ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Σύστημα έκδοσης και διαχείρισης ακαδημαϊκών πιστοποιητικών και βεβαιώσεων με τεχνολογίες blockchain (Blockchain-based academic official documents management system)

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

των

Τζανέτος Δανιήλ
Κόσσυφας Θεόδωρος

Επιβλέπων : Σπύρος Κοκολάκης

Μέλη εξεταστικής επιτροπής: Ιωάννης Χαραλαμπίδης, Μαρία Καρύδα

Σάμος, Φεβρουάριος 2020

Ευχαριστίες

Αρχικά θα θέλαμε να ευχαριστήσουμε τον επιβλέποντα καθηγητή μας κ. Κοκολάκη Σπύρο για την καθοδήγηση και την υποστήριξη που μας έδειξε και την βοήθειά του για την ολοκλήρωση αυτής της διπλωματικής εργασίας.

Έπειτα θα θέλαμε να ευχαριστήσουμε τις οικογένειές μας καθώς και τους φίλους μας για την υπομονή και την στήριξή τους καθ' όλη την διάρκεια της διπλωματικής εργασίας.

Περίληψη

Το διαδίκτυο στις σύγχρονες κοινωνίες αποτελεί σημαντικό κομμάτι στην καθημερινότητα των ανθρώπων καθώς τους προσφέρει αμέτρητες δυνατότητες για να εκπληρώσουν τις ανάγκες τους. Μέσα σε όλες τις αναπτυσσόμενες τεχνολογίες που αυξάνονται με ταχύτατους ρυθμούς ένα στοιχείο του διαδικτύου που παραμένει αναλλοίωτο είναι η αρχιτεκτονική πελάτη-εξυπηρετητή η οποία είναι υπεύθυνη για την λειτουργία των περισσότερων εφαρμογών που διατίθενται στο διαδίκτυο. Στο σημείο αυτό η καινούργια τεχνολογία του blockchain και του Internet of Things (IoT) ενδέχεται να επιφέρουν τεράστιες αλλαγές στο χώρο με έναν πιο αποκεντρωμένο χαρακτήρα.

Το blockchain εκτός από την αρχική του ιδιότητα ως μέσο λειτουργίας του κρυπτονομίσματος bitcoin, έπαιξε σημαντικό ρόλο και σε διαφορετικούς κλάδους του διαδικτύου. Το blockchain είναι ένα ψηφιακό κατανεμημένο δημόσιο καθολικό (ledger) στο οποίο καταγράφονται συναλλαγές και συμφωνίες με τρόπο αδιάβλητο και υποστηρίζεται από ένα δίκτυο ομότιμων κόμβων. Καθοριστικής σημασίας είναι η συνεισφορά του στην δημιουργία αποκεντρωμένων εφαρμογών (DApps – Decentralized Applications), δηλαδή σε εφαρμογές που δεν βασίζονται στην γνωστή αρχιτεκτονική πελάτη-εξυπηρετητή αλλά σε ένα δίκτυο που ονομάζεται δίκτυο ομότιμων κόμβων. Στην πράξη το Ethereum blockchain αποτελεί παράδειγμα της εφαρμογής του blockchain στην δημιουργία αποκεντρωμένων εφαρμογών μέσω έξυπνων συμβολαίων.

Η ανάπτυξη του συστήματος της διπλωματικής αυτής εργασίας θέτει ως κύριες λειτουργίες της εφαρμογής την έκδοση (issue) και υπογραφή (sign) ψηφιακών βεβαιώσεων και πιστοποιητικών που εκδίδουν οι γραμματείες των πανεπιστημιακών τμημάτων, η αυθεντικότητα (verify) των οποίων θα πιστοποιείται με τεχνολογίες Blockchain αντί για τα κλασικά ψηφιακά πιστοποιητικά. Μελετώντας εφαρμογές βασισμένες στην τεχνολογία του blockchain καθώς και στον ακαδημαϊκό κλάδο αντλήσαμε πληροφορίες ώστε να καθορίσουμε τις ανάγκες του συστήματος και να προχωρήσουμε στην ανάλυση και υλοποίηση αυτού.

Το σύστημα που αναπτύξαμε με όνομα «Academic Certificates» βασίζεται στο μοντέλο πελάτη – εξυπηρετητή. Ο πελάτης (client) και ο εξυπηρετητής (server) συνδέονται με το σύστημα μέσω μίας εφαρμογής που δημιουργήσαμε για αυτόν το σκοπό. Η επικοινωνία των δύο, συσχετίζεται με μια βάση δεδομένων, που παρέχει σημαντικές πληροφορίες, απαραίτητες για τη λειτουργικότητα του συστήματος.

Στην διπλωματική εργασία μέσω μιας αποκεντρωμένης εφαρμογής-πληροφοριακό σύστημα αναλύονται και επεξηγούνται σημαντικές έννοιες του αποκεντρωμένου ιστού καθώς και κάποιες ιδιότητες και χρήσεις του.

Abstract

The internet in modern societies is an important part of everyday life as it offers innumerable possibilities to fulfill people's needs. Throughout the emerging technologies that are rapidly growing, an internet component that remains unchanged is the client-server architecture that is responsible for running most of the applications available on the Internet. At this point the new technology of blockchain and Internet of Things (IoT), can bring enormous changes to the field with a more decentralized character.

Blockchain, in addition to its original property as a means of operating bitcoin currency, has also played an important role in different branches of the internet. Blockchain is a digitally distributed public ledger in which transactions and agreements are recorded in a non-invasive manner and supported by a peer-to-peer network. It is critical that it contributes to the creation of decentralized applications (DApps - Decentralized Applications), meaning applications that do not rely on the known client-server architecture but on a network called peer-to-peer network. In fact, the Ethereum blockchain is an example of the implementation of blockchain in creating decentralized applications through smart contracts.

The development of the system of this thesis puts the implementation of the issue and signing of digital certificates which are issued by the secretariats of the university departments, the authentication of which will be certified by Blockchain technologies instead of classic digital certificates. By studying applications based on blockchain technology as well as in the academic industry, we have acquired information to determine the needs of the system and to proceed with its analysis and implementation.

The system we developed named «Academic Certificates» is based on the client - server model. The client and the server are connected to the system through an application that we created for that purpose. The communication of the two is associated with a database that provides important information necessary for system functionality.

The thesis presents both the theoretical issues of the decentralized web and its constituent technologies as well as practical issues through the presentation of the process of developing a decentralized implementation.

Περιεχόμενα

Λίστα εικόνων

Λίστα πινάκων

Λίστα Συντομογραφιών

IoT. . . . . Internet of Things
DApps. . . . . Decentralized Applications
P2P. . . . . Peer To Peer
DHT. . . . . Distributed Hash Table
ISP. . . . . Internet Service Provider
CAD. . . . . Cash Against Documents
IFTTT. . . . . If This Then That
GDPR. . . . . General Data Protection Regulation
DAO. . . . . Decentralized Autonomous Organization
H/Y. . . . . Ηλεκτρονικός Υπολογιστής
KB. . . . . Kilobyte
URI. . . . . Uniform Resource Identifier
HTTP. . . . . Hypertext Transfer Protocol
CAPTCHA. . . . Completely Automated Public Turing test to tell Computers and Humans Apart
CSRF. . . . . Cross-Site Request Forgery

1. Εισαγωγή

1.1 Σκοπός διπλωματικής

Ο βασικός σκοπός αυτής της διπλωματικής εργασίας είναι η ανάπτυξη ενός συστήματος που θα υποστηρίζει την έκδοση και διαχείριση ψηφιακών βεβαιώσεων και πιστοποιητικών που εκδίδουν οι γραμματείες των πανεπιστημιακών τμημάτων, η αυθεντικότητα των οποίων θα πιστοποιείται με τεχνολογίες Blockchain αντί για τα κλασικά ψηφιακά πιστοποιητικά. Η πιο σημαντική περιοχή στην οποία βοηθάει το Blockchain είναι στην διασφάλιση εγκυρότητας μιας συναλλαγής καταγράφοντας την όχι μόνο σε ένα βασικό μητρώο αλλά σε ένα συνδεδεμένο και διανεμημένο σύστημα μητρώων, τα οποία συνδέονται μέσω ενός ασφαλούς μηχανισμού ελέγχου εγκυρότητας.

1.2 Αρχιτεκτονικοί στόχοι

Υπάρχουν ορισμένες βασικές απαιτήσεις και περιορισμοί του συστήματος που έχουν σημαντικό αντίκτυπο στην αρχιτεκτονική. Αυτές είναι:

- Όλες οι λειτουργίες των φοιτητών, καθηγητών και εγγεγραμμένων πρέπει να είναι διαθέσιμες τόσο από τοπικούς υπολογιστές πανεπιστημίων όσο και από απομακρυσμένους υπολογιστές με συνδέσεις μέσω διαδικτύου.
- Το σύστημα θα πρέπει να εξασφαλίζει την πλήρη προστασία των δεδομένων από μη εξουσιοδοτημένη πρόσβαση. Όλες οι απομακρυσμένες προσβάσεις θα υπόκεινται σε έλεγχο ταυτότητας χρήστη και κωδικού πρόσβασης.

1.3 Υποθέσεις και εξαρτήσεις

Μία λίστα με τις υποθέσεις και τις εξαρτήσεις που οδήγησαν στις αρχιτεκτονικές αποφάσεις ακολουθούν παρακάτω:

- Λόγω της διάστασης του έργου και της διανομής των εμπλεκόμενων ομάδων ανάπτυξης, οι τεχνολογίες που χρησιμοποιούνται για κάθε στοιχείο του συστήματος είναι σχετικά ευέλικτες, αλλά η ενσωμάτωση πρέπει να είναι διαφανής για τους χρήστες για όλα τα μέρη.
- Οι χρησιμοποιούμενες τεχνολογίες θα συζητηθούν, θα επανεξεταστούν και θα εγκριθούν στο πλαίσιο της ομάδας και θα είναι επιθυμητή η διατήρηση εύλογου αριθμού διαφορετικών τεχνολογιών
- Θα διεξαχθούν περιοδικές συναντήσεις της ομάδας που συμμετέχει στην ενσωμάτωση των διαφόρων στοιχείων.
- Τις διεργασίες που θα υποστηρίξει το πληροφοριακό σύστημα.
- Τα δεδομένα που παράγονται και τηρούνται στο σύστημα.

- Το νέο πληροφοριακό σύστημα θα πρέπει να διαθέτει μηχανισμούς διαλειτουργικότητας με παραδοσιακές εφαρμογές ή να διαθέτει νέες εφαρμογές που θα αποτελούν την εξέλιξη των παραδοσιακών.
- Απαραίτητη προϋπόθεση είναι να εξασφαλιστεί η διαλειτουργικότητα του συστήματος, δηλαδή η επικοινωνία με τις υπόλοιπες πλατφόρμες (εικονικό φοιτητολόγιο).

1.4 Απαιτήσεις αρχιτεκτονικής

Οι σημαντικότερες απαιτήσεις είναι εκείνες που παίζουν καθοριστικό ρόλο στον προσδιορισμό της αρχιτεκτονικής του συστήματος.

- Το σύστημα οφείλει να διαφυλάσσει τις βεβαιώσεις και τα πιστοποιητικά που διαχειρίζεται σε ένα ξεχωριστό χώρο (back-up) σε περίπτωση αποτυχίας του συστήματος.
- Η επικοινωνία πρέπει να γίνεται ασύγχρονα και να μην χάνεται κανένα αίτημα ή δεδομένα όταν αποτυγχάνουν ένα ή περισσότερα από αυτά τα στοιχεία.
- Το user interface του συστήματος θα βασίζεται σε γλώσσα προγραμματισμού Java.
- Το σύστημα θα πρέπει να χαρακτηρίζεται από διαλειτουργικότητα. Ο όρος διαλειτουργικότητα αναφέρεται στη δυνατότητα επικοινωνίας συστημάτων και εφαρμογών που είναι ασύμβατα, δηλαδή χρησιμοποιούν διαφορετικά πρότυπα μοντελοποίησης πληροφοριών και διαφορετικά πρότυπα επικοινωνίας.

1.5 Επιλογές και περιορισμοί

Υπάρχουν διάφοροι παράγοντες που θέτουν περιορισμούς στην αρχιτεκτονική που αναπτύσσεται. Αυτοί οι αρχιτεκτονικοί περιορισμοί, σε συνδυασμό με τις απαιτήσεις, θα βοηθήσουν στον ορισμό της αρχιτεκτονικής του συστήματος. Η καταγραφή αυτών των περιορισμών θα διευκολύνει την ολοκλήρωση και μπορεί να μειώσει τον κίνδυνο και το κόστος.

- Λόγω της κατανεμημένης φύσης του συστήματος, οι τεχνολογίες ανάπτυξης και βάσεων δεδομένων που χρησιμοποιούνται πρέπει να βασίζονται σε ανοιχτού κώδικα και σε τεχνολογίες γνωρίζει η ομάδα.
- Η χρήση λογισμικού ή συγκεκριμένης τεχνολογίας θα κοινοποιείται στον υπεύθυνο, ο οποίος αποφασίζει αν θα πρέπει να επιτρέπεται.
- Προκειμένου να αυξηθεί η ποιότητα του παραδοθέντος προϊόντος, θα εφαρμοστούν οι ακόλουθες αξιολογήσεις:
    - Αρχιτεκτονική ανασκόπηση με επίκεντρο την προστασία της ιδιωτικής ζωής και της ασφάλειας από το σχεδιασμό.
    - Αναθεώρηση κώδικα.

2. Ανάλυση τεχνολογίας Blockchain

2.1 Περιγραφή του Blockchain

Το blockchain είναι στην ουσία ένα μητρώο (ledger) στο οποίο αποθηκεύονται κι επαληθεύονται πληροφορίες και δεδομένα, τα οποία συνήθως εντάσσονται σε μπλοκ, με τη χρήση κρυπτογραφικών μεθόδων και με τέτοιο τρόπο ώστε να δημιουργείται μία συνεχής αλυσίδα δεδομένων, ενώ κάθε τροποποίηση μίας πληροφορίας που έχει καταγραφεί στο μητρώο να επηρεάζει αναγκαστικά όλες τις μεταγενέστερες καταχωρήσεις.

Η πλατφόρμα του blockchain ανάλογα με το δίκτυο που «τρέχει» (δημόσιο ή εσωτερικό) χωρίζεται σε δύο κατηγορίες, δημόσια και ιδιωτική [7].

Το γεγονός που κάνει την τεχνολογία αυτή να διαφοροποιείται από τα υπάρχοντα μητρώα και από τις βάσεις δεδομένων (databases) είναι ότι πλέον δεν υπάρχει μια κεντρική αρχή αλλά οι κόμβοι (nodes) που αρμοδιότητα τους είναι η τήρηση του μητρώου δεδομένων. Οι κόμβοι δηλαδή είναι χρήστες με λογισμικό που απαιτείται να έχουν εγκατεστημένο έτσι ώστε να υπάρχει δυνατότητα να γίνεται ταυτόχρονη ενημέρωση από όλους με στόχο σε οποιαδήποτε χρονική στιγμή να έχουν όλοι στο μητρώο την ίδια κατάσταση. Για παράδειγμα για την μεταφορά χρημάτων μεταξύ δύο λογαριασμών η επαλήθευση-επιβεβαίωση γίνεται με την επιτυχή και ταυτόχρονη ενημέρωση του μητρώου από τους χρήστες-κόμβους.

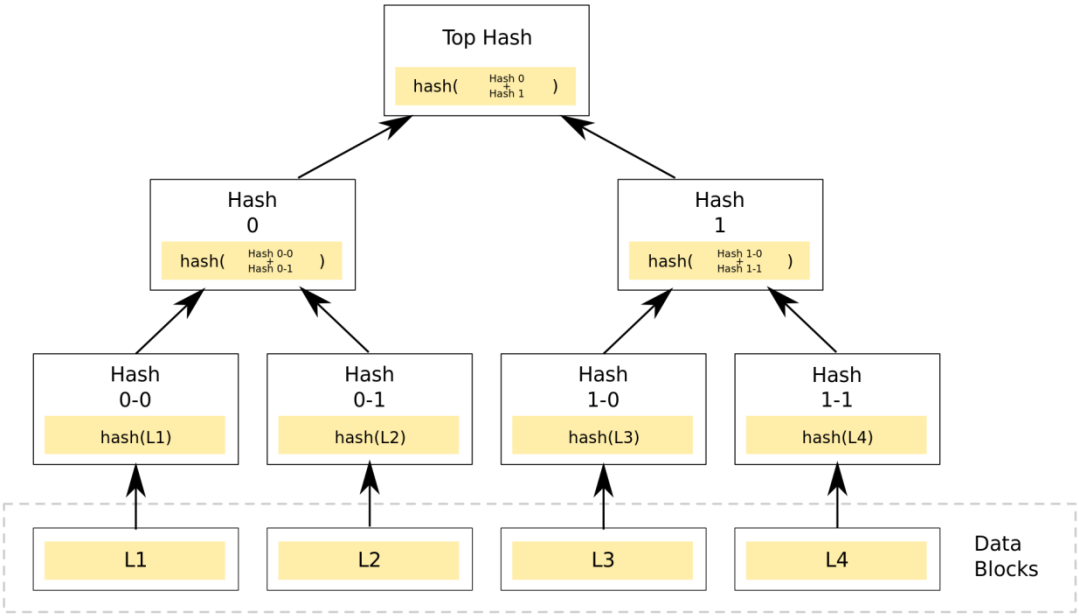
Μεταξύ των κόμβων-χρηστών για να διασφαλιστεί η εμπιστοσύνη και κατ επέκταση η ασφάλεια των στοιχείων που καταχωρούν στο μητρώο πραγματοποιείται μια συμφωνία (consensus). Στην πραγματική χρήση του μητρώου η πληθώρα χρηστών-κόμβων συνεπάγεται και μεγαλύτερο βαθμό διασφάλισης των δεδομένων και εμπιστοσύνης. Όσον αφορά τον βαθμό λειτουργικότητας της βάσης δεδομένων blockchain οι κόμβοι (nodes) οφείλουν να τηρούν και να πράττουν βάσει κάποιων ηθικών και δεοντολογικών αρχών.

Κατά συνέπεια το μητρώο δεδομένων και πληροφοριών λόγω του ότι όλοι οι κόμβοι (nodes) ενημερώνουν ταυτόχρονα και τηρούν αυτό του προσδίδεται ακόμα ένα χαρακτηριστικό-ιδιότητα εκτός από αποκεντρωμένο-decentralized αυτό του διανεμημένο-distributed.

Όσον αφορά την πλατφόρμα του blockchain και συγκεκριμένα το λογισμικό σε αυτήν είναι ανοιχτού κώδικα-open source. Αυτό σημαίνει ότι το λογισμικό θέτει ορισμένες προϋποθέσεις και καθορίζει την διαδικασία που απαιτείται για να δημιουργηθούν τα κρυπτονομίσματα καθώς και τον τρόπο με τον οποίο οι πληροφορίες, τα δεδομένα θα εγγραφούν και θα επαληθευθούν στο μητρώο. Κάθε πλατφόρμα blockchain περιγράφει τις προϋποθέσεις και τους όρους της και τους διανέμει στους χρήστες.

Απαραίτητη προϋπόθεση για την λειτουργία της κάθε μίας από τις πλατφόρμες του blockchain είναι οι αλγόριθμοι βάσει των οποίων πραγματοποιείται η καταχώρηση και η επαλήθευση των πληροφοριών και των δεδομένων καθώς και των κινήσεων που λαμβάνουν χώρα. Για την κρυπτογράφηση μιας συναλλαγής χρησιμοποιείται μία συνάρτηση κατακερματισμού (hash function) η οποία δέχεται ως είσοδο ένα δεδομένο τυχαίου μεγέθους και επιστρέφει έναν ακέραιο σταθερού μεγέθους αναπαράστασης.

Τα στοιχεία μίας συναλλαγής αποθηκεύονται στα data blocks. Τα merkle trees δημιουργούνται έχοντας στα φύλλα τους το hash των data blocks και στους υπόλοιπους κόμβους το hash των παιδιών τους [5].



Εικόνα 2.1 Hash tree/Merkle tree.

Η καταχώρηση των προαναφερθέντων κρυπτογραφικών στοιχείων (hash) στο blockchain, ονομάζεται proof of work και εκτελείται από τους miners οι οποίοι έχουν στην διάθεση τους τεχνολογική υποστήριξη με τεράστια ισχύ παράλληλα με το κατάλληλο λογισμικό.

Πρόκειται για μία διαδικασία κατά την οποία τα μπλοκ των συναλλαγών έχουν δεδομένα των προηγούμενων μπλοκ έως ότου φτάσει στο αρχικό μπλοκ-αρχική συναλλαγή. Αυτό σημαίνει ότι στην περίπτωση όπου πραγματοποιηθεί κάποια αλλαγή σε οποιοδήποτε μπλοκ του blockchain θα υπάρξουν αλλαγές σε όλα τα άλλα μπλοκ. Τα δεδομένα του κάθε μπλοκ θα τροποποιηθούν και ως αποτέλεσμα το hash που θα παραχθεί δεν θα είναι το ίδιο όπως και όλη η αναδρομική ακολουθία με το παραγόμενο hash.

2.2 Ethereum Blockchain

Το blockchain είναι κυρίαρχο συστατικό και υψίστης σημασίας τεχνολογία για την νέα γενιά διαδικτύου καθώς θέτει νέες μεθόδους και λύσεις στο αιώνιο ζήτημα της εμπιστευτικότητας [3].

Βασίζεται σε ένα δίκτυο ομότιμων κόμβων (P2P) στο οποίο δεν χρειάζεται ο μεσάζων-τρίτος στην συναλλαγή αλλά μόνο η πληροφορία από τους υπόλοιπους στο

δίκτυο [6]. Αποθηκεύει ένα αντίγραφο με όλες τις συναλλαγές που έχουν ολοκληρωθεί και τα δεδομένα τους.

