Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Σχεδίαση και υλοποίηση συστήματος ανώνυμης αξιολόγησης. **Design and implementation of an anonymous evaluation system**
Ονοματεπώνυμο Φοιτητή	Στέφανος Μονάχος
Πατρώνυμο	Ιωάννης
Αριθμός Μητρώου	ΜΠΠΛ/16013
Επιβλέπων	Κοτζανικολάου Παναγιώτης, Αναπληρωτής Καθηγητής

Ημερομηνία Παράδοσης 8 Ιουλίου 2020

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)	(υπογραφή)	(υπογραφή)
Κοτζανικολάου Παναγιώτης Αναπληρωτής Καθηγητής	Δουληγέρης Χρήστος Καθηγητής	Ψαράκης Μιχαήλ Επίκουρος Καθηγητής

Περίληψη

Σκοπός της παρούσας εργασίας είναι η υλοποίηση ενός συστήματος αξιολόγησης καθηγητών από φοιτητές το οποίο διασφαλίζει την ανωνυμία των χρηστών του. Στις ηλεκτρονικές αξιολογήσεις υπάρχει ο κίνδυνος να χαθεί η αμεροληψία των φοιτητών εάν δεν διασφαλίζεται η ανωνυμία τους. Χρησιμοποιώντας τεχνικές παρόμοιων συστημάτων, όπως οι ανώνυμες υπογραφές κατά τις μη ανιχνεύσιμες πληρωμές του David Chaum, δημιουργήσαμε ένα σύστημα τέτοιο ώστε οι φοιτητές να μπορούν να αισθάνονται άνετοι να εκφράσουν τις απόψεις τους κατά την αξιολόγηση ενός καθηγητή ηλεκτρονικά. Το πρωτόκολλο που δημιουργήθηκε αναλύθηκε μαθηματικά ως προς την εγκυρότητά του καθώς επίσης εξηγούνται όλα τα θέματα ασφαλείας που μπορεί να δημιουργούνται από την εξασφάλιση της ανωνυμίας των χρηστών ενός τέτοιου πρωτοκόλλου. Το σύστημα αποτελείται από δύο εφαρμογές. Η πρώτη είναι υπεύθυνη για την εξασφάλιση της κατάθεσης αξιολογήσεων μόνο από έγκυρους χρήστες καθώς επίσης και η παράδοση ενός αποδεικτικού εισιτηρίου που θα καταθέτουν οι φοιτητές κατά την αξιολόγησή τους. Σκοπός της δεύτερης εφαρμογής είναι η ανώνυμη κατάθεση της αξιολόγησης, η πιστοποίηση της εγκυρότητας των εισιτηρίων και φυσικά η συλλογή των αξιολογήσεων. Οι εφαρμογές αυτές είναι σημαντικό να τηρούν τις προϋποθέσεις και τους στόχους της αξιολόγησης των καθηγητών, οπότε είναι απαραίτητη η ανάλυση των απαιτήσεων του συστήματος καθώς και η σχεδίασή τους πριν υλοποιηθούν. Μέρος της ανάλυσης αυτής αποτελούν και οι επιλογές των τεχνολογιών που θα διευκολύνουν την δημιουργία των διαδικτυακών εφαρμογών όπως και οι γλώσσες προγραμματισμού που θα χρησιμοποιηθούν είτε για τις απαραίτητες ενέργειες που θα γίνονται από τη μεριά του χρήστη είτε από εκείνη των διακομιστών που θα φιλοξενούν τις εφαρμογές. Εφόσον οι εφαρμογές αυτές είναι διαδικτυακές, χρησιμοποιήθηκαν κατάλληλα εργαλεία για την αντιμετώπιση διαδικτυακών επιθέσεων και για την εξασφάλιση υψηλής ποιότητας κώδικα. Στη συνέχεια θα αναλυθούν και θα παρουσιαστούν τα κύρια τμήματα των εφαρμογών που υλοποιούν το πρωτόκολλο του ανώνυμου συστήματος αξιολόγησης. Τέλος, θα παρουσιαστεί η χρήση του συστήματος κατά μία ανώνυμη αξιολόγηση.

Abstract

The purpose of this thesis statement is to implement a student-instructor evaluation system that ensures the anonymity of the users. In evaluations that can be submitted electronically, there is always the risk of losing the confidence of expressing an opinion honestly. To avoid such occurrences, the system must provide anonymity to the users. By using techniques of similar systems, like David Chaum's bind signatures that can ensure untraceable payments, we designed and implemented an anonymous evaluation system so that students can express their opinions of their respective teachers impartially. The protocol created was analyzed mathematically for its integrity. In addition, we recited the problems that can occur by providing the users of the system with full anonymity like using a ticket that can be bought or using multiple tickets of the student to evaluate one instructor. Our anonymous evaluation system consists of two modules. The first module is responsible for the blind signatures that ensure the anonymity of the students by providing a ticket that cannot be reversed or reproduced. The goal of the second module is the anonymous submission of the evaluations, the attestation of a ticket's integrity and the collection of the evaluations. It is of great importance that these two applications are meeting the requirements and objectives of typical teacher evaluations, so it is deemed necessary to analyze and design the system requirements before the implementation. Part of this analysis is the choice of technologies that will facilitate the creation of the web applications as well as the programming languages that will be used on either the front end or the back end of the applications. Since these applications will be hosted online, appropriate tools have been used to counter the more well-known attacks and ensure high code quality. The main parts of the applications that are implementing the anonymous evaluation system protocol will then be analyzed and presented. Finally, the use of the system during an anonymous evaluation will be presented.

Πίνακας περιεχομένων

1. Εισαγωγή

Ένα σύστημα αξιολόγησης καθηγητών αποτελεί μία διαδικασία μέτρησης της ποιότητας της εκπαίδευσης που παρέχεται στους φοιτητές, καθώς επίσης βοηθάει και στην επιτήρηση των καθηγητών. Είναι λογικό η αξιολόγηση ενός φοιτητή να πρέπει να γίνεται χωρίς ενδοιασμούς και με κύριο μέλημα την ελευθερία της έκφρασης των απόψεων του και με απόλυτο σεβασμό στην ιδιωτικότητα του.

Ως ιδιωτικότητα ορίζεται η ικανότητα ενός ατόμου, ή μίας κοινωνικής ομάδας, στο να απομονώσει τον εαυτό του ή τις προσωπικές του πληροφορίες ώστε να μπορεί να εκφραστεί επιλεκτικά (J. Herring, 2014).

Όταν μιλάμε για ιδιωτικότητα πληροφοριών εννοούμε τον έλεγχο των προσωπικών δεδομένων που μπορούν να συγκεντρωθούν για αποθήκευση, διάδοση ή επεξεργασία. Επομένως, για να διασφαλίζεται η ιδιωτικότητα των πληροφοριών των χρηστών πρέπει να δίνεται ιδιαίτερη βαρύτητα στην υπεράσπιση των προσωπικών στοιχείων του χρήστη. Ως απαιτήσεις ασφάλειας των προσωπικών δεδομένων είναι απαραίτητη η εξασφάλιση της εμπιστευτικότητας (προστασία από αποκάλυψη των προσωπικών δεδομένων σε τρίτους, μη εγκεκριμένους χρήστες), της ακεραιότητας (αποφυγή μη εξουσιοδοτημένων εγγραφών ή τροποποιήσεων των δεδομένων), της διαθεσιμότητας (προστασία από έλλειψη διαθεσιμότητας των δεδομένων), αυθεντικότητα (δυνατότητα ταυτοποίησης των δεδομένων) και της μη αποποίησης (μη παροχή άρνησης των δεδομένων από μία σχετική οντότητα).

Με τον όρο ανωνυμία εννοούμε την προστασία της ταυτότητας ενός ατόμου, που κάνει μία ορατή ενέργεια σε άλλους ανθρώπους. Η πιο διαδεδομένη ενέργεια που διατηρείται η ανωνυμία είναι η ψήφος σε εκλογές, όπου προστατεύεται η ταυτότητα του ανθρώπου που κατέθεσε την ψήφο, αλλά η ψήφος προσμετράτε κανονικά.

Επομένως, η βασική διαφορά της ανωνυμίας με την ιδιωτικότητα είναι ότι στην πρώτη διασφαλίζεται η ταυτότητα του ατόμου που ενεργεί, ενώ στην δεύτερη προστατεύονται τα προσωπικά του δεδομένα από επεξεργασία, τροποποίηση και διάδοση.

Όπως στις ψηφοφορίες, είναι σημαντικό και στις αξιολογήσεις των φοιτητών, να παρέχεται η δυνατότητα να εκφραστούν επιλεκτικά. Στην περίπτωση που καταπατάται η ανωνυμία μίας αξιολόγησης θα μπορούσε να επηρεάσει την σωστή συνεργασία των φοιτητών με τον καθηγητή που αξιολογούν ή ακόμα και να δημιουργηθούν προσωπικά προβλήματα μεταξύ τους.

Όπως με όλες τις διαδικασίες, η διεξαγωγή τους μέσω ηλεκτρονικών συστημάτων αποφέρει πολλά πλεονεκτήματα για τους χρήστες. Ένα σημαντικό πλεονέκτημα είναι ότι οι χρήστες του συστήματος δεν βασίζονται σε φυσικές εγκαταστάσεις αλλά μπορούν να χρησιμοποιήσουν τα εν λόγω συστήματα με οποιαδήποτε συσκευή

επιθυμούν. Επίσης, ο χρόνος είναι ένα ακόμα πολύ βασικό πλεονέκτημα καθώς οι προσφέρεται μεγαλύτερη ευελιξία στους χρήστες του συστήματος. Φυσικά, η μεταφορά τέτοιων διαδικασιών σε ηλεκτρονική μορφή, μπορεί να είναι και πιο οικονομικές ως προς την διεξαγωγή τους. Η επεξεργασία των δεδομένων μέσω ηλεκτρονικού συστήματος, μπορεί να βοηθήσει στην εξαγωγή συμπερασμάτων και στην εύκολη πρόσβαση από τους ενδιαφερόμενους.

2. Επισκόπηση του χώρου

Στο παρόν κεφάλαιο θα εξεταστούν ανώνυμα συστήματα που εξασφαλίζουν την ιδιωτικότητα και την εμπιστευτικότητα των δεδομένων των χρηστών τους καθώς επίσης και οι κρυπτογραφικές τεχνικές που χρησιμοποιούν. Κατά κύριο λόγο, χρησιμοποιούνται οι τυφλές υπογραφές (blind signatures) και τα ανώνυμα διαπιστευτήρια (Anonymous Credentials).

2.1 Ανώνυμα συστήματα με τη χρήση τυφλών υπογραφών

2.1.1 Σύστημα μη ανιχνεύσιμων πληρωμών του Chaum

Ο David Chaum ήταν ο πρώτος που έθιξε την ανάγκη για ανωνυμία και ιδιωτικότητα στις ηλεκτρονικές συναλλαγές και στις προτιμήσεις των καταναλωτών. Συγκεκριμένα πρότεινε ένα σύστημα πληρωμών που είχε την δυνατότητα να μην μπορεί να ανιχνευτούν οι πληροφορίες του πληρωτή, του χρόνου και του ποσού της συναλλαγής από τρίτους. Ταυτόχρονα όμως, υπάρχει η δυνατότητα παροχής αποδεικτικών στοιχείων για τις πληρωμές αυτές. Σε περίπτωση όμως κλοπής των μέσων πληρωμής μπορεί να σταματήσουν οι συναλλαγές.

Το συγκεκριμένο σύστημα στηρίζεται στις τυφλές υπογραφές, οι οποίες αποτελούν και δημιουργία David Chaum για την κατασκευή του συγκεκριμένου συστήματος.

Οι τυφλές υπογραφές στηρίζονται στην ασύμμετρη κρυπτογραφία που χρησιμοποιούν δύο κλειδιά, το δημόσιο κλειδί και το ιδιωτικό. Το δημόσιο κλειδί είναι προσβάσιμο από οποιονδήποτε και χρησιμοποιείται για την κρυπτογραφία προς τον ιδιοκτήτη του ζεύγους κλειδιού ή για την επικύρωση της υπογραφής του, ενώ το ιδιωτικό κλειδί χρησιμοποιείται για την αποκρυπτογράφηση και την δημιουργία υπογραφών από τον ιδιοκτήτη του ζεύγους και είναι προσβάσιμο μόνο από τον ίδιο.

Για την δημιουργία μίας τυφλής υπογραφής, επιλέγεται ένας τυχαίος αριθμός x έτσι ώστε η r(x) να μπορεί να δημιουργήσει το c(x) και στη συνέχεια να το στείλει σε αυτόν που θα να το υπογράψει.

Στη συνέχεια, ο υπογράφων χρησιμοποιώντας το ιδιωτικό του κλειδί και το c(x) δημιουργώντας το s'(c(x)) που αποτελεί μία έγκυρη υπογραφή του.

Χρησιμοποιώντας την c', ο αιτούμενος για την υπογραφή δημιουργεί το c'(s'(c(x))) = s'(x) που είναι η υπογραφή του αρχικού μηνύματος x. Έτσι ο οποιοσδήποτε μπορεί να πιστοποιήσει την υπογραφή χρησιμοποιώντας το δημόσιο κλειδί του υπογράφων χρησιμοποιώντας την r(s'(s(x))).

Χρησιμοποιώντας το παραπάνω πρωτόκολλο των τυφλών υπογραφών ο David Chaum πρότεινε το παρακάτω σύστημα μη ανιχνεύσιμων πληρωμών.

1. Ο πληρωτής παράγει έναν τυχαίο αριθμό από την r(x) και δημιουργεί το σημείωμα για την τράπεζα c(x).
2. Ο πληρωτής προωθεί το σημείωμα του βήματος 1 στην τράπεζα.
3. Η τράπεζα υπογράφει το σημείωμα ηλεκτρονικά χρησιμοποιώντας την s'(x) και χρεώνει τον πληρωτή.
4. Το τυφλά υπογεγραμμένο μήνυμα s'(c(x)) αποστέλλεται στον πληρωτή.
5. Ο πληρωτής αποβάλλει τον τυχαίο αριθμό από το τυφλά υπογεγραμμένο σημείωμα, δηλαδή c'(s'(c(x))).
6. Ο πληρωτής ελέγχει την εγκυρότητα της υπογραφής, δηλαδή εάν s(s'(x))=x.
7. Για την πληρωμή, αρκεί η αποστολή του υπογεγραμμένου μηνύματος s'(x) από τον πληρωτή στον πληρωτέο.
8. Η εγκυρότητα του σημειώματος μπορεί να ελεγχθεί με τη χρήση της s(x), δηλαδή s(s'(x))=x.
9. Ο πληρωτέος στέλνει το υπογεγραμμένο μήνυμα s'(x) στην τράπεζα.
10. Η τράπεζα ελέγχει εάν είναι έγκυρο το υπογεγραμμένο μήνυμα s(s'(x)), εάν δεν είναι σταματάει η διαδικασία.
11. Η τράπεζα ελέγχει εάν έχει ξαναχρησιμοποιηθεί το μήνυμα, εάν δεν έχει το αποθηκεύει ως χρησιμοποιημένο.
12. Η τράπεζα πιστώνει τον πληρωτέο σύμφωνα με το ποσό του σημειώματος.
13. Η τράπεζα ενημερώνει τον πληρωτή για την επιτυχία της πληρωμής.

