
Με τα στοιχεία που του έχουν αποσταλεί στο email μπορεί να συνδεθεί στο σύστημα. Μετά την αυθεντικοποίηση του οδηγείται στην κεντρική σελίδα ψηφοφόρου. Εκεί έχει την επιλογή αν θέλει να αλλάξει τον προσωπικό κωδικό του, να συμμετάσχει σε μια ενεργή ψηφοφορία ή να δει τα αποτελέσματα μιας ολοκληρωμένης.

Αν επιλέξει να συμμετάσχει σε μια ψηφοφορία μεταφέρεται σε μια σελίδα στην οποία μπορεί να δει την περιγραφή του δημοψηφίσματος και τις επιλογές που έχει. ΝΑΙ, ΟΧΙ, ΛΕΥΚΟ .Από αυτές τις επιλογές μπορεί να επιλέξει μόνο μία και να προχωρήσει στο επόμενο βήμα. Στο δεύτερο βήμα για ασφάλεια ζητείται από τον ψηφοφόρο να επιβεβαιώσει την επιλογή του. Αν κάνει μια διαφορετική επιλογή μεταφέρεται ξανά στο προηγούμενο βήμα και η διαδικασία ξεκινά από την αρχή. Αν επιβεβαιώσει μεταφέρεται στο τρίτο βήμα. Στο βήμα αυτό εμφανίζεται η τελική επιλογή του ψηφοφόρου και έχει την επιλογή ή να προχωρήσει στην ψηφοφορία ή αν έχει αλλάξει επιλογή να ξεκινήσει την διαδικασία από τη αρχή οπότε και μεταφέρεται στο πρώτο βήμα ψηφοφορίας. Αν θέλει να προχωρήσει στην ψηφοφορία με αυτήν την επιλογή εμφανίζεται ένα παράθυρο για μια τελική επιβεβαίωση και αν επιλέξει να προχωρήσει ξανά η ψήφος του καταχωρείται στην Β.Δ και μεταφέρεται στην αρχική σελίδα ψηφοφοριών.

Από την στιγμή που έχει ψηφίσει σε μια ψηφοφορία αυτή γίνεται ανενεργή για εκείνον και θα πρέπει να περιμένει μέχρι την ολοκλήρωσή της ώστε να δει τα αποτελέσματα.

Στην θέαση των αποτελεσμάτων ο ψηφοφόρος μπορεί να δει εκτός από τα αποτελέσματα και το ποσοστό συμμετοχής που υπήρξε στο συγκεκριμένο δημοψήφισμα.

Αν θέλει και υπάρχει και άλλη ενεργή ψηφοφορία στην οποία δεν έχει ψηφίσει μπορεί να ψηφίσει και σε αυτήν αλλιώς μπορεί να αποσυνδεθεί από το σύστημα.



Σχήμα 14. Διάγραμμα ροής δεδομένων ψηφοφόρου

3.5 Εργαλεία υλοποίησης εφαρμογής

Για την υλοποίηση της εφαρμογής θα προτιμηθούν εργαλεία ανοιχτού κώδικα τα οποία παρέχουν οικονομία αλλά δεν υστερούν καθόλου σε σχέση με τα εμπορικά σε θέματα ταχύτητας, αξιοπιστίας και ασφάλειας. Ο εξυπηρετητής διαδικτυακών εφαρμογών που θα χρησιμοποιήσουμε είναι ο Apache Tomcat, η εφαρμογή σχεσιακής βάσης δεδομένων που θα χρησιμοποιηθεί είναι η MySql και μεταγλωττιστής των αρχείων του κώδικα της εφαρμογής θα είναι μεταγλωττιστής Java. Οι συγκεκριμένες εφαρμογές διαχειρίζονται τα servlet και τα servlet beans ενώ συνεργάζονται κατάλληλα ώστε να μπορεί να δημιουργηθεί η αρχιτεκτονική MVC. Από πλευράς ασφάλειας ενημερώνονται συνεχώς με βάση τους τρέχοντες κινδύνους και υποστηρίζουν τα πιο σύγχρονα πρωτόκολλα κρυπτογράφησης.

3.5.1 Servlet Container (Tomcat)

Ο servlet container είναι ένας εξειδικευμένος webserver που υποστηρίζει την εκτέλεση των servlet. Συνδυάζει τη βασική λειτουργικότητα ενός διακομιστή web με ορισμένες ειδικές βελτιστοποιήσεις Java/servlet και επεκτάσεις – όπως ένα ενσωματωμένο Περιβάλλον Χρόνου Εκτέλεσης της Java (Java Runtime Environment), και την ικανότητα να μεταφράζει αυτόματα συγκεκριμένες διευθύνσεις URL σε συγκεκριμένες αιτήσεις σε servlet. Τα επιμέρους servlets καταχωρούνται σε ένα servlet container, παρέχοντάς του συγκεκριμένες πληροφορίες σχετικά με τις λειτουργίες που παρέχουν και ποιο URL θα χρησιμοποιούν για τον εντοπισμό τους. Το servlet container στη συνέχεια είναι σε θέση να αρχικοποιήσει το servlet ανάλογα με τις ανάγκες και να παραδίδει τις αιτήσεις προς το servlet κατά την άφιξή του.

Στην συγκεκριμένη εφαρμογή χρησιμοποιείται ο Apache Tomcat, ένας servlet container που αναπτύχθηκε από το Apache Software Foundation (ASF). Ο Tomcat υλοποιεί τα Java Servlet και τις προδιαγραφές JavaServer Pages (JSP) από την Sun Microsystems και παρέχει ένα περιβάλλον διακομιστή web HTTP "καθαρής Java" για την εκτέλεση κώδικα Java.

Για την δική μας εφαρμογή χρησιμοποιούμε την έκδοση 6.0.43 του Apache-Tomcat [72].

3.5.2 Σχεσιακή βάση δεδομένων (MySQL)

Μια σχεσιακή βάση δεδομένων αντιστοιχεί δεδομένα χρησιμοποιώντας τα κοινά χαρακτηριστικά που διαπιστώθηκαν στο σύνολο των δεδομένων. Η ομαδοποίηση των δεδομένων που προκύπτει είναι οργανωμένη και πολύ πιο εύκολη να κατανοηθεί. Η MySQL είναι μια σχεσιακή βάση δεδομένων ανοιχτού κώδικα και χρησιμοποιείται σε πολλά δωρεάν προγράμματα λογισμικού.

Για την δική μας εφαρμογή χρησιμοποιούμε την έκδοση 5.0.45 της MySQL. Για τη διαχείριση της MySQL χρησιμοποιούμε τις κονσόλες MySQLAdministrator 1.2.17 και Navicat Lite 8.2.14. [73]

3.5.3 MySQL connector/J 5.1.7

Ο MySQL Connector προσφέρει συνδεσιμότητα μεταξύ της MySQL και εφαρμογών πελάτη που αναπτύχθηκαν στη γλώσσα προγραμματισμού Java μέσω ενός οδηγού JDBC [74]

3.5.4 Μεταγλωττιστής Java

Για την μεταγλώττιση των αρχείων java σε κλάσεις χρησιμοποιούμε την έκδοση j2sdk 1.7.0_75 της Java. [75].

3.5.5 Display tag library 1.2

Η Display tag library είναι μια σουίτα ανοικτού κώδικα με προσαρμοσμένες ετικέτες (tags) που παρέχουν υψηλού επιπέδου παρουσίαση δεδομένων σε μορφή πίνακα στο διαδίκτυο. Η συγκεκριμένη βιβλιοθήκη έχει σχεδιαστεί και είναι απόλυτα συμβατή με τις αρχές της αρχιτεκτονικής MVC. Δηλαδή μπορεί και λαμβάνει από ένα servlet ένα JavaBean το οποίο περιέχει λίστες αντικειμένων [76]. Με την συγκεκριμένη σουίτα θα γίνεται η εξαγωγή των λιστών αντικειμένων από το Model της εφαρμογής και η παρουσίασή τους σε πίνακες στο View.

3.5.6 OpenSSL

Το OpenSSL Project είναι μια συλλογική προσπάθεια για την ανάπτυξη μιας ισχυρής, πλήρως εξοπλισμένης, ισάξιας με τις εμπορικές, εργαλειοθήκης η οποία είναι ανοιχτού κώδικα και υποστηρίζει τα πρωτόκολλα Secure Sockets Layer (SSL v2 / v3) και Transport Layer Security (TLS v1.0 / v1.1 / V1 .2). Είναι στην ουσία μια γενικής χρήσης βιβλιοθήκη κρυπτογράφησης την οποία μπορεί να διαχειρίζεται μια παγκόσμια κοινότητα εθελοντών που χρησιμοποιούν το Διαδίκτυο για να επικοινωνούν με ασφάλεια. [77]. Ο εξυπηρετητής της εφαρμογής μας ο Apache Tomcat συνεργάζεται με την συγκεκριμένη βιβλιοθήκη και μας βοηθά στην εγκαθίδρυση της κρυπτογραφημένης συνεδρίας ανάμεσα στους χρήστες και στο σύστημα.

3.5.7 JavaMail

Η JavaMail είναι μια ανοιχτού κώδικα βιβλιοθήκη γραμμένη σε γλώσσα java η οποία μπορεί να υποστηρίξει την αποστολή μηνυμάτων ηλεκτρονικού ταχυδρομείου μέσα από μια εφαρμογή Java. Στο δικό μας σύστημα χρησιμοποιούμε την έκδοση javamail-1.4.4 η οποία είναι συμβατή με την έκδοση Apache-Tomcat και Java compiler που χρησιμοποιούμε [78] .

3.5.8 Kaptcha

Η Kaptcha είναι επίσης μια βιβλιοθήκη κλάσεων Java η οποία χρησιμοποιώντας την τεχνολογία MVC και Java Servlets μας βοηθά στο να δημιουργήσουμε τον έλεγχο captcha (Completely Automated Public Turing test to tell Computers and Humans Apart). Με τον έλεγχο αυτό ελέγχουμε πως τα στοιχεία που αποστέλλονται από την φόρμα εγγραφής δεν τα στέλνει κάποια αυτοματοποιημένη εργασία αλλά κάποιος χρήστης ο οποίος μπορεί να αναγνωρίσει τυχαίους χαρακτήρες που παράγονται από την βιβλιοθήκη και να τους πληκτρολογήσει [79] .

Μετά την ολοκλήρωση της ανάλυσης και του σχεδιασμού του διαδικτυακού συστήματος ψηφοφορίας πιστεύουμε πως έχουν καλυφτεί οι αρχικές απαιτήσεις και τα κριτήρια ασφαλείας που διατυπώθηκαν στο πρώτο κεφάλαιο. Με την χρήση των παραπάνω εργαλείων και σύμφωνα με τον σχεδιασμό που έχουμε κάνει μπορούμε να προχωρήσουμε στην υλοποίηση της εφαρμογής.

4 Υλοποίηση - Παρουσίαση εφαρμογής

4.1 Εγκατάσταση απαιτούμενου λογισμικού εφαρμογής

Σύμφωνα με τις προδιαγραφές της ανάλυσης απαιτήσεων σε λογισμικό του συστήματος πρέπει να εγκατασταθεί ένας servlet container ο Tomcat. Από την επίσημη ιστοσελίδα του λογισμικού προσφέρεται δωρεάν. Χχρησιμοποιήσαμε την έκδοση apache-tomcat-6.0.43. Κατεβάσαμε το bin αρχείο της έκδοσης και το εγκαταστήσαμε στο c:\path του συστήματος μας. Για να λειτουργήσει σωστά χρειάζεται να ορίσουμε κάποιες μεταβλητές του συστήματος:

```
CATALINA_HOME:C:\apache-tomcat-6.0.43;
ορίζουμε το home path του λογισμικού
CLASSPATH:.;C:\ServletDevel;C:\apache-tomcat-
6.0.43\common\lib\servlet-api.jar; C:\apache-tomcat-
6.0.43\common\lib\mysql-connector-java-6.0.43-bin.jar
```

Ορίζουμε στο σύστημα το πού βρίσκονται οι βιβλιοθήκες java (jar) που χρειάζεται να χρησιμοποιεί ο tomcat. Είτε όταν θέλει να μεταγλωττίσει τις σελίδες jsp είτε όταν θέλει να χρησιμοποιήσει την mysql ή να διαχειριστεί τα servlets. Στο C:\ServletDevel γράφουμε όλα τα αρχεία java τα οποία δημιουργούμε και έπειτα πρέπει να μεταγλωττιστούν σε class files.