Στην συνέχεια θα εξετάσουμε το Ethereum και θα αναφέρουμε τις διαφορές του έναντι του Bitcoin, του κρυπτονομίσματος που «γέννησε» την τεχνολογία του blockchain.

Η κύρια διαφορά μεταξύ του Ethereum με το Bitcoin είναι το γεγονός ότι η πλατφόρμα Ethereum παρέχει περισσότερες δυνατότητες (έξυπνα συμβόλαια) κυρίως για την δημιουργία εφαρμογών αποκεντρωμένου ιστού με προτεραιότητα την ασφάλεια και την διασφάλιση της εμπιστευτικότητας των δεδομένων έναντι με το Bitcoin το οποίο συνίσταται για τις οικονομικές συναλλαγές του κρυπτονομίσματος αυτού.

Το Ethereum blockchain βασίζεται σε μία αρχιτεκτονική κατά την οποία κάθε ένας κόμβος εκτελεί, τηρεί και αποθηκεύει ίδιες συναλλαγές. Στην συνέχεια τα δεδομένα-συναλλαγές συλλέγονται και καταχωρούνται στο blockchain [20].

2.3 Δίκτυα ομότιμων κόμβων

Στην ενότητα αυτή, θα γίνει μία σύντομη παρουσίαση της αρχιτεκτονικής ομότιμων κόμβων, αφού αποτελεί την αρχιτεκτονική πάνω στην οποία βασίζεται η τεχνολογία του blockchain.

Την χρονική περίοδο που διανύουμε στο διαδίκτυο δύο είναι οι κύριες αρχιτεκτονικές που χρησιμοποιούνται για την ανάπτυξη και λειτουργία εφαρμογών. Αυτή του πελάτη εξυπηρετητή (client-server) με πληθώρα εφαρμογών σε αυτό το μοντέλο και αυτή των ομότιμων κόμβων (P2P) [2]. Η διαφορά των δύο είναι κυρίως η ύπαρξη του server (εξυπηρετητή) όπου σε αυτή του client-server είναι απαραίτητη ο οποίος δέχεται από τους χρήστες αιτήματα για να ολοκληρώσουν τις ενέργειες τους έναντι της αρχιτεκτονικής των ομότιμων κόμβων στην οποία δεν χρειάζεται.

Στο δίκτυο ομότιμων κόμβων (P2P) οι υπολογιστές (peers) υπό τις εντολές των χρηστών αλληλεπιδρούν, επικοινωνούν σε ζεύγη χωρίς την μεσολάβηση κάποιου τρίτου για παροχή υπηρεσίας. Οι κόμβοι στο δίκτυο P2P είναι ταυτόχρονα και clients αλλά και server λόγω του ότι έχουν την δυνατότητα να διατηρούν την σωστή λειτουργία του δικτύου παρέχοντας πόρους αλλά και καταναλώνοντας, εξαλείφοντας με αυτόν τον τρόπο την αρχή που θα παρείχε υπηρεσίες και θα οργάνωνε το δίκτυο. Κατά την υλοποίηση μίας εφαρμογής είναι σύνηθες να χρησιμοποιούνται και οι δύο αυτές αρχιτεκτονικές σε μία με όνομα υβριδική αρχιτεκτονική. Τα δίκτυα ομότιμων χωρίζονται σε δύο μεγάλες κατηγορίες, στα αδόμητα και στα δομημένα.

Τα αδόμητα P2P δίκτυα δημιουργούνται μέσω συνδέσεων των κόμβων χωρίς να είναι προκαθορισμένη η σύνδεση αυτών, χωρίς κάποια οργάνωση και αρχή και τέλος χωρίς τα δεδομένα να έχουν κάποια σχέση με τους κόμβους από τους οποίους τα συλλέγουμε. Τα πλεονεκτήματα αυτών είναι η ευκολία για την δημιουργία τους μέσω της παραπάνω διαδικασίας, οι αλλαγές τοπικά για καλύτερα αποτελέσματα και η δυνατότητα που έχουν για εξέλιξη και δημιουργία σε διαφορετικές «κινήσεις» των κόμβων. Ένα σημαντικό μειονέκτημα είναι η σπατάλη πόρων στο δίκτυο όταν γίνεται αίτημα για αναζήτηση πληροφοριών και η επιθυμητή απάντηση στο αίτημα καθώς απαιτείται ο αρμόδιος κόμβος για συλλογή δεδομένων πράγμα που δεν είναι σίγουρο ότι θα πραγματοποιηθεί με επιτυχία.

Τα δομημένα P2P δίκτυα διαμορφώνονται με τέτοιο τρόπο ώστε στην διαδικασία αιτήματος αναζήτησης δεδομένων ανεξάρτητα από ποιον κόμβο να εγγυάται το καλύτερο και επιθυμητό αποτέλεσμα.

Μία μορφή-τύπος δομημένων δικτύων (P2P) επιλύει τον DHT κατανεμημένο πίνακα κατακερματισμού βάσει του οποίου μεταφέρονται στον κόμβο που πρέπει τα δεδομένα. Ένας κατανεμημένος πίνακας κατακερματισμού είναι μια κατηγορία αποκεντρωμένου κατανεμημένου συστήματος που παρέχει μια υπηρεσία αναζήτησης παρόμοια με ένα πίνακα κατακερματισμού. Ο κατανεμημένος πίνακας κατακερματισμού DHT χρησιμοποιεί έναν τροποποιημένο αλγόριθμο της συνάρτησης κατακερματισμού για την παραπάνω διαδικασία όπου αποστέλλονται τα δεδομένα στον κάθε κόμβο από τον οποίο επίσης με τον DHT γίνεται και η αναζήτηση αυτών των δεδομένων γεγονός που κάνει τα δομημένα δίκτυα καλύτερα έναντι των αδόμητων [20].

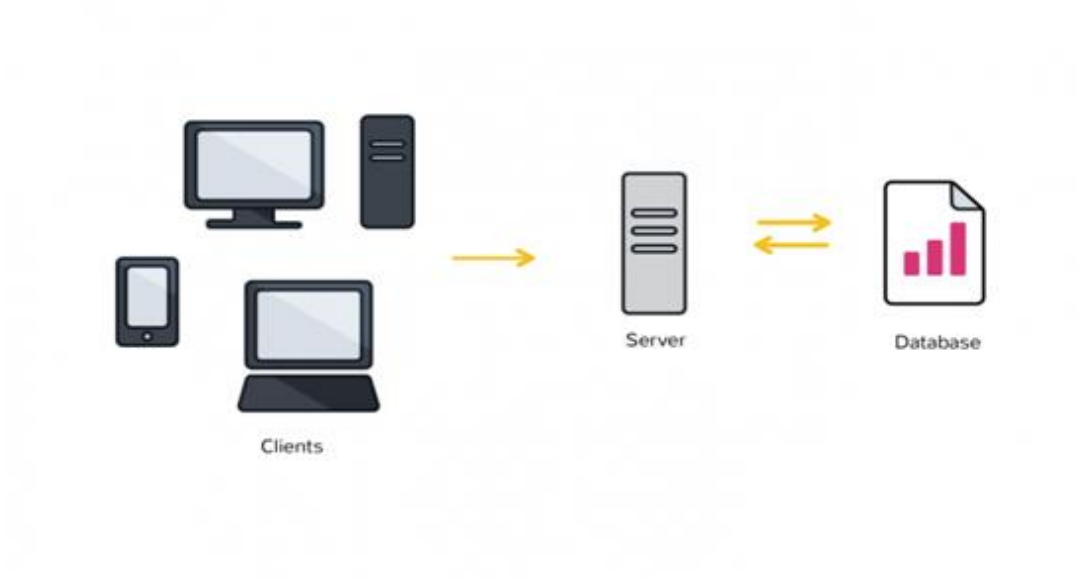
Τα βασικότερα πλεονεκτήματα της αρχιτεκτονικής ομότιμων:

- Αυτο-κλιμακωσιμότητα (self-scalability) του δικτύου. Ταυτόχρονα με την αύξηση των κόμβων στο δίκτυο αυξάνεται και η ανάγκη σε πόρους τους οποίους πόρους όμως καλύπτουν οι ίδιο οι κόμβοι πράγμα το οποίο συμβαίνει λόγω της αρχιτεκτονικής του δικτύου αυτού.
- Μείωση κόστους για τον λόγο ότι δεν χρειάζεται να εγκατασταθεί κάποιος ιδιαίτερος τεχνολογικός εξοπλισμός ούτε υλοποίηση και διαμόρφωση κάποιου εξυπηρετητή.
- Μη ύπαρξη μοναδικού σημείου αστοχίας του δικτύου. Στην αρχιτεκτονική πελάτη εξυπηρετητή ένα λάθος που θα υπάρξει στον εξυπηρετητή έχει ως αποτέλεσμα την μη λειτουργία της εφαρμογής σε αντίθεση με μία βλάβη σε έναν κόμβο.

Μερικές από τις βασικές προκλήσεις που αντιμετωπίζουν οι εφαρμογές αρχιτεκτονικής ομότιμων:

- Ασφάλεια. Λόγω της κατανεμημένης και ανοικτής φύσης τους, οι εφαρμογές P2P μπορούν να δημιουργήσουν προβλήματα ασφαλείας.
- Η ανάκτηση δεδομένων ή η δημιουργία αντιγράφων ασφαλείας είναι δύσκολο να πραγματοποιηθεί για τον λόγο ότι κάθε υπολογιστής πρέπει να διαθέτει δικό του σύστημα δημιουργίας αντιγράφων ασφαλείας.
- Λόγω του ότι το σύστημα είναι αποκεντρωμένο δεν διαθέτει κάποιον κεντρικό διαχειριστή για αυτό και καθίσταται δύσκολη η ρύθμιση της κίνησης του δικτύου [19].

2.4 Διαφορές Blockchain και βάσης δεδομένων



Εικόνα 2.2 Αρχιτεκτονική πελάτη-εξυπηρετητή.

Η διαφορά ανάμεσα σε μια παραδοσιακή βάση δεδομένων και ένα blockchain εντοπίζεται στην αρχιτεκτονική και στον τρόπο που είναι δομημένες οι τεχνολογίες. Η αρχιτεκτονική πελάτη-εξυπηρετητή χρησιμοποιείται κυρίως από βάσεις δεδομένων οι οποίες εκτελούνται στο Word Wide Web.

Ο χρήστης έχει την δυνατότητα με τα απαραίτητα δικαιώματα να τροποποιήσει τα συσχετιζόμενα με τον λογαριασμό του δεδομένα-πληροφορίες που βρίσκονται σε έναν διακομιστή ο οποίος εξυπηρετεί όλους τους υπόλοιπους. Οι χρήστες που θα ζητήσουν πρόσβαση στη βάση μετά την κάθε αλλαγή που μπορεί να πραγματοποιηθεί λαμβάνουν την ανανεωμένη βάση με τις νέες πληροφορίες είτε νέες εγγραφές, καταχωρήσεις είτε διαγραφές.

Οι διαχειριστές ανά πάσα χρονική στιγμή είναι οι μόνοι οι οποίοι μπορούν να επιτρέπουν την είσοδο στη βάση και παραχωρήσεις δικαιωμάτων ώστε να γίνονται επεξεργασίες από τους χρήστες καθώς και διατήρηση αυτών για ασφάλεια της βάσης η οποία συνεπάγεται προστασία και διασφάλιση εγκυρότητας για το σύστημα ολόκληρο.

Στην περίπτωση της βάσης δεδομένων blockchain ο κάθε χρήστης-κόμβος είναι υπεύθυνος να ενημερώνει τη βάση το οποίο έχεις ως στόχο να έχουν όλοι στο δίκτυο P2P την ίδια κατάσταση και την μέγιστη ασφάλεια.

2.4.1 Αποκεντρωμένος έλεγχος

Στο blockchain οι χρήστες χωρίς να γνωρίζουν τα άτομα που μοιράζονται πληροφορίες πραγματοποιούν συναλλαγές χωρίς κάποια κεντρική αρχή-διαχειριστή ο οποίος θα διαχειρίζεται τις πληροφορίες. Αντιθέτως όλοι οι κόμβοι-χρήστες λειτουργούν με βάση την εμπιστοσύνη μεταξύ τους με στόχο να έχουν την ίδια κατάσταση με όλους στο δίκτυο των ομότιμων κόμβων.

Το πλεονέκτημα αυτής της διαδικασίας είναι ότι δεν εξαρτώνται οι χρήστες από κάποιους που είναι υπεύθυνοι για την κεντρική βάση και με αυτόν τον τρόπο μειώνουν τις πιθανότητες να γίνει εσκεμμένα κάποια καταστροφή ή υποκλοπή στα δεδομένα. Οι χρήστες είναι υπεύθυνοι να τηρούν οι ίδιοι τις αλλαγές ενεργώντας με τρόπο τέτοιο που να διασφαλίζεται η ακεραιότητα της αλυσίδας.

Οι περισσότερες κεντρικές βάσεις δεδομένων διατηρούν πληροφορίες που είναι ενημερωμένες σε μια συγκεκριμένη στιγμή.

Οι βάσεις δεδομένων Blockchain είναι σε θέση να διατηρούν πληροφορίες που είναι σχετικές τώρα, αλλά και όλες τις πληροφορίες που έχουν προηγηθεί. Δημιουργούν ένα ιστορικό από συναλλαγές με πληροφορίες όπου είχαν πραγματοποιηθεί και συνεχίζουν με τις νέες εγγραφές και καταχωρήσεις [20].

2.4.2 Εκτέλεση

Ενώ τα blockchain μπορούν να χρησιμοποιηθούν ως συστήματα εγγραφής και είναι ιδανικά ως πλατφόρμες συναλλαγών, θεωρούνται αργές ως βάσεις δεδομένων σε σύγκριση με το τι είναι δυνατό για την τεχνολογία ψηφιακών συναλλαγών που βλέπουμε σήμερα με τις κάρτες Visa και PayPal.

Αν και σίγουρα θα υπάρξουν βελτιώσεις σε αυτές τις επιδόσεις, η φύση της τεχνολογίας blockchain απαιτεί να θυσιάζεται κάποια ταχύτητα. Ο τρόπος με τον οποίο τα κατανεμημένα δίκτυα χρησιμοποιούνται σε τεχνολογία blockchain σημαίνει ότι δεν μοιράζονται και επεξεργάζονται ισχυρή δύναμη επεξεργασίας, εξυπηρετούν από κοινού το δίκτυο και στη συνέχεια συγκρίνουν τα αποτελέσματα της δουλειάς τους με το υπόλοιπο δίκτυο έως ότου υπάρξει συναίνεση ότι κάτι συνέβη.

Αντίθετα οι κεντρικές βάσεις δεδομένων αυξάνουν την ταχύτητα τους συνέχεια δίνοντας στην ψηφιακή εποχή νέες τεχνολογικές καινοτομίες.

2.4.3 Εμπιστευτικότητα

Ένα από τα κύρια χαρακτηριστικά μίας βάσης δεδομένων blockchain είναι η εμπιστευτικότητα. Το χαρακτηριστικό αυτό προσδίδει τις εξής ιδιότητες:

- Στην βάση δεδομένων blockchain ο καθένας μπορεί να καταχωρήσει ένα νέο μπλοκ και να επεξεργαστεί χωρίς να πάρει άδεια από κάποιον διαχειριστή.
- Στο blockchain μπορεί το δίκτυο να διαμορφωθεί έτσι ώστε μόνο αυτοί που έχουν την άδεια να μπορούν να διαχειριστούν την βάση δεδομένων όπως γίνεται στην περίπτωση της κεντρικής βάσης δεδομένων.
- Για να γίνει απόκρυψη των πληροφοριών στο blockchain οι κόμβοι-χρήστες απαιτείται να χρησιμοποιήσουν ισχυρή κρυπτογράφηση και κατ επέκταση επιβαρύνονται με ένα υπολογιστικό βάρος.

2.5 Συνοπτικά

Τα βασικά χαρακτηριστικά της τεχνολογίας blockchain τα οποία προκύπτουν από την δομή της είναι τα εξής:

- Αποκεντρωμένη/διανεμημένη τήρηση του μητρώου των δεδομένων από περισσότερους χρήστες / κόμβους.
- Εμπιστοσύνη μεταξύ των χρηστών οι οποίοι είναι σε θέση ανά πάσα στιγμή να επαληθεύσουν οποιαδήποτε καταχώρηση ή συναλλαγή στο δίκτυο.
- Διαφάνεια καθώς όλες οι συναλλαγές είναι καταγεγραμμένες στο μητρώο και είναι δημόσια προσβάσιμες ανά πάσα στιγμή.
- Ασφάλεια καθώς είναι εξαιρετικά δύσκολο να τροποποιηθούν ήδη καταχωρημένες συναλλαγές.

Η εμπιστοσύνη για τις συναλλαγές μεταξύ των χρηστών και η υποχρέωση για την διασφάλιση των δεδομένων και πληροφοριών είναι αρμοδιότητα των χρηστών και όχι κάποιου διαχειριστή μιας κεντρικής βάσης δεδομένων [20].

2.6 Εφαρμογές της τεχνολογίας του Blockchain

2.6.1 Χρηματοπιστωτικές / Ασφαλιστικές υπηρεσίες

Η τεχνολογία του blockchain έχει εφαρμογές σε πολλές υπηρεσίες εκτός από το πιο σύνηθες παράδειγμα χρήσης ως μέσο πληρωμών χωρίς ενδιάμεσο.

Ο τρόπος επεξεργασίας συναλλαγών είναι αρκετά δαπανηρός με ιδιαίτερο βαθμό δυσκολίας το οποίο συνεπάγεται καθυστέρηση στην διαδικασία. Υπάρχει πληθώρα ατόμων που απαιτούνται για να διεκπεραιωθεί η διαδικασία τα οποία άτομα έχουν τα δικά τους αρχεία με αντίστοιχες πληροφορίες και καταχωρήσεις δεδομένων. Η τεχνολογία blockchain απλοποιεί σε μεγάλο βαθμό την διαδικασία ενώ καθιστά περιττή την ανάγκη ύπαρξης ενδιάμεσων προσώπων. Ο χρόνος που απαιτείται για τις συναλλαγές λόγω του ότι δεν υπάρχουν ενδιάμεσοι με δικά τους αρχεία μειώνεται σε μεγάλο βαθμό.

Επίσης το blockchain συνεισφέρει στην διαδικασία επιβεβαίωσης πληρωμών καθώς επιταχύνει την διαδικασία και την μετατρέπει σε απλούστερη.

Στην διεθνή μεταφορά εμπορευμάτων-Cash Against Documents (CAD), ο παραλήπτης πρέπει να περιμένει ένα χρονικό διάστημα για να λάβει το εμπόρευμα για τον λόγο ότι ο μεταφορέας μέχρι να πάρει την βεβαίωση ότι τα χρήματα έχουν καταβληθεί δεν μπορεί να στείλει την παραγγελία. Με την τεχνολογία του blockchain εξαλείφοντας τον ενδιάμεσο δηλαδή την τράπεζα η διαδικασία αυτή μεταξύ αποστολέα και παραλήπτη θα διαρκέσει ελάχιστα λεπτά [7].

2.6.2 Τήρηση μητρώων

Η τεχνολογία blockchain αποτελεί ουσιαστικά έναν νέο τρόπο καταχώρησης και αποθήκευσης πληροφοριών. Πραγματοποιείται με τέτοιο τρόπο ώστε να δημιουργείται μία αλληλένδετη αλυσίδα δεδομένων με παλιές και νέες καταχωρήσεις (ιστορικό) αποτρέποντας διπλές εγγραφές, κακόπιστες καταχωρήσεις. Σημαντική εφαρμογή της είναι στην τήρηση μητρώων, όπως για παράδειγμα το κτηματολόγιο, το ληξιαρχείο, μητρώο εταιρειών, φορολογικό μητρώο. Παράλληλα η συνεισφορά του blockchain σε καταχωρήσεις λογιστικών γραφείων όπου η αποφυγή λαθών και η ακεραιότητα εγγραφών είναι ύψιστης σημασίας θα ήταν μεγάλη με αξιοσημείωτο αντίκτυπο. Από την στιγμή που τα έγγραφα με τα αρχεία και τα δεδομένα των χρηστών θα καταχωρηθούν στην βάση είναι δύσκολο να κλαπούν ακόμα και αν επιχειρήσει κάποιος από το δίκτυο λόγω του ότι «σπάνε σε μικρότερα κομμάτια» τα οποία διανέμονται σε ολόκληρο το δίκτυο [7].

2.6.3 Έξυπνα συμβόλαια (smart contracts)

Ο όρος «έξυπνα συμβόλαια» αναφέρεται σε ψηφιοποιημένα συμβόλαια στα οποία έχει ενσωματωθεί κώδικας υπό τη μορφή if –this – then – that, (εν συντομία IFTTT), τα οποία εκτελούνται αυτόματα αν πληρούνται οι προϋποθέσεις που έχουν τεθεί. Ένα τέτοιο παράδειγμα είναι ο διακόπτης εκκίνησης (starter interrupter) ο οποίος λειτουργεί με αυτόν τον τρόπο έχει δηλαδή ένα smart contract και δεν επιτρέπει να ξεκινήσει η διαδικασία εάν δεν έχουν τηρηθεί προηγουμένως οι απαιτούμενες προϋποθέσεις. Σε ένα έξυπνο συμβόλαιο και από τις δύο πλευρές γίνεται διαπραγμάτευση για τους όρους όπως προδιαγραφές των προϊόντων, ποσότητα, τίμημα, χρόνο και τόπο εκπλήρωσης μέσω του blockchain.