2.1.2 Το σύστημα Sensus

Το σύστημα Sensus χρησιμοποιεί τυφλές υπογραφές για την διεξαγωγή ηλεκτρονικών ερευνών και ψηφοφοριών που διαφυλάσσουν την ανωνυμία και την ιδιωτικότητα των ερωτηθέντων.

Η διεξαγωγή ηλεκτρονικών ψηφοφοριών και δημοσκοπήσεων προσφέρει πολλά πλεονεκτήματα από τις συμβατικές μεθόδους που απαιτούν την φυσική παρουσία του ερωτηθέντος. Ένα πρωταρχικό πλεονέκτημα είναι η ευκολία που παρέχεται στους ψηφοφόρους, οι οποίοι μπορούν να ασκήσουν το εκλογικό τους δικαίωμα στον χρόνο, τον τόπο και με το μέσο που τους βολεύει λόγω της έλλειψης της ανάγκης της φυσικής παρουσίας τους σε συγκεκριμένο χώρο. Για αυτόν τον λόγο υπάρχει η πεποίθηση ότι μία ηλεκτρονική ψηφοφορία θα μπορούσε να ενισχύσει την προσέλευση των ψηφοφόρων και να μειώσει το ποσοστό της αποχής.

Το καθήκον του συστήματος είναι η σωστή διεξαγωγή της εγγραφής των χρηστών στο σύστημα, της επιβεβαίωσης της ταυτότητας των χρηστών ώστε να μην μπορεί να

ψηφίσει μόνο όποιος έχει το δικαίωμα ψήφου και της σωστής συλλογής και καταμέτρησης των ψήφων.

Για την σωστή διεξαγωγή ηλεκτρονικών εκλογών είναι απαραίτητο να ικανοποιούνται οι ιδιότητες της ακρίβειας, της ατρωσίας, της ιδιωτικότητας και της επαληθευσιμότητας.

Με τον όρο ακρίβεια εννοούμε την ικανότητα να μην μπορεί να εγκριθεί μία μη έγκυρη ψήφος, να μην μπορεί να αλλοιωθεί καμία ψήφος και να μην μπορεί μία επικυρωμένη ψήφος να αποκλειστεί από την καταμέτρηση των ψήφων.

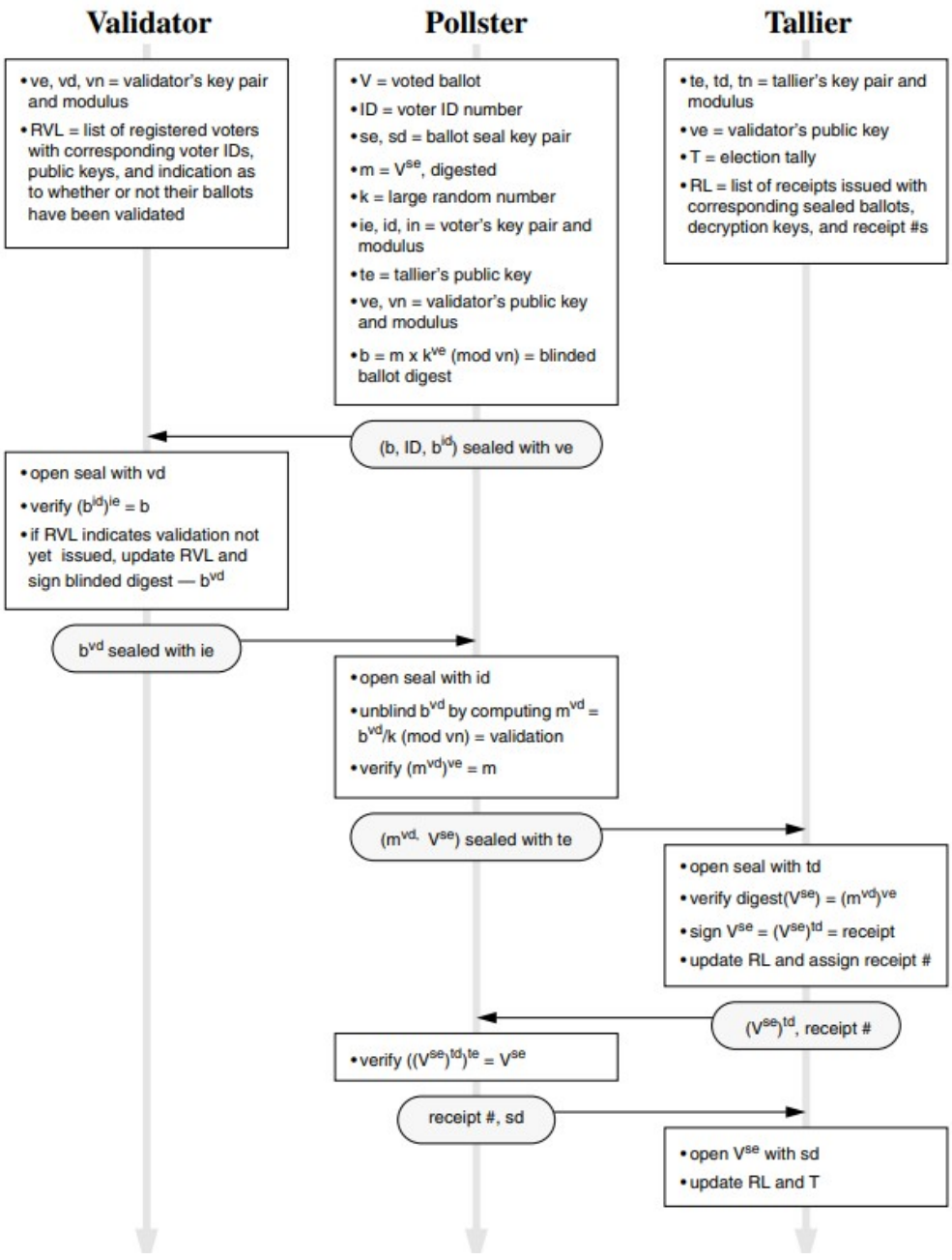
Ένα σύστημα ψηφοφορίας μπορούμε να το χαρακτηρίσουμε άτρωτο εάν επιτρέπεται σε έναν ψηφοφόρο να ψηφίσει μόνο μία φορά και παράλληλα να διασφαλίζεται ότι η ψήφος αυτή προέρχεται από έγκυρο ψηφοφόρο.

Φυσικά, η ψήφος αποτελεί ευαίσθητη πληροφορία των ψηφοφόρων και για την διασφάλιση της ιδιωτικότητας τους πρέπει να μην μπορεί κανείς να διασταυρώσει την ψήφο με τον ψηφοφόρο, επιπροσθέτως ούτε και ο ίδιος ο ψηφοφόρος.

Η τελευταία ιδιότητα που αναφέρθηκε είναι η επαληθευσιμότητα, με την οποία εννοούμε την ικανότητα από όλους να μπορούν να επιβεβαιώσουν ότι όλοι οι ψήφοι έχουν επικυρωθεί και καταμετρηθεί σωστά.

Το πρωτόκολλο ορίζει τρία βασικά τμήματα για την σωστή του διεξαγωγή. Το πρώτο τμήμα είναι υπεύθυνο για την επαλήθευση (validator), το δεύτερο για την καταγραφή των ψήφων (tallier) και το τρίτο για την καταμέτρηση και τον υπολογισμό των αποτελεσμάτων.

Στην παρακάτω εικόνα παρουσιάζεται ένα διάγραμμα συχνοτήτων, που εκφράζει τα απαραίτητα βήματα για την υλοποίηση και την επικοινωνία των τριών αυτών τμημάτων για το πρωτόκολλο Sensus, όπως ακριβώς παρουσιάζεται στο σχετικό δημοσιευμένο άρθρο.



Εικόνα 1. Το πρωτόκολλο Sensus

Πηγή: LF Cranor, RK Cytron, 1997. *Sensus : A security-conscious electronic polling system for the Internet*. Proceedings of the Hawaii International Conference on System Sciences

2.2 Διαπιστευτήρια μέσω χαρακτηριστικών

Τα συστήματα αυτά βασίζονται στην ιδέα ότι μπορεί να υπάρξει έγκυρη υπογραφή που να παρουσιάζει μόνο τα απαραίτητα χαρακτηριστικά για το διαπιστευτήριο. Ένα απλό παράδειγμα θα ήταν το να αποδείξει ο χρήστης ότι είναι αρκετά μεγάλος ηλικιακά για να παρακολουθήσει μία ταινία. Στην προκειμένη περίπτωση, το μόνο διαπιστευτήριο που χρειάζεται είναι ότι η ηλικία είναι μεγαλύτερη από μία συγκεκριμένη τιμή. Επομένως, πληροφορίες όπως το ονοματεπώνυμο του χρήστη είναι περιττές και προσβάλουν την ιδιωτικότητα του χρήστη.

Τα βήματα για να επιτευχθούν τα παραπάνω είναι τα εξής:

1. Ο χρήστης (Prover) ζητάει από τον εκδότη (Issuer) τα διαπιστευτήριά του τα οποία και παραλαμβάνει.
2. Ο χρήστης ενημερώνεται σχετικά με την πολιτική πρόσβασης από τον ελεγκτή (verifier) των διαπιστευτηρίων του, για την ενέργεια που θέλει να κάνει. Στην περίπτωση που προαναφέρθηκε, την ηλικία του.
3. Ο χρήστης χρησιμοποιεί την κατάλληλη κρυπτογραφία για να υπολογίσει το αποδεικτικό που χρειάζεται. Συνήθως, χρησιμοποιούνται CL (Camenisch-Lysyanskaya) υπογραφές, οι οποίες χρησιμοποιούν ελλειπτικές καμπύλες. Στο βήμα αυτό, είναι και που ο χρήστης επιλέγει ποιες πληροφορίες σχετικά με αυτόν θα παρουσιάσει.
4. Ο χρήστης στέλνει το υπολογισμένο αυτό token στην οντότητα που χρειάζεται να ταυτοποιήσει τα στοιχεία του. Ο ελεγκτής θα λάβει μόνο τις απαραίτητες πληροφορίες σχετικά με την ενέργεια που θέλει να κάνει ο χρήστης (ή ότι θέλησε να παρουσιάσει στο βήμα 3). Εάν τα διαπιστευτήρια είναι επαρκή, θα επιτρέψει στον χρήστη να κάνει την επιθυμητή του ενέργεια.

2.2.1 IBM Idenity Mixer (IDEMIX)

Η IBM έχει δημιουργήσει το δικό της σύστημα που χρησιμοποιεί διαπιστευτήρια βασισμένα σε χαρακτηριστικά.

Στην ουσία το συγκεκριμένο σύστημα χρησιμοποιεί τα βήματα που αναφέρθηκαν παραπάνω αλλά με την διαφορά ότι στο συγκεκριμένο σύστημα το ιδιωτικό κλειδί του χρήστη μπορεί να έχει παραπάνω από ένα δημόσιο. Το κάθε δημόσιο χρησιμοποιείται για διαφορετικές υπηρεσίες. Για παράδειγμα, θα μπορούσε ένας χρήστης να έχει

διαφορετικά κλειδιά για να συμπληρώσει την φορολογική του δήλωση και για να ανανεώσει ένα δίπλωμα οδήγησης.

Εφόσον σε κάθε συναλλαγή χρησιμοποιείται διαφορετικό κλειδί, δεν μπορούν να ταυτοποιηθούν ούτε συναλλαγές που χρησιμοποίησαν ένα συγκεκριμένο κλειδί. Εν ολίγοις, δεν μπορεί να βγει το συμπέρασμα ότι ο κάτοχος ενός κλειδιού έκανε συγκεκριμένες ενέργειες.

Η συγκεκριμένη υλοποίηση είναι διαθέσιμη σαν πειραματική υπηρεσία στην «IBM Bluemix cloud» πλατφόρμα ώστε να μπορεί να χρησιμοποιηθεί εύκολα από προγραμματιστές ο συγκεκριμένος τρόπος αυθεντικοποίησης.

2.3 Σύγκριση των δύο μεθόδων

Στις τυφλές υπογραφές, ο εκδότης του εισιτηρίου υπογράφει ένα υπαρκτό μήνυμα του οποίου δεν γνωρίζει το περιεχόμενό του. Στα ανώνυμα διαπιστευτήρια ο εκδότης γνωρίζει τα χαρακτηριστικά και τα προσφέρει στον χρήστη, ο οποίος με την σειρά του επιλέγει ποια θα αποκρύψει από τον ελεγκτή.

Ο ελεγκτής των διαπιστευτηρίων γνωρίζει ολόκληρο το μήνυμα στις τυφλές υπογραφές, ενώ στα ανώνυμα διαπιστευτήρια γνωρίζει μόνο τα απαραίτητα στοιχεία. Στην πρώτη μέθοδο ο έλεγχος των απαραίτητων χαρακτηριστικών πρέπει να γίνει από τον εκδότη. Δηλαδή, να ταυτοποιήσει εάν ο συγκεκριμένος χρήστης πληροί τις προϋποθέσεις και να χρησιμοποιήσει την υπογραφή του. Στην δεύτερη μέθοδο, τα διαπιστευτήρια δεν ελέγχονται από τον εκδότη αλλά από τον ελεγκτή, στον οποίο προσφέρονται μόνο οι απαραίτητες πληροφορίες, με μορφή που μπορεί να αποδείξει ότι είναι από τον εκδότη.

Μία σημαντική διαφορά είναι η επιλογή των μεθόδων κρυπτογράφησης σχετικά με τις υλοποιήσεις των δύο μεθόδων. Και στις δύο μεθόδους χρησιμοποιείται ασύμμετρη κρυπτογραφία αλλά στις τυφλές υπογραφές οι υλοποιήσεις βασίζονται κυρίως στις υπογραφές RSA. Στην περίπτωση των διαπιστευτηρίων μέσω χαρακτηριστικών, προτιμώνται οι CL υπογραφές, οι οποίες είναι βασισμένες σε ελλειπτικές καμπύλες.

3. Ανάλυση και σχεδίαση

Αρχικά πρέπει να αναλύσουμε τις απαιτήσεις του συστήματος και στην συνέχεια να προβούμε στην κατάλληλη σχεδίαση ώστε να μπορούν να καλυφθούν οι εν λόγω απαιτήσεις. Θα πρέπει στις απαιτήσεις του συστήματος να λαμβάνεται υπόψιν η δυνατή καταγραφή που μπορεί να γίνει σε μία εφαρμογή μέσω είτε σε εγγραφές στη βάση δεδομένων ή σε οποιοδήποτε σύστημα επιτήρησης των κινήσεων ενός χρήστη στην εφαρμογή (π.χ. σύστημα logging).

3.1 Ανάλυση απαιτήσεων συστήματος

Το σύστημα ανώνυμης αξιολόγησης αποσκοπεί στην βελτιστοποίηση των καθηγητών και της ύλης των μαθημάτων τους. Αυτό μπορεί να επιτευχθεί μέσω της συμπλήρωσης ερωτηματολογίων από τους άμεσα ενδιαφερόμενους, δηλαδή τους φοιτητές.

Εφόσον οι αξιολογήσεις μπορούν να επηρεάσουν τον καθηγητή ως προς τον φοιτητή, είτε θετικά είτε αρνητικά, είναι αναγκαιότητα η απόκρυψη της αξιολόγησης από τον καθηγητή και γενικότερα από οποιονδήποτε έχει πρόσβαση στην εφαρμογή ή στους διακοσμητές του.