Για την μεταγλώττιση χρησιμοποιήσαμε τον μεταγλωττιστή jdk1.7.0_75 τον οποίο και κατεβάσαμε από την επίσημη ιστοσελίδα της java http://java.sun.com/ για τον οποίο και πάλι ορίσαμε μεταβλητές συστήματος

```
JAVA_HOME : C:\ jdk1.7.0_75
```

Για την εκτέλεση των εφαρμογών αυτών από το λειτουργικό μας σύστημα πρέπει να ορίσουμε στη μεταβλητή συστήματος τα paths στα οποία βρίσκονται τα εκτελέσιμα αρχεία των εφαρμογών.

```
C:\jdk1.7.0_75\bin;C:\apache-tomcat\6.0.43\bin;
C:\ServletDevel\; C:\mysql\bin
```

Για την εγκατάσταση της mysql χρησιμοποιήσαμε αρχεία από το επίσημο site της mysql www.mysql.com. Τα mysql-5.0.45-win32 και mysql-connector-java-5.1.7 ο οποίος περιέχει βιβλιοθήκες σύνδεσης jdbc με την εφαρμογή μας. Η σύνδεση με την βάση Mysql από τον κώδικα της εφαρμογής μας γίνεται με την κλάση condb.class την οποία και κάνουμε import από οποιαδήποτε αρχείο χρειαστούμε σύνδεση με την mysql.

```
package ivote;
import java.util.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
```

```
public class condb{
   public static Connection getConnection() throws
Exception {
    String driver = "org.gjt.mm.mysql.Driver";
    String url = "jdbc:mysql://localhost/ivote";
    String username = "root";
    String password = "1234";
    Class.forName(driver);
    Connection conn = DriverManager.getConnection(url,
username, password);
    return conn;
  }
}
```

Τα αρχεία της εφαρμογής βρίσκονται στον φάκελο *ROOT* στο φάκελο webapps του tomcat. Για να εκτελεστούν χρειάζεται να πληκτρολογήσουμε το dns name που ακούει ο tomcat. Για τοπική εκτέλεση στον server εκτελούμε localhost και ο tomcat εμφανίζει την αρχική μας σελίδα index.html.

Οι σελίδες html (στατικές σελίδες) βρίσκονται στον *ROOT\i vote* και μπορούν να καλεστούν απευθείας από κάποιον χρήστη στο διαδίκτυο.

Οι σελίδες JSP βρίσκονται στον φάκελο ROOT\WEB-INF\view και δεν μπορούν να καλεστούν από το διαδίκτυο παρά μόνο να τις καλέσει η εφαρμογή.

Οι κλάσεις που ανήκουν στο model και στον controller βρίσκονται στα paths *ROOT\WEB-INF\classes\ivote\model* και *ROOT\WEB-INF\classes\ivote\controller* αντίστοιχα.

Τα servlet καλούνται από τις σελίδες JSP σύμφωνα με την ροή της λειτουργίας του συστήματος. Αν κάποιος χρήστης προσπαθήσει να τα καλέσει περνώντας μεταβλητές που αυτός θέλει μέσα από το URL ελέγχεται το session του και το id του και αν δεν έχει δικαίωμα να τα εκτελέσει προωθείται στην αρχική σελίδα. Επίσης, οι τιμές που εισάγονται ελέγχονται και αν προσπαθήσει να περάσει με αυτόν τον τρόπο μεταφέρεται και πάλι στην κεντρική σελίδα.

Στον φάκελο *C:\apache-tomcat-6.0.43\server\lib* τοποθετούμε όσες επιπλέον βιβλιοθήκες χρησιμοποιούμε και είναι απαραίτητες για την λειτουργία του Tomcat και της εφαρμογής μας. Εκεί τοποθετείται η βιβλιοθήκη mail.jar της javamail.

Ο κώδικας της εφαρμογής και τα αρχεία εγκατάστασής της θα παραδοθούν σε cd-rom μαζί με την τεκμηρίωση της. Στις επόμενες ενότητες θα παρουσιαστούν μεμονωμένα κάποια αποσπάσματα του κώδικα τα οποία χρησιμοποιήθηκαν για την επίτευξη της ασφάλειας του συστήματος.

4.2 Δημιουργία πιστοποιητικού

Ο Apache Tomcat όπως προείπαμε συνεργάζεται με την εργαλειοθήκη ανοιχτού κώδικα OpenSSL για να επιτύχουμε την εγκαθίδρυση κρυπτογραφημένων ασφαλών επικοινωνιών μεταξύ των χρηστών και του συστήματος. Σε αυτήν την ενότητα θα περιγραφεί η ρύθμιση

του εξυπηρετητή με σκοπό να απαντάει μόνο σε αιτήσεις ασφαλούς σύνδεσης από τους χρήστες στην πόρτα 443. Στο αρχείο ρύθμισης του Tomcat στο Path C:\apache-tomcat-6.0.43\conf\server.xml απενεργοποιούμε μετατρέποντας σε σχόλιο τις γραμμές κώδικα του αρχείου που έχουν να κάνουν με την επικοινωνία HTTP και ενεργοποιούμε με τις παρακάτω εντολές την επικοινωνία μέσω πρωτοκόλλων TLSv1.1 και TLSv1.2

```
<Connector port="443" protocol="HTTP/1.1"
SSLEnabled="true"
maxThreads="150" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS"
keystorePass="baktisnikos"/>
```

Με αυτόν τον τρόπο και καθορίζοντας και στους δρομολογητές του δικτύου μας πως οποιοσδήποτε καλεί τον εξυπηρετητή της εφαρμογής θα μπορεί να συνδέεται με αυτόν μόνο μέσω της πόρτας 443 διασφαλίζουμε πως από τη στιγμή που κάποιος συνδεθεί με τον εξυπηρετητή θα ξεκινήσει το TLS Handshake της διαδικτυακής εφαρμογής του με τον Server μας. Σε περίπτωση που κάποιος προσπαθήσει να συνδεθεί με απλή επικοινωνία HTTP δεν θα μπορέσει να εκτελέσει την εφαρμογή.

Κατά την διαδικασία του TLS HandsShake ο εξυπηρετητής πρέπει να στείλει στον χρήστη ένα πιστοποιητικό με τα στοιχεία του και το δημόσιο κλειδί του. Θα δημιουργήσουμε ένα πιστοποιητικό για την επίδειξη της εφαρμογής αλλά στο πλαίσιο της παρουσίασης δεν θα το πιστοποιήσουμε μέσω μιας αρχής πιστοποίησης [80]. Για τον λόγο αυτό οι web clients των χρηστών θα εμφανίζουν προειδοποίηση πως το πιστοποιητικό δεν είναι ασφαλές.

Αν είχαμε ένα πιστοποιημένο από αρχή πιστοποίησης πιστοποιητικό και θέλαμε να το εισάγουμε στο σύστημα θα εκτελούσαμε την εντολή αντικαθιστώντας τα ονόματα των αρχείων με του δικού μας.

```
openssl pkcs12 -export -in mycert.crt -inkey mykey.key -
out mycert.p12 -name tomcat -CAfile myCA.crt  -caname root
-chain
```

Για την δημιουργία δικού μας πιστοποιητικού θα χρησιμοποιήσουμε το εργαλείο keytool της Java. Με την παρακάτω εντολή ξεκινά η διαδικασία δημιουργίας του κλειδιού ενώ ορίζουμε πως ο αλγόριθμος είναι ο RSA. Μετά μας ζητείται να δώσουμε έναν κωδικό (δίνουμε τον κωδικό που δηλώνουμε και στο αρχείο conf του tomcat) για το κλειδί που δημιουργούμε και τα στοιχεία που θέλουμε να περιέχει.

keytool" -genkey -alias tomcat -keyalg RSA

Όταν το πιστοποιητικό είναι έτοιμο από έναν web client καλώντας τον εξυπηρετητή και αφού έχει εγκαθιδρυθεί η ασφαλής σύνδεση μπορούμε να δούμε το πιστοποιητικό που δημιουργήσαμε. Στην ενότητα παρουσίασης της εφαρμογής παρατίθεται screenshot με το πιστοποιητικό και με έναν έλεγχο των δεδομένων που κυκλοφορούν στο δίκτυο μέσω του whireshark ώστε να διαπιστώσουμε πώς κρυπτογραφούνται τα δεδομένα.

Κρυπτογράφηση PKCS #1 RSA Modulus (2048 bits):
ac 37 a9 30 43 ee f5 d8 05 24 77 f8 51 d1 f0 ea
ba bc f1 01 4a 83 9b a2 6b 96 d5 84 39 ff f3 be
8d 84 a4 78 a1 82 84 02 8b b2 e2 1a ad f9 2b 6f
52 ce 36 9e 1b ab 37 a6 5a c2 8b 54 32 c4 92 72
45 3e 65 24 fb ec 35 ef 5e 0a 1c ed 92 08 50 fa
ef e2 86 c7 85 94 fe eb da 41 7e a0 bd a8 75 69
f6 51 91 b8 7a e4 68 38 62 fd 96 bc 72 9e d8 d7
f2 42 d5 1a 53 52 c9 28 98 11 43 a3 29 38 df a0
bd 10 32 61 a1 62 0e 82 db db 60 f7 e9 90 c2 29
8e d8 19 95 a5 40 c7 f5 e8 f7 c6 13 ee 02 01 58
a6 01 a1 c1 85 bb 5d 08 44 7f 61 13 f6 65 a1 4d
fa 5b 15 22 5c e0 f9 a2 54 fb 10 de 65 c4 d5 d0
7b 7f 00 6a b3 43 1a 07 f6 54 2c 4b 9f 2a 1d 0b
42 08 eb f8 6d c1 05 93 f1 69 76 1f b4 ae 56 a6
a7 ca 22 5d 00 83 87 88 2a 00 e2 28 11 66 93 0d
f4 fe cf fe f1 9f 3f 2c 09 0b 0d 86 b3 c3 51 6f

Πίνακας 3. Δημόσιο κλειδί εξυπηρετητή

4.3 Αυτοματοποιημένη δημιουργία credentials

Όπως επισημάνθηκε κατά το σχεδιασμό τα στοιχεία σύνδεσης των ψηφοφόρων δημιουργούνται αυτόματα από το σύστημα μόλις επιβεβαιωθούν τα στοιχεία του. Αυτό συμβαίνει με την εκτέλεση του κώδικα της ρουτίνας writeIvoter (int voterId) της κλάσης ivoter. Σε αυτήν την ρουτίνα με την παρακάτω εντολή κόβουμε τις τιμές των μεταβλητών που περιέχουν τα στοιχεία του χρήστη και ενώνοντας τα κομμάτι σχηματίζεται το username.

```
Username=voter.surname.substring(0,2)
+voter.name.substring(0,2)+"."
+String.valueOf(voter.afm).substring(2,7)
+voter.fathersname.substring(0,1)+"."
+voter.at.substring(2,6)
+ voter.registeredPass.substring(1,3);
```

Λόγω του ότι υπάρχει κάποια πιθανότητα από σύμπτωση να παραχθεί το ίδιο username για δύο χρήστες και να υπάρχει πρόβλημα κατά την σύνδεση με τις παρακάτω εντολές αν βρεθεί ίδιο username στο σύστημα προσθέτουμε στο καινούριο που μόλις δημιουργείται έναν τυχαίο διψήφιο αριθμό. Αν συμπέσει και πάλι προσθέτουμε ακόμα έναν κ.ο.κ.

```
String query = "select username from ivoters where
username='"+username + "' ";
rs = stmt.executeQuery(query);
while(rs.next())
{
Random rnd = new Random();
int rndInt = rnd.nextInt(voter.afm);
```

```
username= username +
String.valueOf(rndInt).substring(0,2);
query = "select username from ivoters where username = '"
+ username + "' ";
rs = stmt.executeQuery(query);
}
```

Ο κωδικός χρήστη παράγεται εντελώς τυχαία με την χρήση της ρουτίνας generatePswd της κλάσης PswGenerator. Σε αυτήν την ρουτίνα στέλνουμε σαν εισόδους το ελάχιστο και το μέγιστο μέγεθος ,τον αριθμό των γραμμάτων, το σύνολο των αριθμών και το σύνολο ων συμβόλων που θέλουμε να έχει ο νέος κωδικός. Για λόγους ασφαλείας τα σύμβολα που θα χρησιμοποιηθούν έχουμε δηλώσει πως είναι τα !@#$^*_+-" . Με την βοήθεια τυχαίων αριθμών παράγουμε τυχαίους χαρακτήρες και επιστρέφουμε τον κωδικό.

```
public static char[] generatePswd(int minLen, int maxLen,
int noOfCAPSAlpha,int noOfDigits, int noOfSplChars) {
if(minLen > maxLen)
throw new IllegalArgumentException("Min. Length > Max.
Length!");
if( (noOfCAPSAlpha + noOfDigits + noOfSplChars) > minLen )
throw new IllegalArgumentException
("Min. Length should be atleast sum of (CAPS, DIGITS, SPL
CHARS) Length!");
Random rnd = new Random();
int len = rnd.nextInt(maxLen - minLen + 1) + minLen;
char[] pswd = new char[len];
int index = 0;
for (int i = 0; i < noOfCAPSAlpha; i++)
{
index = getNextIndex(rnd, len, pswd);
pswd[index] =
ALPHA_CAPS.charAt(rnd.nextInt(ALPHA_CAPS.length()));
}
for (int i = 0; i < noOfDigits; i++)
{
index = getNextIndex(rnd, len, pswd);
pswd[index] = NUM.charAt(rnd.nextInt(NUM.length()));
}
for (int i = 0; i < noOfSplChars; i++)
{
index = getNextIndex(rnd, len, pswd);
pswd[index] =
SPL_CHARS.charAt(rnd.nextInt(SPL_CHARS.length()));
}
for(int i = 0; i < len; i++)
{
if(pswd[i] == 0)
{pswd[i] = ALPHA.charAt(rnd.nextInt(ALPHA.length()));}
}
return pswd;
}
```

Ο κωδικός που παράγουμε έχει μήκος 16 χαρακτήρων οι οποίοι ανήκουν μοιρασμένα στην κάθε ομάδα χαρακτήρων που δημιουργεί η ρουτίνα. Αφού εξαχθούν τα στοιχεία σύνδεσης αποθηκεύονται στην Βάση Δεδομένων και αποστέλλονται στον χρήστη μέσω email όπως θα δούμε στην επόμενη ενότητα.