Το παράδειγμα του διακόπτη εκκίνησης είναι ακόμη πιο χαρακτηριστικό των δυνατοτήτων του συνδυασμού των έξυπνων συμβολαίων και της τεχνολογίας blockchain. Αντί ο προγραμματισμός του λογισμικού συμβολαίου (contract ware) να καθορίζεται από τον δανειστή, θα καθορίζεται και θα εκτελείται από την πλατφόρμα blockchain. Κανένα από τα μέλη δεν χρειάζεται να εμπιστεύεται το άλλο για την εκτέλεση του συμβολαίου αλλά την ουδέτερη πλατφόρμα blockchain, η οποία θα εκτελεί τους όρους όταν πληρωθούν οι προϋποθέσεις [7].

2.6.4 Διακυβέρνηση

Η ψηφιακή διακυβέρνηση και η ηλεκτρονική ψηφοφορία πραγματοποιείται πολύ πιο εύκολα και με πολύ μεγαλύτερη ασφάλεια λόγω της κρυπτογράφησης των δεδομένων των χρηστών άρα αυτόματα η διαδικασία κλοπής και μεταποίησης είναι δυσκολότερη με τους χρήστες πλέον να έχουν δυνατότητα να γνωρίζουν και να επιβεβαιώνουν για τους ψήφους τους (πχ. αν μετρήθηκαν) [7].

Ακόμα ένα σημαντικό πρόβλημα που παρατηρείται και είναι σε θέση να αντιμετωπίσει η τεχνολογία είναι η χρηματοδότηση των μη κερδοσκοπικών εταιρειών και συγκεκριμένα εάν η καταβολή των χρημάτων από τους δωρητές χρησιμοποιείται για το σκοπό που αυτοί τα έστειλαν αφού οι ίδιοι θα μπορούν να παρακολουθούν την κίνηση τους. Επίσης όσον αφορά τα κεφάλαια και το πώς αυτά θα μοιραστούν το blockchain καθιστά την διαδικασία πιο εύκολη με την παρακολούθηση τους.

2.6.5 Διαχείριση ψηφιακής ταυτότητας

Η δημιουργία ψηφιακής ταυτότητας με την τεχνολογία blockchain βρίσκει αντίκρισμα στην ασφάλεια, στην διασφάλιση της εγκυρότητας και στην εμπιστοσύνη των δεδομένων χρησιμοποιώντας κρυπτογράφηση καθώς και διαχείριση των προσωπικών δεδομένων όποτε αυτοί επιθυμούν. Τα δεδομένα μετά την κρυπτογράφηση καταχωρούνται στο blockchain και ύστερα ο κάθε χρήστης μπορεί

όταν επιθυμεί να τα στείλει όπου αυτός θελήσει. Τον αριθμό hash μετά την κρυπτογράφηση μπορεί να παρέχεται από έναν πάροχο υπηρεσίας χωρίς αυτός να έχει αρμοδιότητες πάνω στα δεδομένα. Σε μία τέτοια περίπτωση όπου ο χρήστης είναι το επίκεντρο και διαχειρίζεται για αυτόν προσδίδεται ο όρος «αυτό – κυριαρχική ταυτότητα» (self – sovereign identity).

Συμπερασματικά με την ψηφιακή ταυτότητα θα έχουμε πρόσβαση, εγγραφή κλπ. σε ποικίλους κλάδους, οργανισμούς λόγω της ιδιότητας της να περικλείει διαφόρων ειδών δεδομένα και πληροφορίες όπως για παράδειγμα τα απαραίτητα στοιχεία για την άδεια οδήγησης για την ταυτότητα, τους διάφορους κωδικούς που χρησιμοποιούμε στην καθημερινότητα μας [7].

2.6.6 Διαδίκτυο των πραγμάτων

Το blockchain παρέχει ασφάλεια στις πληροφορίες που μεταδίδονται κρυπτογραφώντας τα δεδομένα αυτά στη βάση. Αυτό χρειάζεται γιατί οι έξυπνες συσκευές μεταφέρουν πληροφορίες ενώ βρίσκονται στο διαδίκτυο είτε κατά την μεταξύ τους επικοινωνία είτε μεταξύ συσκευών και χρηστών. Αποτέλεσμα της τεχνολογίας του internet of things είναι καλύτερη απόδοση των συσκευών και μη άσκοπη σπατάλη ενέργειας επειδή ελέγχονται ανά πάσα στιγμή από οπουδήποτε μέσω των χρηστών [7].

2.6.7 Διαχείριση εφοδιαστικής αλυσίδας

Η τεχνολογία blockchain χάριν στις ιδιότητες της δίνει νέες δυνατότητες στην οργάνωση, διανομή και παρακολούθηση των προϊόντων στην εφοδιαστική αλυσίδα. Παραδείγματα όπως η κατανομή των προϊόντων στις αποθήκες και το πώς φτάνουν εκεί, αισθητήρες που βρίσκονται στα προϊόντα και δίνουν πληροφορίες για την τοποθεσία αλλά και για την κατάσταση στην οποία βρίσκονται πραγματοποιούνται λόγω καταχωρήσεων που βρίσκονται στην βάση του blockchain. Όλες αυτές οι πληροφορίες που συλλέγονται διαχειρίζονται από το blockchain και προστατεύονται καθώς όλοι τηρούν το αρχείο ώστε να είναι ενημερωμένο με δεδομένα που έχουν όλοι οι κόμβοι την ίδια χρονική στιγμή [7].

2.7 Νομικές πτυχές της τεχνολογίας του Blockchain

2.7.1 Προσωπικά δεδομένα

Η τεχνολογία του blockchain οφείλει να εξεταστεί σε ότι αφορά τα προσωπικά δεδομένα και την προστασία αυτών. Στις ανοιχτές πλατφόρμες δεν υπάρχει γνώση για

το ποιος εκτελεί τις ενέργειες και για τον λόγο αυτό ενέργειες όπως το να διαγράφει ή να ενημερώνει κανείς είναι δύσκολο να γίνουν σε βάσεις δεδομένων όπως ορίζει ο Κανονισμός για την προστασία των προσωπικών δεδομένων (GDPR) [7].

2.7.2 Εικονικά νομίσματα / κρυπτονομίσματα

Η νομική αντιμετώπιση των κρυπτονομισμάτων οφείλει να εφαρμοστεί βάσει του χαρακτηρισμού που θα τους αποδοθεί από το νόμο αναλύοντας πρώτα την φύση τους ότι πρόκειται για περιουσιακά στοιχεία και όχι για τα κλασικά νομίσματα με ό,τι αυτό συνεπάγεται. Η αξία των κρυπτονομισμάτων είναι συσχετισμένη με την πλατφόρμα του blockchain καθώς για την εκτέλεση ενός έξυπνου συμβολαίου (smart contract) απαιτούνται tokens και εξαρτάται από την κοινότητα που απευθύνεται και από τις δυνατότητες του πρωτοκόλλου blockchain [7].

2.7.3 Δίκαιο προστασίας καταναλωτή

Τα κρυπτονομίσματα επιφέρουν πολλά μειονεκτήματα στους χρήστες όπως έλλειψη διαφάνειας λόγω της δυσκολίας των χρηστών να κατανοήσουν βασικές ιδιότητές τους και κίνδυνο να χάσουν σημαντικά χρηματικά ποσά για τον λόγο ότι αντιμετωπίζουν τα κρυπτονομίσματα ως μέσο αποθήκευσης και όχι τόσο ως μέσο πληρωμών και συναλλαγών

Επειδή τα κρυπτονομίσματα πολλές φορές συγχέονται με το ηλεκτρονικό χρήμα, οι νόμοι που έχουν θεσπιστεί για αυτά είναι ημιτελείς και ασαφείς κυρίως λόγω του ότι δεν ανήκουν σε μια συγκεκριμένη "γεωγραφική περιοχή'' ώστε να εφαρμοστεί το εκάστοτε δίκαιο. Παράλληλα τα κρυπτονομίσματα δεν έχουν εδραιωθεί ακόμα στην καθημερινότητα των ανθρώπων πράγμα το οποίο σημαίνει ότι μπορεί ανά πάσα στιγμή να σταματήσει η έκδοση ή η ισχύς τους.

Η χρήση κρυπτονομισμάτων έχει ως πλεονέκτημα την άμεση συναλλαγή μεταξύ των δύο χωρίς την μεσολάβηση ενδιάμεσου κάνοντας την διαδικασία γρηγορότερη. Από την άλλη πλευρά όμως ελλοχεύουν και κίνδυνοι απάτης από τις οποίες δεν υπάρχει δυνατότητα επιστροφής χρημάτων επειδή δεν υπάρχει ένας υπεύθυνος κεντρικός μηχανισμός που να διαχειρίζεται τις συναλλαγές [7].

2.7.4 Έξυπνα συμβόλαια (smart contracts)

Χρησιμοποιώντας τις γλώσσες προγραμματισμού αποτρέπουμε κυρίως λάθη που σχετίζονται με παρερμηνείες γεγονός που μικραίνει την αβεβαιότητα για το πώς θα ερμηνευθούν οι όροι ενός έξυπνου συμβολαίου.

Τα έξυπνα συμβόλαια για να είναι εφικτό να αλλάξουν τους όρους εάν γίνει κάποια νομοθετική αλλαγή αφού έχουν ενταχθεί στο blockchain θα πρέπει να

αλληλεπιδρούν με κρατικές βάσεις ώστε να ενημερώνονται αυτόματα από το νομοθετικό πλαίσιο που ισχύει.

Επίσης δύσκολο είναι να προσδιοριστεί και το εφαρμοστέο δίκαιο λόγω της διασυνοριακής φύσης της πλατφόρμας του blockchain μεταξύ των συναλλασσόμενων.

Τέλος, η πιο σημαντική νομική πτυχή από την εφαρμογή των έξυπνων συμβολαίων είναι αδιαμφισβήτητα το στάδιο της εκτέλεσης και κατά πόσο οι αυστηροί όροι αυτόματης εκπλήρωσης των συνεπειών σε βάρος του μη συμμορφούμενου μέρους συνάδουν με το δημόσιο χαρακτήρα των πράξεων εκτέλεσης [7].

2.7.5 Οργανισμοί – νομικά πρόσωπα

Μέσα από την δραστηριοποίηση του blockchain στην ψηφιακή διακυβέρνηση και σε άλλους κλάδους δημιουργήθηκε ένα νέο μοντέλο οργανισμού με το όνομα «αποκεντρωμένος αυτόνομος οργανισμός» ή DAO του οποίου η λειτουργία βασίζεται σε έξυπνα συμβόλαια στην πλατφόρμα του blockchain. Πρόκειται λοιπόν για οργανισμό στον οποίο έχουν αρμοδιότητες άτομα από διαφορετικές χώρες με κανόνες και όρους διαφορετικούς το οποίο γεννά πολλά νομικά ζητήματα [7].

2.7.6 Εξαγορές - Συγχωνεύσεις - Νομικοί έλεγχοι

Το blockchain αποθηκεύοντας στο μητρώο όλες τις καταχωρήσεις, λογιστικά έγγραφα, εταιρικά έγγραφα, ισολογισμούς θέτει κατάλληλες συνθήκες για την αυτοματοποιημένη, ασφαλή και έμπιστη διαδικασία για τους νομικούς ελέγχους και την οικονομική κατάσταση των εταιρειών. Στην σημερινή κοινωνία στο χώρο των νομικών υπηρεσιών μέσω τεχνολογιών τεχνητής νοημοσύνης αντικαθίστανται η ανθρώπινη εργασία με τα μειονεκτήματα της ώστε να παράγονται αποτελέσματα και να γίνεται έλεγχος σε τεράστιο όγκο δεδομένων σε πολύ μικρότερο χρονικό διάστημα [7].

2.7.7 Δικονομία: αποδεικτικά μέσα

Προκειμένου ο δικαστής να επαληθεύει τα στοιχεία που του έχουν αποσταλεί με βεβαιότητα και χωρίς ανάμειξη άλλων μελών θα πρέπει τα στοιχεία αυτά να καταχωρούνται στο blockchain. Αυτό μπορεί να συμβεί εύκολα λόγω της ιδιότητας της βάσης δεδομένων η οποία είναι δημόσια και μπορεί να έχει πρόσβαση για τις απαραίτητες πληροφορίες [7].

3. Vision

3.1 Σκοπός

Σκοπός του παρόντος εγγράφου είναι να καθορίσει τις απαιτήσεις υψηλού επιπέδου του συστήματος έκδοσης βεβαιώσεων και πιστοποιητικών.

3.2 Πεδίο Εφαρμογής

Αυτό το έγγραφο ισχύει για το σύστημα έκδοσης βεβαιώσεων και πιστοποιητικών το οποίο θα αναπτυχθεί από το τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων. Το τμήμα πληροφορικής θα αναπτύξει αυτό το σύστημα για την αυθεντικοποίηση των βεβαιώσεων και των πιστοποιητικών.

3.3 Τοποθέτηση

Το σύστημα που θα υλοποιηθεί θα τροποποιήσει το back–end μέρος του ήδη υπάρχοντος συστήματος με ένα νέο βασισμένο σε νέες τεχνολογίες (blockchain).

3.3.1 Δήλωση προβλήματος

Το πρόβλημα είναι	Η δυσκολία ελέγχου της εγκυρότητας ενός πιστοποιητικού και τα πλαστά πιστοποιητικά
Επηρεάζει	Τους φοιτητές και τους καθηγητές
Και έχει ως αντίκτυπο	Την πιθανή υποκλοπή προσωπικών αρχείων
Μια επιτυχημένη λύση θα ήταν	Η χρήση της τεχνολογίας blockchain

Πίνακας 3.1.: Δήλωση προβλήματος.

3.3.2 Δήλωση θέσης προϊόντος

Προς	Τους φοιτητές και το Πανεπιστήμιο Αιγαίου
Οι οποίοι	Χρησιμοποιούν το σύστημα για την έκδοση των βεβαιώσεων και πιστοποιητικών

Το νέο σύστημα	Είναι ένα εργαλείο (πλατφόρμα)
Το οποίο	Χρησιμοποιεί τεχνολογίες blockchain και την πλατφόρμα Ethereum
Σε αντίθεση	Με το υπάρχον σύστημα
Το νέο σύστημα θα παρέχει	Διαφάνεια προς τους χρήστες, εμπιστοσύνη για την ασφάλεια των προσωπικών δεδομένων, μεγαλύτερη ευκολία χρήσης καθώς και αποτελεσματικότητα

Πίνακας 3.2.: Δήλωση θέσης προϊόντος.

3.4 Περιγραφές ενδιαφερόμενων

Αυτή η ενότητα περιγράφει τους χρήστες του συστήματος.

3.4.1 Σύνοψη ενδιαφερόμενων

Name	**Description**	**Responsibilities**
Φοιτητές	Φοιτητές	Λήψη βεβαιώσεων και πιστοποιητικών
Γραμματεία	Γραμματεία	Έλεγχος και έγκριση βεβαιώσεων
Διαχειριστές συστήματος	Σύστημα	Συντήρηση συστήματος

Πίνακας 3.3.: Σύνοψη ενδιαφερόμενων.

3.4.2 Περιβάλλον χρήστη

Η κοινότητα των χρηστών του Πανεπιστημίου είναι μια μεγάλη περίπλοκη κοινότητα που απαιτεί την ευελιξία, την ασφάλεια και τον χρόνο ανταπόκρισης που μπορεί να προσφέρει ένα νέο σύστημα αυθεντικοποίησης

Κάθε χρήστης θα μπορεί να διεκπεραιώσει οποιαδήποτε ενέργεια με το σύστημα σε πολύ μικρό χρονικό διάστημα όπως και με το τρέχων σύστημα

Δεν υπάρχει κάποιος περιορισμός σχετικά με την πρόσβαση στο σύστημα καθώς αυτή γίνεται και μέσω των τοπικών Η/Υ του πανεπιστημίου αλλά και από απομακρυσμένες συνδέσεις εκτός αυτού.

Το νέο σύστημα θα χρησιμοποιεί την πλατφόρμα του Ethereum με χρήση τεχνολογιών blockchain και θα χαρακτηρίζεται από διαλειτουργικότητα καθώς θα παίρνει στοιχεία από πλατφόρμες όπως το Icarus (πλατφόρμα ενημέρωσης φοιτητών) κ.α.

3.5 Επισκόπηση προϊόντος

Αυτή η ενότητα παρέχει μια επισκόπηση των δυνατοτήτων του συστήματος, των διεπαφών με το εξωτερικό σύστημα βάσης δεδομένων και τη διαμόρφωση του συστήματος.

Το «Academic Certificates» θα αντικαταστήσει το υπάρχον σύστημα αυθεντικοποίησης. Το νέο σύστημα θα διασυνδέεται με την Βάση Δεδομένων και τον κατάλογο μαθημάτων μέσω από τις διάφορες πλατφόρμες που χρησιμοποιεί ήδη το Πανεπιστήμιο.

3.5.1 Ανάγκες και δυνατότητες

Need	**Priority**	**Features**
Ένα νέο σύστημα αυθεντικοποιήσης	Υψηλή	Χρήση της τεχνολογίας blockchain, πλατφόρμα ethereum
Επικοινωνία με άλλες πλατφόρμες του Πανεπιστημίου	Μέτρια	Άμεση επικοινωνία μεταξύ των εφαρμογών, διαλειτουργικότητα της εκάστοτε πλατφόρμας
Ευκολία στη χρήση	Υψηλή	Φιλικό περιβάλλον προς το χρήστη
Ασφαλής σύνδεση	Υψηλή	Σύνδεση μέσω συνθηματικών του χρήστη

Πίνακας 3.4.: Ανάγκες και δυνατότητες.

3.6 Απαιτήσεις συστήματος

- Το σύστημα θα διασυνδέεται με το Σύστημα Βάσεων Δεδομένων και τους Καταλόγους μαθημάτων.

3.6.1 Απαιτήσεις απόδοσης

- Το σύστημα θα υποστηρίζει πολλούς ταυτόχρονους χρήστες ανά πάσα στιγμή.
- Το σύστημα παρέχει πρόσβαση στην βάση δεδομένων καταλόγου μαθημάτων.

3.6.2 Περιβαλλοντικές Απαιτήσεις

- Καμία.

3.7 Περιορισμοί

Το σύστημα δεν απαιτεί ανάπτυξη ή προμήθεια υλικού. Οι διαθέσιμες πληροφορίες περιορίζονται στον τύπο των δεδομένων που υποστηρίζονται από την υπάρχουσα Βάση Δεδομένων και τους καταλόγους μαθημάτων.

3.7.1 Ευχρηστία

Στην υποενότητα αυτή καθορίζονται οι απαιτήσεις που σχετίζονται με την ευκολία που μπορεί να χρησιμοποιηθεί το σύστημα.

- Η έκδοση και αυθεντικοποίηση βεβαιώσεων και πιστοποιητικών πραγματοποιείται με μεγάλη ευκολία μέσω της χρήσης του μενού από τον χρήστη.

3.7.2 Αξιοπιστία

Αυτή η υποενότητα καθορίζει τις απαιτήσεις σχετικά με την αξιοπιστία του συστήματος (π.χ. μέσος χρόνος μεταξύ των βλαβών).

- Σε περίπτωση όπου μία συναλλαγή αποτύχει λόγω πτώσης του συστήματος τα προσωπικά στοιχεία του χρήστη παραμένουν ασφαλή. Ύστερα η διαδικασία ξεκινά από την αρχή ώστε να διεκπεραιώσει ο χρήστης τις ενέργειες που επιθυμεί.

3.7.3 Χωρητικότητα

Αυτή η υποενότητα υποδεικνύει τις απαιτήσεις σχετικά με τον ελάχιστο αριθμό αντικειμένων που μπορεί να υποστηρίξει το σύστημα:

- Το σύστημα πρέπει να υποστηρίζει μεγάλο αριθμό ταυτόχρονων συναλλαγών.
- Το σύστημα πρέπει να υποστηρίζει μεγάλο όγκο χρηστών.

3.7.4 Χρόνος καθυστέρησης

Αυτή η υποενότητα υποδεικνύει τις απαιτήσεις σχετικά με το μέγιστο χρόνο που επιτρέπεται στο σύστημα να εκτελέσει συγκεκριμένες εργασίες (δηλ. Λειτουργίες του συστήματος).

- Ο χρήστης θα πρέπει να είναι σε θέση να εκτελέσει ό,τι διεργασία επιθυμεί σε μικρό χρονικό διάστημα και το σύστημα να παρέχει τον μικρότερο δυνατό χρόνο απόκρισης.

3.7.5 Χρόνος απόκρισης

Το παρόν πεδίο καθορίζει τις απαιτήσεις σχετικά με το μέγιστο χρόνο που επιτρέπεται στο σύστημα να ανταποκρίνεται σε αιτήματα:

- Όλες οι αποκρίσεις του συστήματος θα γίνονται εντός πέντε (5) δευτερολέπτων.

3.7.6 Υποστήριξη συστήματος

Αυτή η ενότητα υποδεικνύει τυχόν απαιτήσεις που θα βελτιώσουν την υποστήριξη και την συντήρηση του ενσωματωμένου συστήματος.

- Το σύστημα θα πρέπει να είναι ικανό να πραγματοποιεί μια αναβάθμιση χωρίς να χρειαστεί να τεθεί εκτός λειτουργίας.