Οι αξιολογήσεις πρέπει να γίνονται μέσα από ερωτηματολόγια, ειδικά διαμορφωμένα από τον φορέα του πανεπιστημίου, του οποίου οι απαντήσεις θα γίνονται με βάση την κλίμακα Likert. Ο φοιτητής θα καλείται να απαντήσει σε γενικές ερωτήσεις σχετικά με την διεξαγωγή του μαθήματος, την υλοποίηση των εργαστηρίων, τον φόρτο εργασίας του καθηγητή και γενικότερα για την συνολική εντύπωση του φοιτητή για τον καθηγητή και το συγκεκριμένο του μάθημα. Επίσης, πρέπει να παρέχεται η δυνατότητα εισαγωγής γενικού σχολίου από τον φοιτητή. Με αυτόν τον τρόπο θα μπορεί να καταγγείλει αρνητικές ενέργειες και συμπεριφορές του καθηγητή κατά την περάτωση του μαθήματός του ή να επικροτήσει καινοτομίες και καλές πρακτικές.

Η μέθοδος Likert θα χρησιμοποιηθεί για την ποσοτικοποίηση της κάθε αξιολόγησης. Το σύνολο των αξιολογήσεων θα πρέπει να είναι μετρήσιμο ώστε να μπορούν να γίνουν κατάλληλες αναλύσεις δεδομένων. Για την προστασία των δεδομένων θα πρέπει ο καθηγητής να μην μπορεί να επηρεάσει κανένα από τα στοιχεία της αξιολόγησης.

Ο φοιτητής πρέπει να μπορεί να διατηρήσει την ανωνυμία του κατά την ψηφοφορία και να μπορεί να αξιολογήσει ελεύθερα τον κάθε καθηγητή στα μαθήματα που έχει παρακολουθήσει. Λαμβάνοντας υπόψιν ότι σε ένα μάθημα μπορεί να υπάρξει παραπάνω από ένας καθηγητής, πρέπει να μπορεί να συμπληρώσει διαφορετικά το ερωτηματολόγιο για τον κάθε καθηγητή.

Φυσικά, πρέπει ο χρήστης της εφαρμογής να μην μπορεί να συμπληρώσει το ερωτηματολόγιο όσες φορές επιθυμεί, αλλά ανάλογα με πόσα μαθήματα και αντίστοιχους καθηγητές έχει επιλέξει στο συγκεκριμένο εξάμηνο φοίτησής του.

Επομένως, κρίνεται απαραίτητη η αυθεντικοποίηση του χρήστη ώστε να μπορούν να εξακριβωθούν οι δυνατότητές του μέσα από το σύστημα και ο αριθμός των αξιολογήσεων που έχει δικαίωμα να συμπληρώσει. Πρέπει επίσης, να έχει δικαίωμα να συμπληρώσει μόνο ένα ερωτηματολόγιο για κάθε συνδυασμό μαθήματος και καθηγητή.

Ένας ακόμα περιορισμός, πρέπει να είναι το χρονικό περιθώριο που θα δίνεται στον φοιτητή από τον φορέα για την αξιολόγηση. Η αξιολόγηση θα πρέπει να γίνεται σε συγκεκριμένο χρονικό περιθώριο, μετά την ολοκλήρωση των μαθημάτων και πριν την έκδοση των αποτελεσμάτων του καθώς μπορεί να επηρεαστεί από την βαθμολογία του.

Σημαντική είναι και η ευκολία χρήσης των φοιτητών της εφαρμογής, παρέχοντας τους την δυνατότητα να αξιολογήσουν στον δικό τους χρόνο και με την δική τους άνεση μέσα στο κατάλληλο χρονικό περιθώριο που καθορίζεται από τον φορέα. Για την ευκολία χρήσης θα πρέπει η εφαρμογή να λειτουργεί με εύκολο και συμβατό τρόπο σε κινητές συσκευές ή σε ηλεκτρονικούς υπολογιστές. Επίσης, πρέπει να μπορεί ο κάθε χρήστης να χρησιμοποιήσει την εφαρμογή από τον φυλλομετρητή της επιλογής του, χωρίς περιορισμούς.

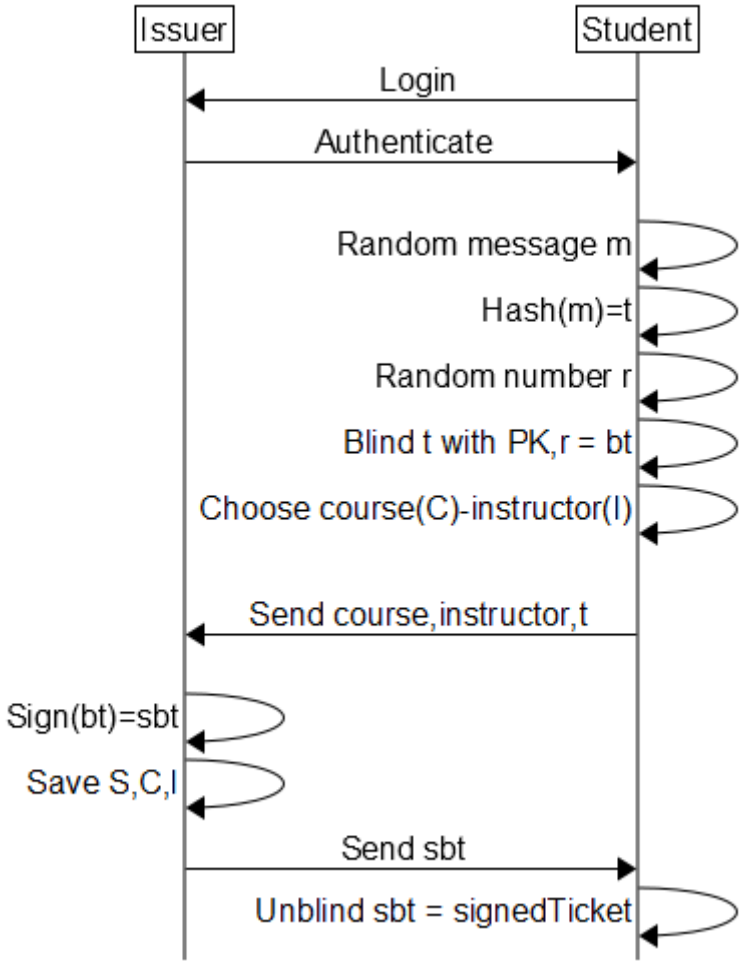
Το γραφικό περιβάλλον έχει ιδιαίτερα σημαντικό ρόλο όσον αφορά την ευχρηστία της εφαρμογής. Έτσι, πρέπει να είναι σχεδιασμένο ώστε ο φοιτητής να καταλαβαίνει τις δυνατότητές της και να ενημερώνεται μέσω μηνυμάτων στην οθόνη του για την σωστή ή μη χρήση της.

3.2 Σχεδίαση πρωτοκόλλου ανώνυμης αξιολόγησης

Παρακάτω, παρατίθεται το πρωτόκολλο που εξασφαλίζει την ανωνυμία του φοιτητή κατά την διαδικασία της αξιολόγησης, καθώς και όλες οι απαραίτητες ενέργειες που χρειάζονται.

Για την διαδικασία της αξιολόγησης θα χρησιμοποιήσουμε δύο διαφορετικές εφαρμογές που μπορούν να φιλοξενηθούν σε διαφορετικούς διακομιστές.

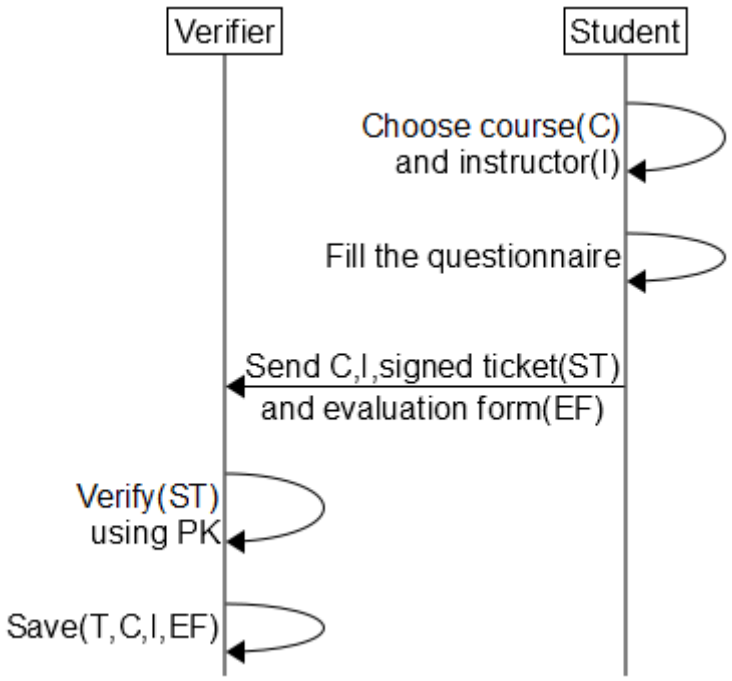
Η πρώτη εφαρμογή χρησιμοποιείται για την έκδοση ενός εισιτηρίου στον χρήστη. Για την δημιουργία αυτού του εισιτηρίου ακολουθείτε το παρακάτω πρωτόκολλο.



Εικόνα 2. Διάγραμμα πρωτοκόλλου δημιουργίας υπογεγραμμένου εισιτηρίου

Η δεύτερη εφαρμογή χρησιμοποιείται για την αξιολόγηση ενός καθηγητή και ενός επιλεγμένου μαθήματός του. Ο φοιτητής χρειάζεται το υπογεγραμμένο εισιτήριο που παράγεται από το παραπάνω διάγραμμα.

Παρακάτω, υπάρχει το διάγραμμα που αντικατοπτρίζει τις απαραίτητες ενέργειες του φοιτητή και της εφαρμογής για μία επιτυχημένη αξιολόγηση.



Εικόνα 3 Διάγραμμα πρωτοκόλλου καταβολής αξιολόγησης

3.3 Σχεδίαση του συστήματος ανώνυμης αξιολόγησης

Για το σύστημα ανώνυμης αξιολόγησης, απαιτείται η υλοποίηση δύο εφαρμογών.

Στην πρώτη εφαρμογή, ο χρήστης μετά την αυθεντικοποίησή του, επιλέγει ένα τυχαίο μήνυμα από πέντε λέξεις μίας λίστας.

Χρησιμοποιώντας μία κατάλληλη συνάρτηση κατακερματισμού, ο χρήστης παράγει το hash του υπαρκτού αυτού μηνύματος. Το νέο αυτό μήνυμα αποτελεί το ticket του χρήστη για την συμπλήρωση και κατάθεση του ερωτηματολογίου του. Η αποστολή αυτού του ticket όμως θα πρόδιδε την ταυτότητά του εάν γινόταν στο συγκεκριμένο στάδιο του πρωτοκόλλου καθώς ο διακομιστής θα μπορούσε να το αποθηκεύσει και να το διασταυρώσει με αυτόν.

Για να επιτευχθεί η απόκρυψη του νέου αυτού μηνύματος, ο χρήστης χρησιμοποιεί έναν τυχαίο αριθμό r και το δημόσιο κλειδί του διακομιστή για να το τυφλώσει. Με αυτόν τον τρόπο ο διακομιστής δεν γνωρίζει το αρχικό μήνυμα γιατί δεν έχει την πληροφορία της τιμής του αριθμού r, ώστε να δοκιμάσει να δημιουργήσει το τυφλό αυτό μήνυμα. Η χρήση του δημοσίου κλειδιού, όταν υπογράψει ο διακομιστής το μήνυμα θα απομείνει μόνο το υπογεγραμμένο τυφλό μήνυμα. Μετά την περάτωση των παραπάνω ενεργειών, ο χρήστης έχει ένα τυφλό μήνυμα, που μπορεί να αποκρυπτογραφήσει με την χρήση

του αριθμού r μόνο αφού το έχει υπογράψει ο διακομιστής με την χρήση του ιδιωτικού κλειδιού.

Ο φοιτητής στη συνέχεια επιλέγει τον καθηγητή και το μάθημα που τον ενδιαφέρει να αξιολογήσει και τα στέλνει μαζί με το τυφλό κρυπτογραφημένο μήνυμα στον διακομιστή της πρώτης εφαρμογής.

Ο διακομιστής Α με τη σειρά του υπογράφει το συγκεκριμένο μήνυμα και έτσι απομένει μόνο το τυφλό εισιτήριο που αποτελεί το προϊόν της συνάρτησης κατακερματισμού του τυχαίου μηνύματος που παράχθηκε από τον χρήστη, αλλά κρυπτογραφημένο με τον τυχαίο αριθμό r. Επίσης, μπορεί να αποθηκεύσει την επιλογή του φοιτητή ως προς τον καθηγητή και το μάθημα. Έτσι, μπορεί να αποφύγει την αίτηση του φοιτητή για τον ίδιο συνδυασμό και την παραγωγή άπειρων υπογραφών για αυτόν.

Ο διακομιστής Α στέλνει το τυφλό υπογεγραμμένο μήνυμα στον φοιτητή.

Κατά την παραλαβή του υπογεγραμμένου αυτού μηνύματος, ο φοιτητής χρησιμοποιεί τον τυχαίο αριθμό r έτσι ώστε να απομείνει μόνο το υπογεγραμμένο hash του αρχικού μηνύματος. Ο φοιτητής αποθηκεύει το έγκυρο και υπογεγραμμένο πλέον μήνυμα για να το χρησιμοποιήσει σε οποιοδήποτε χρόνο (εντός της προθεσμίας αξιολογήσεων) επιθυμεί.

Η δεύτερη εφαρμογή του συστήματος είναι υπεύθυνη για την συμπλήρωση και αποθήκευση της φόρμας αξιολόγησης του φοιτητή.

Κατά την είσοδό του στην δεύτερη εφαρμογή, καλείται να συμπληρώσει το ερωτηματολόγιο μορφής Likert, καθώς και το προαιρετικό σχόλιό του. Στη συνέχεια, συμπληρώνει μία δεύτερη μικρότερη φόρμα όπου επιλέγει τον καθηγητή και το μάθημα που επιθυμεί και καταθέτει το υπογεγραμμένο μήνυμα που απέκτησε από τον διακομιστή Α. Μετά την αποστολή των παραπάνω δεδομένων στον διακομιστή Β ο φοιτητής έχει τελειώσει με την διαδικασία της αξιολόγησης.

Ο διακομιστής Β αρκεί να χρησιμοποιήσει το δημόσιο κλειδί του διακομιστή Α για να ελέγξει την εγκυρότητα του ticket. Εάν είναι έγκυρο τότε αποθηκεύει τις επιλογές του ερωτηματολογίου και το σχόλιο του χρήστη στη βάση δεδομένων του συστήματος και σημειώνει το συγκεκριμένο μήνυμα ως χρησιμοποιημένο.

3.4 Μαθηματική προσέγγιση του πρωτοκόλλου

Είναι σημαντικό να εξεταστεί εάν το συγκεκριμένο πρωτόκολλο μπορεί να αποφέρει την επιθυμητή ανωνυμία και μαθηματικά. Με την μαθηματική αυτή απόδειξη αποδεικνύουμε την δυνατότητα υλοποίησης του συστήματος και την επιτυχή απόκρυψη της πληροφορίας που θα συνέδεε ένα ερωτηματολόγιο με έναν φοιτητή.