4.4 Αποστολή Credentials με email

Η αποστολή email γίνεται με την βοήθεια της java βιβλιοθήκης javamail.jar.Για να την χρησιμοποιήσουμε στο σύστημα χρησιμοποιούμε τις παρακάτω εντολές. Σε αυτές αρχικά δηλώνουμε τα στοιχεία του mailserver που θα χρησιμοποιήσουμε και το email του χρήστη. Έπειτα δημιουργούμε το μήνυμα που θέλουμε να εμφανιστεί στον παραλήπτη στο οποίο προσθέτουμε τα στοιχεία σύνδεσης που έχουν δημιουργηθεί. Τέλος, ενθυλακώνουμε τα αντικείμενα που χρειάζονται στο αντικείμενο transport της κλάσης Transport της βιβλιοθήκης javamail.jar και αποστέλλουμε το email.

```
String USER_NAME = "baknik@hotmail.com";
String PASSWORD ="......";
String RECIPIENT = voter.email;
String from = USER_NAME;
String pass = PASSWORD;
String[] to = { RECIPIENT };

String subject = "Σύστημα ηλεκτρονικής ψηφοφορίας";
String body = "<html><body>Γειά σας κ/κα "+ voter.surname
+ " " + voter.name+"</br> Ευχαριστούμε για την εγγραφή σας
στο σύτημα ηλεκτρονικής ψηφοφορίας<br /> <b>Username :</b>
" + username + "<br /><b>Password :</b>" + password +"<br
/> Χρησιμοποιήστε τα παραπάνω στοιχεία για να συνδεθείτε
στο σύστημα και να ψηφίσετε. </br></body></html> " ;
Properties props = System.getProperties();
String host = "smtp.live.com";
props.put("mail.smtp.starttls.enable", "true");
props.put("mail.smtp.host", host);
props.put("mail.smtp.user", from);
props.put("mail.smtp.password", pass);
props.put("mail.smtp.port", "587");
props.put("mail.smtp.auth", "true");
Session session = Session.getDefaultInstance(props);
MimeMessage message = new MimeMessage(session);
try {
message.setFrom(new InternetAddress(from));
[] toAddress = new InternetAddress[to.length];
for( int i = 0; i < to.length; i++ )
{toAddress[i] = new InternetAddress(to[i]);}
for( int i = 0; i < toAddress.length; i++)
{message.addRecipient(Message.RecipientType.TO,
toAddress[i]);}
message.setSubject(subject);
message.setContent(body, "text/html; charset=UTF-8");
Transport transport = session.getTransport("smtp");
transport.connect(host, from, pass);
transport.sendMessage(message,
message.getAllRecipients());
transport.close();
}
```

4.5 Έλεγχοι ασφαλείας

Κατά τη διάρκεια του σχεδιασμού στο προηγούμενο κεφάλαιο μιλήσαμε για τους διπλούς ελέγχους που πραγματοποιούνται στα δεδομένα που εισέρχονται στο σύστημα. Οι έλεγχοι αυτοί πραγματοποιούνται με την κλήση της ρουτίνας check στην αρχή κάθε controller του συστήματος ο οποίος είναι υπεύθυνος να επεξεργαστεί τα δεδομένα και να μας προωθήσει ανάλογα στις διαδικασίες του συστήματος. Θα δώσουμε ένα παράδειγμα ελέγχου του πεδίου ΑΦΜ καθώς και σε όλα τα υπόλοιπα πεδία πραγματοποιείται ο ανάλογος έλεγχος με βάση τα δεδομένα που περιμένουμε να εισαχθούν ανά περίσταση. Στο ΑΦΜ περιμένουμε να εισαχθεί ένας ακέραιος αριθμός μεγέθους 9 ψηφίων. Δεν επιτρέπουμε να εισαχθούν στο σύστημα ούτε χαρακτήρες αλλά κυρίως σύμβολα τα οποία με την κατάλληλη χρήση μπορεί να προκαλέσουν δυσλειτουργίες. Ο πρώτος έλεγχος γίνεται στο web client του χρήστη με τη χρήση javascript. Για τον λόγο ότι κάποιος είναι πολύ εύκολο να έχει απενεργοποιήσει στο web client που χρησιμοποιεί την χρήση js τα δεδομένα ελέγχονται και στον controller πριν εκτελέσει οποιαδήποτε άλλη λειτουργία:

```
flag =Check.check(request.getParameter("afm"),flag,"afm");
```

Με την κλήση της ρουτίνας check στέλνουμε ως παράμετρο την τιμή που έχει εισαγάγει ο χρήστης, την μεταβλητή flag(=0) η οποία είναι και αυτή που θα επιστραφεί αλλαγμένη αν προκύψει πρόβλημα και τον τύπο του πεδίου ο οποίος σε αυτήν την περίπτωση είναι afm.

```
String ok="";
int mhkos = parString.length();
if (field=="afm" )
{ok = "0123456789";
  if ( mhkos < 9 ||  mhkos > 9 )
    {flag=1;}
}
for (int j=0; j<mhkos; j++)
{if (ok.indexOf(parString.substring(j,j+1) )==-1 )
    {flag=1;}
}
return(flag);
```

Αν ο χρήστης έχει εισάγει μόνο 9ψηφιο αριθμό επιστρέφεται η τιμή 0 και ο controller συνεχίζει την εκτέλεση εντολών αν όχι επιστρέφεται η τιμή 1 και ο controller προωθεί τον χρήστη εκτός συστήματος εμφανίζοντας κατάλληλο μήνυμα.

Μετά από αυτόν τον έλεγχο θα πρέπει να γίνει άλλος ένας ο οποίος αφορά την εξουσιοδότηση των χρηστών. Ο controller λοιπόν αμέσως μετά και πριν κάνει κάποια επεξεργασία των δεδομένων ελέγχει αν ο χρήστης με το συγκεκριμένο id είναι όντως συνδεδεμένος στο σύστημα με την ορθή διαδικασία της αυθεντικοποίησης των στοιχείων του και αν επιτρέπεται να εκτελεί τις λειτουργίες τις οποίες ζητήθηκαν. Στο παράδειγμα που ακολουθεί ελέγχεται αν έχει δικαίωμα ένας χρήστης να εκτελέσει διαδικασίες ως υπεύθυνος της αρχής πιστοποίησης. Αρχικά διαβάζουμε το sessionid του. Έπειτα ελέγχουμε με την βοήθεια της ρουτίνας authenticate της κλάσης sessions αν υπάρχει η εγγραφή με το sessionid αυτό και το userid του με πρόθεμα ca στην Β.Δ. Αν δεν υπάρχει σημαίνει ή πως το session με το οποίο είχε συνδεθεί έληξε και έχει αλλάξει ή πως προσπαθεί να αποκτήσει πρόσβαση κάποιος χωρίς την κατάλληλη εξουσιοδότηση. Σε αυτή

την περίπτωση οδηγείται εκτός συστήματος στην κατάλληλη σελίδα. Αν ταιριάζει με την εγγραφή στη Β.Δ. ο controller εκτελεί την διαδικασία που του ζητήθηκε.

```
HttpSession session = request.getSession(true); Sessions
nowsession                                              =
Sessions.authenticate(""+session.getId(),"ca"+user.getUser
Id());
if          ((nowsession          ==          null)          )
{    address    =    "/WEB-INF/view/expired.jsp";          }
else
{//εκτέλεση της διαδικασίας που ζητήθηκε ……}
```

Όπως αναφέρθηκε στην αρχή της ενότητας ο κώδικας και η εγκατάσταση της εφαρμογής βρίσκεται σε συνοδευτικό CD-rom της συγκεκριμένη εγκατάστασης. Παρουσιάστηκαν κάποια τμήματα του κώδικα τα οποία είχαν να κάνουν με λειτουργίες του συστήματος οι οποίες όπως είχε φανεί κατά τον σχεδιασμό ήταν αρκετά σημαντικές για την ασφαλή λειτουργία του συστήματος. Στην επόμενη ενότητα, όπου ακολουθεί σύντομη παρουσίαση της εφαρμογής, θα παρουσιάσουμε τα παραπάνω αλλά και κάποιες ακόμα λειτουργίες του υλοποιημένου πλέον συστήματος.

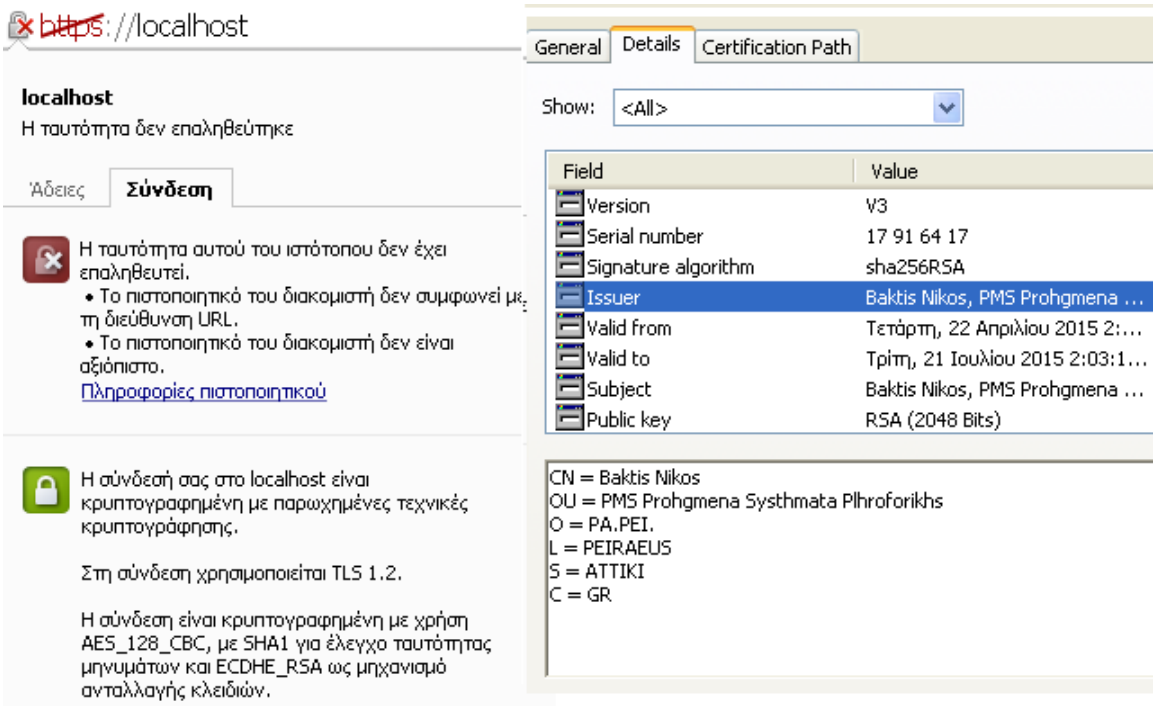
4.6 Παρουσίαση εφαρμογής

Στην αρχική οθόνη της εφαρμογής οι χρήστες μπορούν να επιλέξουν είτε να μεταφερθούν στην περιοχή ψηφοφορίας ώστε να εγγραφούν και να ψηφίσουν, είτε στην περιοχή διαχείρισης ώστε να συνδεθούν σαν έναν από τους τρεις διαχειριστικούς ρόλους του συστήματος.

Εικόνα 2. Αρχική σελίδα συστήματος

Όπως βλέπουμε στο url της σελίδας χρησιμοποιείται πρωτόκολλο HTTPS. Λόγω του ότι δεν είναι πιστοποιημένο από κάποια αρχή πιστοποίησης η εφαρμογή του χρήστη εμφανίζει σφάλμα. Για την εκπόνηση της συγκεκριμένης πτυχιακής δεν θα χρησιμοποιηθεί έγκυρο πιστοποιητικό αλλά αν η εφαρμογή τεθεί σε λειτουργία αυτό επιβάλλεται ώστε να αναγνωρίζουν οι χρήστες και οι εφαρμογές που χρησιμοποιούν πως συνδέονται με τον

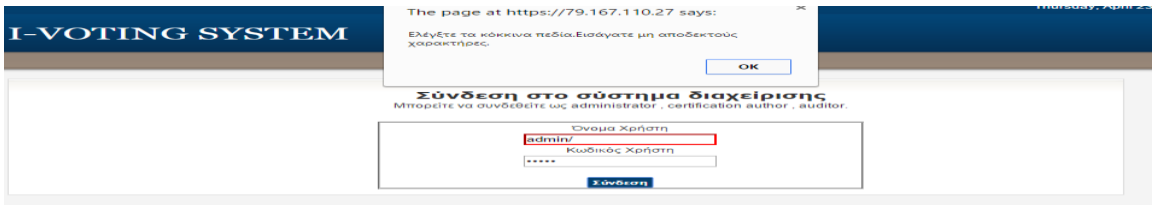
σωστό εξυπηρετητή. Αν ζητήσουμε από την εφαρμογή να δούμε το πιστοποιητικό μπορούμε να δούμε τις πληροφορίες που περιέχει.