4. Ανάλυση εφαρμογής

4.1 Περιγραφή Blockcerts

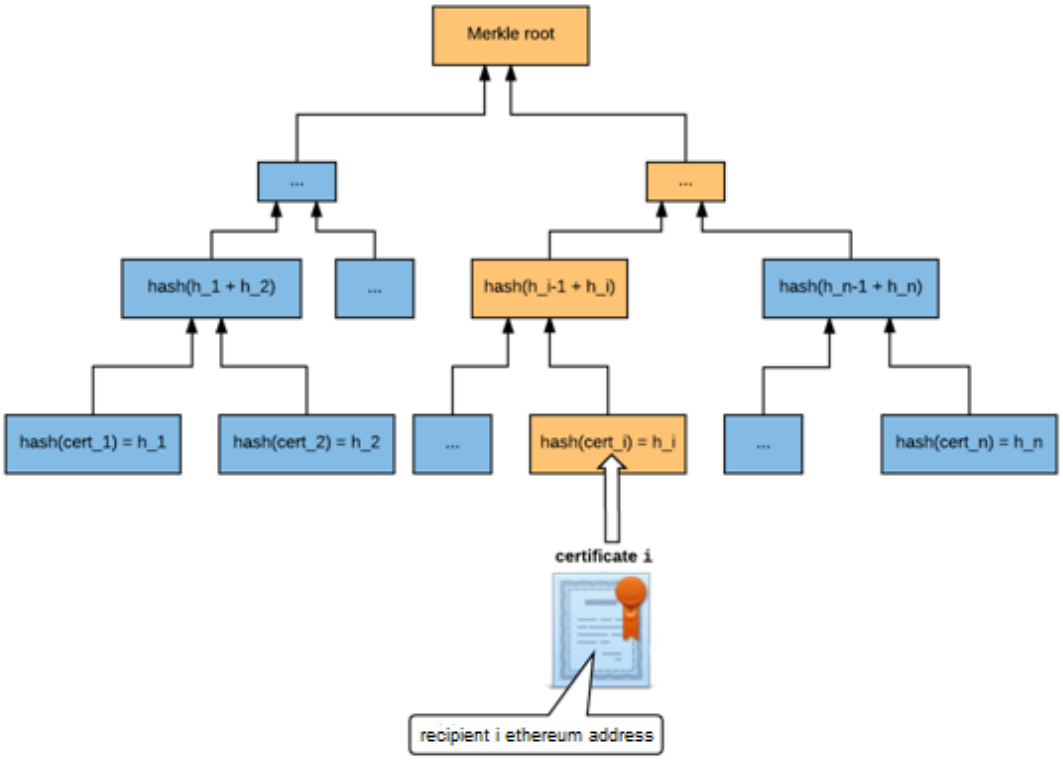
Το Blockerts είναι ένα ανοικτό πρότυπο για τη δημιουργία, την έκδοση, την προβολή και την επαλήθευση των πιστοποιητικών που βασίζονται σε blockchain. Αυτά τα ψηφιακά αρχεία καταγράφονται σε blockchain, κρυπτογραφικά υπογεγραμμένα, ανθεκτικά στις παραβιάσεις και κοινά. Ο στόχος είναι να δημιουργηθεί ένα κύμα καινοτομίας που δίνει στα άτομα τη δυνατότητα να κατέχουν και να μοιράζονται τα δικά τους επίσημα αρχεία. Ο αρχικός σχεδιασμός βασίστηκε σε πρωτότυπα που αναπτύχθηκαν στο MIT Media Lab και στο Learning Machine το 2016 με επικεφαλής τον Philipp Schmidt και τον Juliana Nazare (με τη βοήθεια πολλών άλλων, συμπεριλαμβανομένων των Guy Zyskind και Jeremy Rubin). Η Kim Hamilton ήταν η επικεφαλής της τεχνολογίας και διευθύντρια για το βασικό λογισμικό και πρότυπο που εκδόθηκε και ο Chris Downie ανέπτυξε την εφαρμογή για το iOS [21].

4.2 Πώς λειτουργεί η έκδοση παρτίδων

Παρόλο που είναι δυνατή η έκδοση ενός πιστοποιητικού με μία συναλλαγή, είναι πολύ πιο αποδοτικό να χρησιμοποιείται μία συναλλαγή για την έκδοση μιας παρτίδας πιστοποιητικών.

Ο εκδότης δημιουργεί ένα δένδρο Merkle [16] που περιλαμβάνει hashes πιστοποιητικών και καταχωρεί τη ρίζα του δέντρου στο πεδίο extraData στη συναλλαγή Ethereum.

Υποθέστε ότι η παρτίδα περιέχει n πιστοποιητικά και κάθε πιστοποιητικό i περιέχει τις πληροφορίες του παραλήπτη i. Ο εκδότης κρυπτογραφεί κάθε πιστοποιητικό και τα συνδυάζει σε ένα δέντρο Merkle. Η ρίζα του δέντρου Merkle, η οποία έχει κρυπτογράφηση 256-bit, εκδίδεται στο blockchain Ethereum.



Εικόνα 4.1 Merkle tree.

Το πιστοποιητικό Blockchain που έχει δοθεί στον παραλήπτη i περιέχει μια μορφοποιημένη υπογραφή (Merkle Proof Signature Suite), αποδεικνύοντας ότι το πιστοποιητικό i περιέχεται στο δέντρο Merkle.

Αυτή η υπογραφή περιέχει:

- Το ID της συναλλαγής Ethereum που αποθηκεύει τη ρίζα Merkle.
- Την ρίζα του Merkle στο blockchain.
- Το hash για το πιστοποιητικό του παραλήπτη i.
- Το path από το πιστοποιητικό του παραλήπτη i στη ρίζα του δέντρου.

Η διαδικασία επαλήθευσης (verify) εκτελεί υπολογισμούς για να ελέγξει ότι:

- Το hash του πιστοποιητικού i αντιστοιχεί στην τιμή στην υπογραφή.
- Το path του δέντρου είναι έγκυρο.
- Η ρίζα του δέντρου (Merkle) που είναι αποθηκευμένη στο blockchain αντιστοιχεί με την τιμή στην υπογραφή.

Αυτά τα βήματα αποδεικνύουν ότι το πιστοποιητικό δεν έχει παραβιαστεί από τότε που εκδόθηκε.

4.3 Το hash ενός πιστοποιητικού

Τα περιεχόμενα του JSON πιστοποιητικού Blockchain χωρίς τον κόμβο υπογραφής είναι το πιστοποιητικό που δημιούργησε ο εκδότης. Αυτή είναι η τιμή που απαιτείται να κρυπτογραφηθεί για την αντιστοίχιση με την απόδειξη. Επειδή δεν υπάρχουν εγγυήσεις σχετικά με την μορφοποίηση του JSON, πρώτα κανονικοποιούμε το πιστοποιητικό (χωρίς την υπογραφή) έναντι του σχήματος JSON LD. Αυτό μας επιτρέπει να αποκτήσουμε ένα ντετερμινιστικό hash σε όλες τις πλατφόρμες.

4.4 Τι πρέπει να είναι σε μια παρτίδα;

Ο τρόπος ορισμού μιας παρτίδας μπορεί να διαφέρει, αλλά πρέπει να οριστεί έτσι ώστε να αλλάζει σπάνια. Για παράδειγμα το "φοιτητές 2019 " σε σχέση με το "φοιτητές" (το τελευταίο θα έπρεπε να ενημερώνεται κάθε χρόνο). Το μέγεθος της παρτίδας περιορίζεται από το μέγιστο μέγεθος συναλλαγής 100KB που επιβάλλεται από το δίκτυο Ethereum. Αυτό θα ανέλθει σε περίπου 2.000 παραλήπτες ανά παρτίδα πιστοποιητικού.

4.5 Δομή συναλλαγών

Μια συναλλαγή εκτελείται για κάθε παρτίδα πιστοποιητικών. Δεν υπάρχει όριο στον αριθμό των πιστοποιητικών που μπορεί να περιλαμβάνονται σε μια παρτίδα, οπότε οι παρτίδες συνήθως ορίζονται σε λογικές ομάδες, όπως το "Απόφοιτοι τμήματος Μ.Π.Ε.Σ 2019".

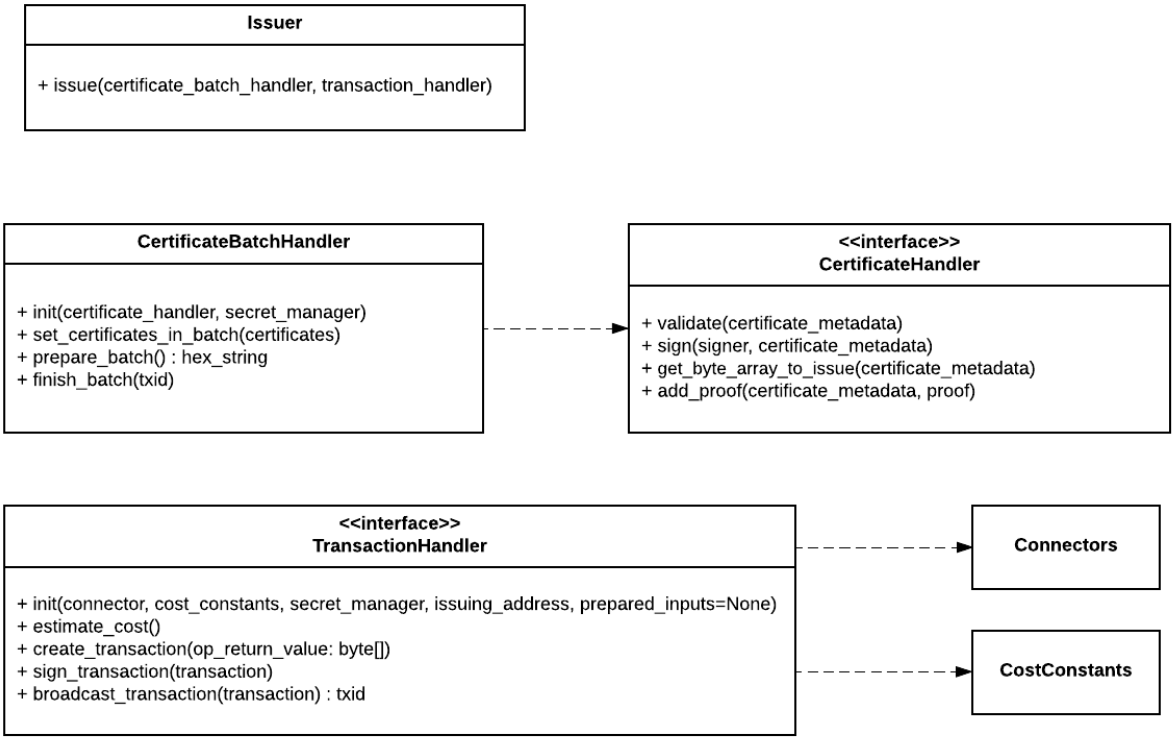
Η δομή της συναλλαγής είναι η ακόλουθη:
Είσοδος:
Ελάχιστο ποσό ethereum από τη διεύθυνση ethereum του Εκδότη.
Έξοδοι:
Πεδίο extraData, όπου αποθηκεύεται ένα hash της παρτίδας των πιστοποιητικών.
Προαιρετικό: αλλαγή σε διεύθυνση εκδότη.

Η έξοδος extraData είναι παρόμοια με την έξοδο OP_RETURN του bitcoin και χρησιμοποιείται για να αποδείξει την εγκυρότητα της παρτίδας πιστοποιητικών. Αυτή η έξοδος αποθηκεύει δεδομένα, τα οποία είναι το hash της ρίζας Merkle της παρτίδας πιστοποιητικών. Ανά πάσα στιγμή, μπορούμε να αναζητήσουμε αυτή την τιμή στο blockchain για να επιβεβαιώσουμε μια συναλλαγή.

Η διεύθυνση Ethereum και η χρονική σφραγίδα του εκδότη από τη συναλλαγή είναι επίσης κρίσιμες για τη διαδικασία επαλήθευσης [16].

4.6 Σχεδιασμός κλάσης

4.6.1 Βασικές κλάσεις έκδοσης



Εικόνα 4.2 Βασικές κλάσεις έκδοσης.

Η διαδικασία έκδοσης είναι αρκετά απλή. Βασίζεται στις κλάσεις Certificate Handler και Transaction Handler για να πραγματοποιήσει την εργασία της εξαγωγής των δεδομένων προκειμένου να γίνει η έκδοση στο blockchain.

Η κλάση Certificate Batch Handler διαχειρίζεται τα πιστοποιητικά που θα εκδοθούν στο blockchain. Επίσης, είναι υπεύθυνη για την έκδοση των πιστοποιητικών με την αντίστοιχη σειρά ζήτησης γεγονός που είναι ιδιαίτερα σημαντικό διότι τα στοιχεία του δέντρου (Merkle) πρέπει να συσχετιστούν με το σωστό πιστοποιητικό.

❖ prepare_batch

- Εκτελεί τα προπαρασκευαστικά βήματα για τα πιστοποιητικά της παρτίδας, συμπεριλαμβανομένης της διαμόρφωσης των δεδομένων που θα "μπουν" στο blockchain.
- Επιστρέφει τον δεκαεξαδικό πίνακα byte που θα πάει στο blockchain.

❖ Το finish_batch διασφαλίζει ότι κάθε πιστοποιητικό ενημερώνεται με τις πληροφορίες συναλλαγής στο blockchain.

Η κλάση Certificate Handler είναι υπεύθυνη για την ανάγνωση και την ενημέρωση ενός συγκεκριμένου πιστοποιητικού που προσδιορίζεται από το certificate_metadata.

Η κλάση Transaction Handler ασχολείται με την τοποθέτηση των δεδομένων στο blockchain [16].

4.7 Διαδικασία επαλήθευσης πιστοποιητικών

Η βιβλιοθήκη Python cert-verifier επαληθεύει τα πιστοποιητικά Blockchain. Παρακάτω ακολουθεί η ανάλυση της διαδικασίας επαλήθευσης των πιστοποιητικών.

4.7.1 Blockchain πιστοποιητικό

Ένα πιστοποιητικό Blockchain περιέχει:

- Το περιεχόμενο που πρέπει να επαληθευτεί.
- Τη θέση πρόσθετων "εισόδων" που απαιτούνται για την επαλήθευση.

Ένα πιστοποιητικό Blockchain πρέπει επίσης να περιέχει ένα πεδίο certificate.signature.anchors, το οποίο πρέπει να έχει τουλάχιστον μία anchor εγγραφή. Μία εγγραφή anchor για μια Ethereum συναλλαγή φαίνεται παρακάτω.

"type": "ETHData",

"sourceId":
"0xc952833ba9c1b630282df24b3ca56e30963e75a70005ca95190a050492298f51"

Μία συναλλαγή του Ethereum μπορεί να ληφθεί από μια υπηρεσία όπως το etherscan.io και η γενική της μορφή θα είναι:

https://etherscan.io/tx/<transaction_id>

4.7.2 Ταυτότητα εκδότη

Το πεδίο badge.issuer.id στο πιστοποιητικό αναφέρει που θα βρείτε τις τρέχουσες πληροφορίες του εκδότη σχετικά με το ποια κλειδιά ισχύουν. Πρόκειται για ένα URI HTTP, το οποίο περιέχει μια σειρά δημοσίων κλειδιών που αξιώνει ο εκδότης.

```
{
```

```
 ...
 "publicKey": [
  {
   "created": "2012-01-03T14:34:57+0000",
   "revoked": "2012-05-01T18:11:19+0000",
   "id": "ecdsa-koblitz-pubkey:16wyA4kLFiaQSEE9xZEFTEMXTzWsGf4Zki"
  },
  {
   "created": "2016-01-03T14:34:57+0000",
   "id": "ecdsa-koblitz-pubkey:1Q3P94rdNyftFBEKiN1fxmt2HnQgSCB619"
}
 ],
 ...
}
```

Αυτές οι πληροφορίες είναι απαραίτητες για να ελεγχθούν τα δημόσια κλειδιά που αξιώνει ο εκδότης με τις πληροφορίες από τη συναλλαγή.

4.7.3 Πληροφορίες ανάκλησης εκδότη

Το πεδίο badge.issuer.revocationList στο πιστοποιητικό αναφέρει που να αποκτήσετε τον κατάλογο ανακληθέντων πιστοποιητικών του εκδότη (assertions). Για παράδειγμα:

```
{
 ...
 "revokedAssertions": [
  {
   "id": "urn:uuid:93019408-acd8-4420-be5e-0400d643954a",
   "revocationReason": "Honor code violation"
```

```
  },
  {
    "id": "urn:uuid:8e0b8a28-beff-43de-a72c-820bc360db3d",
"revocationReason": "Issuedinerror."
  }
 ]
}
```

Μπορεί να επιτραπούν και άλλες υλοποιήσεις ανάκλησης, ανάλογα με τις υλοποιήσεις που επιτρέπονται από το blockchain και την "καταλληλότητα" του εκδότη.

4.7.4 Έλεγχος ακεραιότητας πιστοποιητικού

Ο έλεγχος της ακεραιότητας ενός πιστοποιητικού εξασφαλίζει ότι δεν έχει παραβιαστεί από έναν τρίτο. Η διαδικασία του ελέγχου γίνεται σε 3 βήματα.

1. Επικύρωση της υπογραφής του Merkle στο πιστοποιητικό.

Χρησιμοποιείται μία μορφή υπογραφής Verifiable Claims Merkle Proof 2017, η οποία βασίζεται στο Chainpoint 2.0. Σύμφωνα με τις προδιαγραφές, η υπογραφή του Merkle μπορεί να επαληθευτεί στέλνοντας το αντικείμενο στο signature πεδίο σε οποιονδήποτε verifier συμβατό με Chainpoint 2.0, αφού αντικατασταθεί το Merkle Proof 2017 με τον τύπο Chainpoint SHA256v2.

2. Σύγκριση του hash του τοπικού πιστοποιητικού με την τιμή του hash στην υπογραφή.

Χρησιμοποιείται η κανονικοποίηση JSON-LD για να εξασφαλιστεί μια συνεπή σειρά "εισόδου" JSON. Αυτό εξασφαλίζει συνεπή hash για τους verifiers. Ανάλογα με την προδιαγραφή υπογραφής JSON-LD, αυτό λειτουργεί ως εξής:

- Αφαίρεση του "signature" τμήματος από το πιστοποιητικό.
- Το JSON-LD κανονικοποιεί το αποτέλεσμα
    - Με ρυθμίσεις: {'algorithm': 'URDNA2015', 'format': 'application/nquads'}
- SHA-256 το αποτέλεσμα

Η προκύπτουσα τιμή θα πρέπει να ταιριάζει με την τιμή στο signature.targetHash.

3. Σύγκριση της τιμής merkleRoot στο πιστοποιητικό με την ίδια τιμή στη συναλλαγή.

Οι πληροφορίες της συναλλαγής αποκτούν το αρχείο blockchain του περιεχομένου. Αυτό το βήμα συγκρίνει την τιμή της συναλλαγής με την τιμή στο πιστοποιητικό.

Τα στοιχεία της συναλλαγής για μια συναλλαγή Ethereum στη διεύθυνση https://ropsten.etherscan.io/tx/0xc952833ba9c1b630282df24b3ca56e30963e75a70005ca95190a050492298f51 έχουν δεδομένα στο Input Data πεδίο.

```
{
   ...

"input":"0x4f48e91f0397a49a5b56718a78d681c51932c8bd9242442b94bcfb93434957db"
```

Αφαιρώντας το 0x πρόθεμα, η είσοδος 4f48e91f0397a49a5b56718a78d681c51932c8bd9242442b94bcfb93434957db πρέπει να ταιριάζει με αυτό που παρέχεται στο signature.merkleRoot πεδίο [16].

4.7.5 Έλεγχος αυθεντικότητας πιστοποιητικού

Σε αυτόν τον έλεγχο επιβεβαιώνεται ότι το πιστοποιητικό συντάχθηκε από τον εκδότη. Αυτό επαληθεύεται διασφαλίζοντας ότι το κλειδί υπογραφής για τη συναλλαγή αξιώνεται πράγματι από τον εκδότη και το κλειδί ήταν έγκυρο κατά τη στιγμή της έκδοσης της συναλλαγής.

Χρησιμοποιείται η χρονική σήμανση (timestamp) και η διεύθυνση (input address) από τις πληροφορίες των συναλλαγών καθώς και η ταυτότητα του εκδότη.

Ένα παράδειγμα ακολουθεί παρακάτω, όπου το πεδίο "addr" έχει την διεύθυνση (input address) και το πεδίο "time" τη χρονική σήμανση (timestamp).

```
"inputs":[
 {
```

```
    "prev_out":{
       ...
       "addr":"1Q3P94rdNyftFBEKiN1fxmt2HnQgSCB619",
       ...
     }
  }
]
...
"time":1475524375,
```

Αυτό το δημόσιο κλειδί είναι έγκυρο. Ο εκδότης ισχυρίζεται ότι το δημόσιο κλειδί "1Q3P94rdNyftFBEKiN1fxmt2HnQgSCB619" είναι έγκυρο από 03 Ιανουαρίου 2016 14:34:57 GMT.

Η συναλλαγή πραγματοποιήθηκε στις 03 Οκτωβρίου 2016 19:52:55 GMT, η οποία είναι μετά την ημερομηνία δημιουργίας και το κλειδί δεν είχε λήξει ή ανακλήθηκε κατά την πραγματοποίηση της συναλλαγής.

```
{
 ...
 "publicKey": [
   {
     "created": "2012-01-03T14:34:57+0000",
     "revoked": "2012-05-01T18:11:19+0000",
     "id": "ecdsa-koblitz-pubkey:16wyA4kLFiaQSEE9xZEFTEMXTzWsGf4Zki"
   },
   {
     "created": "2016-01-03T14:34:57+0000",
     "id": "ecdsa-koblitz-pubkey:1Q3P94rdNyftFBEKiN1fxmt2HnQgSCB619",
"expires": "2017-01-03T14:34:57+0000",
```

```
  }
 ],
 ...
}
```

Αυτό αποκλείει εξαιρετικές (και ενδεχομένως δόλιες) περιπτώσεις, όπως:

- Το δημόσιο κλειδί δεν ζητείται από τον εκδότη.
- Η συναλλαγή εκδόθηκε αφού έγινε ανάκληση ή "λήξη" του δημοσίου κλειδιού.