Αρχικά, ορίζουμε ότι το ιδιωτικό κλειδί που χρησιμοποιείται για την υπογραφή του μηνύματος του χρήστη είναι SK(d,n) ενώ το δημόσιο κλειδί του, που χρησιμοποιείται για την κρυπτογράφηση του μηνύματος, είναι PK(e,n).

Έτσι, κατά την τύφλωση του μηνύματος το blinded_ticket θα είναι ως εξής:

$$blinded\_ticket = (tr^e) \mod n$$

Στη συνέχεια, ο διακομιστής Α θα υπογράψει το συγκεκριμένο τυφλό μήνυμα.

$$signed\_blinded\_ticket = (blinded\_ticket)^d \mod n$$

Μετά την αντικατάσταση του blinded_ticket στην παραπάνω εξίσωση:

$$signed\_blinded\_ticket = (tr^e)^d \mod n$$

Λόγω της σχέσης του ιδιωτικού και του δημοσίου κλειδιού RSA, προκύπτει:

$$signed\_blinded\_ticket = t^d r \mod n$$

Στη συνέχεια, μετά της αποστολή του παραπάνω μηνύματος, ο χρήστης μπορεί να αποβάλλει το r και να κρατήσει την έγκυρη υπογραφή του μηνύματος από τον διακομιστή Α ως εξής:

$$signed\_ticket = signed\_blinded\_ticket \ r^{-1} \mod n$$

$$signed\_ticket = t^d \mod n$$

Επομένως, με τη χρήση του δημοσίου κλειδιού, μπορεί να ταυτοποιηθεί η εγκυρότητα της υπογραφής του διακομιστή Α και να παραμείνει μόνο το αρχικό εισιτήριο για την συμπλήρωση του ερωτηματολογίου.

3.5 Σχεδίαση βάσης δεδομένων

Απαραίτητη πριν την υλοποίηση της εφαρμογής κρίνεται η σχεδίαση της βάσης δεδομένων που θα χρησιμοποιούν οι εφαρμογές για αποθήκευση δεδομένων.

Η βάση δεδομένων σχεδιάστηκε με γνώμονα μία ήδη υπάρχουσα βάση δεδομένων ενός πανεπιστημίου που χρησιμοποιείται για την αποθήκευση δεδομένων σχετικά με τις σχέσεις μεταξύ φοιτητών, καθηγητών και μαθημάτων. Οι σχέσεις αυτές απαντούν σε ερωτήματα όπως ποια μαθήματα παρακολουθεί ο κάθε φοιτητής, ποιοι καθηγητές διδάσκουν το εκάστοτε μάθημα κτλ. Θεωρούμε κοινή την βάση δεδομένων για ολόκληρο το πανεπιστήμιο και όχι διαφορετική βάση για κάθε ένα από τα τμήματα που ανήκουν σε αυτό.

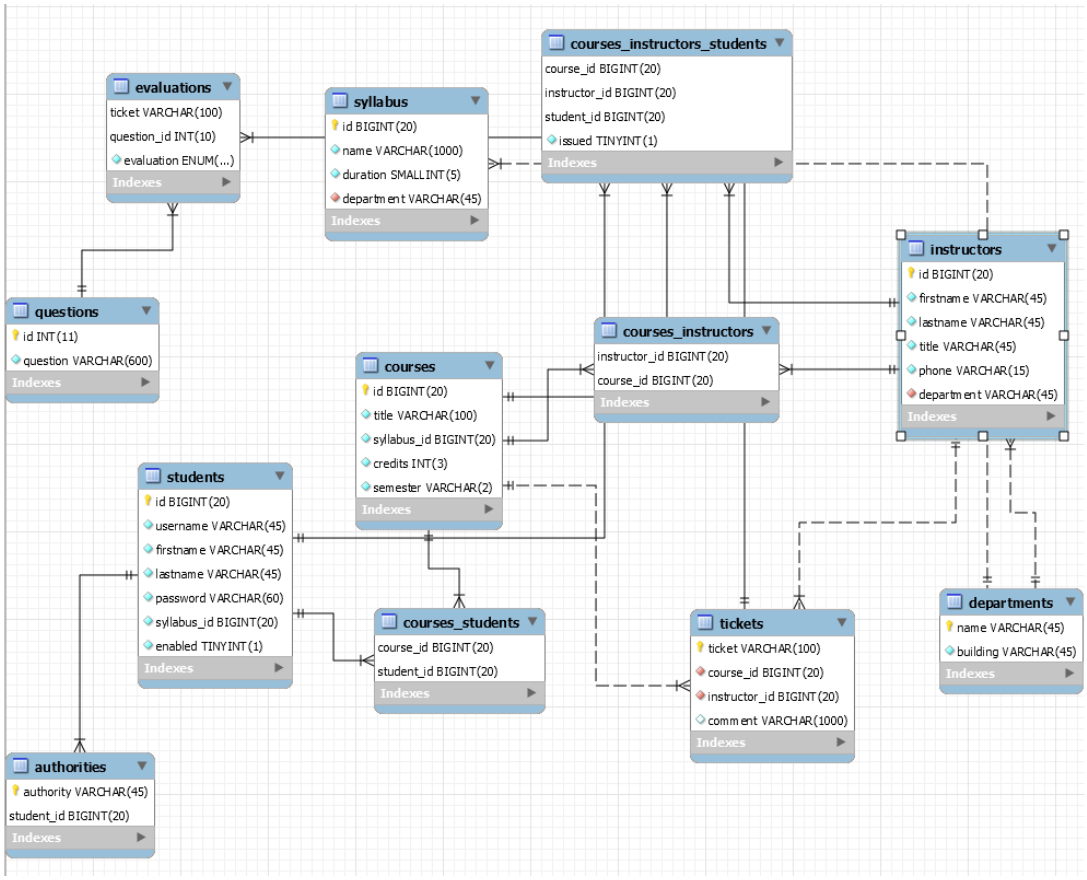
Στην ήδη υπάρχουσα βάση δεδομένων, υπάρχουν πληροφορίες σχετικά με τους φοιτητές, τους καθηγητές και τα μαθήματα που διδάσκονται στο πανεπιστήμιο. Επίσης υπάρχουν πληροφορίες σχετικά με τα τμήματα του πανεπιστημίου καθώς και των προγραμμάτων σπουδών που υπάρχουν σε κάθε ένα από αυτά.

Σε αυτήν τη βάση δεδομένων, θα προσθέσουμε τις οντότητες και τις συσχετίσεις που απαιτούνται ώστε να μπορεί να αποθηκευτεί η αξιολόγηση του κάθε καθηγητή για κάθε μάθημα που διδάσκει από τον εκάστοτε φοιτητή, καθώς και τις απαραίτητες προϋποθέσεις ώστε το κάθε εισιτήριο για αξιολόγηση να μην μπορεί να επαναχρησιμοποιηθεί. Επίσης, λαμβάνεται υπόψιν ότι ο κάθε φοιτητής έχει δικαίωμα να πάρει ένα εισιτήριο για κάθε συνδυασμό καθηγητή και μάθημα που παρακολουθεί ώστε να μην μπορεί να δημιουργήσει άπειρα εισιτήρια.

Με την εξασφάλιση της ανωνυμίας των αξιολογήσεων, απαραίτητη είναι η ποσοτικοποίηση των δεδομένων από το κάθε ερωτηματολόγιο Likert διότι πιθανές αναλύσεις δεδομένων σε αυτές είναι αποδεκτές από το GDPR.

Τέλος, υποθέτουμε πως τις αξιολογήσεις τις χρησιμοποιούν μόνο φοιτητές, οπότε οι απαραίτητες εγγραφές στη βάση για την αυθεντικοποίηση και την εξουσιοδότηση των καθηγητών δεν υπάρχουν για τους καθηγητές.

Παρακάτω, παρατίθεται το σχεδιάγραμμα της βάσης δεδομένων που χρησιμοποιείται και από τις δύο εφαρμογές του συστήματός μας.



Εικόνα 4 Σχεσιακή βάση δεδομένων του συστήματος

4. Ανάλυση ασφάλειας

Στο κεφάλαιο αυτό αναλύονται όλες οι απαραίτητες προϋποθέσεις που χρειάζονται για να θεωρηθεί ασφαλής η υλοποίηση του πρωτοκόλλου.

4.1 Γεννήτρια ψευδοτυχαίων αριθμών

Στο πρωτόκολλο αναφέρεται ότι ο χρήστης της εφαρμογής θα πρέπει να διαλέξει έναν τυχαίο αριθμό για να δημιουργηθεί η τυφλή υπογραφή την οποία θα χρησιμοποιήσει ώστε να κρατήσει την υπογραφή του διακομιστή διατηρώντας την ανωνυμία του.

Είναι μείζονος σημασίας αυτός ο τυχαίος αριθμός r να μην μπορεί να προβλεφθεί από τον διακομιστή ή από τρίτους ώστε να διατηρηθεί η ανωνυμία. Επίσης με την γεννήτρια αυτή θα υπολογιστεί και το τυχαίο μήνυμα m το οποίο ουσιαστικά υπογράφεται από τον διακομιστή Α.

Φυσικά, η συγκεκριμένη διαδικασία γίνεται μέσα από τον φυλλομετρητή του χρήστη, επομένως πρέπει να βρεθεί ένας τρόπος που να παράγονται τυχαίοι αριθμοί χωρίς να είναι εύκολο να προβλεφθούν από έναν επιτιθέμενο.

Για αυτόν τον λόγο θα χρησιμοποιηθεί η window.crypto.getRandomValues αντί για την Math.Random.

Παρακάτω, φαίνεται η συνάρτηση που χρησιμοποιεί την εν λόγω συνάρτηση ώστε να παραχθεί ένας τυχαίος αριθμός μεταξύ ενός εύρους τιμών.

```
function getRandomInt(min, max) {
    var randomBuffer = new Uint32Array( length: 1);

    window.crypto.getRandomValues(randomBuffer);

    var randomNumber = randomBuffer[0] / (0xffffffff + 1);

    min = Math.ceil(min);
    max = Math.floor(max);
    return Math.floor( x: randomNumber * (max - min + 1)) + min;
}
```

Εικόνα 5 Γεννήτρια ψευδοτυχαίων αριθμών

4.2 Ασφάλεια φοιτητικών λογαριασμών

Η προστασία των φοιτητικών λογαριασμών στο επίπεδο εφαρμογής είναι πολύ σημαντική. Στην περίπτωση κλοπής συνθηματικών πρόσβασης, ο επιτιθέμενος μπορεί να χρησιμοποιήσει όλες τις δυνατές αξιολογήσεις του φοιτητή. Λόγω της ανωνυμίας που παρέχεται από το σύστημα ανώνυμης αξιολόγησης μία τέτοια ενέργεια δεν είναι πλήρως αντιστρέψιμη. Αφενός μπορούμε να επιτρέψουμε στον φοιτητή να ξαναδημιουργήσει τις φόρμες αξιολόγησης αλλά αφετέρου δεν μπορούμε να γνωρίζουμε τα αποτελέσματα των αξιολογήσεων από τα εισιτήρια που παράχθηκαν με τον εν λόγω λογαριασμό. Επομένως, πρέπει να διασφαλίσουμε τους λογαριασμούς των φοιτητών από κοινές επιθέσεις σε αυτούς.

Αρχικά

Οι πιο διαδεδομένες απόπειρες κλοπής συνθηματικών είναι του τύπου brute force attack. Ο επιτιθέμενος προσπαθεί να πετύχει το σωστό αλφαριθμητικό χρησιμοποιώντας είτε έτοιμες λίστες από κοινούς κωδικούς πρόσβασης, είτε από παραγόμενη λίστα κωδικών με βάση προσωπικά στοιχεία του ιδιοκτήτη του λογαριασμού (Dictionary attacks).

Τέτοιες επιθέσεις περιορίζονται σημαντικά εάν περιοριστούν οι πιθανές απόπειρες εισόδου. Στην εφαρμογή που δημιουργούνται τα εισιτήρια που χρειάζονται για τις αξιολογήσεις, μπορούμε να περιορίσουμε τις αποτυχημένες προσπάθειες πρόσβασης σε έναν αριθμό της επιλογής μας. Στην περίπτωση που υπερβεί ο λογαριασμός αυτόν τον αριθμό προσπαθειών απενεργοποιείται ο λογαριασμός του προσωρινά.

```
@Transactional
@Override
public void updateLoginAttempts(String username) {
    Student student = userDetailsDao.findUserByUsername(username);
    student.setLoginAttempts(student.getLoginAttempts()+1);
    if(student.getLoginAttempts() >= Constants.MAXIMUM_LOGIN_ATTEMPTS){
        student.setEnabled(false);
    }
}
```

Εικόνα 6 Ανανέωση προσπαθειών εισόδου

Επιπλέον, κάθε φορά που ο χρήστης κάνει μία επιτυχημένη προσπάθεια εισόδου στο σύστημα, πρέπει να ανανεωθεί η αντίστοιχη τιμή στη βάση δεδομένων.

```
@Transactional
@Override
public void successfulLogin(String username) {
    Student student = userDetailsDao.findUserByUsername(username);
    student.setEnabled(true);
    student.setLoginAttempts(0);
}
```

Εικόνα 7 Επαναφορά προσπαθειών εισόδου

4.3 Προστασία πληροφορίας από τρίτους

Κατά την επικοινωνία με τους διακομιστές, πρέπει να εξασφαλίζεται ότι δεν μπορεί να παρέμβει τρίτος στην επικοινωνία, είτε για να τροποποιήσει είτε για να αποθηκεύσει πληροφορία. Ο τύπος μίας τέτοιας επίθεσης ονομάζεται man-in-the-middle.

Για την προστασία από τέτοιες επιθέσεις, χρησιμοποιείται το πρωτόκολλο SSL/TLS (Secure Sockets Layer / Transport Layer Security). Με τα πρωτόκολλα αυτά, ο φοιτητής θα μπορεί να γνωρίζει την ταυτότητα του διακομιστή επαληθεύοντάς την από μία αρχή πιστοποίησης. Η συγκεκριμένη διαδικασία γίνεται μέσω πιστοποιητικών που έχουν εκδοθεί από την αρχή πιστοποίησης.

Στις εφαρμογές μας, θα επιτρέψουμε μόνο την χρήση του πρωτοκόλλου HTTPS (Hypertext Transfer Protocol Secure) η οποία χρησιμοποιεί το πρωτόκολλο TLS ώστε να αποφύγουμε τις επιθέσεις τύπου man-in-the-middle.

Η συγκεκριμένη ενέργεια θα γίνει μέσω των spring security filters και παρακάτω φαίνεται η ενδεικτική φωτογραφία με τις κατάλληλες ρυθμίσεις.

```
@Override
protected void configure(HttpSecurity http) throws Exception {
  http.requiresChannel().anyRequest().requiresSecure() ChannelSecurityConfigurer<H>.ChannelRequestMatcherRegistry
  .and() HttpSecurity
  .authorizeRequests().antMatchers( ...antPatterns: "/*").permitAll() ExpressionUrlAuthorizationConfigurer<H>.ExpressionInterceptUrlRegistry
  .and() HttpSecurity
  .csrf().disable();
}
```

Εικόνα 8 Ρυθμίσεις για την χρήση του πρωτοκόλλου TLS

4.4 Ανάλυση ασφάλειας του πρωτοκόλλου

Η ανωνυμία που εξασφαλίζεται μέσα από το πρωτόκολλο επιτρέπει στον φοιτητή να εκφράσει την άποψή του αμερόληπτα και χωρίς φόβους. Παρόλα αυτά, δημιουργούνται προβλήματα όταν δεν μπορούμε να περιορίσουμε τον φοιτητή στο να μπορεί να αξιολογήσει μόνο καθηγητές και μαθήματα που παρακολουθεί.