Εικόνα 3. Πιστοποιητικό X.509 διαδικτυακού συστήματος ψηφοφορίας

Παρατηρούμε και πάλι πως το πιστοποιητικό δεν μπορεί να επαληθευτεί λόγω της μη πιστοποίησής του από μια έγκυρη αρχή πιστοποίησης. Παρατηρούμε, επίσης, πως έχει εγκατασταθεί ασφαλής σύνδεση με κρυπτογραφία μέσω του πρωτοκόλλου TLS 1.2. Χρησιμοποιείται ο αλγόριθμος RSA για την ασύμμετρη κρυπτογραφία και για την συμμετρική αφού έχει πραγματοποιηθεί η ανταλλαγή κλειδιών και έχει επέλθει συμφωνία ανάμεσα στον εξυπηρετητή και την εφαρμογή του χρήστη ο αλγόριθμος συμμετρικής κρυπτογραφίας AES. Επίσης μπορούμε να δούμε πως το πιστοποιητικό λήγει στις 21 Ιουλίου του 2015 και πως το δημόσιο κλειδί του εξυπηρετητή είναι 256 bytes.

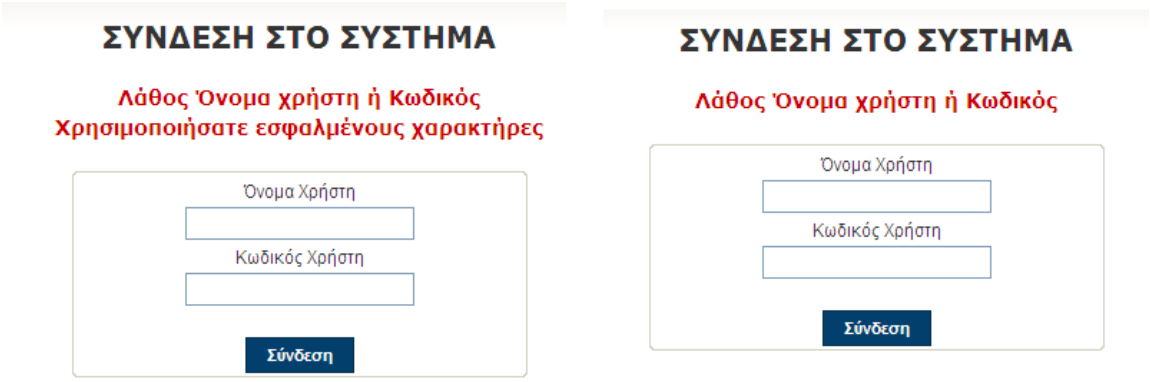
Αρχικά θα συνδεθούμε στο σύστημα ως διαχειριστής για να παρουσιαστούν οι λειτουργίες που μπορεί να εκτελέσει και θα ελέγξουμε συγχρόνως ότι τα δεδομένα μας αποστέλλονται κρυπτογραφημένα. Στην οθόνη σύνδεσης της περιοχής διαχείρισης πληκτρολογούμε το όνομα χρήστη κα τον κωδικό του διαχειριστή. Την στιγμή που θα ζητήσουμε να συνδεθούμε πραγματοποιείται ο έλεγχος με javascript για να διαπιστωθεί πως τα πεδία είναι σωστά συμπληρωμένα. Αν έχουν εισαχθεί εσφαλμένοι χαρακτήρες (όπως η /) η εφαρμογή δεν μας επιτρέπει να συνεχίσουμε και εμφανίζει μήνυμα λάθους.



Εικόνα 4. Μήνυμα λάθους κατά των έλεγχο των δεδομένων με χρήση javascript

Πολλές εφαρμογές επιτρέπουν στου χρήστες του συστήματος να απενεργοποιήσουν την εκτέλεση των εντολών javascript. Καταβαίνουμε ότι μπορεί πολύ εύκολα κάποιος να προσπεράσει τον συγκεκριμένο έλεγχο. Ο έλεγχος αυτός γίνεται κυρίως ώστε να αποφύγουμε λάθη κατά την πληκτρολόγηση πριν αποσταλούν τα δεδομένα στον εξυπηρετητή. Αν απενεργοποιήσουμε την εκτέλεση των εντολών javascipt και δοκιμάσουμε ξανά τα δεδομένα θα αποσταλούν στον εξυπηρετητή αλλά θα ελεγχθούν ξανά πριν προχωρήσουμε σε οποιαδήποτε άλλη διεργασία. Αν έχουμε εισαγάγει μη επιτρεπτούς χαρακτήρες ο controller που είναι υπεύθυνος για τον συγκεκριμένο έλεγχο θα μεταφέρει τον χρήστη στην κατάλληλη σελίδα και θα εμφανιστεί μήνυμα λάθους.

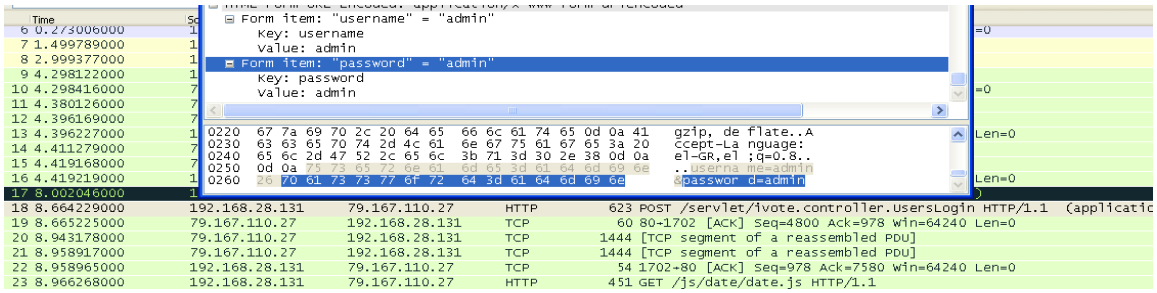
Αφού έχουν ελεγχθεί τα δεδομένα που εισήλθαν στο σύστημα γίνεται αυθεντικοποίηση των στοιχείων που πληκτρολογήσαμε ώστε να πραγματοποιηθεί η σύνδεση του χρήστη. Αν τα στοιχεία δεν ταιριάζουν με τα στοιχεία που έχουμε αποθηκευμένα στη βάση δεδομένων ζητείται από τον χρήστη να προσπαθήσει ξανά χωρίς να γίνεται γνωστό σε ποιο πεδίο έχει κάνει λάθος.



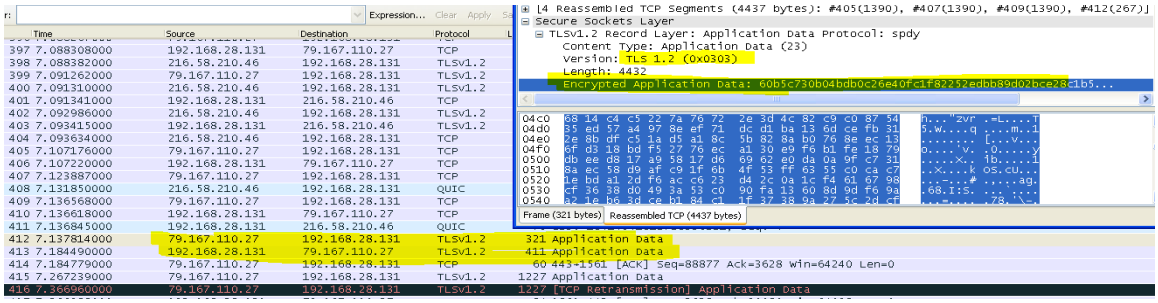
Εικόνα 5. Μηνύματα λάθους κατά την είσοδο στο σύστημα.

Αν δεν έχει συμβεί τίποτα από τα παραπάνω αλλά έχουν εισαχθεί τα σωστά στοιχεία πρόσβασης ο χρήστης μεταφέρεται στην κεντρική σελίδα, του ενός εκ των τριών ρόλων διαχείρισης, για την οποία έχει εξουσιοδότηση. Πριν δούμε τις διαδικασίες που εκτελούν οι τρεις αυτοί ρόλοι θα ελεγχθεί όπως είπαμε και πιο πριν, αν τα δεδομένα που εισαγάγαμε κατά την σύνδεσή μας στο σύστημα είχαν κρυπτογραφηθεί χρησιμοποιώντας ένα εργαλείο ανίχνευσης των πακέτων που λαμβάνει και αποστέλλει ο Η/Υ μας. Παρατηρούμε πως αν εκτελέσουμε την σύνδεση στο σύστημα ενώ είμαστε συνδεδεμένοι με μη κρυπτογραφημένη συνεδρία μπορούμε πολύ εύκολα να δούμε μέσω του εργαλείου τα στοιχεία σύνδεσης που είχαμε εισαγάγει στα πεδία της φόρμας σύνδεσης. Αν όμως επικοινωνήσουμε με τον εξυπηρετητή με την χρήση του πρωτοκόλλου HTTPS και εγκαθιδρυθεί ασφαλής συνεδρία τα δεδομένα δεν είναι εύκολο να διαβαστούν στα πακέτα του δικτύου καθώς έχουν κρυπτογραφηθεί. Με τους παραπάνω τρόπους εξασφαλίζουμε την εμπιστευτικότητα, την αυθεντικοποίηση και την εξουσιοδότηση τρεις από τις βασικές απαιτήσεις ασφαλείας όπως τις προσδιορίσαμε στο κεφάλαιο της ανάλυσης. Εξασφαλίζεται επίσης η διαθεσιμότητα και η ακεραιότητα καθώς κάποιες από τις επιθέσεις σε διαδικτυακά συστήματα βασίζονται είτε σε επιθέσεις ενδιάμεσου χρήστη (man in the middle) την οποία αποφεύγουμε εν μέρει με την κρυπτογράφηση και το

ψηφιακό πιστοποιητικό και σε επιθέσεις με εισαγωγή μη επιτρεπτών χαρακτήρων με σκοπό είτε να περιορίσουν την λειτουργία του συστήματος είτε να αποκτήσουν πρόσβαση στην βάση δεδομένων αυτού.

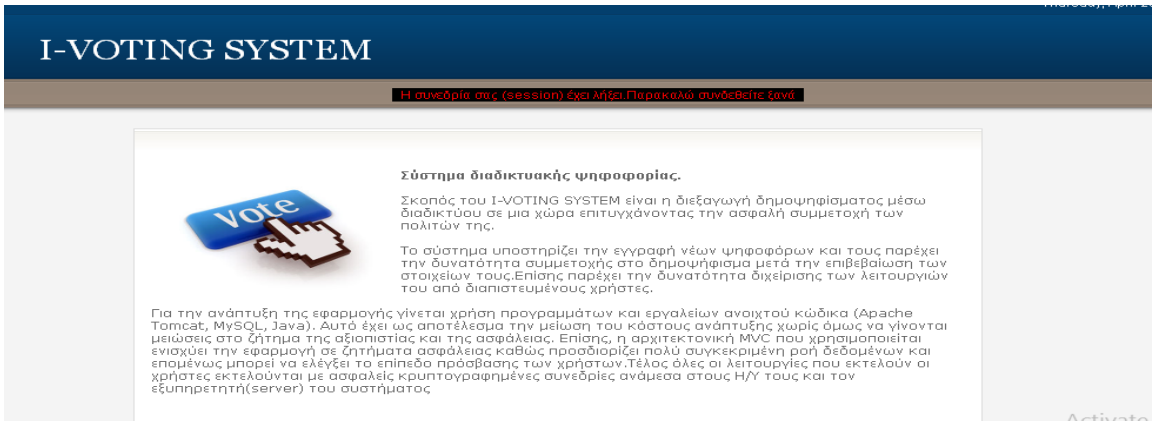


Εικόνα 6. Μη κρυπτογραφημένη σύνδεση με HTTP



Εικόνα 7. Κρυπτογραφημένη σύνδεση με HTTPS - TLS1.2

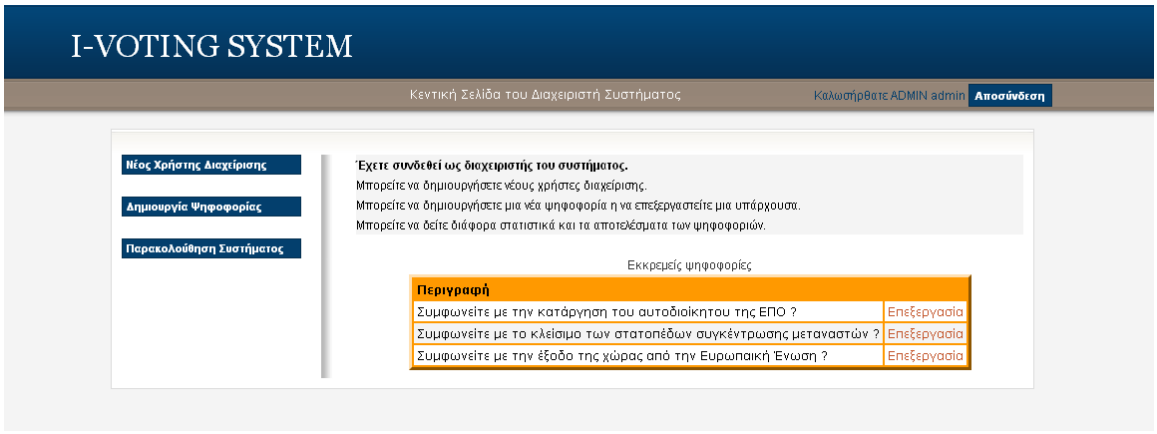
Άλλος ένα μηχανισμός ασφαλείας που χρησιμοποιούμε όπως είπαμε είναι ο έλεγχος των sessions ώστε να γνωρίζουμε ποιος είναι συνδεδεμένος και να μην μπορεί κάποιος να εκτελέσει εργασία του συγκεκριμένου χρήστη με άλλο session. Αν προσπαθήσουμε να εισάγουμε μέσω Link μια άλλη τιμή στα id που μεταφέρουμε ανάμεσα στα διάφορα servlets αυτό θα ελεγχθεί και θα μεταφερθούμε εκτός συστήματος καθώς δεν θα έχουμε εξουσιοδότηση για αυτήν την λειτουργία. Ούτως ή άλλως με την χρήση των μεθόδων doPost ο Tomcat δεν επιτρέπει να επισκεφτούμε καμία σελίδα παρά μόνο αν μας οδηγήσει εκεί ένας controller. Αλλά σε περίπτωση που σε κάποιο σημείο της εφαρμογής χρησιμοποιηθεί η μέθοδος doGet αυτό δεν θα επιτραπεί με την βοήθεια του παραπάνω μηχανισμού. Το ίδιο θα συμβεί και εάν έχει λήξει το session ή αν ο χρήστης συνδεθεί και από κάποια άλλη εφαρμογή και δημιουργήσει καινούριο session.