Μια κρίσιμη διάκριση σε αυτό το παράδειγμα είναι ότι η συναλλαγή θεωρείται έγκυρη παρόλο που το κλειδί έληξε. Και όμως στην περίπτωση αυτή, αυτό δεν αποτελεί κάποιο πρόβλημα αφού η συναλλαγή πραγματοποιήθηκε όταν το κλειδί ήταν ενεργό.

Η "λήξη" ενός κλειδιού διαφέρει από την "λήξη" ενός πιστοποιητικού. Η "λήξη" των κλειδιών είναι μια καλή μέθοδος ασφάλειας για τους εκδότες [16].

4.7.6 Έλεγχος ανάκλησης από τον εκδότη

Οι πληροφορίες ανάκλησης εκδότη περιέχουν τη λίστα ανακληθέντων πιστοποιητικών (assertions).

Θεωρείται ότι ένα πιστοποιητικό ανακαλείται εάν οποιαδήποτε εγγραφή id στον πίνακα "revokedAssertions" περιέχει το id του πιστοποιητικού. Το αναγνωριστικό του πιστοποιητικού είναι διαθέσιμο στο πεδίο "id" του πιστοποιητικού.

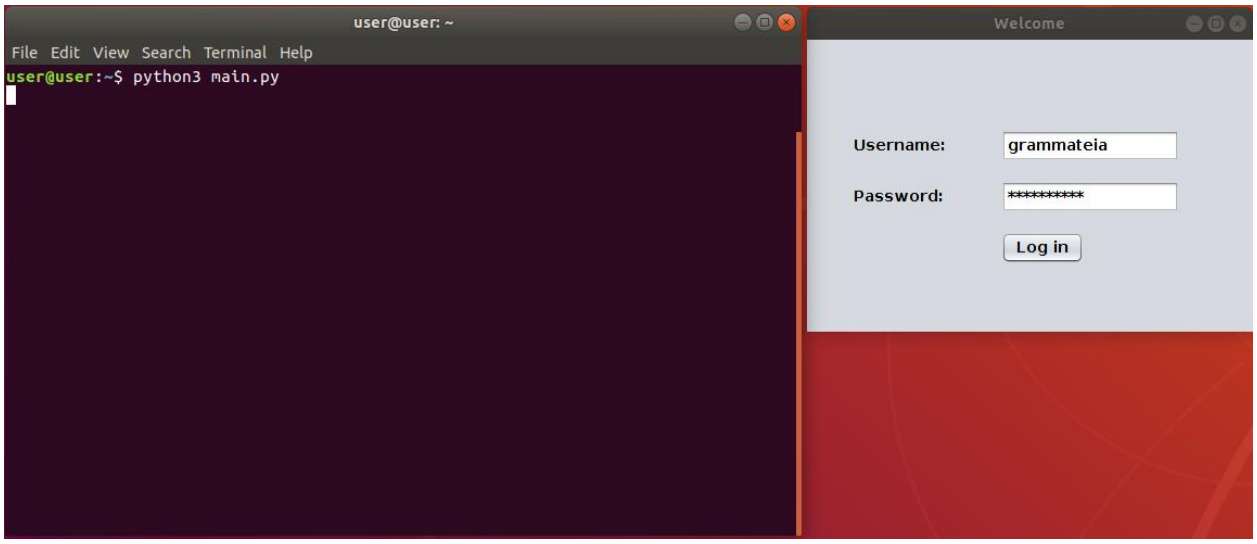
Αν το πιστοποιητικό έχει ανακληθεί, το προαιρετικό πεδίο "revocationReason" μπορεί να παρέχει περισσότερες πληροφορίες σχετικά με τους λόγους για τους οποίους το πιστοποιητικό ανακλήθηκε.

4.7.7 Έλεγχος ότι το πιστοποιητικό δεν έχει λήξει

Το πιστοποιητικό μπορεί επίσης να περιέχει μία ημερομηνία λήξης. Εάν υπάρχει η ημερομηνία, τότε η διαδικασία επαλήθευσης πρέπει να συγκρίνει αυτήν την τιμή (η οποία βρίσκεται στο πεδίο "expires"), έναντι της τρέχουσας ώρας.

5. Περιγραφή εφαρμογής

Αρχικά ανοίγουμε το terminal και τρέχουμε ένα python script το οποίο περιμένει την σύνδεση κάποιου χρήστη στη βάση. Έπειτα από το Netbeans ανοίγουμε την εφαρμογή μας η οποία μας προτρέπει να συνδεθούμε στη βάση, πληκτρολογώντας το όνομα και τον κωδικό του χρήστη.



Εικόνα 5.1 Αρχικό μενού για την είσοδο του χρήστη.

Περίπτωση Χρήσης #1	Είσοδος Χρήστη	
Goal in Context	Η είσοδος του χρήστη στο σύστημα	
Scope & Level	Υποσύστημα αυθεντικοποίησης, Βασική λειτουργία	
Preconditions	Καμία	
Success End Condition	Είσοδος του χρήστη στο σύστημα	
Failed End Condition	Αποτυχία εισόδου του χρήστη στο σύστημα	
Primary Actor	Χρήστης	
Description	**Step**	**Action**
	1	Ο χρήστης εκτελεί το πρόγραμμα
	2	Το σύστημα εμφανίζει το παράθυρο εισόδου στον χρήστη
	3	Ο χρήστης συμπληρώνει τα απαιτούμενα πεδία για την είσοδό του στο σύστημα

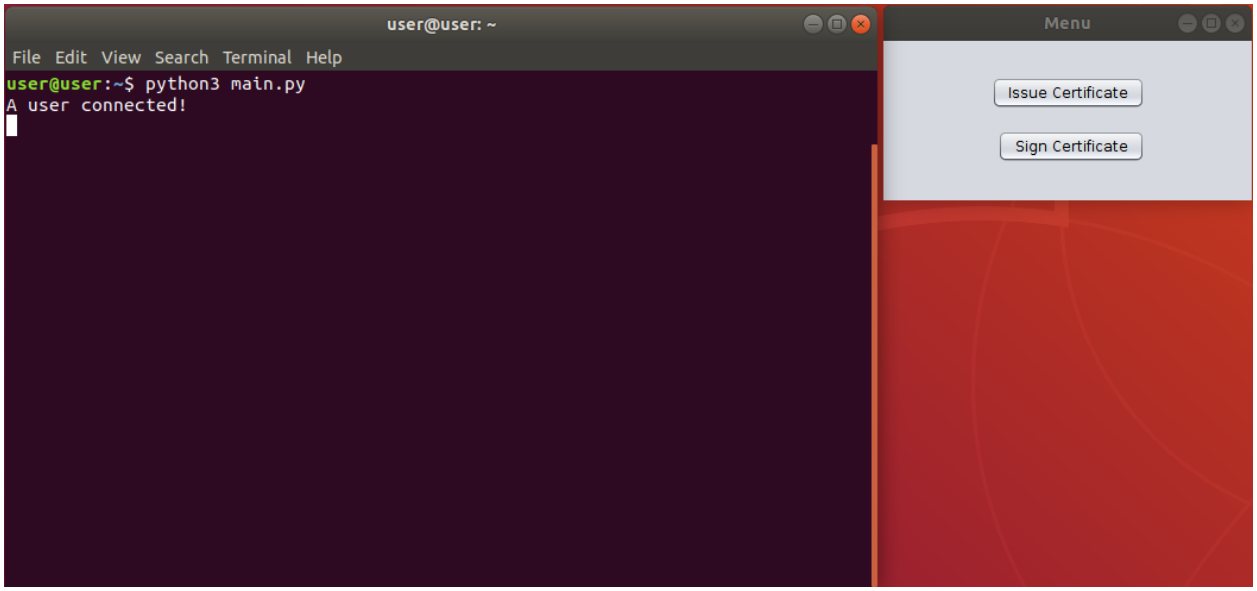
	4	Το σύστημα ελέγχει την εγκυρότητα των στοιχείων του χρήστη
	5	Το σύστημα ελέγχει την ομάδα στην οποία ανήκει ο χρήστης (Grammateia/Foititis)
	6	Το σύστημα πραγματοποιεί τη σύνδεση με τον χρήστη και εμφανίζει το παράθυρο με τις ενέργειες που είναι διαθέσιμες
Extensions	**Step**	**Branching Action**
	4α	Η αυθεντικοποίηση των στοιχείων του χρήστη είναι ανεπιτυχής
		4α1. Το σύστημα ενημερώνει τον χρήστη για την ανεπιτυχή αυθεντικοποίηση των στοιχείων του
Sub-Variations	**Step**	**Branching Action**
	N/A	N/A

Πίνακας 5.1.: Περίπτωση Χρήσης #1: Είσοδος Χρήστη

ID	**Category**	**Description**
ΜΛΑ1.1	Ευχρηστία	Το σύστημα θα πρέπει να παρέχει πληροφορίες στο χρήστη οποτεδήποτε η σύνδεση είναι ανεπιτυχής
ΜΛΑ1.2	Ευχρηστία	Το σύστημα θα πρέπει να διατηρεί τα στοιχεία που εισήγαγε ο χρήστης εκτός από τον κωδικό πρόσβασης, οποτεδήποτε η σύνδεση είναι ανεπιτυχής
ΜΛΑ1.3	Απόδοση	Μετά την επιτυχή σύνδεση του χρήστη, το σύστημα θα πρέπει να εκτελεί αμέσως την επόμενη ενέργεια
ΜΛΑ1.4	Ασφάλεια	Το σύστημα θα πρέπει να παρέχει μηχανισμούς κρυπτογράφησης, για τα δεδομένα που αποστέλλονται μέσω του δικτύου
ΜΛΑ1.5	Υποστήριξη	Το σύστημα θα πρέπει να υποστηρίζει μελλοντική εφαρμογή ενός συστήματος CAPTCHA

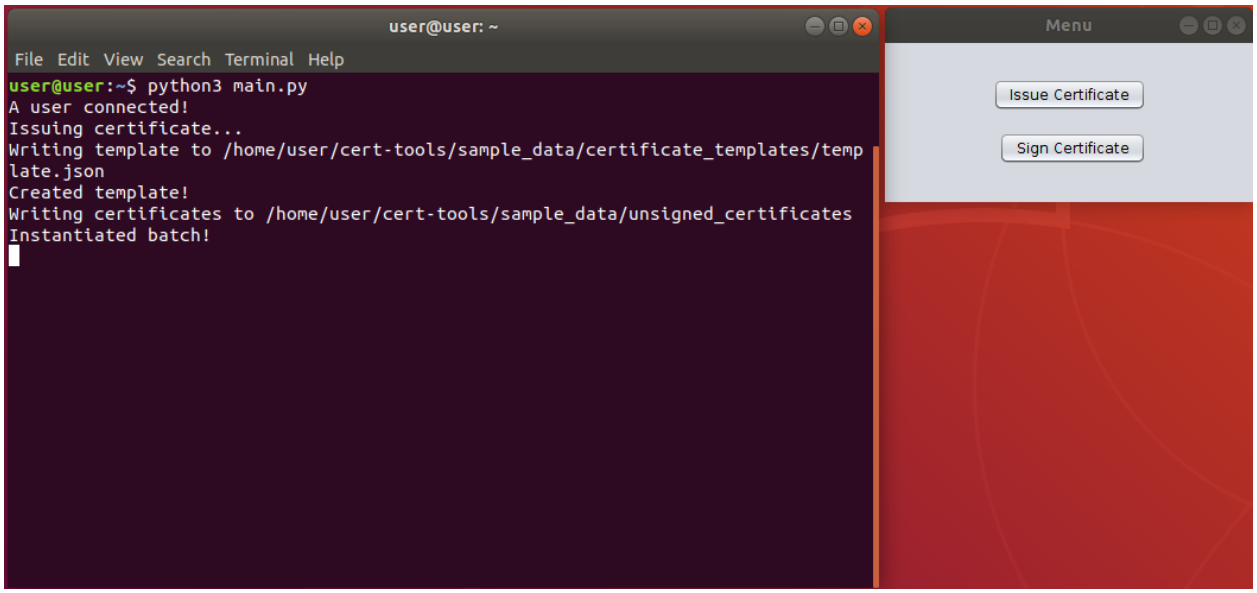
Πίνακας 5.2.: Περίπτωση Χρήσης #1: Μη Λειτουργικές Απαιτήσεις

Αναλόγως με την ομάδα στη οποία βρίσκεται ο χρήστης ο οποίος προσπαθεί να συνδεθεί στη βάση, εμφανίζεται και το ανάλογο μενού. Κατά την σύνδεση ενός χρήστη στη βάση, εμφανίζεται και ένα μήνυμα επιτυχής σύνδεσης στο terminal. Στην περίπτωση που συνδέεται η Γραμματεία εμφανίζεται το μενού που φαίνεται στην παρακάτω εικόνα.



Εικόνα 5.2 Μενού χρήστη που ανήκει στην ομάδα της γραμματείας.

Αναλόγως με την ενέργεια που θα επιλεχθεί θα κληθεί και το αντίστοιχο script για την εκτέλεση της. Επιλέγοντας να τρέξουμε την ενέργεια issue certificate γίνεται δημιουργία ενός template και έκδοση του πιστοποιητικού με βάση αυτό.



Εικόνα 5.3 Εκτέλεση της ενέργειας "Issue Certificate".

Περίπτωση Χρήσης #2	Έκδοση Πιστοποιητικού	
Goal in Context	Η έκδοση πιστοποιητικού από το σύστημα	
Scope & Level	Υποσύστημα έκδοσης πιστοποιητικών, Βασική λειτουργία	
Preconditions	Είσοδος χρήστη που ανήκει στην ομάδα “Grammateia”	
Success End Condition	Έκδοση πιστοποιητικού από το σύστημα	
Failed End Condition	Αποτυχία έκδοσης πιστοποιητικού από το σύστημα	
Primary Actor	Χρήστης	
Description	**Step**	**Action**
	1	Ο χρήστης επιλέγει την ενέργεια “Issue Certificate”
	2	Το σύστημα δημιουργεί ένα template του πιστοποιητικού
	3	Το σύστημα δημιουργεί μια παρτίδα πιστοποιητικών με τα στοιχεία των χρηστών που υπάρχουν στο σύστημα
Extensions	**Step**	**Branching Action**
	2α	Η δημιουργία του template είναι ανεπιτυχής
		2α1. Το σύστημα ενημερώνει τον χρήστη για την ανεπιτυχή δημιουργία του template
	3α	Η δημιουργία της παρτίδας πιστοποιητικών είναι ανεπιτυχής
		3α1. Το σύστημα ενημερώνει τον χρήστη για την ανεπιτυχή δημιουργία της παρτίδας πιστοποιητικών
Sub-Variations	**Step**	**Branching Action**
	N/A	N/A

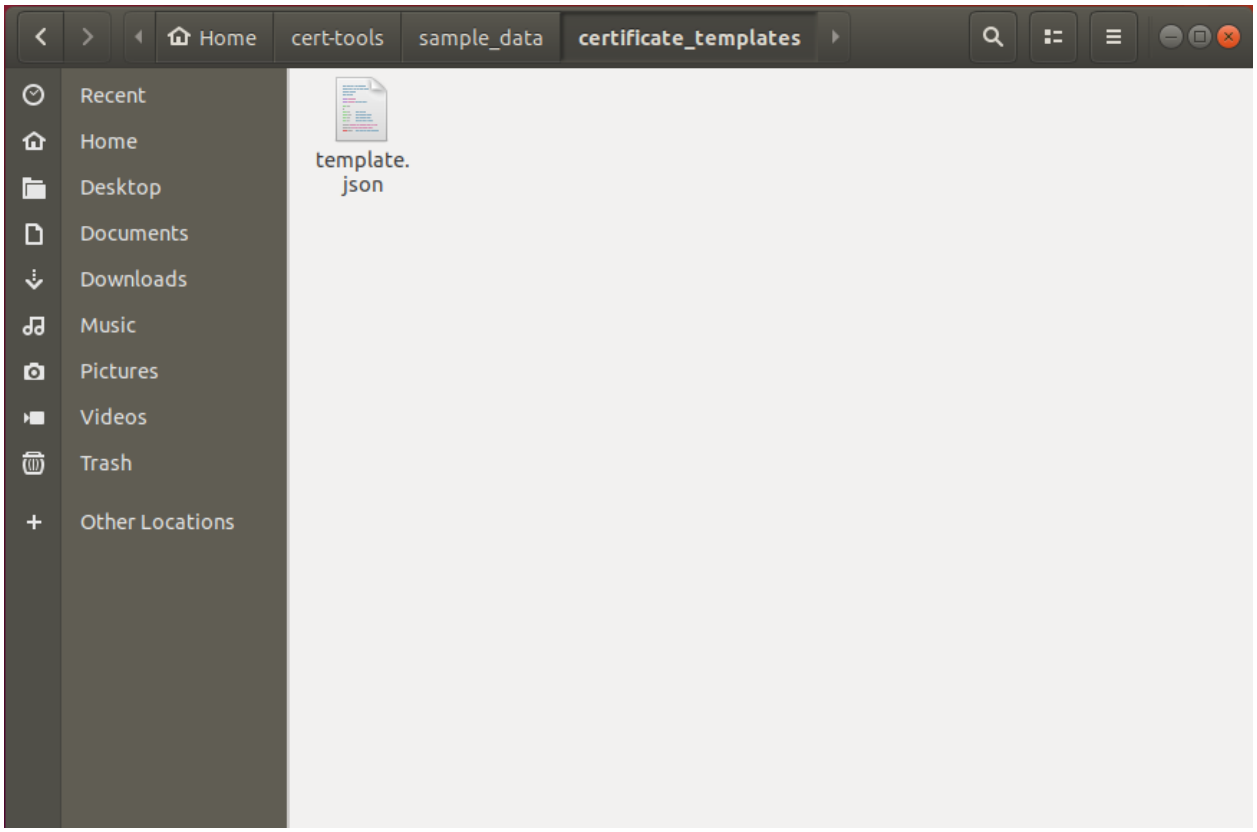
Πίνακας 5.3.: Περίπτωση Χρήσης #2: Έκδοση Πιστοποιητικού

ID	**Category**	**Description**
ΜΛΑ2.1	Ευχρηστία	Το σύστημα θα πρέπει να παρέχει πληροφορίες στο χρήστη οποτεδήποτε η δημιουργία της παρτίδας πιστοποιητικών είναι ανεπιτυχής

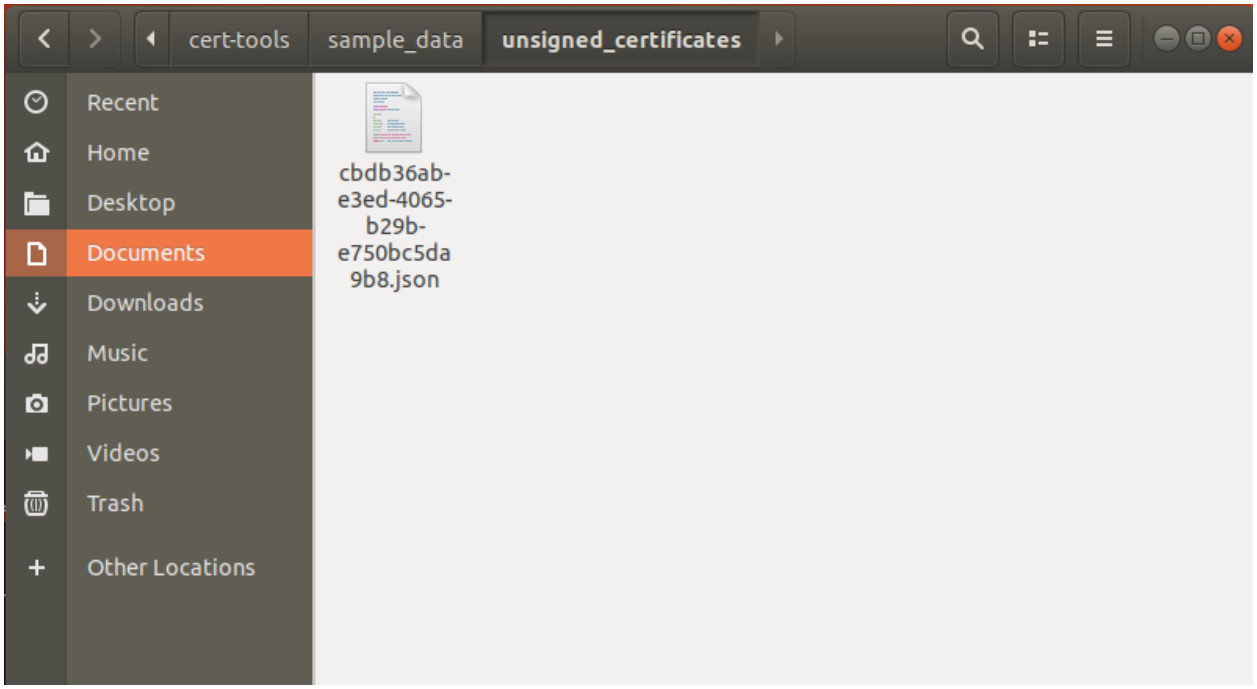
ΜΛΑ2.2	Απόδοση	Μετά την επιτυχή δημιουργία του template, το σύστημα θα πρέπει να εκτελεί αμέσως την επόμενη ενέργεια
ΜΛΑ2.3	Ασφάλεια	Το σύστημα θα πρέπει να παρέχει μηχανισμούς κρυπτογράφησης, για τα δεδομένα που αποστέλλονται μέσω του δικτύου
ΜΛΑ2.4	Ασφάλεια	Το σύστημα θα πρέπει να παρέχει μηχανισμούς ασφαλείας έναντι επιθέσεων CSRF (Cross-Site Request Forgery) και άλλων σχετικών επιθέσεων

Πίνακας 5.4.: Περίπτωση Χρήσης #2: Μη Λειτουργικές Απαιτήσεις

Στις εικόνες που ακολουθούν παρακάτω μπορούμε να δούμε το template που δημιουργήθηκε καθώς και το πιστοποιητικό.