Ένας φοιτητής μπορεί να χρησιμοποιήσει όλα του τα εισιτήρια για να αξιολογήσει τον ίδιο καθηγητή και μάθημα είτε με κακόβουλες είτε με καλοπροαίρετες αξιολογήσεις επηρεάζοντας σημαντικά τα δεδομένα που προκύπτουν για την συνολική αξιολόγηση του καθηγητή και του μαθήματός του. Τέτοιες περιπτώσεις μπορούν μεν να διαπιστωθούν μέσω της ανάλυσης της σχέσης του μαθήματος και του συνόλου των φοιτητών που παρακολουθούν το εκάστοτε μάθημα, αλλά δεν γίνεται να αντιστραφούν ή να αφαιρεθούν με ακρίβεια από τη βάση δεδομένων. Η αφαίρεση των αξιολογήσεων είναι αδύνατη καθώς δεν υπάρχει τρόπος να εξακριβωθεί εάν όλες οι αρνητικές ή θετικές αξιολογήσεις αποτελούν μέρος της αξιολόγησης, και θα μπορούσαν να αφαιρεθούν πραγματικές αξιολογήσεις από φοιτητές που παρακολουθούν το συγκεκριμένο μάθημα.

Επίσης, θα μπορούσε ένας φοιτητής να χρησιμοποιήσει τα εισιτήρια που δικαιούται άλλος φοιτητής μέσω συνεννόησής τους ή ακόμα και αγοράς. Δηλαδή, δεν υπάρχει τρόπος να εξακριβωθεί ποιος φοιτητής χρησιμοποίησε τα εισιτήρια αλλά μόνο ποιος φοιτητής έκανε αίτηση και για ποια μαθήματα και καθηγητές.

Άλλο ένα πρόβλημα που δημιουργείται με την εξασφάλιση της ανωνυμίας της αξιολόγησης, είναι η περίπτωση κλοπής συνθηματικών πρόσβασης φοιτητών. Παρότι έχουν προληφθεί αρκετοί τύποι επιθέσεων για την προστασία των φοιτητικών λογαριασμών, ένας κωδικός πρόσβασης μπορεί να κλαπεί και λόγω επιπολαιότητας του κατόχου του. Ο επιτιθέμενος, μπορεί να χρησιμοποιήσει όλα τα εισιτήρια που δικαιούται ο κάτοχος του λογαριασμού και να τα αξιοποιήσει με όποιον τρόπο θέλει. Στην περίπτωση που διαπιστωθεί η κλοπή του λογαριασμού, δεν μπορούν να αντιστραφούν οι αξιολογήσεις που δημιουργήθηκαν, ούτε να καταργηθούν τα δημιουργημένα εισιτήρια. Αυτό συμβαίνει διότι η ανωνυμία που εξασφαλίζεται δεν επιτρέπει στο να αναγνωριστεί ποια αξιολόγηση έγινε από ποιόν φοιτητή. Μπορεί όμως, να ξαναδοθεί στον φοιτητή το δικαίωμα έκδοσης νέων εισιτηρίων ώστε να μπορεί να τα χρησιμοποιήσει για καινούριες αξιολογήσεις.

Στην περίπτωσή μας, το σύστημα έχει δομηθεί ως σύστημα πανεπιστημίου, το οποίο επιτρέπει στους φοιτητές να αξιολογήσουν μέχρι και καθηγητές άλλων προγραμμάτων σπουδών και τμημάτων.

4.5 Αντιμετώπιση προβληματικών γεγονότων

Για όλες τις περιπτώσεις που αναφέρονται στο προηγούμενο κεφάλαιο, κρίνεται απαραίτητο να υπάρχει η δυνατότητα του εντοπισμού τους, ώστε να μπορούν να καταπολεμηθούν και να προβλεφθούν στο μέλλον.

Θα πρέπει να υπάρξει τρόπος να εξακριβωθούν προβλήματα που ενδεχομένως ισχυρίζονται οι φοιτητές ότι έγιναν σε μία συγκεκριμένη περίοδο. Για αυτόν τον λόγο θα πρέπει να υπάρχει ένα αυτόματο αρχείο καταγραφής των ενεργειών που συμβαίνουν στην εκάστοτε εφαρμογή.

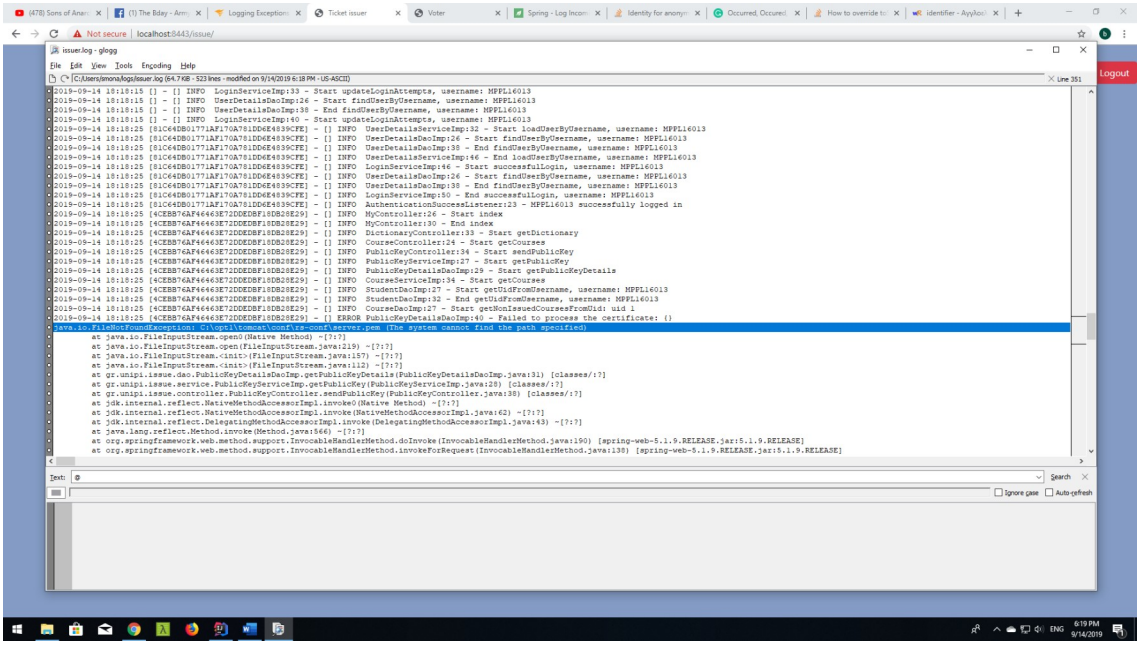
Αυτό το σύστημα καταγραφής, μας επιτρέπει να αναλύσουμε τι ακριβώς έγινε και σε ποια χρονική περίοδο, ώστε να εντοπιστούν τα προβλήματα που έγιναν κατά τον ορισμό των εισιτηρίων ή την ίδια την αξιολόγηση.

Έτσι, σε περίπτωση παραπόνων των χρηστών (όπως π.χ. ότι δεν έγινε σωστή η δημιουργία του εισιτηρίου) μπορούμε να εξακριβώσουμε τι ακριβώς έγινε, και να προβούμε στις κατάλληλες ενέργειες και την ενημέρωση των φοιτητών.

Για αυτό το σύστημα χρησιμοποιήθηκε η βιβλιοθήκη log4j2 η οποία μας επιτρέπει να καταγράψουμε όλα τα προβλήματα που μπορεί να δημιουργηθούν. Μπορούμε να καταγράψουμε και την σύνοδο (session) που υπάρχει κατά την περάτωση των απαραίτητων ενεργειών που χρειάζονται κατά την ανώνυμη αξιολόγηση. Αυτό θα βοηθήσει σημαντικά στο να διασταυρωθούν οι ενέργειες που έγιναν από τον χρήστη.

Είναι άξιο αναφοράς πως η ύπαρξη αυτού του συστήματος καταγραφής ενεργειών δεν μπορεί να καταπατήσει την ανωνυμία που προσφέρει το σύστημα, καθώς η ψηφοφορία γίνεται σε άλλο διακομιστή και τα αναγνωριστικά των συνόδων είναι διαφορετικά. Δηλαδή, εξακολουθεί να μην υπάρχει τρόπος να διασταυρωθεί η έκδοση ενός εισιτηρίου για έναν φοιτητή με την αξιολόγησή του.

Μία πιθανή περίπτωση εξακρίβωσης τέτοιου γεγονότος, θα μπορούσε να είναι ο ισχυρισμός του φοιτητή ότι δεν μπορεί να εκδώσει ένα εισιτήριο ο διακομιστής. Στην παρακάτω εικόνα μπορούμε να δούμε ότι ο χρήστης MPPL16013, με αναγνωριστικό 1, κατάφερε επιτυχώς να εισαχθεί στην εφαρμογή, αλλά ο διακομιστής δεν κατάφερε να δώσει τα στοιχεία του δημοσίου κλειδιού του, λόγω λάθος ρυθμίσεων στον διακομιστή.



Εικόνα 9 Παράδειγμα καταπολέμησης προβλημάτων μέσω του αρχείου καταγραφών.

4.6 Ποιότητα κώδικα και προστασία από κλασικές επιθέσεις

Κατά τη δημιουργία της εφαρμογής θα χρησιμοποιηθεί το πρόγραμμα SonarQube, το οποίο εντοπίζει αδυναμίες ασφαλείας στο επίπεδο εφαρμογής μέσα από τον κώδικά μας, καθώς επίσης και γενικότερα προγραμματιστικά σφάλματα που μπορεί να δημιουργήσουμε μεγάλα προβλήματα κατά την λειτουργία των διαδικτυακών εφαρμογών στο σύστημα.
Επίσης θα χρησιμοποιηθεί και το πρόγραμμα OWASP ZAP (v. 2.8.0) για να εντοπιστούν τυχόν αδυναμίες που δεν είναι καταγεγραμμένες στο σύστημα του SonarQube.
Έτσι θα διασφαλίσουμε την ασφάλεια της εφαρμογής στις πιο συχνές επιθέσεις (όπως π.χ. SQL Injections), καθώς και θα διασφαλίσουμε περαιτέρω την ασφάλεια της εφαρμογής μας.
Αξίζει να αναφερθεί ότι στις πιο συχνές επιθέσεις που συμβαίνουν στο διαδίκτυο ανήκουν και οι επιθέσεις συνόδου (session attacks), όπως π.χ. session hijacking, session fixation κτλ. Με την πρόβλεψη της δυνατότητας τέτοιων επιθέσεων, διαφυλάσσουμε περαιτέρω την ιδιωτικότητα των χρηστών που είναι και η κύρια μέριμνα του συστήματος ανώνυμης αξιολόγησης μαθημάτων.

Η πολιτική μας θα είναι να περιορίσουμε στο 0 όλα τα sonar bugs και vulnerabilities και να ασχοληθούμε σε δεύτερο χρόνο με τα λιγότερο σημαντικά ευρήματα του SonarQube σε δεύτερο χρόνο, διότι αφορούν περισσότερο τη συντήρηση της εφαρμογής.
Όσον αφορά τα προβλήματα που θα εντοπιστούν από το OWASP ZAP, θα φροντίσουμε να περιορίσουμε τα προβλήματά μας ώστε να αποφύγουμε όλες τις πολύ διαδεδομένες επιθέσεις.

5. Υλοποίηση

5.1 Τεχνολογίες υλοποίησης εφαρμογών

Σημαντική είναι και η επιλογή των τεχνολογιών που θα χρησιμοποιηθούν στην υλοποίηση των εφαρμογών του ανώνυμου συστήματος αξιολόγησης ενός πανεπιστημίου.

Καθώς μιλάμε για διαδικτυακές εφαρμογές απαραίτητο είναι να χρησιμοποιηθούν οι γλώσσες html, CSS και JavaScript.

Θα διευκολύνει ιδιαίτερα την υλοποίηση των διεπαφών χρήσης, το πακέτο bootstrap. Έτσι, θα επιτύχουμε εύκολη χρήση των εφαρμογών και μέσω διαφορετικών συσκευών από τον ηλεκτρονικό υπολογιστή (όπως π.χ. κινητά τηλέφωνα και tablets).

Οι ενέργειες που απαιτούνται για την έκδοση του εισιτηρίου που θα χρησιμοποιηθεί για την αξιολόγηση, καθώς και οι ενέργειες που απαιτούνται από αυτήν από τον χρήστη πρέπει να γίνουν μέσω JavaScript ώστε να είναι συμβατός ο κώδικας με κάθε φυλλομετρητή που επιλέγει να χρησιμοποιήσει ο χρήστης. Για την διευκόλυνση των παραπάνω ενεργειών θα χρησιμοποιηθεί η βιβλιοθήκη jQuery. Για την υλοποίηση της εφαρμογής θα πρέπει να χρησιμοποιηθεί και η βιβλιοθήκη jsbn (JavaScript Big Numbers), διότι υπάρχουν υπολογισμοί με νούμερα μεγαλύτερα από το εύρος που αποθηκεύει η JavaScript.

Η γλώσσα προγραμματισμού που θα χρησιμοποιούν οι διακομιστές θα είναι η νεότερη έκδοση της Java (v. 11.0.2). Για την ευκολότερη διαχείριση των απαραίτητων πακέτων που θα χρησιμοποιηθούν και θα αναφερθούν παρακάτω, χρειάζεται να χρησιμοποιηθεί ένα εργαλείο κατασκευής (build tool). Για αυτόν τον ρόλο θα χρησιμοποιηθεί το Apache Maven 3.6.1.

Η επιλογή της Java είναι αλληλένδετη με τα θετικά που προσφέρει το Spring Framework και η εύκολη πρόσβαση που παρέχεται από το μοτίβο σχεδιασμού MVC (Model View Controller) για το διαδίκτυο Spring Web MVC (v. 5.1.16).

Επίσης μέσω του JPA (Java Persistence API), μπορούμε να επιτύχουμε εύκολες, σωστές και ασφαλές συναλλαγές (transactions) με την βάση δεδομένων του πανεπιστημίου. Για την περαιτέρω διευκόλυνση στις κινήσεις που γίνονται μέσω της βάσης δεδομένων θα χρησιμοποιηθεί και η βιβλιοθήκη Hibernate για το ORM (Object Relational Mapping). Εξαιτίας του Hibernate θα μπορούμε να συνδυάσουμε τα model Object μας με τις εγγραφές στη βάση δεδομένων που θα βοηθήσει ιδιαίτερα στην υλοποίηση.

Φυσικά, δεν θα μπορούσαμε να μιλήσουμε για ασφάλεια σε επίπεδο εφαρμογής χωρίς να χρησιμοποιήσουμε μία βιβλιοθήκη για την διαχείριση των συνόδων, καθώς και γενικότερα για την αυθεντικοποίηση και εξουσιοδότηση των εφαρμογών. Η βιβλιοθήκη που επιλέχτηκε είναι η Spring Security (v. 5.1.16).