Εικόνα 8. Μεταφορά χρήστη εκτός συστήματος μετά τον έλεγχο του session

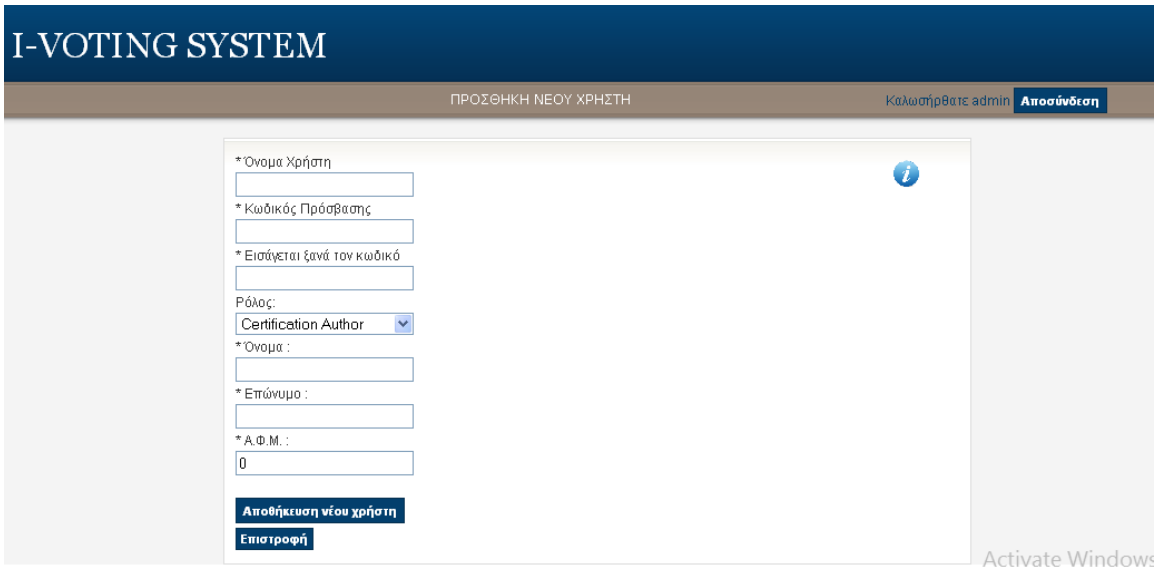
Όλοι οι χρήστες ελέγχονται και οι διεργασίες που εκτελούν ελέγχονται με τους παραπάνω τρόπους .Ο κάθε χρήστης όμως εκτελεί τις δικές του διεργασίες.

Ο admin, όπως έχουμε σχεδιάσει, μπορεί να δημιουργήσει καινούριους χρήστες και νέες ψηφοφορίες. Επίσης μπορεί να παρακολουθήσει κάποια στατιστικά του συστήματος και να δει τα αποτελέσματα των ψηφοφοριών. Στην κεντρική του σελίδα εκτός των άλλων μπορεί να δει τις εκκρεμείς ψηφοφορίες και να τις επεξεργαστεί ώστε να τις ενεργοποιήσει.



Εικόνα 9. Κεντρική σελίδα admin.

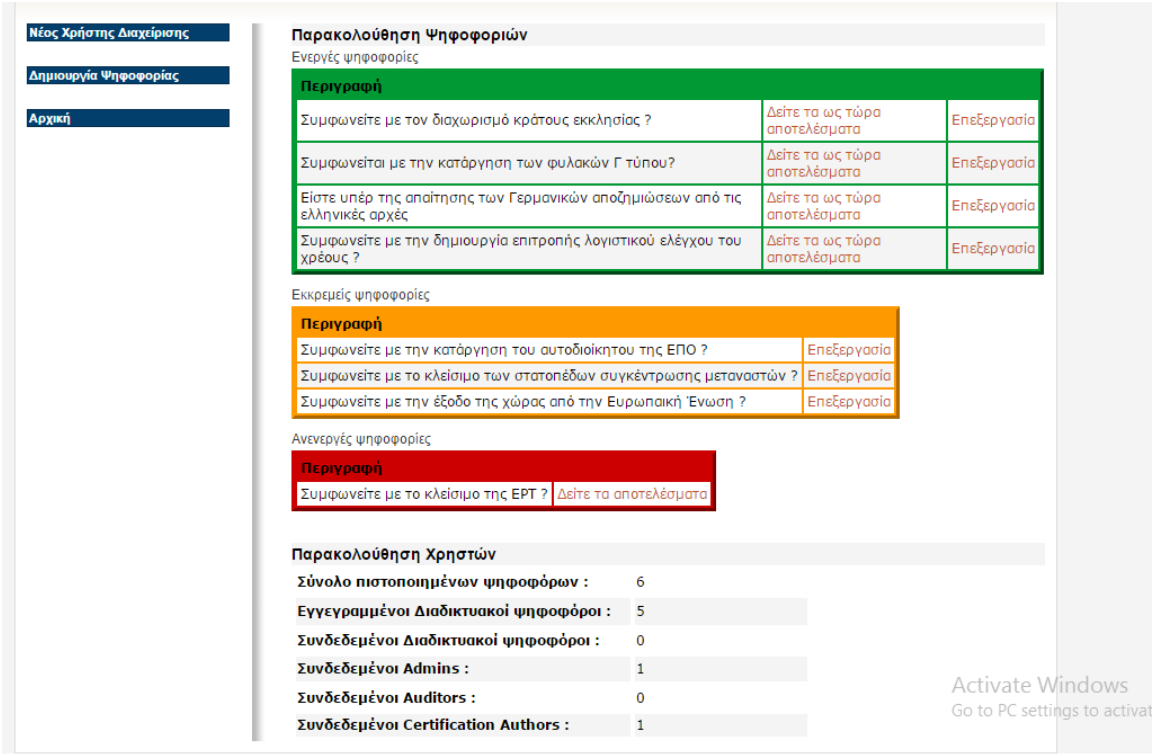
Κατά την δημιουργία νέων χρηστών ή ψηφοφοριών ελέγχονται και πάλι τα πεδία εισαγωγής χαρακτήρων για τυχόν εσφαλμένους. Επίσης, κατά την εισαγωγή νέων κωδικών ζητείται πάντα επιβεβαίωση για να αποφευχθεί λάθος κατά την πληκτρολόγηση.



Εικόνα 10. Μεταφορά χρήστη εκτός συστήματος μετά τον έλεγχο του session

Στην οθόνη παρακολούθησης ο διαχειριστής μπορεί να δει όλες τις ψηφοφορίες, τα αποτελέσματά τους και στατιστικά για αυτές αλλά και γενικότερα για το σύστημα. Μπορεί να δει τον αριθμό των πιστοποιημένων και των εγγεραμμένων ψηφοφόρων ενώ επίσης μπορεί να δει και τον αριθμό των χρηστών που είναι συνδεδεμένοι στο σύστημα. Στις

ψηφοφορίες εκτός από τα αποτελέσματα βλέπει το ποσοστό συμμετοχής και μπορεί να συγκρίνει το άθροισμα των καταγεγραμμένων ψήφων με το άθροισμα των ψηφοφόρων που έχουν καταγραφεί στο σύστημα πως ψήφισαν ώστε να διαπιστώσει κάποια παρατυπία στην ψηφοφορία.

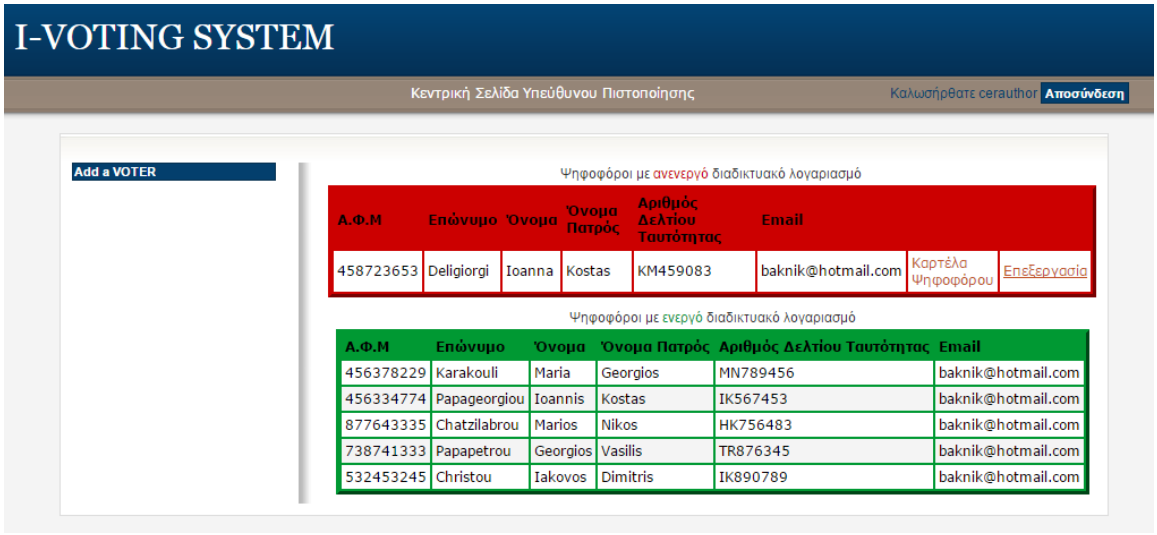


Εικόνα 11. Οθόνη παρακολούθησης συστήματος

Εικόνα 12. Αποτελέσματα και στατιστικά ψηφοφορίας

Ο χρήστης παρακολούθησης συνδέεται στο σύστημα και έχει δικαίωμα να εκτελέσει μόνο την διεργασία της θέασης των αποτελεσμάτων με κάποιους περιορισμούς. Βλέπει τις ίδιες οθόνες με τον διαχειριστή εκτός από τους χρήστες που είναι συνδεδεμένοι καθώς ο βασικός σκοπός αυτού του ρόλου είναι η θέαση της πορείας των ψηφοφοριών.

Ο υπεύθυνος αρχής πιστοποίησης από την άλλη έχει εντελώς διαφορετικό ρόλο στο σύστημα. Στην κεντρική του οθόνη βλέπει μια λίστα με τους πιστοποιημένους ψηφοφόρους και μπορεί να επεξεργαστεί τα στοιχεία μόνο αυτών οι οποίοι δεν έχουν εγγραφεί ακόμα στο σύστημα.