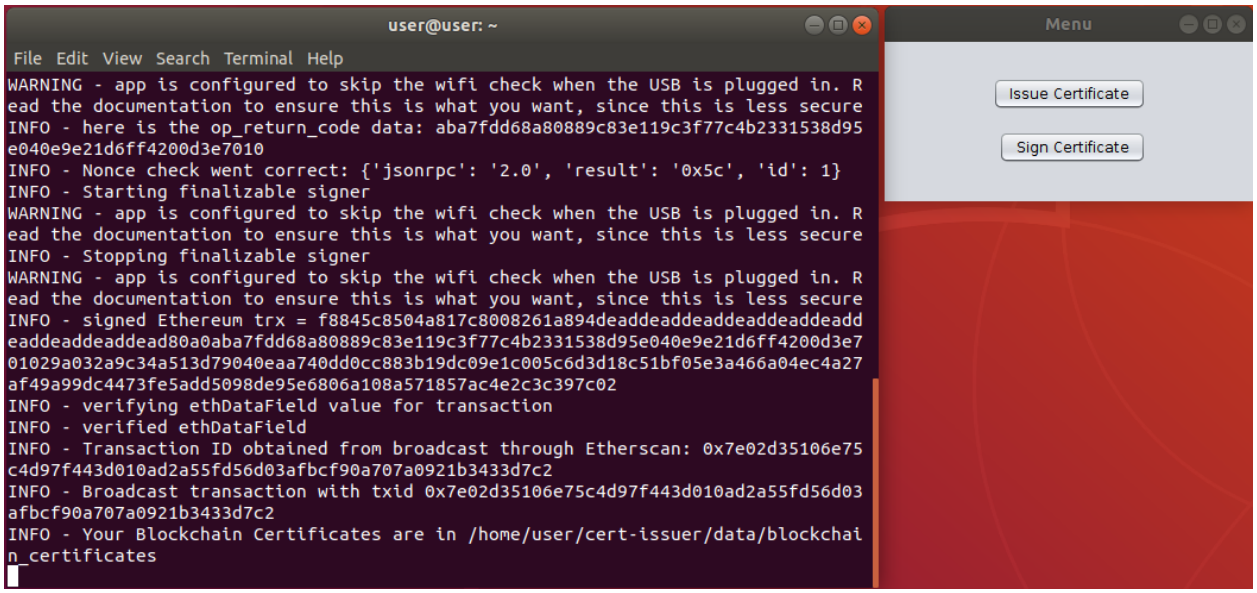


Εικόνα 5.4 Το template που δημιουργήθηκε.



Εικόνα 5.5 Το πιστοποιητικό που δημιουργήθηκε.

Στη συνέχεια εκτελώντας την ενέργεια sign certificate υπογράφεται η παρτίδα πιστοποιητικών που έχουμε δημιουργήσει.



Εικόνα 5.6 Εκτέλεση της ενέργειας "Sign Certificate".

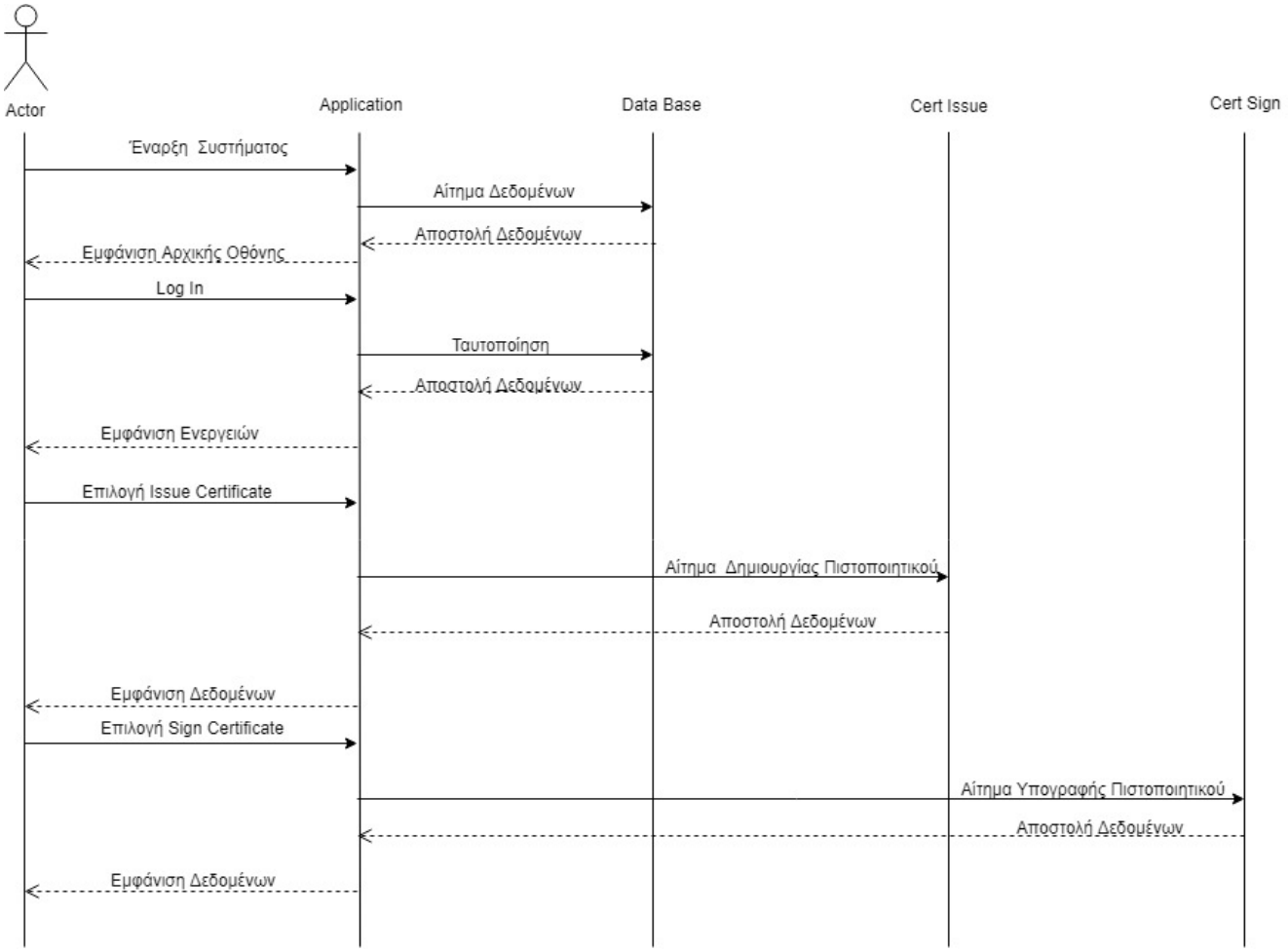
Περίπτωση Χρήσης #3	Υπογραφή Πιστοποιητικού	
Goal in Context	Η υπογραφή του πιστοποιητικού από το σύστημα	
Scope & Level	Υποσύστημα υπογραφής πιστοποιητικών, Βασική λειτουργία	
Preconditions	Είσοδος χρήστη που ανήκει στην ομάδα “Grammateia”	
Success End Condition	Υπογραφή πιστοποιητικού από το σύστημα	
Failed End Condition	Αποτυχία υπογραφής πιστοποιητικού από το σύστημα	
Primary Actor	Χρήστης	
Description	**Step**	**Action**
	1	Ο χρήστης επιλέγει την ενέργεια “Sign Certificate”
	2	Το σύστημα υπογράφει την παρτίδα πιστοποιητικών που βρίσκονται στον φάκελο
Extensions	**Step**	**Branching Action**
	2α	Η υπογραφή της παρτίδας πιστοποιητικών είναι ανεπιτυχής
		2α1. Το σύστημα ενημερώνει τον χρήστη για την ανεπιτυχή υπογραφή της παρτίδας πιστοποιητικών
Sub-Variations	**Step**	**Branching Action**
	N/A	N/A

Πίνακας 5.5.: Περίπτωση Χρήσης #3: Υπογραφή Πιστοποιητικού

ID	**Category**	**Description**
ΜΛΑ3.1	Ευχρηστία	Το σύστημα θα πρέπει να παρέχει πληροφορίες στο χρήστη οποτεδήποτε η υπογραφή της παρτίδας πιστοποιητικών είναι ανεπιτυχής
ΜΛΑ3.2	Απόδοση	Μετά την επιτυχή υπογραφή του πιστοποιητικού, το σύστημα θα πρέπει να εκτελεί αμέσως την επόμενη ενέργεια
ΜΛΑ3.3	Ασφάλεια	Το σύστημα θα πρέπει να παρέχει μηχανισμούς κρυπτογράφησης, για τα δεδομένα που αποστέλλονται μέσω του δικτύου

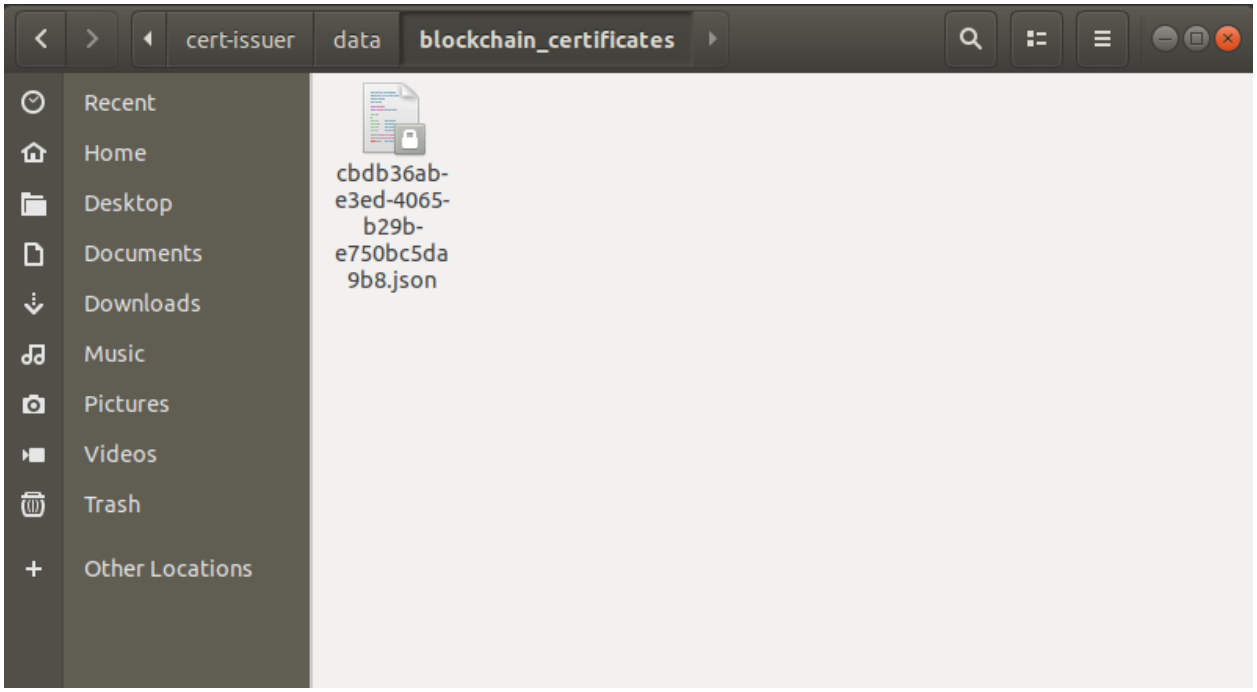
ΜΛΑ3.4	Ασφάλεια	Το σύστημα θα πρέπει να παρέχει μηχανισμούς ασφαλείας έναντι επιθέσεων CSRF και άλλων σχετικών επιθέσεων

Πίνακας 5.6.: Περίπτωση Χρήσης #3: Μη Λειτουργικές Απαιτήσεις

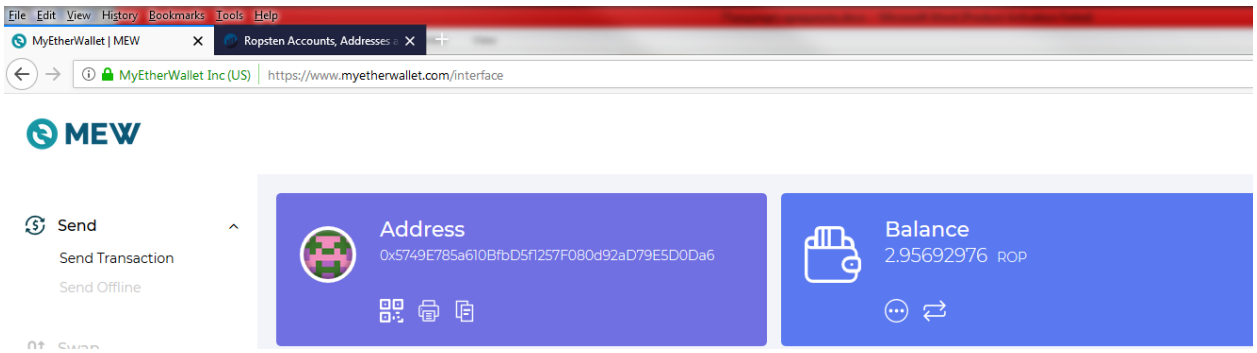


Εικόνα 5.7 Διάγραμμα ακολουθίας (Έκδοση και υπογραφή)

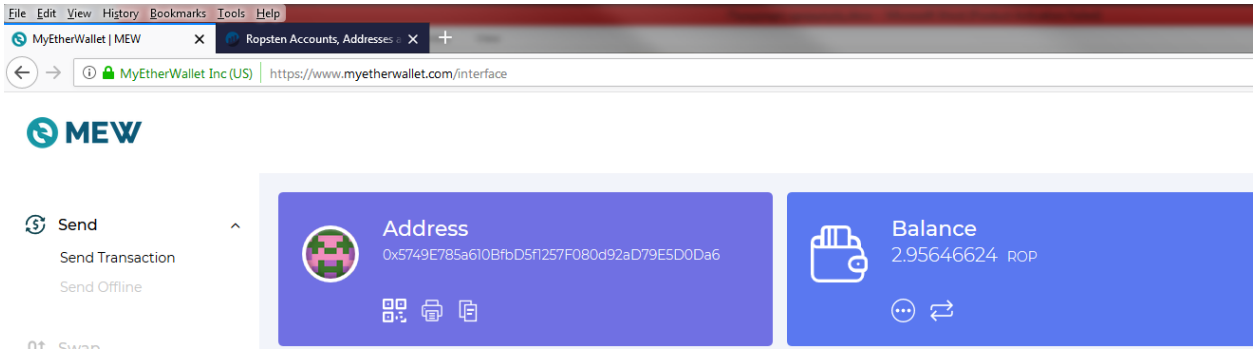
Μετά την ολοκλήρωση της υπογραφής του πιστοποιητικού, μπορούμε να δούμε το υπογεγραμμένο πιστοποιητικό, την αφαίρεση του ποσού που χρειάστηκε η συναλλαγή καθώς και περαιτέρω πληροφορίες για αυτή.



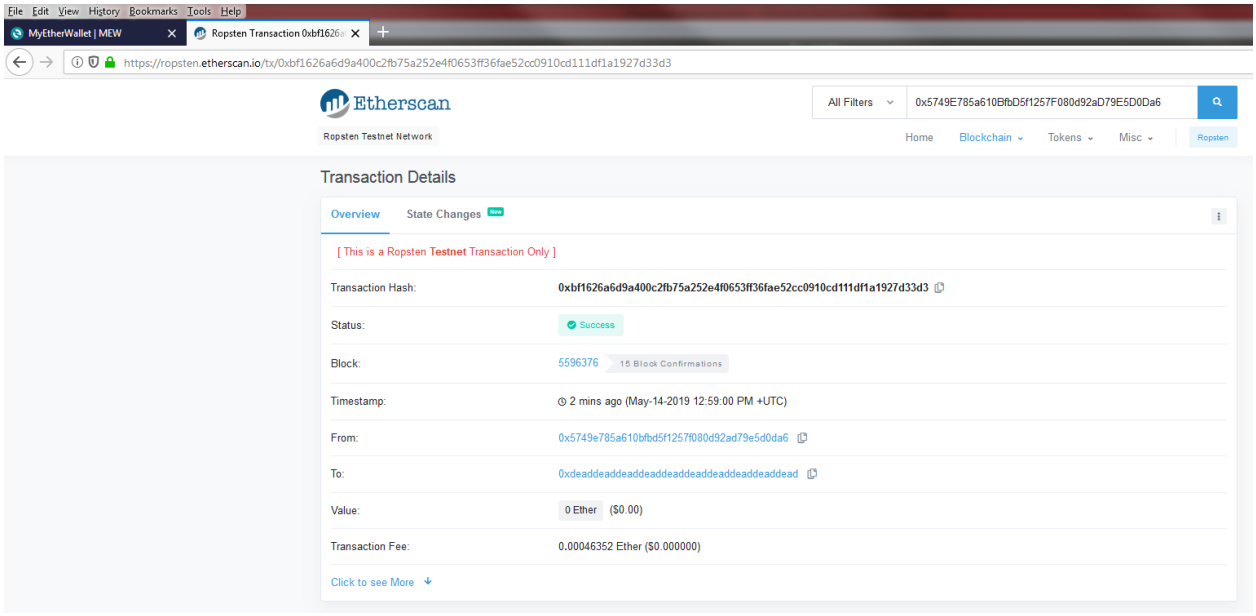
Εικόνα 5.8 Το υπογεγραμμένο πιστοποιητικό.



Εικόνα 5.9 Το υπόλοιπό μας πριν την υπογραφή του πιστοποιητικού.

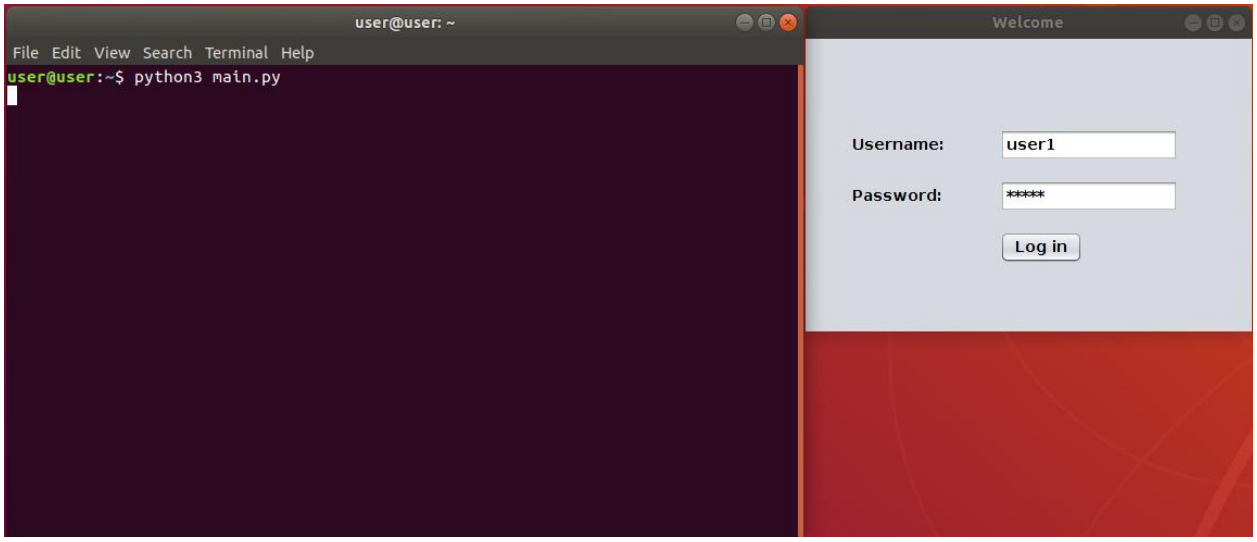


Εικόνα 5.10 Το υπόλοιπό μας μετά την υπογραφή του πιστοποιητικού.

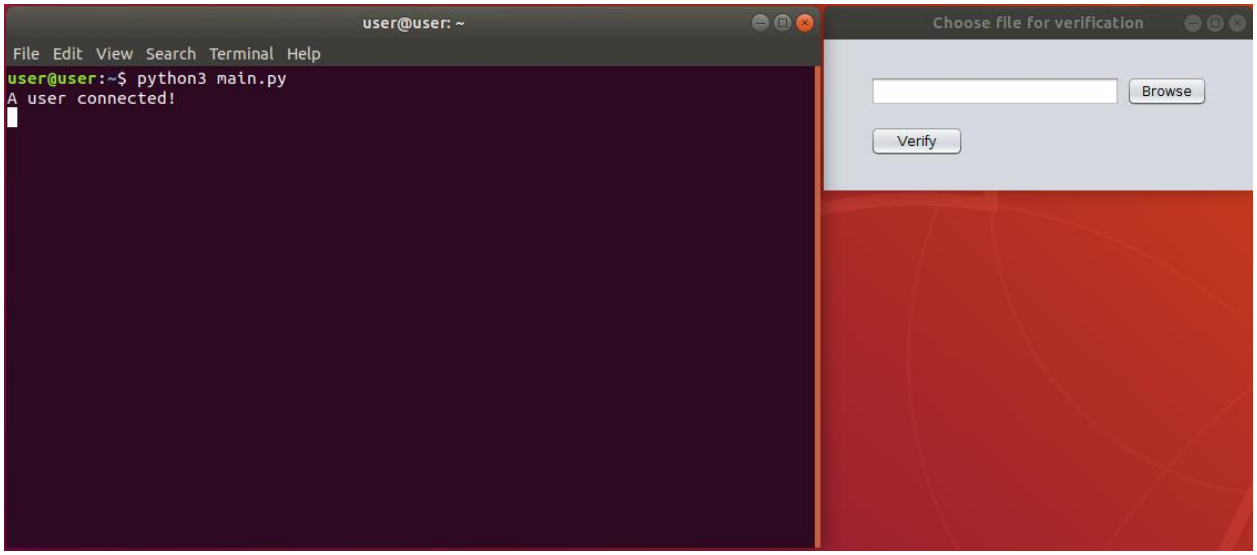


Εικόνα 5.11 Πληροφορίες της συναλλαγής.