5.2 Έκδοση εισιτηρίου

Το πρώτο βήμα είναι να επιλέξουμε από το λεξικό του συστήματος έναν αριθμό τυχαίων λέξεων ώστε να βεβαιωθούμε ότι το μήνυμά μας θα είναι υπαρκτό. Για την επιλογή των τυχαίων αυτών λέξεων θα χρησιμοποιηθεί η ασφαλής γεννήτρια τυχαίων αριθμών που θα χρησιμοποιήσουμε για να δημιουργηθεί και ο τυχαίος αριθμός r. Η συνάρτηση που δημιουργεί τέτοιους αριθμούς θα αναλυθεί περαιτέρω στη συνέχεια του κεφαλαίου.

```
function generateM(){
    var message = "";
    for(var i=0;i<5;i++){
        var r = getRandomInt(0,dictionary.length);
        var word = dictionary[r].replace(/"/g,"");
        word = word.charAt(0).toUpperCase() +word.slice(1);
        message = message + word;

    }
    return message;

}
```

Εικόνα 10. Παραγωγή αρχικού μηνύματος m

Το μήνυμα που προκύπτει από τις τυχαίες αυτές λέξεις θα τον εισάγουμε σε μία συνάρτηση κατακερματισμού ώστε να μην μπορεί να διαπιστωθεί ποιες λέξεις χρησιμοποιήθηκαν για την δημιουργία αυτού του εισιτηρίου με εύκολο και προσβάσιμο τρόπο. Για την επιλογή της συνάρτησης κατακερματισμού, επιλέχθηκε η SHA256, η οποία και δεν έχει συγκρούσεις στα αποτελέσματά της, αλλά είναι και γρήγορη υπολογιστικά και δεν μπορούμε ποτέ να υπολογίζουμε σε μεγάλη υπολογιστική δύναμη του χρήστη. Το προϊόν της συνάρτησης κατακερματισμού είναι και το εισιτήριο t που θα χρησιμοποιήσει ο φοιτητής για την αξιολόγηση ενός μαθήματος – καθηγητή.

```
function sha256(str) {
      // We transform the string into an arraybuffer.
      var buffer = new TextEncoder("utf-8").encode(str);
      return crypto.subtle.digest( algorithm: "SHA-256", buffer).then( onfulfilled: function (hash) {
        return hex(hash);
      });
}
```

Εικόνα 11. Υπολογισμός του t μέσω της SHA256

Για την επιτυχή έκδοση ενός εισιτηρίου απαραίτητη είναι η δημιουργία ενός ασφαλούς ψευδοτυχαίου αριθμού από τη μεριά του χρήστη της εφαρμογής που στην περίπτωσή μας είναι ο φοιτητής. Με την χρήση της βιβλιοθήκης windo.crypto θα επιτύχουμε να μην μπορεί να προβλεφθεί αυτός ο αριθμός από κανέναν επιτιθέμενο, ακόμα και αν έχει πρόσβαση στους διακομιστές του συστήματος. Είναι σημαντικό να μπορούμε να ελέγξουμε το εύρος αυτού του τυχαίου αριθμού και κατά συνέπεια την τυχαιότητά του. Παρακάτω παρατίθεται ο κώδικας δημιουργίας ενός τέτοιου αριθμού r.

```
function getRandomInt(min, max) {
    var randomBuffer = new Uint32Array( length: 1);

    window.crypto.getRandomValues(randomBuffer);

    var r = randomBuffer[0] / (0xffffffff + 1);

    min = Math.ceil(min);
    max = Math.floor(max);
    return Math.floor( x: r * (max - min + 1)) + min;
}
```

Εικόνα 12. Γεννήτρια τυχαίων αριθμών

Στη συνέχεια, απομένει να χρησιμοποιηθεί αυτός ο τυχαίος αριθμός r σε συνδυασμό με το δημόσιο κλειδί του διακομιστή για να τυφλωθεί το εισιτήριο. Το προϊόν της συνάρτησης κατακερματισμού είναι αλφαριθμητικό οπότε χρησιμοποιούμε τον αντίστοιχο που αντιστοιχεί στα bytes της μεταβλητής που χρησιμοποιούμε.

```
function blind(msg,r){
    return msg.multiply(r.modPow(e,n)).mod(n);
}
```

Εικόνα 13. Απόκρυψη εισιτηρίου μέσω του r

Ο διακομιστής Α παραλαμβάνει το προϊόν των παραπάνω συναρτήσεων και το υπογράφει. Μετά από αυτή την ενέργεια στέλνει πίσω στο χρήστη το προϊόν της υπογραφής.

```
// Signs the blinded ticket
@Override
public BigInteger signMessage(BigInteger message) throws
        UnrecoverableKeyException, KeyStoreException, NoSuchAlgorithmException, CertificateException, IOException {
    logger.info( s: "Start signMessage, message {}", message);

    PrivateKeyDetails privateKey = privateKeyDetailsDao.getPrivateKeyDetails();
    BigInteger blindSignature = message.modPow(privateKey.getPrivateExponent(), privateKey.getModulus());

    logger.info( s: "End signMessage, message {}, blindSignature {}", message, blindSignature);
    return blindSignature;
}
```

Εικόνα 14. Υπογραφή του τυφλού εισιτηρίου από τον διακομιστή

Αφού επιστραφεί αυτή η τυφλή υπογραφή στον φυλλομετρητή του χρήστη, πρέπει να επιβεβαιωθεί ότι από αυτή την τυφλή υπογραφή μπορεί να προκύψει το εισιτήριο t.

```
function verify(msg){
    return msg.modPow(e,n).equals(convertStringToInteger(t));
}
```

Εικόνα 15. Επαλήθευση τυφλής υπογραφής από τον χρήστη

Τέλος, εάν επιβεβαιωθεί ότι η τυφλή υπογραφή είναι σωστή και μπορεί να συνδεθεί με το εισιτήριο t, απομένει να υπολογιστεί το προϊόν της υπογραφής του t ώστε ο χρήστης να μπορεί να το χρησιμοποιήσει κατά την αξιολόγησή του.

```
function unblind(msg,r){
    return msg.multiply(r.modInverse(n)).mod(n);
}
```

Εικόνα 16. Αποβολή του τυχαίου αριθμού r από την τυφλή υπογραφή

5.3 Χρήση εισιτηρίου και αξιολόγηση

Κατά την αξιολόγηση, ο χρήστης θα χρησιμοποιήσει το μήνυμα και την υπογραφή του εισιτηρίου που εκδόθηκε από την εφαρμογή του διακομιστή Α.

Για την ακρίβεια αποστέλνονται τα απαραίτητα δεδομένα σε μορφή JSON με την HTTP μέθοδο POST.

```
var data = {
        "courseId": course_id,
        "instructorId": instructor_id,
        "message": m,
        "signedTicket": signed_ticket,
        "eval": JSON.stringify(evaluation),
        "comment": comment
};
$.post("vote",data,
```

Εικόνα 17. Αποστολή δεδομένων από τον χρήστη κατά την αξιολόγηση

Στη συνέχεια, ο διακομιστής Β αφού παραλάβει το παραπάνω JSON Object, θα πρέπει να διαπιστώσει την εγκυρότητα της υπογραφής ξεκινώντας από την παραγωγή του SHA256 με είσοδο το μήνυμα m που έστειλε ο φοιτητής. Το αποτέλεσμα της συνάρτησης κατακερματισμού αποτελεί το εισιτήριο t χωρίς την υπογραφή του διακομιστή Α.

```java
// Generates the ticket from the original message m (as a big integer)
@Override
public BigInteger generateTicket(String msg) throws NoSuchAlgorithmException {
    logger.info( s: "Start generateTicket, messsage: {}", msg);
    String hash = generateHash(msg);
    BigInteger bigInt = new BigInteger(hash.getBytes());
    logger.info( s: "End generateTicket, message: {}, ticket: {}", msg, bigInt);
    return bigInt;
}
```

Εικόνα 18. Παραγωγή εισιτηρίου από το αρχικό μήνυμα m

Η παραγωγή του t από τον διακομιστή χρησιμοποιείται για να επαληθευτεί αν είναι σωστός συνδυασμός το μήνυμα m και το εισιτήριο t, όπως φαίνεται στην παρακάτω εικόνα.

```java
@Override
public boolean isValid(String msg, BigInteger signedTicket) throws NoSuchAlgorithmException,
        FileNotFoundException, CertificateException {
    logger.info( s: "Start isValid, message: {}, signedTicket: {}", msg, signedTicket);
    BigInteger ticket = generateTicket(msg);
    PublicKeyDetails publicKey = publicKeyDao.getPublicKeyDetails();

    BigInteger verifiedTicket = signedTicket.modPow(publicKey.getExponent(), publicKey.getModulus());
    if(ticket.equals(verifiedTicket)) {
        logger.info( s: "End isValid, result: {}, message: {}, signedTicket: {}",  o: "true", msg, signedTicket);
        return true;
    }
    logger.warn( s: "End isValid, result: {}, message: {}, signedTicket: {}",  o: "false", msg, signedTicket);
    return false;
}
```

Εικόνα 19. Έλεγχος εγκυρότητας εισιτηρίου

Φυσικά, πρέπει να αποφεύγεται να χρησιμοποιείται το εισιτήριο από τον χρήστη παραπάνω από μία φορά οπότε πρέπει να αποθηκεύεται στη βάση δεδομένων. Παρακάτω φαίνεται ο έλεγχος εάν το εισιτήριο υπάρχει στη βάση δεδομένων του πανεπιστημίου.

```
// Checks if the ticket is already used
@Transactional
public boolean isUsed(String ticket) {
    logger.info( s: "Start isUsed, ticket: {}", ticket);
    try {

        ticketDao.isUsed(ticket);

    }catch(NoResultException ex) {
        logger.info( s: "End isUsed, result: {}, ticket: {}", o: "false", ticket);
        return false;
    }catch(Exception ex){
        logger.error( s: "An unkown error occured on isUsed", ex);
    }
    logger.warn( s: "End isUsed, result: {}, ticket: {}", o: "true", ticket);
    return true;
}
```

Εικόνα 20. Έλεγχος επαναχρησιμοποίησης εισιτηρίου

Παρακάτω φαίνεται η αποθήκευση της αξιολόγησης του φοιτητή στη βάση δεδομένων μας.

```
// Submits the evaluation of the user for one of the instructors and his course
@Override
public void submitEvaluation(Evaluation evaluation) throws
        ConstraintViolationException, DataIntegrityViolationException{
    logger.info( s: "Start submitEvaluation, {}", evaluation);
    sessionFactory.getCurrentSession().save(evaluation);

    logger.info( s: "End submitEvaluation, {}", evaluation);
}
```

Εικόνα 21. Εισαγωγή αξιολόγησης στη βάση δεδομένων

Τέλος, αφού όλες οι παραπάνω ενέργειες έγιναν χωρίς προβλήματα και επιτυχώς, θα αποθηκευτεί το εισιτήριο ώστε να αποφευχθεί η επαναχρησιμοποίησή του.

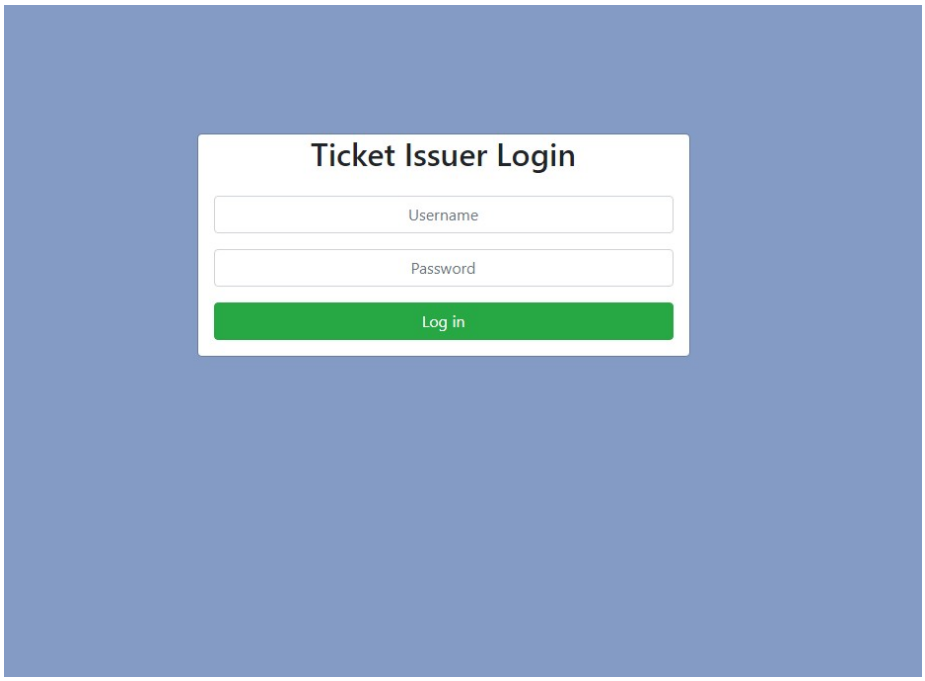
```java
// Submits the ticket on the db so that it can't be used again
@Override
public boolean submitTicket(Ticket ticket) {
    logger.info( s: "Start submitTicket, ticket: {}", ticket);
    Session session = sessionFactory.getCurrentSession();
    session.save(ticket);
    logger.info( s: "End submitTicket, ticket: {}", ticket);
    return true;
}
```

Εικόνα 22. Εισαγωγή εισιτηρίου στη βάση δεδομένων

6. Παρουσίαση

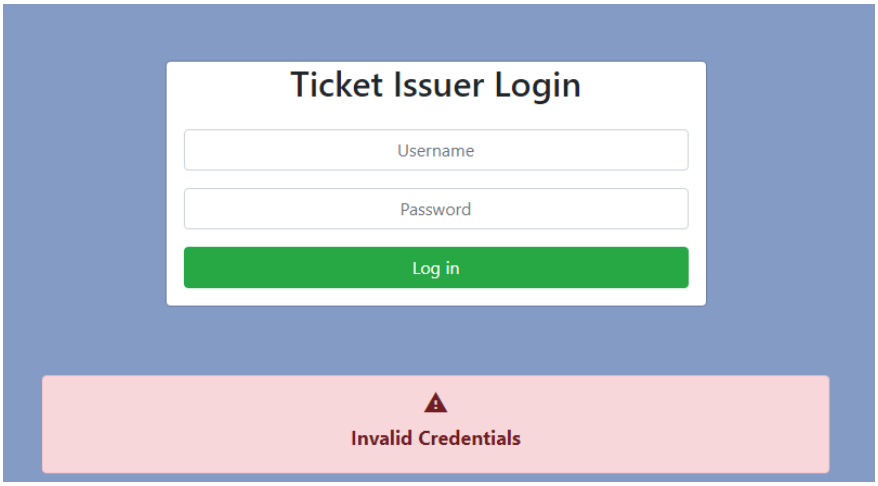
6.1 Εισαγωγή στην εφαρμογή έκδοσης εισιτηρίων

Πριν την είσοδό του στο σύστημα, ο χρήστης πρέπει να πληκτρολογήσει το username του και τον κωδικό πρόσβασής του στην παρακάτω οθόνη.