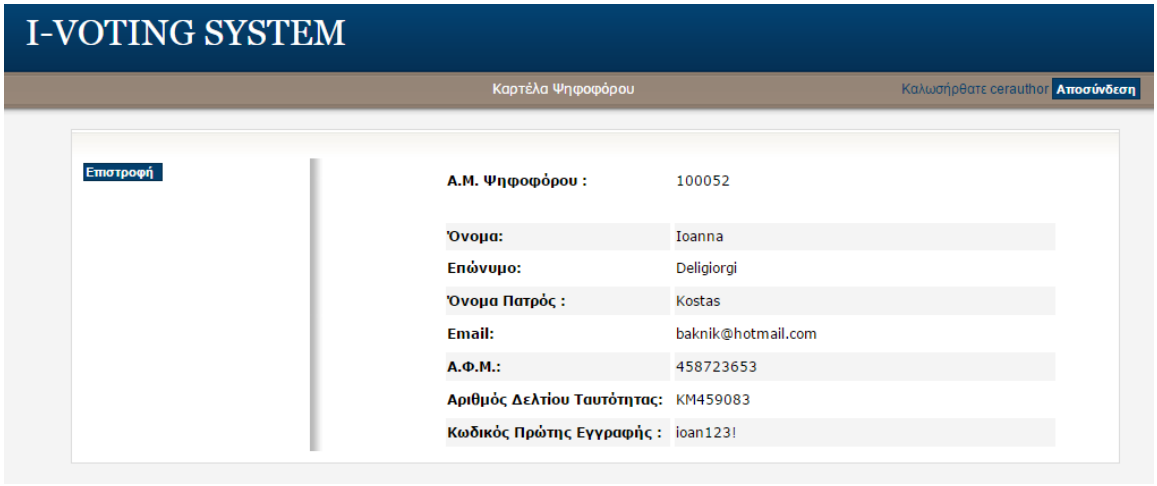


Εικόνα 13. Κεντρική οθόνη υπεύθυνου αρχής πιστοποίησης

Η βασική διεργασία αυτού του ρόλου είναι να πιστοποιεί νέους χρήστες. Οι τιμές που εισάγονται στα πεδία της φόρμας ελέγχονται με τον ίδιο τρόπο όπως περιγράψαμε και παραπάνω. Ένας επιπλέον έλεγχος που πραγματοποιείται εδώ είναι ο έλεγχος του ΑΦΜ. Με την χρήση της τεχνολογίας AJAX ο υπεύθυνος πιστοποίησης κάθε φορά που πληκτρολογεί ενημερώνεται αν είναι έγκυρα τα στοιχεία που εισάγει και όταν έχει εισαχθεί ολόκληρο το ΑΦΜ αν δεν είναι ξανά καταγεγραμμένο στην εφαρμογή. Με αυτόν τον τρόπο αλλά και τους ελέγχους στα πεδία εισαγωγής γίνεται έλεγχος σφαλμάτων και πιθανών παρατυπιών.

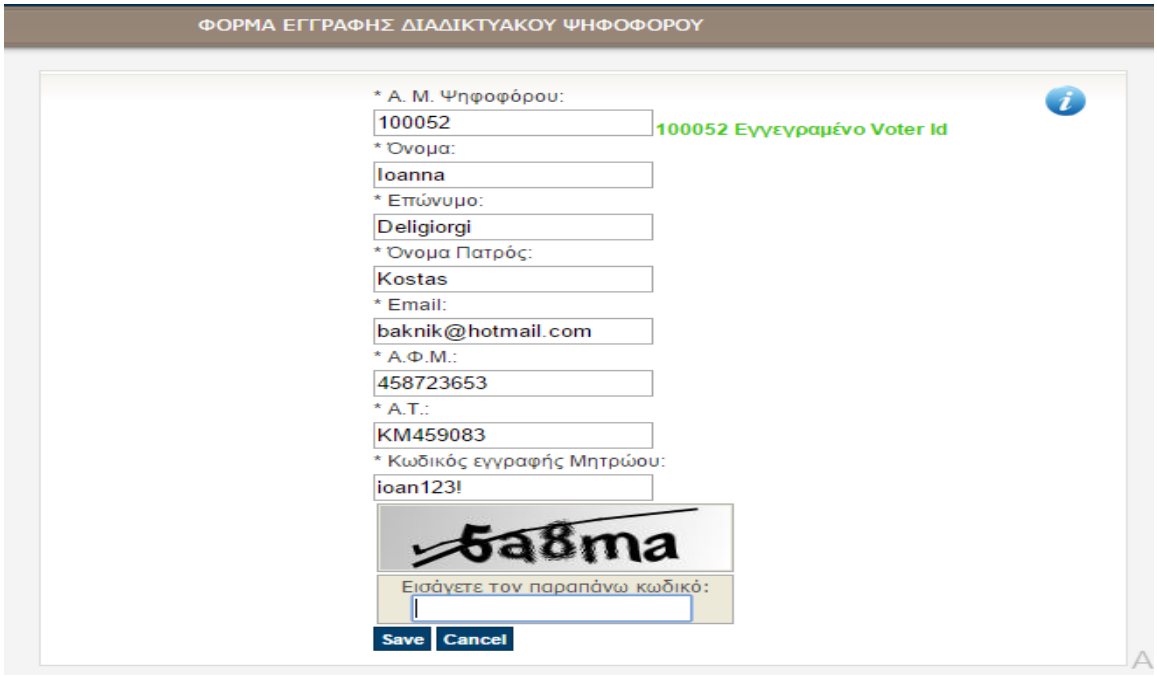
Εικόνα 14. Οθόνη πιστοποίησης ψηφοφόρου.

Όταν ολοκληρώσει την εγγραφή μπορεί να δει και να εκτυπώσει τα στοιχεία του υποψήφιου στην ανάλογη οθόνη.



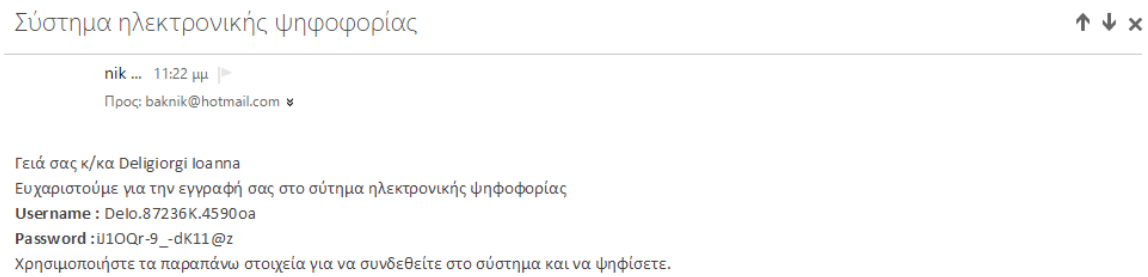
Εικόνα 15. Οθόνη πιστοποίησης ψηφοφόρου.

Ο πολίτης που έχει παραλάβει την εκτύπωση πιστοποίησης μπορεί να εγγραφεί στο σύστημα. Θα πρέπει να μεταφερθεί στον τομέα ψηφοφορίας του συστήματος. Εκεί αν επιλέξει εγγραφή θα εισέλθει στην φόρμα εγγραφής. Και σε αυτήν γίνεται έλεγχος δεδομένων. Εδω χρησιμοποιείται η τεχνολογία AJAX για να ενημερώσει τον ψηφοφόρο πως ο Αριθμός Μητρώου είναι αποθηκευμένος στο σύστημα οπότε και μπορεί να εγγραφεί.



Εικόνα 16. Οθόνη εγγραφής ψηφοφόρου.

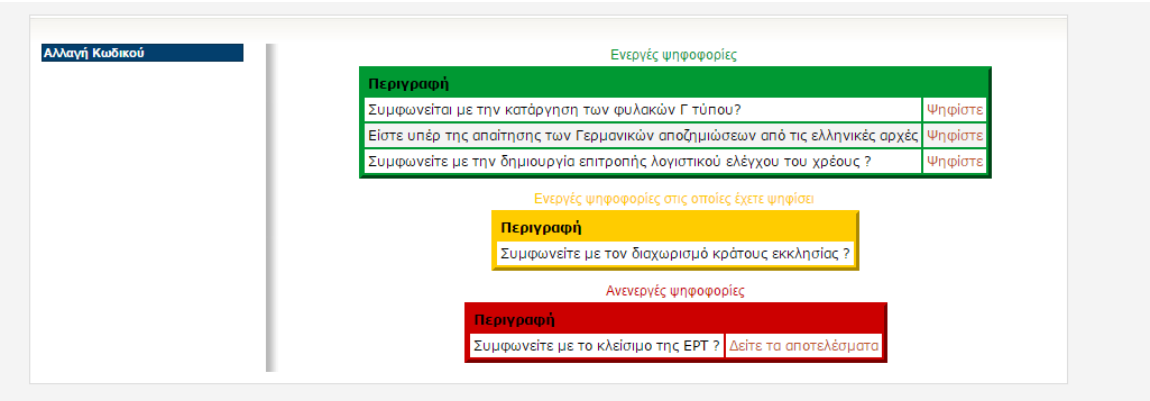
Όπως βλέπουμε και στην εικόνα 16 ο ψηφοφόρος θα πρέπει να συμπληρώσει τα στοιχεία του όπως ακριβώς τα παρέλαβε από την αρχή πιστοποίησης και να επαληθεύσει τον κωδικό που είχε παραδώσει κατά την πιστοποίηση του και έναν κωδικό ο οποίος εμφανίζεται στην οθόνη. Ο κωδικός κατά την εγγραφή χρησιμοποιείται ως πρώτος έλεγχος στο να μην μπορεί κάποιος που ξέρει τα προσωπικά στοιχεία ενός πολίτη να κάνει εγγραφή στο σύστημα πριν από αυτόν. Ο κωδικός της εικόνας χρησιμοποιείται για να αποφύγουμε επιθέσεις από μηχανές που ίσως να προσπαθήσουν να εγγράψουν πλαστούς ψηφοφόρους. Αν πληκτρολογήσει λάθος κάποιο από τα στοιχεία του ή τον κωδικό της οθόνης του ζητείται να προσπαθήσει ξανά .Αν επιβεβαιωθούν όλα τα στοιχεία του αποστέλλεται email στο προσωπικό λογαριασμό email που έχει καταχωρήσει.



Εικόνα 17. Email με στοιχεία εγγραφής.

Με τα στοιχεία αυτά μπορεί να συνδεθεί και να μεταφερθεί στην κεντρική του οθόνη.Για την σύνδεση του διαδικτυακού ψηφοφόρου λειτουργούν οι ίδιοι μηχανισμοί ασφαλείας όπως και των υπολοίπων χρηστών. Στην κεντρική οθόνη μπορεί να αλλάξει τον κωδικό το οποίο εξασφαλίζει πλέον πως κανείς άλλος εκτός από τον ψηφοφόρο δεν μπορεί να συνδεθεί για λογαριασμό του. Για την αλλαγή κωδικού θα πρέπει να εισάγει τον κωδικό που του στάλθηκε με email και δύο φορές έναν καινούριο.

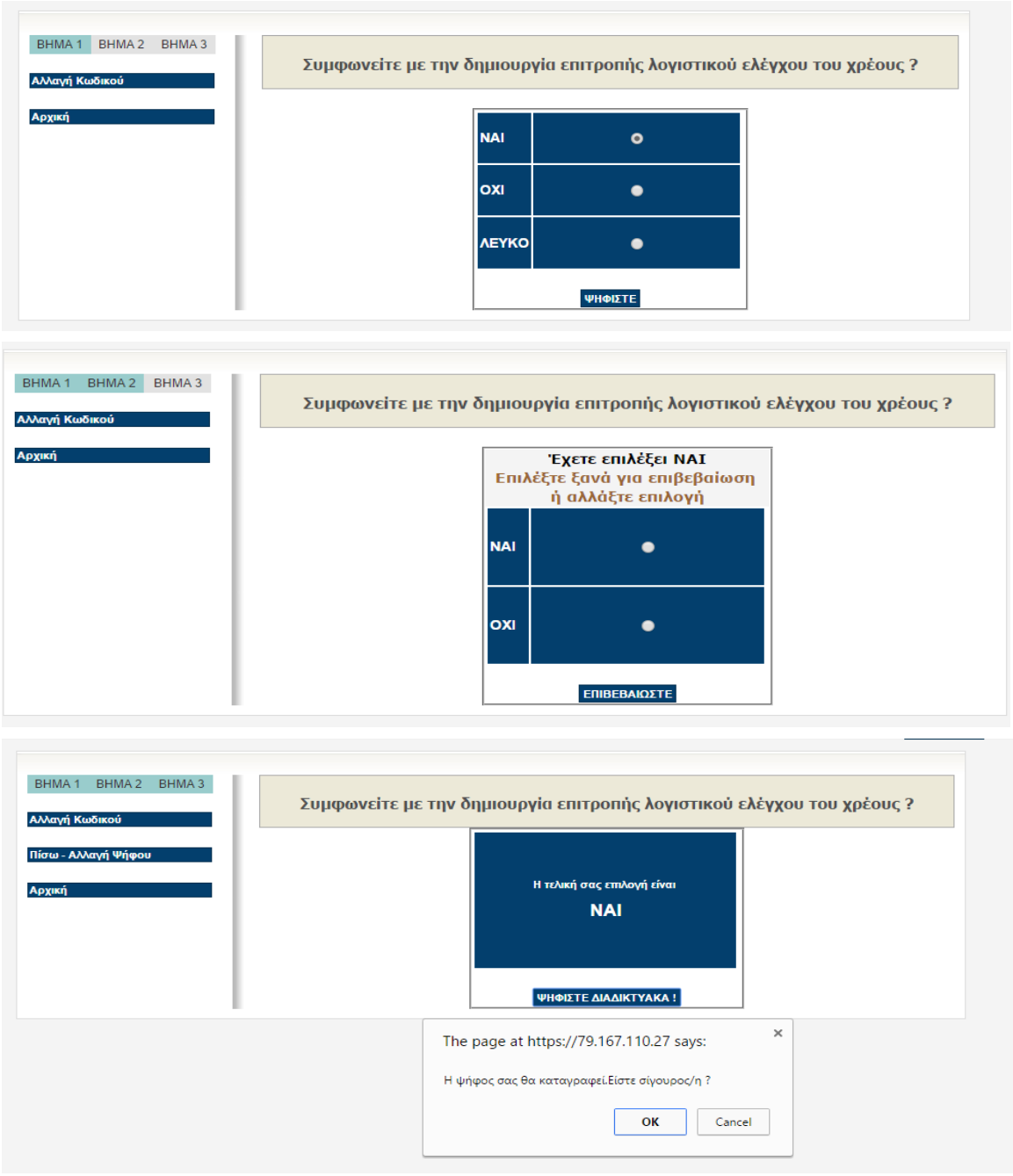
Πλέον μπορεί να προχωρήσει στην διαδικασία της ψηφοφορίας. Στην κεντρική οθόνη του ψηφοφόρου εμφανίζονται όλες οι ενεργές ψηφοφορίες και έχει επιλογή να ψηφίσει μόνο όσες δεν έχει συμμετάσχει ξανά.