Κάνοντας μια νέα σύνδεση με έναν χρήστη που ανήκει στην ομάδα των φοιτητών εμφανίζεται ένα διαφορετικό μενού το οποίο φαίνεται παρακάτω.

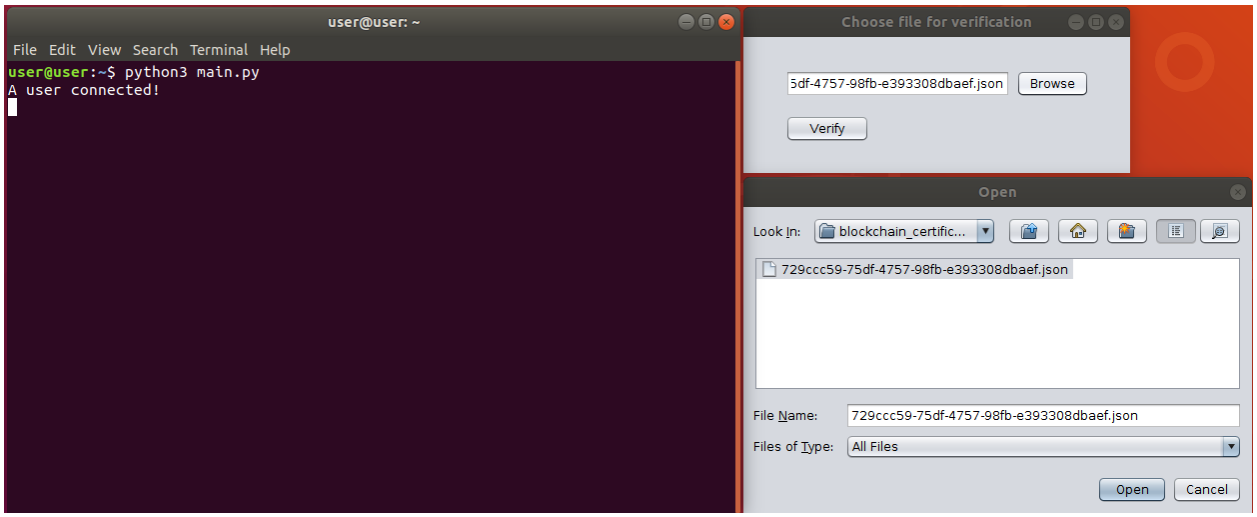


Εικόνα 5.12 Πραγματοποίηση νέας σύνδεσης.

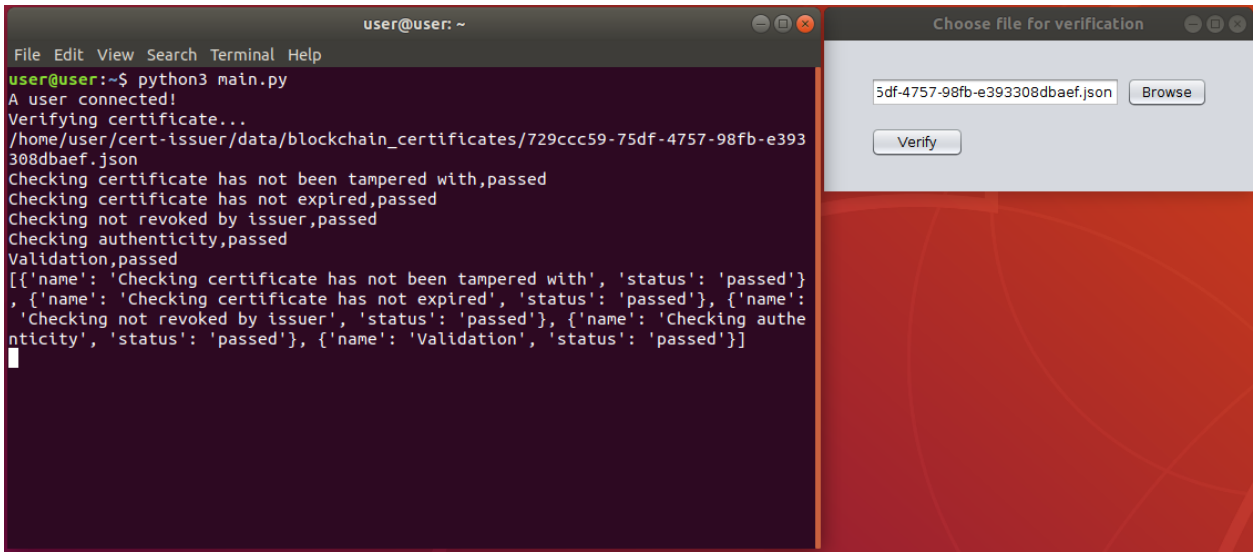


Εικόνα 5.13 Μενού χρήστη που ανήκει στην ομάδα των φοιτητών.

Με το πλήκτρο “browse” ο χρήστης επιλέγει το πιστοποιητικό στο οποίο θέλει να γίνει η αυθεντικοποίηση και πατά το πλήκτρο “verify”.



Εικόνα 5.14 Επιλογή πιστοποιητικού για αυθεντικοποίηση.



Εικόνα 5.15 Εκτέλεση της ενέργειας “Verify”.

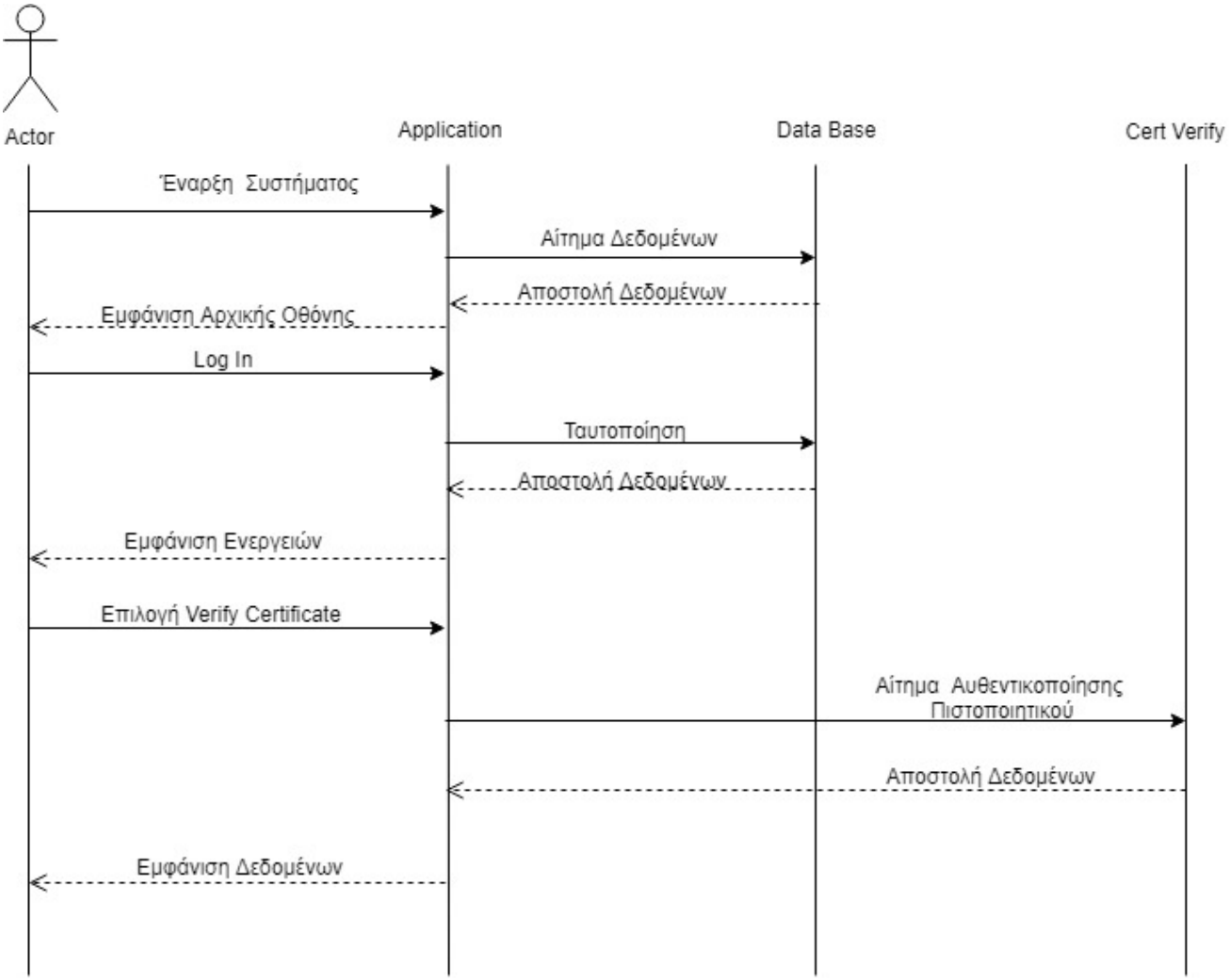
Περίπτωση Χρήσης #4	Αυθεντικοποίηση Πιστοποιητικού
Goal in Context	Η αυθεντικοποίηση του πιστοποιητικού από το σύστημα
Scope & Level	Υποσύστημα αυθεντικοποίησης πιστοποιητικών, Βασική λειτουργία
Preconditions	Είσοδος χρήστη που ανήκει στην ομάδα “Foititis”
Success End Condition	Αυθεντικοποίηση πιστοποιητικού από το σύστημα

Failed End Condition	Αποτυχία αυθεντικοποίησης πιστοποιητικού από το σύστημα	
Primary Actor	Χρήστης	
Description	**Step**	**Action**
	1	Ο χρήστης πατώντας το κουμπί “Browse” επιλέγει το πιστοποιητικό για αυθεντικοποίηση
	2	Ο χρήστης επιλέγει την ενέργεια “Verify”
	3	Το σύστημα ελέγχει την εγκυρότητα του αρχείου που επιλέχθηκε
	4	Το σύστημα πραγματοποιεί την αυθεντικοποίηση του πιστοποιητικού
Extensions	**Step**	**Branching Action**
	3α	Το αρχείο που επιλέχθηκε δεν είναι έγκυρο
		3α1. Το σύστημα ενημερώνει τον χρήστη για την μη συμβατότητα του αρχείου
	4α	Η αυθεντικοποίση του πιστοποιητικού είναι ανεπιτυχής
		4α1. Το σύστημα ενημερώνει τον χρήστη για την ανεπιτυχή αυθεντικοποίση και εμφανίζει το test στο οποίο απέτυχε η αυθεντικοποίηση του πιστοποιητικού
Sub-Variations	**Step**	**Branching Action**
	N/A	N/A

Πίνακας 5.7.: Περίπτωση Χρήσης #4: Αυθεντικοποίηση Πιστοποιητικού

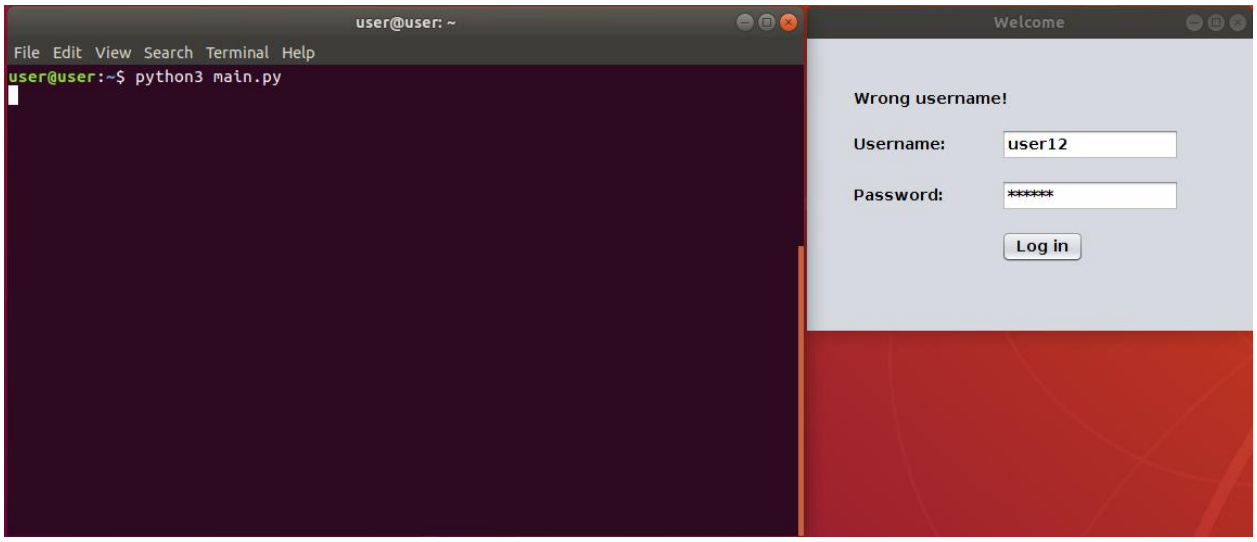
ID	**Category**	**Description**
ΜΛΑ4.1	Ευχρηστία	Το σύστημα θα πρέπει να παρέχει πληροφορίες στο χρήστη οποτεδήποτε η αυθεντικοποίση του πιστοποιητικού είναι ανεπιτυχής
ΜΛΑ4.2	Ασφάλεια	Το σύστημα θα πρέπει να παρέχει μηχανισμούς κρυπτογράφησης, για τα δεδομένα που αποστέλλονται μέσω του δικτύου
ΜΛΑ4.3	Ασφάλεια	Το σύστημα θα πρέπει να παρέχει μηχανισμούς ασφαλείας έναντι επιθέσεων CSRF και άλλων σχετικών επιθέσεων

Πίνακας 5.8.: Περίπτωση Χρήσης #4: Μη Λειτουργικές Απαιτήσεις

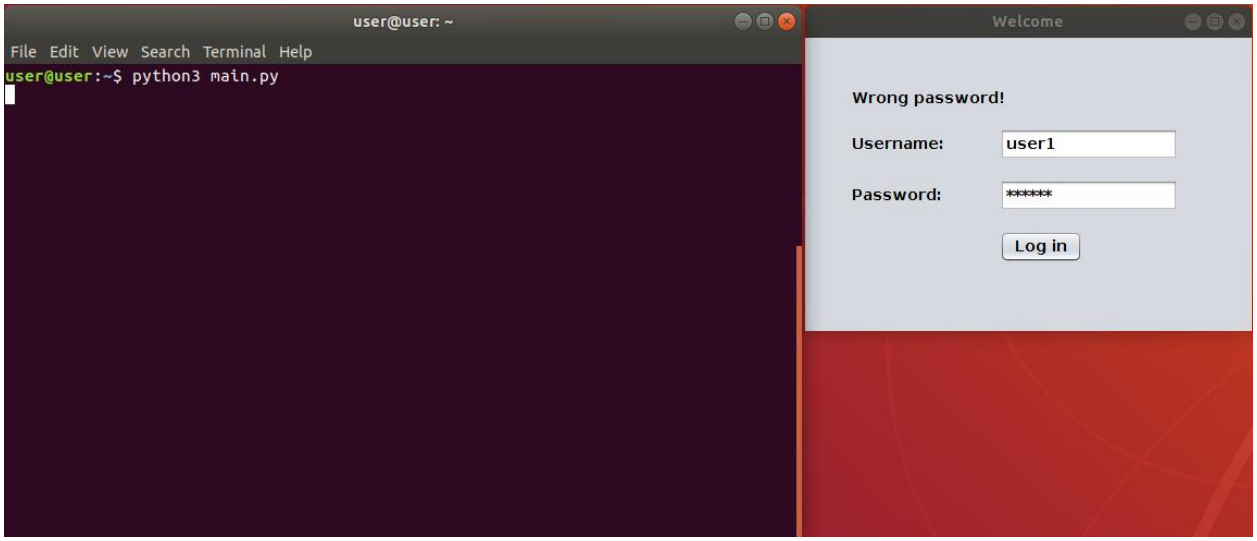


Εικόνα 5.16 Διάγραμμα ακολουθίας (Αυθεντικοποίηση)

Σε περίπτωση που πληκτρολογηθεί λάθος username ή password εμφανίζεται το αντίστοιχο μήνυμα όπως φαίνεται στις παρακάτω εικόνες.

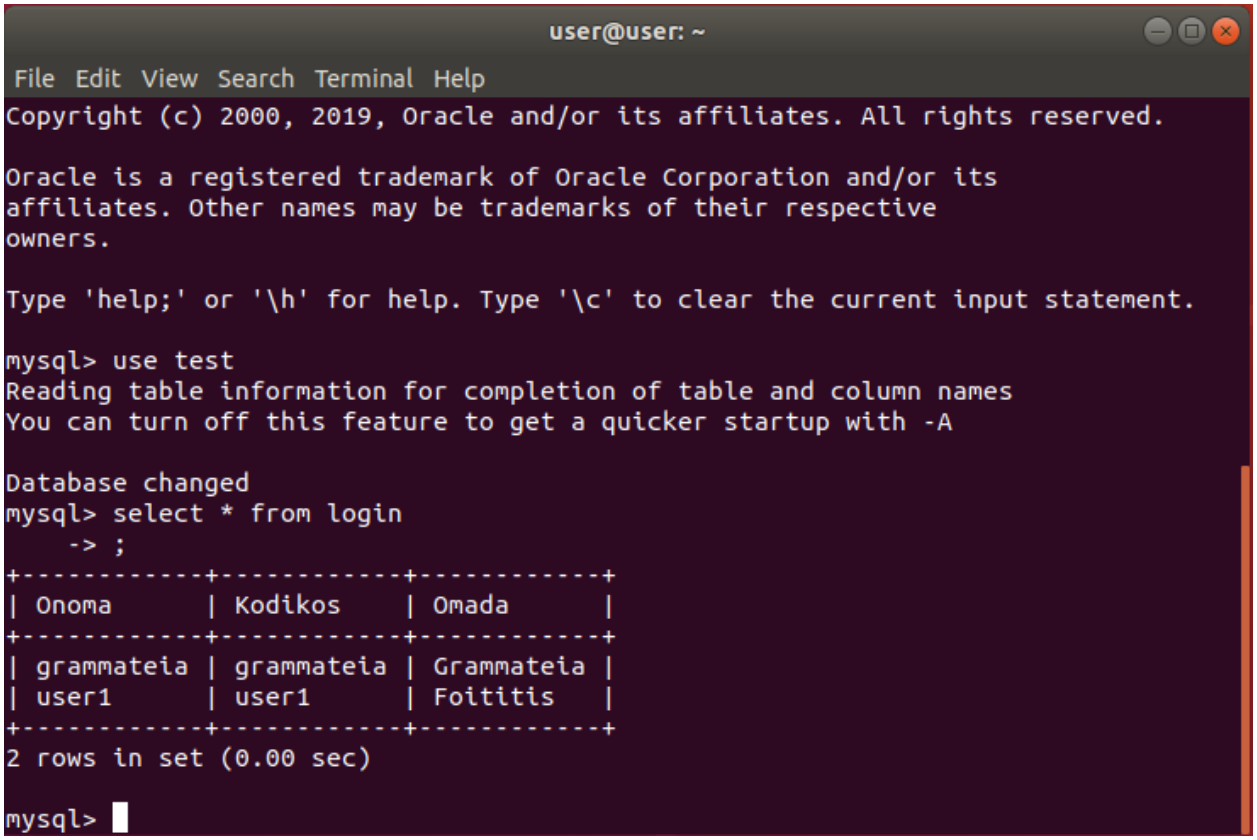


Εικόνα 5.17 Προσπάθεια εισόδου με λανθασμένο όνομα χρήστη.



Εικόνα 5.18 Προσπάθεια εισόδου με λανθασμένο κωδικό χρήστη.

Στην παρακάτω εικόνα φαίνεται η βάση δεδομένων που έχουμε δημιουργήσει με την οποία γίνεται αυθεντικοποίηση των χρηστών ώστε να πραγματοποιηθεί η σύνδεση.



Εικόνα 5.19 Βάση δεδομένων για την καταχώρηση των χρηστών.

6. Αξιολόγηση εφαρμογής

6.1 Ασφάλεια

Για την αξιολόγηση της εφαρμογής Academic Certificates εξετάσαμε την ασφάλεια σε επίπεδο των λειτουργιών, των δεδομένων και της πρόσβασης στο δίκτυο. Τα αποτελέσματα μας οδήγησαν στο συμπέρασμα ότι τα πιστοποιητικά είναι ασφαλή, τα δεδομένα αυθεντικοποίησης είναι αξιόπιστα, αμετάβλητα και επαληθεύσιμα, τα βασικά δεδομένα της πιστοποίησης είναι ασφαλή και ιδιωτικά ακόμα και στην περίπτωση που σταματήσει η χρήση της τεχνολογίας του blockchain και τέλος ότι το σύστημα μειώνει τον κίνδυνο παραβίασής του.

6.2 Αδυναμίες

Στην διαδικασία της αυθεντικοποίησης του πιστοποιητικού το πανεπιστήμιο θα πρέπει να πληρώνει ένα ποσό σε miners για την επιβεβαίωση της εγκυρότητας του πιστοποιητικού στο blockchain. Επίσης, λόγω του ότι η τεχνολογία του blockchain είναι ακόμα σε αρχικό στάδιο, δεν είναι ακόμα πλήρως αποδεκτή από το ευρύ κοινό.