Εικόνα 23. Οθόνη login

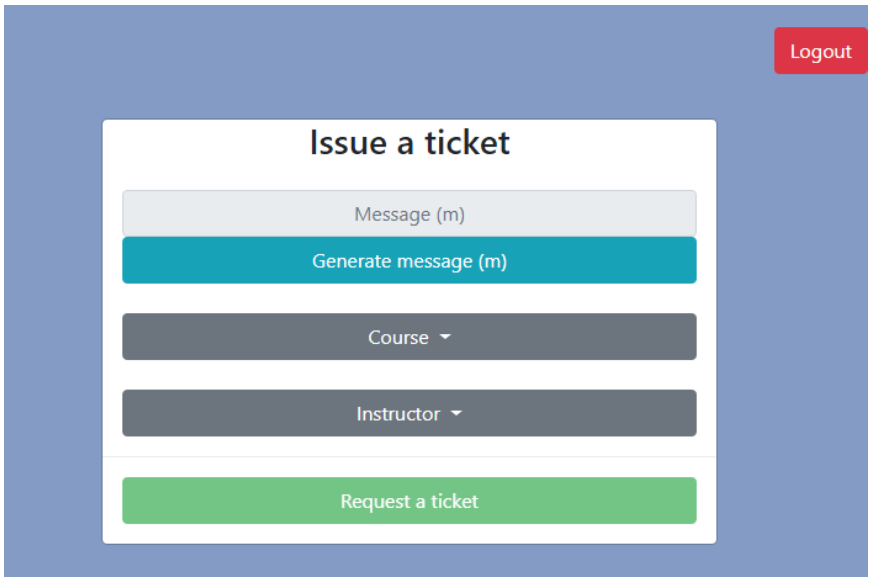
Στην περίπτωση που ο χρήστης εισάγει λάθος κωδικό πρόσβασης στο σύστημα, θα ενημερωθεί όπως φαίνεται στην παρακάτω φωτογραφία αναλόγως.



Εικόνα 24. Ενημέρωση χρήστη για λάθος εισαγωγή στοιχείων

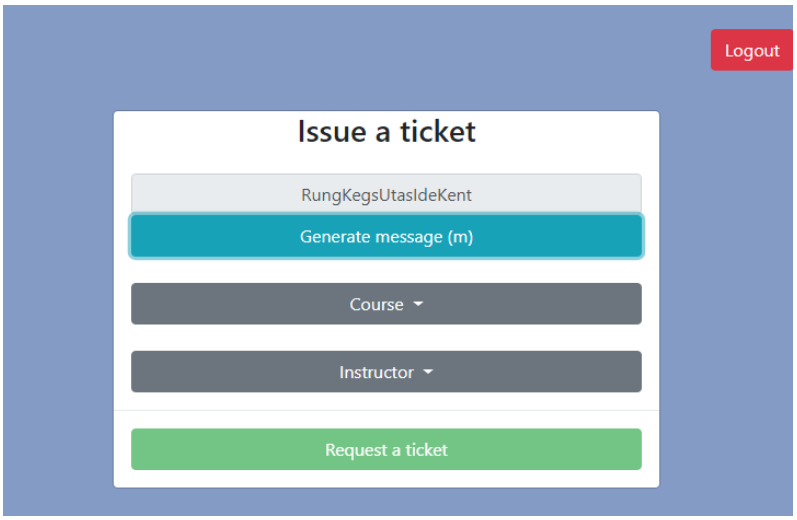
6.2 Έκδοση εισιτηρίων

Μετά την επιτυχημένη είσοδο στο σύστημα, εμφανίζεται στον χρήστη οθόνη που περιέχει όλους τους δυνατούς συνδυασμούς μαθημάτων – καθηγητών που μπορεί να εκδώσει εισιτήριο.



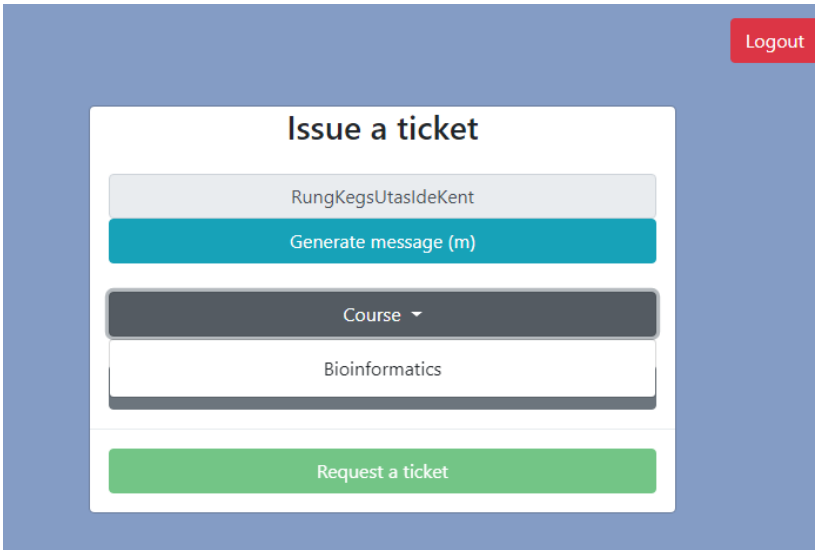
Εικόνα 25. Αρχική οθόνη έκδοσης εισιτηρίων

Με το πάτημα του κουμπιού που αναγράφει «Generate message (m)» ο φοιτητής μπορεί να κατασκευάσει τον συνδυασμό των λέξεων από το λεξικό, το οποίο αποτελεί το πρώτο βήμα για την έκδοση του εισιτηρίου του.



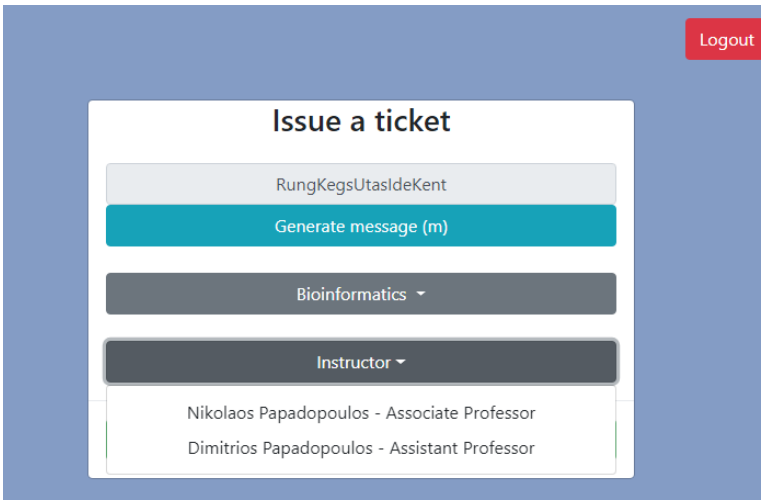
Εικόνα 26. Υπολογισμός αρχικού μηνύματος m

Στη συνέχεια, ο φοιτητής επιλέγει από το dropdown «Course» ένα από τα μαθήματα που παρακολούθησε στο εξάμηνο και δεν έχει εκδώσει ακόμα εισιτήριο.



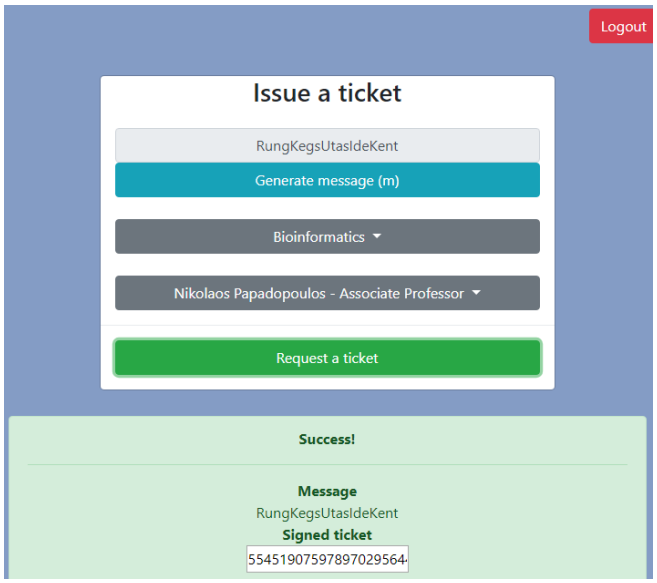
Εικόνα 27. Επιλογή μαθήματος για έκδοση εισιτηρίου

Αφού επιλεχθεί το μάθημα που ενδιαφέρεται να αξιολογήσει ο φοιτητής, μπορεί να επιλέξει έναν από τους καθηγητές που διδάσκουν το συγκεκριμένο μάθημα πατώντας το κουμπί που αναφέρεται στους εκπαιδευτικούς.



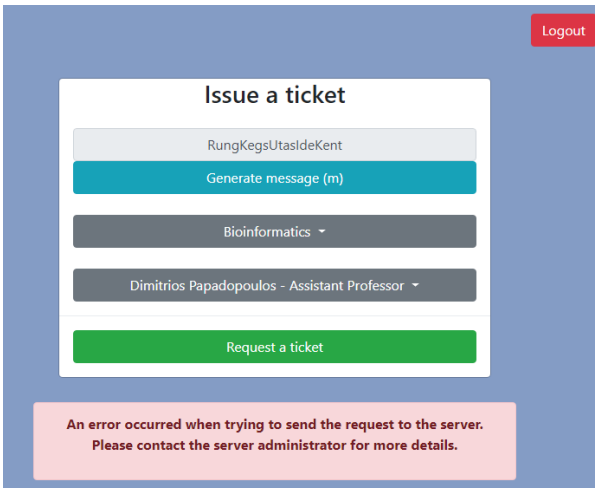
Εικόνα 28. Επιλογή καθηγητή προς αξιολόγηση

Πατώντας το κουμπί «Request a ticket» γίνονται όλες οι εναπομείναντες ενέργειες για την έκδοση της τυφλής υπογραφής, καθώς επίσης και η επαλήθευσή τους από την μεριά του χρήστη.



Εικόνα 29. Εμφάνιση υπογεγραμμένου εισιτηρίου

Σε περίπτωση σφάλματος (όπως π.χ. δοκιμή επανέκδοσης εισιτηρίου) εμφανίζεται στην οθόνη του χρήστη το κατάλληλο μήνυμα για να τον ειδοποιήσει. Ένα παράδειγμα φαίνεται στην παρακάτω οθόνη.



Εικόνα 30. Παράδειγμα μηνύματος σφάλματος κατά την έκδοση

6.3 Κατάθεση αξιολόγησης στο σύστημα

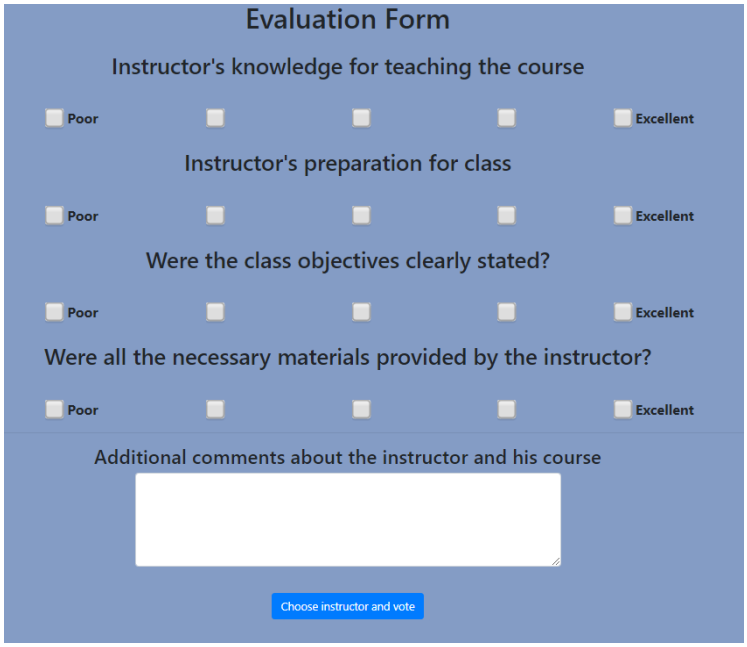
Μετά την επιτυχή έκδοση της ανώνυμης αυτής υπογραφής, ο χρήστης απομένει να οδηγηθεί μέσω του φυλλομετρητή του στην εφαρμογή που είναι υπεύθυνη για την αξιολόγηση των καθηγητών και των αντίστοιχων μαθημάτων που διδάσκουν.

Η εφαρμογή αυτή δεν προαπαιτεί αυθεντικοποίηση, επομένως με το που εισαχθεί στον φυλλομετρητή το κατάλληλο URL, θα οδηγηθεί απευθείας στην φόρμα για την αξιολόγηση.

Η φόρμα αυτή διατηρείται δυναμικά μέσω της βάσης δεδομένων όσον αφορά τις ερωτήσεις που περιέχει και φτιάχνεται δυναμικά η αντίστοιχη HTML μέσω JavaScript. Έτσι, η ανάπτυξη και η διατήρηση του ερωτηματολογίου μπορεί να γίνει με εύκολο και προσιτό τρόπο.

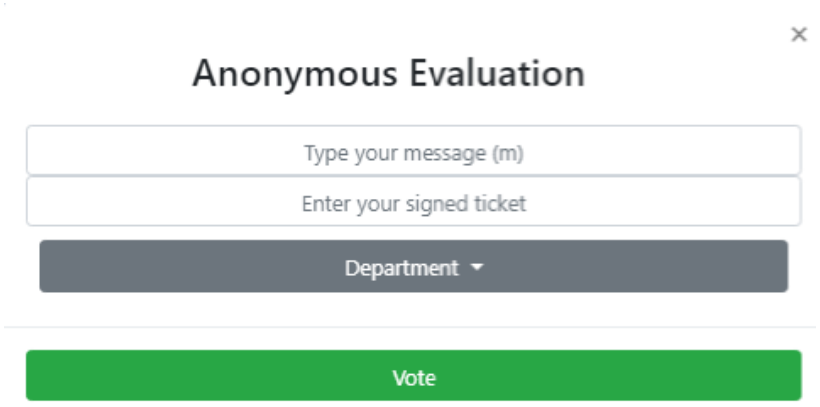
Ο φοιτητής αρκεί να συμπληρώσει την φόρμα όπως αυτός επιθυμεί και να συμπληρώσει το προαιρετικό σχόλιο για τον καθηγητή και το μάθημα.

Παρακάτω, φαίνεται ένα παράδειγμα μίας φόρμας αξιολόγησης.



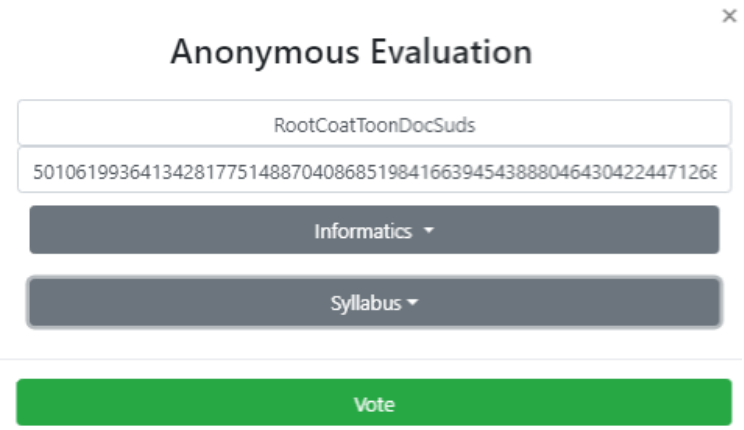
Εικόνα 31. Φόρμα αξιολόγησης

Αφού συμπληρωθεί η παραπάνω φόρμα, ο χρήστης πρέπει να εισάγει το αρχικό μήνυμα m και το εισιτήριο που του εκδόθηκε στην παρακάτω φόρμα.