Εικόνα 18. Email με στοιχεία εγγραφής.

Επιλέγοντας την ψηφοφορία στην οποία θέλει να συμμετάσχει μεταφέρεται στο πρώτο βήμα τη ψηφοφορίας. Η ψηφοφορία όπως έχουμε πει κατά τον σχεδιασμό εκτελείται σε

τρία βήματα. Στο πρώτο ο ψηφοφόρος κάνει την επιλογή του και ακολουθεί διπλή επιβεβαίωση μέσω των επόμενων βημάτων. Σε κάθε βήμα μπορεί αν το αποφασίσει να αλλάξει επιλογή και να ξεκινήσει ξανά από το πρώτο βήμα. Με αυτόν τον τρόπο αποφεύγουμε τις λανθασμένες ψήφους από βιαστική κίνηση ή δυσκολία προσαρμογής στην εφαρμογή.



Εικόνα 19. Επιβεβαίωση τριών βημάτων κατά την ψηφοφορία.

Όταν ολοκληρωθεί η διαδικασία επιστρέφει στην αρχική του σελίδα όπου μπορεί να συμμετάσχει στις υπόλοιπες ψηφοφορίες

5 Συμπεράσματα

5.1 Σύγκριση με το υπάρχων εκλογικό σύστημα

Συνοψίζοντας, μετά την δημιουργία ενός διαδικτυακού συστήματος το οποίο μπορεί να καλύψει τα κριτήρια ενός ασφαλούς και δίκαιου συστήματος ψηφοφορίας και να υπόκειται στο νομικό πλαίσιο το οποίο πρέπει να διέπει ένα οποιοδήποτε δημοκρατικό σύστημα ψηφοφορίας, μπορούμε να πούμε πως έχει έρθει ο καιρός να προχωρήσει έστω σε πιλοτικές δοκιμές χρήσης ενός τέτοιου συστήματος και η δική μας χώρα όπως έχουν ήδη κάνει αρκετές άλλες. Επίσης αποδεικνύεται πως θα πρέπει να ενταθεί η έρευνα και οι προσπάθειες από διάφορους φορείς ώστε να δημιουργηθεί ένα τέτοιο σύστημα ψηφοφορίας το οποίο θα κερδίζει την εμπιστοσύνη των πολιτών λόγω της αξιοπιστίας του.

Τα πλεονεκτήματα από την χρήση ενός διαδικτυακού συστήματος σε σχέση με το εκλογικό σύστημα που χρησιμοποιεί τώρα το μεγαλύτερο ποσοστό των χωρών στον πλανήτη θα είναι αρκετά. Τα αποτελέσματα δεν θα χρειάζονται κόπο και χρόνο μέχρι να καταμετρηθούν όλες οι ψήφοι αλλά θα καταμετρούνται και θα εξάγονται αυτόματα. Οι ψηφοφορίες δεν θα είναι ανάγκη να διεξάγονται μόνο κάποιες συγκεκριμένες ημέρες και ώρες αλλά πολύ πιο συχνά και ο χρόνος όπου θα είναι «ανοιχτές οι κάλπες» μπορεί να είναι μεγαλύτερος. Το κόστος της διαδικασίας μιας διαδικτυακής ψηφοφορίας μετά την πρώτη εγκατάσταση του συστήματος και αφού θα έχουν εγγραφεί οι ψηφοφόροι θα είναι μηδαμινό σε σχέση με το κόστος διεξαγωγής μιας ψηφοφορίας του τωρινού συστήματος εκλογών. Σε αυτό το σημείο αξίζει να αναφέρουμε τα ποσά που δαπανήθηκαν για τις τελευταίες εκλογικές διαδικασίες στην χώρα μας.

Στις βουλευτικές εκλογές του 2012 σύμφωνα με το Υπουργείο Εσωτερικών το κόστος των εκλογών έφτασε κοντά στα 50 εκατομμύρια ευρώ. Από αυτά τα 41,5 ξοδεύτηκαν σε αποζημιώσεις. Οι αποζημιώσεις των 3296 υπαλλήλων του Υπ. Εσωτερικών και της αυτοδιοίκησης που χρησιμοποιήθηκαν στην εκλογική διαδικασία ανήλθαν στα 5,5 εκατ. Ευρώ. Οι δαπάνες που ξοδεύτηκαν όμως ώστε να πληρωθούν οι δικαστικοί αντιπρόσωποι, έφοροι και γραμματείς των εφορευτικών επιτροπών χωρίς τα οδοιπορικά ανήλθαν στα 23,8 εκατ. Ευρώ [81]. Σύμφωνα με αυτά τα στοιχεία και αν υποθέσουμε πως σε ένα διαδικτυακό εκλογικό σύστημα θα χρειάζονται μόνο οι υπάλληλοι των αρχών πιστοποίησης οι οποίοι θα μπορούσε να υπολογιστεί πως είναι οι υπάλληλοι του Υπ. Εσωτερικών το κράτος θα κέρδιζε από κάθε εκλογική διαδικασία πάνω από 20 εκ. ευρώ. Στις ίδιες εκλογές τα παραβάν και οι κάλπες είχαν κοστίσει 1,5 και 3,2 εκ ευρώ αντίστοιχα. [83]

Στις βουλευτικές εκλογές του 2015 δαπανήθηκαν κοντά στα 3 εκ. ευρώ για εκλογικούς φακέλους, ψηφοδέλτια, εκλογικούς σάκους και μίσθωση φορτηγών για την μεταφορά τους.[82] [83]

Δηλαδή σε κάθε εκλογική διαδικασία ξοδεύονται γύρω στα 4 εκ ευρώ εκτός των αποζημιώσεων σε ανθρώπους που εργάζονται για τις εκλογές. Αυτό το χρηματικό ποσό θα ήταν υπεραρκετό για να αγοραστεί όλος ο εξοπλισμός που θα χρειαζόταν ένα διαδικτυακό εκλογικό σύστημα. Το σημαντικότερο όλων βέβαια είναι πως αυτά τα χρήματα θα ξοδεύονταν μόνο μια φορά και από κει και πέρα θα χρειαζόταν συντήρηση του συστήματος.

Υπολογίζουμε λοιπόν πως το σύνολο των εξόδων μιας σημερινής εκλογικής διαδικασίας το οποίο, εκτός των μισθών των υπαλλήλων του Υπουργείου οικονομικών οι οποίοι θα χρειάζονταν και σε ένα διαδικτυακό σύστημα ψηφοφορίας , ανέρχεται κοντά στα 25 εκατομμύρια ευρώ. Αν αναλογιστούμε και ότι λόγω της οικονομικής κατάστασης στη χώρα έχουν γίνει μεγάλες περικοπές στα εκλογικά έξοδα είναι 100% σίγουρο πως ένα διαδικτυακό σύστημα ψηφοφορίας δεν θα χρειαζόταν ούτε τα μισά χρήματα για να λειτουργήσει.

Πέρα από το οικονομικό κομμάτι ένα διαδικτυακό εκλογικό σύστημα έχει και άλλα πλεονεκτήματα. Υπάρχει καλύτερη εξυπηρέτηση προς τους ψηφοφόρους καθώς μειώνεται η γραφειοκρατία των εκλογών και μπορούν να συμμετάσχουν είτε είναι ετεροδημότες είτε είναι κάτοικοι του εξωτερικού. Οι συγκεκριμένες κατηγορίες πολιτών δυστυχώς και λόγω της οικονομικής κρίσης είναι σχεδόν απίθανο σήμερα να μεταβούν στον τόπο τους μόνο και μόνο για να ψηφίσουν. Ενώ με ένα σύστημα διαδικτυακής ψηφοφορίας δεν θα υπήρχε ούτε αυτό το πρόβλημα.

Τέλος άλλη μια κατηγορία πολιτών η οποία δυστυχώς είναι σαν αποκλεισμένη από τις ψηφοφορίες είναι τα άτομα με κινητικά ή άλλα προβλήματα. Αν μπορούν να ψηφίσουν από έναν οποιονδήποτε Η/Υ σίγουρα θα ήταν πιο εύκολο να συμμετάσχουν.

Όπως είναι λογικό το διαδικτυακό σύστημα έχει και κάποια μειονεκτήματα σε σχέση με το σημερινό σύστημα ψηφοφορίας τα οποία θέλουν αρκετό χρόνο και προσπάθεια ώστε να ξεπεραστούν.

Μεγάλο πρόβλημα οπού θα χρειαστεί αρκετή προσπάθεια για να ξεπεραστεί θα είναι η αύξηση καχυποψίας των πολιτών ως προς την εγκυρότητα των αποτελεσμάτων και την διατήρηση της ιδιωτικότητας της απόφασής τους. Εκτός από αυτό άλλο ένα μεγάλο θέμα θα είναι ο αυξημένος κίνδυνος κακόβουλων επιθέσεων με σκοπό την παραποίηση των αποτελεσμάτων ή την πρόκληση δυσλειτουργίας – μη λειτουργίας του συστήματος. Τέλος ένα ακόμα μειονέκτημα θα είναι η δημιουργία διακρίσεων ανάμεσα σε άτομα που έχουν πρόσβαση και μπορούν να διαχειριστούν εφαρμογές διαδικτύου και άλλα που δεν έχουν αυτή την δυνατότητα. Βέβαια το τελευταίο με τον καιρό τείνει να εξαλειφθεί αλλά μέχρι να μπορέσουμε να πούμε με βεβαιότητα πως όλοι οι πολίτες μπορούν να διαχειριστούν μια διαδικτυακή εφαρμογή έχουμε ακόμα μέλλον μπροστά μας.

Για να αντιμετωπιστούν αυτά τα μειονεκτήματα θα χρειαστεί να γίνουν αρκετές πιλοτικές δοκιμές και σίγουρα για κάποιες εκλογικές διαδικασίες το διαδικτυακό σύστημα να λειτουργήσει παράλληλα με το κλασικό σύστημα ψηφοφορίας ώστε να κερδίσει σιγά σιγά την εμπιστοσύνη των πολιτών. Θεωρούμε πάντως πως είναι προβλήματα τα οποία με

σωστή ενημέρωση των πολιτών, σωστό σχεδιασμό και στρατηγική μπορούν να λυθούν και να κερδίσουμε μόνο τα πλεονεκτήματα ενός διαδικτυακού συστήματος ψηφοφορίας.

Η συμμετοχή των πολιτών στις ψηφοφορίες θα αυξανόταν και λόγω του μικρού κόστους και της μικρής προετοιμασίας οι εκλογικές διαδικασίες θα μπορούσαν να γίνονται πιο συχνά και για πιο πολλά ζητήματα. Αυτό θα ενίσχυε το δημοκρατικό πολίτευμα, του οποίου το μέγιστο αγαθό είναι η συμμετοχή των πολιτών στις αποφάσεις.

5.2 Μελλοντικές Βελτιώσεις

Το διαδικτυακό σύστημα ψηφοφορίας που κατασκευάστηκε πληροί τις προϋποθέσεις ασφαλείας και τα κριτήρια ενός δίκαιου και αξιόπιστου συστήματος ψηφοφορίας. Σε καμία περίπτωση όμως, όπως και οποιοδήποτε νέο πληροφοριακό σύστημα, δεν γίνεται να μπει σε κανονική λειτουργία αμέσως. Θα πρέπει να δοκιμαστεί πρώτα σε ψηφοφορίες στις οποίες συμμετέχει μικρός αριθμός ψηφοφόρων ώστε να γίνουν οι κατάλληλοι έλεγχοι κατά την δοκιμαστική περίοδο. Επίσης, κατά τη διάρκεια αυτής της περιόδου θα δοκιμαστεί και σε πραγματικά δεδομένα η αξιοπιστία και η ανοχή του στις επιθέσεις.

Πέρα όμως από την αυτονόητη δοκιμαστική λειτουργία θα πρέπει να γίνει εις βάθος ανάλυση στην ασφάλεια του δικτύου μέσω του οποίου επικοινωνεί ο εξυπηρετητής στο διαδίκτυο και να εξασφαλιστεί η αξιοπιστία και οι τρόποι παρακολούθησης για κακόβουλες κινήσεις στα χαμηλότερα επίπεδα δικτύωσης.