6.3 Μελλοντικές επεκτάσεις

Η τεχνολογία blockchain έχει εδραιωθεί σε πολλούς τομείς όπως χρηματοοικονομικούς, υγείας και κοινωνικές υπηρεσίες. Παράλληλα υπάρχει δυνατότητα επέκτασης και στον εκπαιδευτικό τομέα (Academic Certificates) κάνοντας τις διαδικασίες πιο ασφαλής. Τέλος στον χώρο του πανεπιστημίου το blockchain εκτός από την χρήση του στην αυθεντικοποίηση πιστοποιητικών μπορεί να χρησιμοποιηθεί και στην δήλωση επίσημων εγγράφων [22].

7. Σύνοψη - Συμπεράσματα

Σκοπός της διπλωματικής εργασίας ήταν η εξέταση των νέων τεχνολογιών αποκέντρωσης για την λειτουργία μίας αποκεντρωμένης εφαρμογής (DApp – Decentralized Application), η οποία θα μπορούσε να αποτελέσει κομμάτι του αποκεντρωμένου ιστού. Στην εργασία αυτή λοιπόν, έγινε προσπάθεια χρήσης τεχνολογιών κατανεμημένων συστημάτων με κυρίαρχη αυτή του blockchain για την δημιουργία μίας πρωτοποριακής και χρήσιμης αποκεντρωμένης εφαρμογής. Η εφαρμογή ονομάζεται «Academic Certificates» και υλοποιήθηκε με σκοπό την διασφάλιση εγκυρότητας μιας συναλλαγής καταγράφοντας την όχι μόνο σε ένα βασικό μητρώο αλλά σε ένα συνδεδεμένο και διανεμημένο σύστημα μητρώων, τα οποία συνδέονται μέσω ενός ασφαλούς μηχανισμού ελέγχου εγκυρότητας. Εξετάστηκαν πολλές διαφορετικές τεχνολογίες υλοποίησης blockchain, ώστε να βρεθεί η πιο κατάλληλη και τελικά, επιλέχθηκε η αποκεντρωμένη πλατφόρμα του Ethereum. Μετά από αξιολόγηση των διαφορετικών blockchain, κρίθηκε ότι είναι η καταλληλότερη τεχνολογία για τον σκοπό μας. Αυτό διότι στο Ethereum προσφέρονται τα περισσότερα και πιο πλήρη εργαλεία για την ανάπτυξη και λειτουργία αποκεντρωμένων εφαρμογών.

Η παρούσα διπλωματική εργασία λοιπόν, αποτελεί μία προσπάθεια υλοποίησης βασιζόμενη στην τεχνολογίας του blockchain η οποία τεχνολογία είναι πολλά υποσχόμενη για το μέλλον και μπορεί δημιουργήσει πολλές νέες ευκαιρίες. Τελικό προϊόν της προσπάθειας αυτής, είναι η εφαρμογή «Academic Certificates».

8. Βιβλιογραφία

[1] «What is the Web3? The Decentralized Web - Blockchain,» Available: https://blockchainhub.net/web3-decentralized-web/.

[2] S. Nakamoto, «Bitcoin: A Peer-to-Peer Electronic Cash System,» 2008. Available: https://bitcoin.org/bitcoin.pdf.

[3] «What is Ethereum?,» Available: http://www.ethdocs.org/en/latest/introduction/what-is-ethereum.html.

[4] «devp2p/rlpx.md,» Available: https://github.com/ethereum/devp2p/blob/master/rlpx.md.

[5] K. Christidis και M. Devetsikiotis (2016), «Blockchains and Smart Contracts for the Internet of Things,». Available: http://ieeexplore.ieee.org/document/7467408/.

[6] Wikipedia, Blockchain, online at https://en.wikipedia.org/wiki/Blockchain.

[7] Κωνσταντίνου Λογαρά (2018) Η Τεχνολογία Blockchain, οι εφαρμογές και οι νομικές πτυχές της https://www.lawspot.gr/nomika-nea/h-tehnologia-blockchain-oi-efarmoges-kai-oi-nomikes-ptyhes-tis

[8] Java Object Oriented Language, online at https://en.wikipedia.org/wiki/Java_(programming_language).

[9] Java Api – Oracle Docs, online at https://docs.oracle.com/javase/7/docs/api/

[10] Visual Studio IDE, Code Editor, Azure DevOps, & App Center – Visual Studio https://visualstudio.microsoft.com/

[11] Python, general-purpose programming language. https://en.wikipedia.org/wiki/Python_(programming_language)

[12] GitHub - ethereum/pyethereum: Next generation cryptocurrency network https://github.com/ethereum/pyethereum

[13] Simple And Secured Ethereum Transactions https://pdfs.semanticscholar.org/277e/9d799c38a16a3b2b1b4ce61cd790d46377b6.pdf

[14] A Beginner's Guide to Blockchain Technology (2013) https://www.coindesk.com/information

[15] VirtualBox for Linux Hosts https://www.virtualbox.org/wiki/Linux_Downloads

[16] Blockcerts (2016) https://github.com/blockchain-certificates

[17] Giannis Konstantinidis (2019) - Analysis, Design and Implementation of an Open-Source Web-Based System for GDPR Compliance Assessment

[18] Ζούκας Ευάγγελος, Μπαριαμπάς Λιντιόν Θωμάς (2017) - Ανάπτυξη Εφαρμογής Έξυπνου Τηλεφώνου για την Χαρτογράφηση και Πλοήγηση στα μονοπάτια της Σάμου

[19] Penna Sparrow (2012) - Peer-to-Peer (P2P) : Advantages and Disadvantages https://www.ianswer4u.com/2011/05/peer-to-peer-network-p2p-advantages-and.html

[20] Γεώργιος Ν. Παπαδόδημας (2018), «Ανάπτυξη Έξυπνων Συμβολαίων στο Blockchain και εφαρμογή στο IoT» http://artemis.cslab.ece.ntua.gr:8080/jspui/bitstream/123456789/13689/1/DT2018-0017.pdf

[21] Philipp Schmidt, Director of Learning Innovation, MIT Media Lab (2014) - Blockcerts - An Open Infrastructure for Academic Credentials on the Blockchain https://medium.com/mit-media-lab/blockcerts-an-open-infrastructure-for-academic-credentials-on-the-blockchain-899a6b880b2f

[22] Rujia Li, Yifan Wu, IT Innovation Interns (2017) - Blockchain based Academic Certificate Authentication System Overview https://intranet.birmingham.ac.uk/it/innovation/documents/public/Experiments/Blockchain-based-Academic-Certificate-Authentication-System-Overview.pdf

ΠΑΡΑΡΤΗΜΑ

Κατασκευή Εφαρμογής

Στο κεφάλαιο αυτό βρίσκονται τμήματα κώδικα που προσθέσαμε στον open-source κώδικα που χρησιμοποιήσαμε για τις ανάγκες της διπλωματικής. Ο κώδικας που αφορά τις λειτουργίες της έκδοσης, υπογραφής και αυθεντικοποίησης των πιστοποιητικών βρίσκεται στα παρακάτω links.

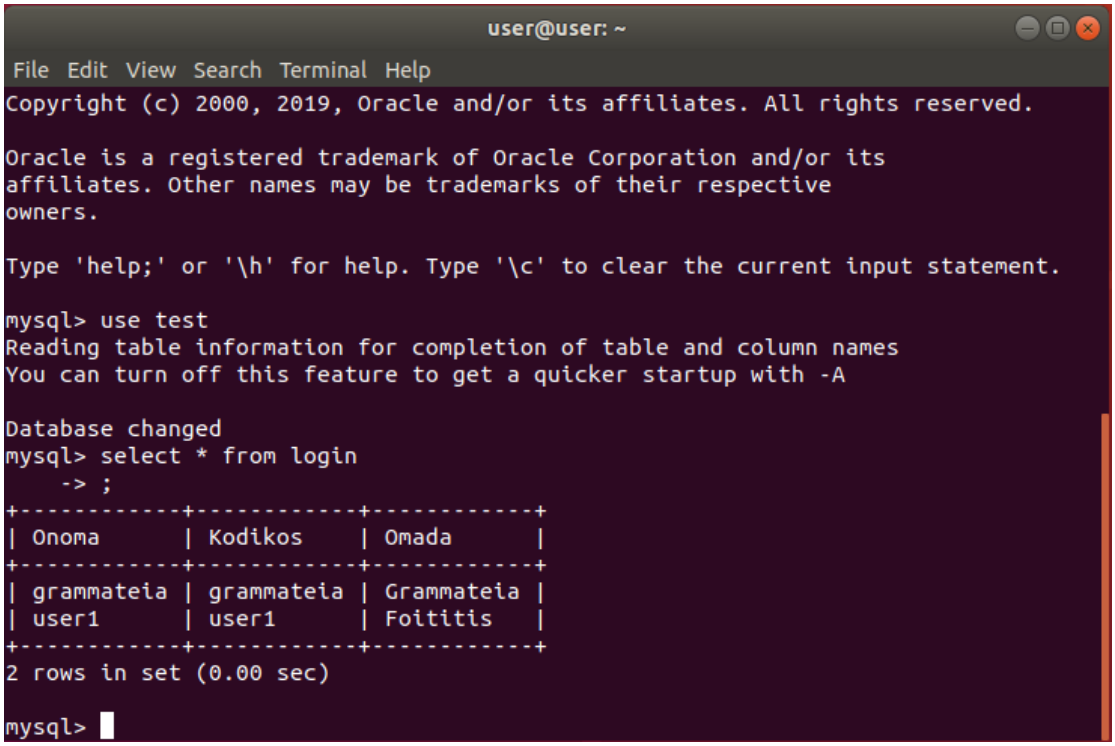
https://github.com/blockchain-certificates/cert-tools

https://github.com/blockchain-certificates/cert-issuer

https://github.com/blockchain-certificates/cert-verifier

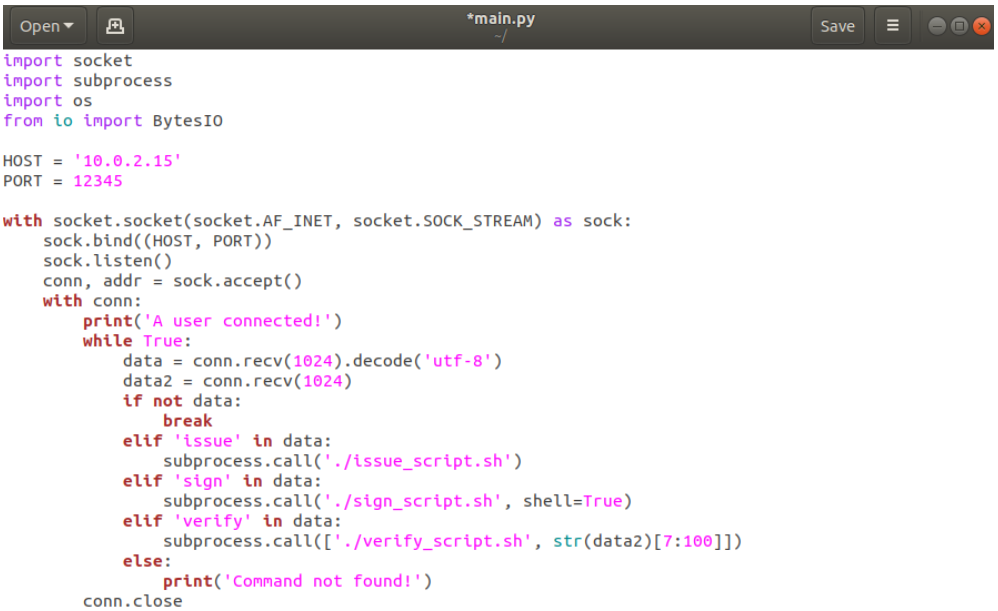
Backend

Οι βασικές λειτουργίες του backend της εφαρμογής βασίζεται σε open-source κώδικα και σε τμήματα κώδικα για την δημιουργία της βάσης δεδομένων που χρησιμοποιήσαμε (εικονικό σύστημα ενημέρωσης φοιτητών). Παρακάτω φαίνεται ένα παράδειγμα της βάσης.



Εικόνα 1 Βάση Δεδομένων.

Παρακάτω φαίνεται ο κώδικας σύνδεσης μεταξύ του frontend και backend με χρήση τεχνολογίας socket τα οποία έχουν υλοποιηθεί σε γλώσσα προγραμματισμού Java και Python αντίστοιχα.



Εικόνα 2 Δημιουργία socket.

Στην συνέχεια απεικονίζεται ο κώδικας που αφορά την εκτέλεση των τριών λειτουργιών της εφαρμογής μέσω των κουμπιών που υπάρχουν στο frontend.

Open Save issue_script.sh ~/

```
#!/bin/bash

echo "Issuing certificate..."

cd

cd cert-tools

create-certificate-template -c conf.ini

instantiate-certificate-batch -c conf.ini

sudo cp -a /home/user/cert-tools/sample_data/unsigned_certificates/. /home/user/cert-issuer/data/
unsigned_certificates
```

Εικόνα 3 Issue script.

Open Save sign_script.sh ~/

```
#!/bin/bash

echo "Signing certificate..."

cd

cd cert-issuer

sudo -H cert-issuer -c conf.ini

#sudo cp -a /home/user/cert-issuer/data/blockchain_certificates/. /home/user/cert-issuer/cert-verifier
```

Εικόνα 4 Sign script.

Open Save verify_script.sh ~/

```
#!/bin/bash

echo "Verifying certificate..."

cd

python3 cert-issuer/cert-verifier/cert_verifier/verifier.py "$1"
```

Εικόνα 5 Verify script.

Frontend

Η σχεδίαση του frontend της εφαρμογής έγινε με χρήση γλώσσας προγραμματισμού Java. Παρακάτω φαίνεται ο κώδικας της αρχικής οθόνης της εφαρμογής (login).

```java
package singinapp;

import java.io.DataOutputStream;
import java.io.IOException;
import java.net.Socket;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

public class UserAuthentication extends javax.swing.JFrame {

    public UserAuthentication() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel2Name = new javax.swing.JLabel();
        jLabel3Password = new javax.swing.JLabel();
        jTextField1Onoma = new javax.swing.JTextField();
        jPasswordField1 = new javax.swing.JPasswordField();
        jButton1 = new javax.swing.JButton();
        jLabel4WrongMsg = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Academic Certificates");

        jLabel2Name.setFont(new java.awt.Font("Dialog", 1, 14)); // NOI18N
        jLabel2Name.setText("Username:");

        jLabel3Password.setFont(new java.awt.Font("Dialog", 1, 14)); // NOI18N
        jLabel3Password.setText("Password:");

        jTextField1Onoma.setFont(new java.awt.Font("Dialog", 1, 14)); // NOI18N

        jPasswordField1.setFont(new java.awt.Font("Dialog", 1, 14)); // NOI18N

        jButton1.setFont(new java.awt.Font("Dialog", 1, 14)); // NOI18N
        jButton1.setText("Log in");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jLabel4WrongMsg.setFont(new java.awt.Font("Dialog", 1, 14)); // NOI18N

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(43, 43, 43)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
                    .addComponent(jLabel4WrongMsg, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addGroup(layout.createSequentialGroup()
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(jLabel2Name)
                            .addComponent(jLabel3Password))
                        .addGap(50, 50, 50)
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                            .addComponent(jButton1)
                            .addComponent(jTextField1Onoma)
                            .addComponent(jPasswordField1, javax.swing.GroupLayout.PREFERRED_SIZE, 164, javax.swing.GroupLayout.PREFERRED_SIZE))))
                .addContainerGap(68, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(45, 45, 45)
                .addComponent(jLabel4WrongMsg, javax.swing.GroupLayout.PREFERRED_SIZE, 20, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel2Name)
                    .addComponent(jTextField1Onoma, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel3Password)
                    .addComponent(jPasswordField1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)
                .addComponent(jButton1)
                .addContainerGap(63, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con = (Connection) DriverManager.getConnection("jdbc:mysql://localhost:3306/test", "root", "");
            String query = "select  Onoma, Kodikos from login where Onoma = '" + jTextField1Onoma.getText() + "'";
            PreparedStatement statement = con.prepareStatement(query);
            ResultSet result = statement.executeQuery();
            if (result.next()) {
                String dbasePassword = result.getString("Kodikos").toString().trim();
                String enteredPassword = new String(jPasswordField1.getText().trim());
                if (dbasePassword.equals(enteredPassword)) {
                    String query2 = "select Omada from login where Onoma = '" + jTextField1Onoma.getText() + "'";
                    PreparedStatement statement2 = con.prepareStatement(query2);
                    ResultSet result2 = statement2.executeQuery();
                    if (result2.next()) {
                        String dbase0mada = result2.getString("Omada").toString().trim();
                        if (dbase0mada.equals("Grammateia")) {
                            this.setVisible(false);
                            jLabel4WrongMsg.setText("Welcome!");
                            Grammateia_gui g = new Grammateia_gui();
                            g.setVisible(true);
                        } else {
                            this.setVisible(false);
                            jLabel4WrongMsg.setText("Welcome!");
                            Student_gui s = new Student_gui();
                            s.setVisible(true);
                        }
                    }
                } else {
                    jLabel4WrongMsg.setText("Wrong password!");
                    jPasswordField1.setText("");
                }
            } else {
                jLabel4WrongMsg.setText("Wrong username!");
                jPasswordField1.setText("");
            }
            statement.close();
            con.close();
        } catch (SQLException se) {
            se.printStackTrace();
        } catch (Exception e) {
            e.printStackTrace();
            jLabel4WrongMsg.setText("Exception occurred while searching in the login table");
        }
    }

    public static void main(String args[]) {
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(UserAuthentication.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(UserAuthentication.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(UserAuthentication.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(UserAuthentication.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new UserAuthentication().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel2Name;
    private javax.swing.JLabel jLabel3Password;
    private javax.swing.JLabel jLabel4WrongMsg;
    private javax.swing.JPasswordField jPasswordField1;
    private javax.swing.JTextField jTextField1Onoma;
    // End of variables declaration
}
```

Εικόνα 6 User authentication GUI.

Στην συνέχεια ανάλογα με την ομάδα που ανήκει ο χρήστης που πραγματοποιεί την σύνδεση εμφανίζεται και το ανάλογο παράθυρο. Στην πρώτη εικόνα φαίνεται ο κώδικας στην περίπτωση που ο χρήστης ανήκει στην ομάδα της γραμματείας και στην δεύτερη όταν ο χρήστης ανήκει στην ομάδα των φοιτητών.

```
package singinapp;

import java.io.DataOutputStream;
import java.io.IOException;
import java.net.Socket;
import java.util.logging.Level;
import java.util.logging.Logger;

public class Grammateia_gui extends javax.swing.JFrame {

    private static DataOutputStream dos;
    /**
     * Creates new form Grammateia_gui
     */
    public Grammateia_gui() throws IOException {
        initComponents();
        Socket sock = new Socket("10.0.2.15", 12345);
        dos = new DataOutputStream(sock.getOutputStream());
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Menu");

        jButton1.setText("Issue Certificate");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jButton2.setText("Sign Certificate");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
                .addContainerGap(91, Short.MAX_VALUE)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                    .addComponent(jButton1)
                    .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 124, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(90, 90, 90))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(30, 30, 30)
                .addComponent(jButton1)
                .addGap(18, 18, 18)
                .addComponent(jButton2)
                .addContainerGap(32, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            dos.writeUTF("issue");
        } catch (IOException ex) {
            Logger.getLogger(Grammateia_gui.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            dos.writeUTF("sign");
        } catch (IOException ex) {
            Logger.getLogger(Grammateia_gui.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    public static void main(String args[]) {
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(Grammateia_gui.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(Grammateia_gui.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(Grammateia_gui.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(Grammateia_gui.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    new Grammateia_gui().setVisible(true);
                } catch (IOException ex) {
                    Logger.getLogger(Grammateia_gui.class.getName()).log(Level.SEVERE, null, ex);
                }
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    // End of variables declaration
}
```

Εικόνα 7 Grammateia GUI.

```java
package singinapp;

import java.io.BufferedInputStream;

import java.io.DataOutputStream;
import java.io.File;
import java.io.FileInputStream;

import java.io.IOException;

import java.io.OutputStream;

import java.net.Socket;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFileChooser;

public class Student_gui extends javax.swing.JFrame {

    private static DataOutputStream dos;
    FileInputStream fis = null;
    BufferedInputStream bis = null;
    OutputStream os = null;
    Socket sock = null;
    String filename;

    /**
     * Creates new form Student_gui
     */
    public Student_gui() throws IOException {
        initComponents();
        sock = new Socket("10.0.2.15", 12345);
        dos = new DataOutputStream(sock.getOutputStream());
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jTextField1 = new javax.swing.JTextField();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Choose file for verification");

        jButton1.setText("Browse");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jButton2.setText("Verify");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(42, 42, 42)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 84, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jTextField1, javax.swing.GroupLayout.DEFAULT_SIZE, 225, Short.MAX_VALUE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jButton1)
                        .addGap(41, 41, 41))))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(33, 33, 33)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jButton1))
                .addGap(18, 18, 18)
                .addComponent(jButton2)
                .addContainerGap(31, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        JFileChooser chooser = new JFileChooser();
        chooser.showOpenDialog(null);
        File f = chooser.getSelectedFile();
        filename = f.getAbsolutePath();
        jTextField1.setText(filename);
        System.out.println(filename);
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            dos.writeUTF("verify");
            dos.writeUTF(filename);
        } catch (IOException ex) {
            Logger.getLogger(Grammateia_gui.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    public static void main(String args[]) {
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(Student_gui.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(Student_gui.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(Student_gui.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(Student_gui.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    new Student_gui().setVisible(true);
                } catch (IOException ex) {
                    Logger.getLogger(Student_gui.class.getName()).log(Level.SEVERE, null, ex);
                }
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JTextField jTextField1;
    // End of variables declaration
}
```

Εικόνα 8 Student GUI.