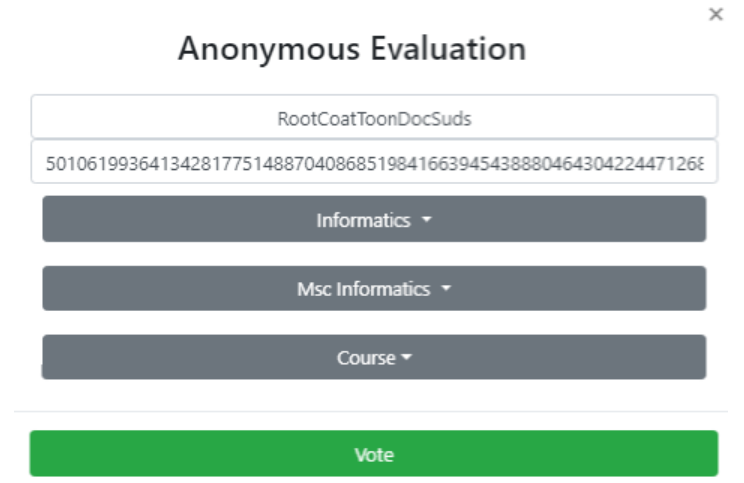
Εικόνα 32. Φόρμα εισαγωγής εισιτηρίου

Στη συνέχεια, μπορεί να επιλέξει το τμήμα στο οποίο ανήκει ο καθηγητής. Έτσι, θα ενεργοποιηθεί το dropdown με όλα τα προγράμματα σπουδών.



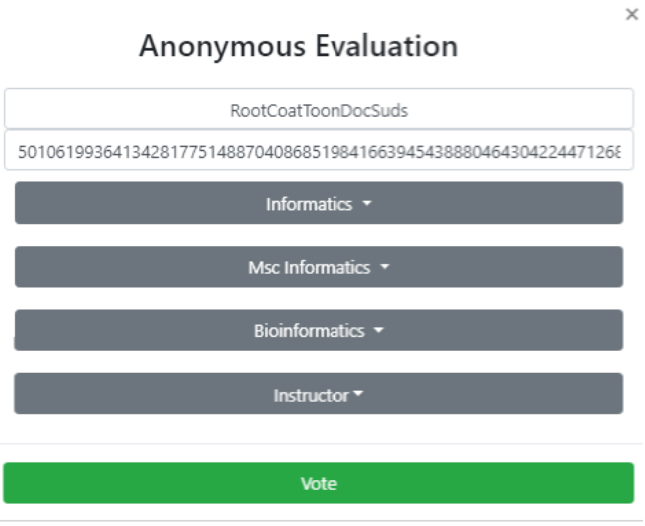
Εικόνα 33. Επιλογή τμήματος

Πλέον, μπορεί να επιλεχθεί το πρόγραμμα σπουδών που ανήκει το μάθημα προς αξιολόγηση και να ενεργοποιηθεί η λίστα με τα αντίστοιχα μαθήματα που μπορεί να επιλέξει ο φοιτητής.



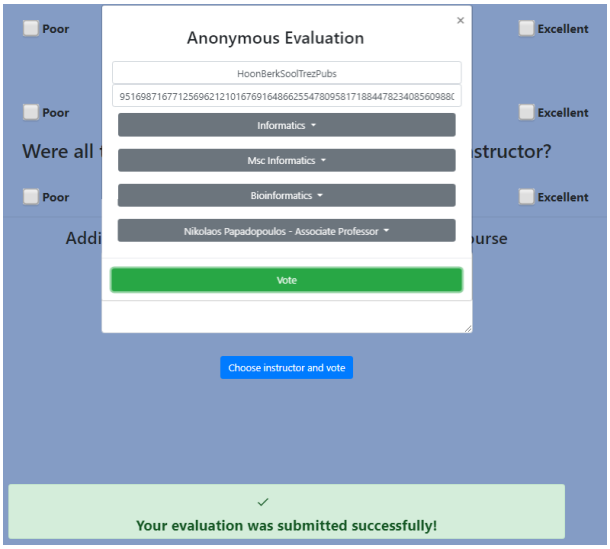
Εικόνα 34. Επιλογή προγράμματος σπουδών

Αντίστοιχα, μετά την επιλογή προγράμματος σπουδών, εμφανίζεται και η επιλογή του μαθήματος που θέλει ο φοιτητής να αξιολογήσει.



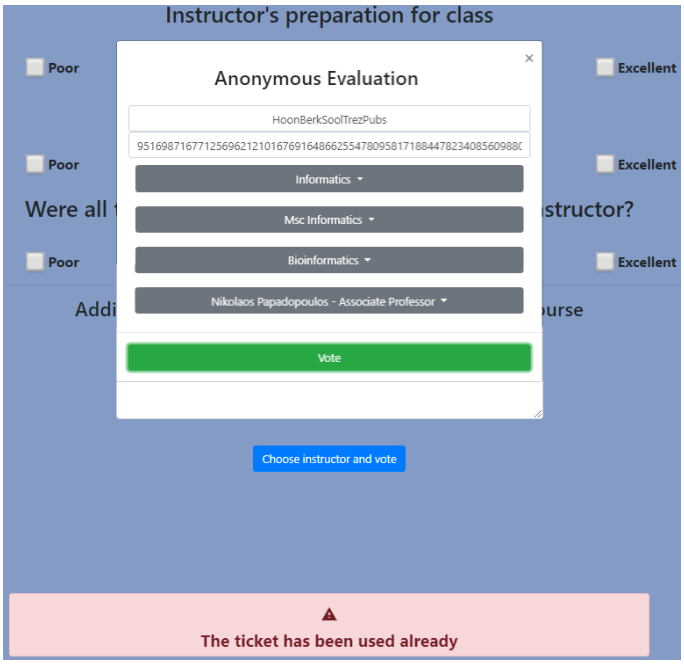
Εικόνα 35. Επιλογή μαθήματος προς αξιολόγηση

Τέλος, όπως φαίνεται στην παραπάνω εικόνα, απομένει η επιλογή του καθηγητή για τον οποίο αναφέρεται ο φοιτητής στην φόρμα. Αφού γίνει και αυτή η τελευταία επιλογή, αρκεί να πατήσει το κουμπί που αναγράφει «Vote» για να την καταθέσει.



Εικόνα 36. Επιτυχής καταχώρηση αξιολόγησης στο σύστημα

Φυσικά, σε περίπτωση αποτυχίας του συστήματος ο χρήστης ενημερώνεται καταλλήλως όπως φαίνεται παρακάτω.



Εικόνα 37. Ενημερωτικό μήνυμα στον χρήστη για σφάλμα κατά την καταχώρηση της αξιολόγησης

7. Συμπεράσματα

Στην παρούσα μεταπτυχιακή διατριβή αναλύθηκε, σχεδιάστηκε και υλοποιήθηκε επιτυχώς ένα σύστημα ανώνυμης αξιολόγησης που χρησιμοποιεί τυφλές υπογραφές για την εξασφάλιση της ανωνυμίας των φοιτητών ώστε να μπορούν να εκφραστούν επιλεκτικά και ελεύθερα.

Το σύστημα πληροί τις προϋποθέσεις της ανάλυσης των απαιτήσεων και παρέχει την ανωνυμία στους φοιτητές ώστε να καταφέρουν να εμπιστευτούν το σύστημα και να μην θεωρούν ότι θα επηρεαστούν από μία τυχόν αρνητική αξιολόγηση.

Ένα ακόμα πλεονέκτημα του συστήματος ανώνυμης αξιολόγησης, είναι ότι τα παραγόμενα δεδομένα από τις αξιολογήσεις των φοιτητών, δεν δέχονται καμία ποινή ως προς την δυνατότητα της εξαγωγής συμπερασμάτων για τους καθηγητές εάν θεωρήσουμε ότι οι αξιολογήσεις έχουν γίνει μόνο από φοιτητές που δικαιούνται εισιτήριο και τα εισιτήρια χρησιμοποιήθηκαν με σωστό τρόπο χωρίς κακόβουλες προθέσεις.

Το σημαντικότερο μειονέκτημα του συστήματος είναι ότι παρέχεται η δυνατότητα εκμετάλλευσης της ανωνυμίας των παρεχόμενων εισιτηρίων από κάποιον που επιθυμεί να αξιολογήσει πολλές φορές τον ίδιο καθηγητή. Επίσης, δεν υπάρχει τρόπος να διαπιστωθεί εάν το άτομο που έκανε αίτηση για το κάθε εισιτήριο δεν θα το παραχωρήσει σε άλλο φοιτητή. Έτσι, θα μπορούσε ένας ενδιαφερόμενος να προμηθεύει τα εισιτήριά του σε ενδιαφερόμενους που θα επιθυμούσαν να καταθέσουν παραπάνω αξιολογήσεις από ότι δικαιούνται.

Ένα ακόμα πρόβλημα που δημιουργείται από το σύστημα είναι ότι ενώ παρέχεται ένα εισιτήριο για κάθε συνδυασμό καθηγητή και μαθήματος που παρακολούθησε κατά τη διάρκεια του εξαμήνου ο φοιτητής, δεν μπορούμε να γνωρίζουμε εάν θα χρησιμοποιηθεί για την αξιολόγηση του συγκεκριμένου συνδυασμού. Ένας φοιτητής θα μπορούσε να χρησιμοποιήσει όλα του τα εισιτήρια για έναν καθηγητή και το μάθημά του.

Επομένως, το σύστημα ενώ παρέχει την επιθυμητή ανωνυμία στους φοιτητές, είναι επιρρεπές σε εκμετάλλευση στην περίπτωση που οι φοιτητές επιθυμούν να αξιολογήσουν είτε θετικά είτε αρνητικά περισσότερες φορές από ότι δικαιούνται.

Για την αντιμετώπιση των παραπάνω προβλημάτων του πρωτοκόλλου, προτείνεται η σύσταση ενός νέου πρωτοκόλλου, το οποίο θα περιλαμβάνει μία νέα οντότητα η οποία θα αντιστοιχεί στον καθηγητή.

Χρησιμοποιώντας ένα ζευγάρι κλειδιών το οποίο θα ανήκει στον εκάστοτε καθηγητή προς αξιολόγηση, ο φοιτητής θα μπορούσε να επαληθεύεται η παρουσία του μέσω μίας υπογραφής για την κάθε του παρακολούθηση.

Κατά την αξιολόγηση λοιπόν, ο φοιτητής θα καλείται να παραδώσει όχι μόνο το διαπιστευτήριο που έλαβε από τον εκδότη, αλλά και μία αλυσίδα από υπογραφές που θα περιγράφουν τον αριθμό των μαθημάτων που έχει παρακολούθησε, αλλά και ότι υπάρχει σύνδεση αυτών των υπογραφών με το διαπιστευτήριο που παραδίδει στο τέλος της διαδικασίας.

Επομένως, χρησιμοποιώντας αυτήν την αλυσίδα υπογραφών, τα προβλήματα όπως ότι ο φοιτητής μπορεί να χρησιμοποιήσει όλα του τα διαπιστευτήρια για να αξιολογήσει έναν και μόνο καθηγητή αντιμετωπίζονται επιτυχώς.

8. Βιβλιογραφία

1. Jonathan Herring (2014), *Medical Law and Ethics*, 5th edition, Oxford University Press, 2014.
2. Massey, A. & Antón, A., 2008. *A Requirements-based Comparison of Privacy Taxonomies* Washington, IEEE CPS.
3. Wallace, Kathleen A, 1999. *Anonymity. Ethics and Information Technology.* The Information Society. 15
4. David Chaum, 1983. *Blind signatures for untraceable payments.* Advances in Cryptology Proceedings of Crypto 82, 199203, 1983
5. LF Cranor, RK Cytron, 1997. *Sensus : a security-conscious electronic polling system for the Internet*. Proceedings of the Hawaii International Conference on System Sciences, IEEE
6. Camenisch J., Lysyanskaya A.,2004. *Signature Schemes and Anonymous Credentials from Bilinear Maps. In: Franklin M. (eds) Advances in Cryptology,* CRYPTO 2004. Lecture Notes in Computer Science, vol 3152. Springer, Berlin, Heidelberg
7. Camenisch J., 2014. *Concepts Around Privacy-Preserving Attribute-Based Credentials. In: Hansen M., Hoepman JH., Leenes R., Whitehouse D. (eds) Privacy and Identity Management for Emerging Services and Technologies. Privacy and Identity 2013*. IFIP Advances in Information and Communication Technology, vol 421. Springer, Berlin, Heidelberg
8. Camenisch, J., Van Herreweghen, E., 2002. *Design and implementation of the idemix anonymous credential system.In: Proceedings of the 9th ACM conference on Computer and communications security, , pp. 21–30*
9. IBM, 2015. *IBM IDENTITY MIXER - Authentication without identification*, Zurich
10. Dennis A., Wixom H, Teggarden D., 2015. *Systems Analysis and Design with UML Version 2.0: An Object-Oriented Approach (2nd edition)*. Wiley
11. Jan L. Harrington, 2016. *Relational Database Design and Implementation, 4th Edition*. Morgan Kaufmann
12. Anish Nath, 2018. *Cryptography for JavaScript Developer: Web Cryptography API, SJCL.*
13. Craig Walls, 2019. *Spring in Action, Fifth Edition.* Manning Publications.
14. Knutson M., Winch R., Mularien P., 2017. *Spring Security - Third Edition: Secure your web applications, RESTful services, and microservice architectures.* Packt Publishing.
15. Cross M., 2011, *Developer's Guide to Web Application Security: A Guide for Developers and Penetration Testers.* Syngress.
16. Daoqi Yang, 2010. *Java Persistence with JPA.* Outskirts Press.
17. Sene, I., Ciss, A.A., Niang, O., 2019. *I2pa: An efficient abc for iot. Cryptography3(2).*
18. Rannenberg, K., Camenisch, J., Sabouri, *2015. A.: Attribute-based credentials for trust. hpIdentity in the InformationSociety, Springer.*
19. Sabouri, A., Rannenberg, K., 2014*: Abc4trust: protecting privacy in identity management by bringing privacy-abcsinto real-life. In: IFIP International Summer School on Privacy and Identity Management, pp. 3–16. Springer*
20. Kluczniak, K., Hanzlik, L., Kubiak, P., Kutyłowski, M., 2015. *Anonymous evaluation system. In: InternationalConference on Network and System Security, pp. 283–299. Springer*

21. Baldimtsi, F., Lysyanskaya, A., 201. *Anonymous credentials light,* Proceedings of the 2013 ACM SIGSACconference on Computer & communications security, pp. 1087–1098.
22. *Benenson, Z., Girard, A., Krontiris, I., Liagkou, V., Rannenberg, K., Stamatiou, Y., 2014. User acceptance ofprivacy-abcs: an exploratory study. In: International Conference on Human Aspects of Information Security,Privacy, and Trust,* Springer, pp. 375–386