Πέρα από την ασφάλεια δικτύου θα πρέπει να μελετηθούν και να χρησιμοποιηθούν πρωτόκολλα κρυπτογράφησης για την αποθήκευση των δεδομένων των χρηστών και των αποτελεσμάτων των ψηφοφοριών στην Βάση Δεδομένων. Με αυτό θα πρέπει να διασφαλιστεί πως κανείς ο οποίος μπορεί να έχει αυτοπρόσωπη παρουσία και σύνδεση με τον εξυπηρετητή του συστήματος δεν θα μπορεί να τροποποιήσει οποιοδήποτε στοιχείο. Το συγκεκριμένο ζήτημα δεν μελετήθηκε και δεν αναπτύχθηκε εκτενώς στην συγκεκριμένη μεταπτυχιακή διατριβή.

Η ασφάλεια του συστήματος θα πρέπει να παρακολουθείται και να αναβαθμίζεται συνεχώς με βάση τις καινούριες απειλές που γίνονται γνωστές ενώ επίσης θα πρέπει να εμβαθύνουμε περισσότερο στην παρακολούθηση του συστήματος και των ψηφοφοριών χωρίς όμως να επηρεάζουμε την εμπιστευτικότητα.

Σε ό,τι έχει να κάνει με τις μελλοντικές βελτιώσεις σχετικά με το λειτουργικό μέρος της εφαρμογής θα μπορούσαν να προστεθούν ψηφοφορίες εκλογών όπου οι επιλογές δεν θα είναι τρεις αλλά τόσες όσες και τα πολιτικά κόμματα που συμμετέχουν ή ακόμα και οι βουλευτές αυτών.

Τέλος θα μπορούσε να επικοινωνεί με τα υπόλοιπα συστήματα ηλεκτρονικής διακυβέρνησης του κάθε κράτους από τα οποία θα μπορούσε να προμηθεύεται τα στοιχεία των ψηφοφόρων ώστε να τα επιβεβαιώνει και σε σχέση με αυτά, πριν κάνει τη δημιουργία νέου ψηφοφόρου ο υπεύθυνος αρχής πιστοποίησης.

6 Βιβλιογραφία

1. Douglas W. Jones ,The Voting and Elections web pages, THE UNIVERSITY OF IOWA Department of Computer ScienceTH, “A Brief Illustrated History of Voting”
2. Brit J. Williams, Merle S. King, “Implementing Voting Systems: the Georgia Method”, COMMUNICATIONS OF THE ACM October 2004/Vol. 47, No. 10 39
3. Peter G. Neumann, “The problems and potentials of voting systems”, COMMUNICATIONS OF THE ACM October 2004/Vol. 47, No. 10
4. Joshua F. Clowers,” E-vote, U I-vote, Why Can't We All Just Vote?!: A Survey of the Changing Face of the American Election “
5. International Experiences of Electronic Voting and Their Implications for New South WalesA report prepared for the New South Wales Electoral Commission. Associate Professor Rodney Smith Department of Government and International Relations University of Sydney July 2006
6. UK Electoral Commission, August 2002, “Modernising Elections --- A strategic evaluation of the 2002 electoral pilot schemes”, “Securing the vote --- Report and recommendations”
7. Alexandros Xenakis and Prof. Ann Macintosh, “E-electoral Administration: Organizational Lessons Learned from the Deployment of E-voting in the UK”
8. Steve Wallace (2005) “M-government Gets Serious”,
9. U.O. Ekong and C.K. Ayo / The Prospects Of M-Voting Implementation In Nigeria
10. Allen Consulting Group 2011, Evaluation of technology assisted voting provided at the NSW State General Election March 2011: report to the New South Wales Electoral Commission,Sydney, July.
11. David Jefferson, Aviel D. Rubin, Barbara Simons AND David Wagner, “Analyzing Internet Voting Security”, COMMUNICATIONS OF THE ACM October 2004/Vol. 47, No. 10 59
12. Dr. David Jefferson, Dr. Aviel D. Rubin, Dr. Barbara Simons, Dr. David Wagner, “A Security Analysis of the Secure Electronic Registration and Voting Experiment (SERVE)”, January 21, 2004
13. Joe Mohen and Julia Glidden, “The case for Internet Voting”, January 2001/Vol. 44, No. 1 COMMUNICATIONS OF THE ACM
14. Expert Study Mission Report.The Carter Center Internet Voting Pilot: Norway’s 2013 Parliamentary Elections,19 March 2014
15. CONSULTATION AND EVALUATION PRACTICES IN THE IMPLEMENTATION OF INTERNET VOTING IN CANADA AND EUROPE, Prepared for Elections Canada by Jon H. Pammett Professor of Political Science Carleton University and Nicole Goodman Assistant Professor of Political Science McMaster University ,November 2013
16. Ülle Madise Professor of Constitutional Law of Tartu , Priit Vinkel Assistant, University of Tartu University Advisor, Elections Department of the Chancellery of Riigikogu ,Constitutionality of Remote Internet Voting: The Estonian Perspective
17. Jordi Barrat i Esteve, Ben Goldsmith and John Turner,International Experience with E-Voting ,Norwegian E-Vote Project,June 2012
18. Estonian National Electoral Committee Statistics , http://www.vvk.ee/voting-methods-in-estonia/engindex/statistics
19. Herb Deutsch and Stephen Berger, “Voting Systems: Standards and Certifications”, COMMUNICATIONS OF THE ACM October 2004/Vol. 47, No. 10
20. By Earl Barr, Matt Bishop, Mark Gondree “Fixing federal e-voting standards “,Communications of the ACM, Vol. 50 No. 3, March 2007
21. Alexandros Xenakis and Prof. Ann Macintosh , “Procedural Security and Social Acceptance in E-voting”. In: HICSS 2005 - 38th Hawaii International Conference on System Sciences. Jan. 2005.
22. IT Governance Institute, Information Security Governance :Guidance for boards of Directors and Executive Management. United States of America: IT Governance Institute, 2006

23. Anderson, Ross J. (2008). Security engineering: a guide to building dependable distributed systems (2nd ed.). Indianapolis, IN: Wiley. p. 1040. ISBN 978-0-470-06852-6.Chapter 2, page 17
24. Christopher Hadnagy . "Social Engineering: The Art of Human Hacking". Retrieved 18 January 2014.
25. McDowell, Mindi (November 4, 2009). "Cyber Security Tip ST04-015 - Understanding Denial-of-Service Attacks". United States Computer Emergency Readiness Team.Archived from the original on 2013-11-04.
26. Distributed Denial of Service Attacks(DDoS) Resources, Pervasive Technology Labs at Indiana University. Advanced Networking Management Lab (ANML). December 3, 2009..
27. A. Bortz, A. Barth, and A. Czeskis. Origin Cookies: Session Integrity for Web Applications. In Web 2.0 Security and Privacy Workshop (W2SP), 2011.
28. Grossman, Jeremiah (July 30, 2006). "The origins of Cross-Site Scripting (XSS)".
29. Williams,Jeff (January 19, 2009). "XSS (Cross Site Scripting) Prevention Cheat Sheet". OWASP. Retrieved February 4, 2009.
30. "Category:OWASP Top Ten Project". OWASP. Retrieved 2011-06-03. https://www.owasp.org/index.php/Top_10_2013-Top_10
31. Sean Michael Kerner (November 25, 2013). "How Was SQL Injection Discovered? The researcher once known as Rain Forrest Puppy explains how he discovered the first SQL injection more than 15 years ago"
32. "SQL Injection Prevention Cheat Sheet". Open Web Application Security Project. Retrieved 3 March 2012.
33. Klein, Christian (September 2004). "Buffer Overflow"
34. Paar, Christof; Pelzl, Jan; Preneel, Bart (2010). Understanding Cryptography: A Textbook for Students and Practitioners. Springer. ISBN 3-642-04100-0.
35. Thai Duong and Juliano Rizzo, "Here Come The Ninjas",2011
36. Ristic, Ivan. "Is BEAST Still a Threat?". 2013
37. Microsoft.Vulnerability in SSL/TLS. Could Allow Information Disclosure (2643584)". 2012-01-10.
38. Ristic, Ivan (Oct 31, 2013). "Apple Enabled BEAST Mitigations in OS X 10.9 Mavericks".
39. Dan Goodin (2012-09-13). "Crack in Internet's foundation of trust allows HTTPS session hijacking". Ars Technica.
40. Dennis Fisher (September 13, 2012). "CRIME Attack Uses Compression Ratio of TLS Requests as Side Channel to Hijack Secure Sessions"
41. Goodin, Dan (1 August 2013). "Gone in 30 seconds: New attack plucks secrets from HTTPS-protected pages"
42. Angelo Prado, Neal Harris, Yoel Gluck . "SSL, gone in 30 seconds: A BREACH beyond CRIME", 2013
43. Leyden, John. "Step into the BREACH: New attack developed to read encrypted web data",2 August 2013
44. Heartbleed Bug: Comodo Urges OpenSSL Users to Apply Patch , 2014 https://blog.comodo.com/e-commerce/heartbleed-bug-comodo-urges-openssl-users-to-apply-patch/
45. Εργαλείο ελέγχου τρωτότητας σε Heartbleed bug . https://sslanalyzer.comodoca.com/heartbleed.html
46. Dan Goodin (4 February 2013). ""Lucky Thirteen" attack snarfs cookies protected by SSL encryption"
47. Nadhem J. AlFardan and Kenneth G. Paterson (4 February 2013). "Lucky Thirteen: Breaking the TLS and DTLS Record Protocols"
48. Bright, Peter (October 15, 2014). "SSL broken, again in POODLE attack".
49. Brandom, Russell (October 14, 2014). "Google researchers reveal new Poodle bug, putting the web on alert"
50. Kelly Jackson Higgins February 24, 2009 (2009-02-24). "SSLStrip Hacking Tool Released"
51. Moxie Marlinspike (2011-12-19) Answers Your Questions - Slashdot". Interviews.slashdot.org.
52. Hodges, Jeff; Jackson, Collin; Barth, Adam (Nov 2012). "Section 5. HSTS Mechanism Overview". RFC 6797
53. Hodges, Jeff; Jackson, Collin; Barth, Adam (Nov 2012). "Section 5.2. HSTS Policy". RFC 6797
54. Network Time Protocol Version 4 https://tools.ietf.org/html/rfc5905
55. Dan Wendlandt, David G. Andersen, Adrian Perrig Carnegie Mellon University. "Perspectives: Improving SSH-style Host Authentication with Multi-Path Probing".
56. Chrome's HSTS preload list. https://hstspreload.appspot.com/
57. Owasp - Certificate and Public Key Pinning https://www.owasp.org/index.php/Certificate_and_Public_Key_Pinning
58. HTTPS Everywhere. https://www.eff.org/https-everywhere
59. T. Dierks, C. Allen, The TLS Protocol version 1.2, IETF, January 2008
60. Microsoft TechNet. "SSL/TLS in Detail" https://technet.microsoft.com/en-us/library/cc785811.aspx

61. IETF Networking Group, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, May 2008
62. Delfs, Hans & Knebl, Helmut (2007). "Symmetric-key encryption". Introduction to cryptography: principles and applications
63. Pelzl & Paar (2010). Understanding Cryptography.
64. IEEE 1363: Standard Specifications for Public-Key Cryptography http://grouper.ieee.org/groups/1363/
65. Christof Paar, Jan Pelzl, "Introduction to Public-Key Cryptography" Springer, 2009
66. JavaBeans tutorial , http://wiki.netbeans.org/NetBeansJavaBeansTutorial
67. Java Servlets tutorial , http://docs.oracle.com/javaee/6/tutorial/doc/bnafd.html
68. Jsp , http://www.oracle.com/technetwork/java/javaee/jsp/index.html
69. JavaScript and AJAX , http://www.w3schools.com/js/
70. World Wide Web Consortium (W3C) , http://www.w3.org/
71. John Deacon ,"Model-View-Controller Architecture" ,May 2009
72. Apache Tomcat v6 , https://tomcat.apache.org/tomcat-6.0-doc/
73. MySql , http://dev.mysql.com/doc/relnotes/mysql/5.0/en/index.html
74. MySql Connector , http://dev.mysql.com/doc/connector-j/en/index.html
75. Java , http://www.oracle.com/technetwork/java/javase/7u75-relnotes-2389086.html
76. DisplayTag Library 1.2, http://displaytag.sourceforge.net/1.2
77. OpenSSL official , https://www.openssl.org/
78. JavaMail, http://www.oracle.com/technetwork/java/javamail/index.html
79. Jon Scott Stevens, Kaptcha code ,2010 , https://code.google.com/p/kaptcha/
80. TLS on Tomcat 6 – How to , https://tomcat.apache.org/tomcat-6.0-doc/ssl-howto.html
81. ΥΠ.ΕΣ - ΕΚΛΟΓΙΚΕΣ ΔΑΠΑΝΕΣ 2012 , http://www.ypes.gr/el/MediaCenter/Minister/Seasonable/?id=ceada352-2f8a-42a2-a554-f6ad40270146
82. Έξοδα εκλογών 2015 , http://news247.gr/eidiseis/politiki/ekloges-gr/to-prwto-kostos-twn-eklogwn-ti-plhrwnoyme-gia-to-yliko-kai-th-metafora-toy.3233403.html
83. Έξοδα εκλογών, http://www.aftodioikisi.gr/proto-thema/poso-tha-kostisoun-oi-ekloges-poies-oi-mexri-stigmis-dapanes-tou-ipes