ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Προπτυχιακό Πρόγραμμα Σπουδών στην Πληροφορική

- Πτυχιακή Εργασία -

Peer-to-peer συναλλαγές μέσω της τεχνολογίας Blockchain

Σπυρίδων Γκέοργκ Κούστας
Α.Μ.: Π2014027

Επιβλέπων: Παναγιώτης Κουρουθανάσης

Κέρκυρα 2018

Επιβλέπων

Παναγιώτης Κουρουθανάσης, Αναπληρωτής Καθηγητής
Ιόνιο Πανεπιστήμιο

Τριμελής Επιτροπή

Παναγιώτης Κουρουθανάσης, Αναπληρωτής Καθηγητής
Ιόνιο Πανεπιστήμιο

Αδαμαντία Πατέλη, Αναπληρώτρια Καθηγήτρια
Ιόνιο Πανεπιστήμιο

Σπυρίδων Σιούτας, Αναπληρωτής Καθηγητής
Ιόνιο Πανεπιστήμιο

Πρόλογος

Σκοπός της παρούσας πτυχιακής εργασίας είναι η εξέταση των εφαρμογών Peer-to-peer lending που κάνουν χρήση της τεχνολογίας του Blockchain. Το Blockchain είναι μια αποκεντρωμένη και κατανεμημένη δομή δεδομένων, ένα αποθηκευτικό μέσο.

Η καινοτομία της τεχνολογίας Blockchain δίνει λύση στο ζήτημα που αφορά την συγχρονισμένη καταγραφή δεδομένων σε ένα κατανεμημένο δίκτυο από ανεξάρτητους και άγνωστους μεταξύ τους κόμβους. Όλοι οι υπολογιστές που συμμετέχουν σε ένα δίκτυο Blockchain έχουν ακριβώς τα ίδια δεδομένα χωρίς να μπορούν να τα επεξεργαστούν. Η ασφάλεια των δεδομένων αυτών επιτυγχάνεται μέσω κρυπτογραφικών συναρτήσεων.

Έχοντας γνώση το πως λειτουργεί το κρυπτονόμισμα Bitcoin και οι ήδη υπάρχουσες πλατφόρμες peer-to-peer lending, επιχειρήθηκε η δημιουργία μιας αποκεντρωμένης κατανεμημένης εφαρμογής η οποία θα προσέφερε στους χρήστες τη δυνατότητα συναλλαγών τόσο κρυπτονομισματικών, όσο και χρηματικών μέσω του πρωτοκόλλου ISO 20022. Η επικοινωνία επιτυγχάνεται σε εξατομικευμένα ανεπτυγμένο ιδιωτικό δίκτυο Blockchain. Με άλλα λόγια δημιουργήθηκε μια διαδικτυακή πλατφόρμα συναλλαγών και ένα προσωπικό ηλεκτρονικό πορτοφόλι για να ανταλλάσσονται τόσο κρυπτονομίσματα, όσο και χρηματικά ποσά. Όλες οι συναλλαγές αποθηκεύονται σε ένα ιδιωτικό Blockchain το οποίο αναπτύχθηκε επίσης στα πλαίσια αυτής της εργασίας.

Εισαγωγή

Με αφορμή τα κρυπτονομίσματα bitcoin και ether ξεκίνησε μια μεγάλη τεχνολογική "επανάσταση" στον κόσμο της τεχνολογίας, η οποία βασίζεται στη χρήση την τεχνολογίας του Blockchain. Τα Blockchain είναι ένα από τα πιο πολυσυζητημένα τεχνολογικά επιτεύγματα με επενδύσεις δισεκατομμυρίων ευρώ και μία μεγάλη βιομηχανία να κτίζεται και να αναπτύσσεται πάνω στις βάσεις της. Κάποια από τα κύρια χαρακτηριστικά του Blockchain που το καθιστούν τόσο ιδιαίτερο, είναι ο αποκεντρωμένος σχεδιασμός και το γεγονός ότι δεν απαιτείται εμπιστοσύνη μεταξύ το χρηστών. Τα δεδομένα που ανταλλάσσονται δεν βρίσκονται σε κεντρικό διακομιστή (server), αλλα σε κατανεμημένο δίκτυο από συνεργαζόμενους κόμβους.

Σκοπός της παρούσας πτυχιακής εργασίας είναι η εξέταση των εφαρμογών Peer-to-peer lending που κάνουν χρήση της τεχνολογίας του Blockchain.

Πιο αναλυτικά, στα πλαίσια της πτυχιακής εργασίας σχεδιάστηκαν και υλοποιήθηκαν τα εξής:

1. Πλατφόρμα Peer-to-peer Lending όπου οι χρήστες μπορούν να ζητούν δάνεια από άλλους χρήστες ή να μεταφέρουν χρήματα και Obolus κρυπτονομίσματα.
2. Ιδιωτικό πορτοφόλι (Wallet) το οποίο χρησιμοποιείται για τις συναλλαγές που αφορούν το ιδιωτικό Blockchain.
3. Το Κρυπτονόμισμα Obolus, το οποίο χρησιμοποιείται για συναλλαγές εντός της πλατφόρμας αλλα και μεταξύ χρηστών
4. Αλγόριθμος ο οποίος επικοινωνεί με το ιδιωτικό δίκτυο Blockchain που αναπτύχθηκε για να λαμβάνει πληροφορίες σχετικά με τις συναλλαγές των χρηστών.

Σχετικά με τη δομή της εργασίας, στο πρώτο κεφάλαιο επιχειρήθηκε η αποσαφήνιση ορισμένων εννοιών που χρησιμοποιούνται ευρέως στην παρούσα εργασία προκειμένου να καταστεί κατανοητό το περιεχόμενό της. Επίσης αναλύθηκε η παρούσα κατάσταση των peer-to-peer lending εφαρμογών και γενικότερα των fintech επιχειρήσεων στην Ευρώπη και την Αμερική.

Στο δεύτερο κεφάλαιο αναγράφονται κάποιοι βασικοί κανόνες τους οποίους ορίζει η βιβλιογραφία σχετικά με την ανάπτυξη peer-to-peer εφαρμογών που βασίζονται στην τεχνολογία Blockchain. Επίσης εξετάζεται η τεχνολογία Blockchain, οι τύποι, η αρχιτεκτονική της αλλά και οι μηχανισμοί που είναι απαραίτητοι προκειμένου να διατηρηθεί η εμπιστοσύνη σε ένα Blockchain δίκτυο.

Στο τρίτο κεφάλαιο παρουσιάζεται η εφαρμογή που επιχειρήθηκε να αναπτυχθεί στα πλαίσια της παρούσας πτυχιακής εργασίας καθώς και το τρόπος ανάπτυξης της. Αναφέρονται τα εργαλεία που χρησιμοποιήθηκαν, οι τεχνικές δυσκολίες που αντιμετωπίστηκαν και ευκαιρίες για βελτίωση. Επίσης παρουσιάζεται το διάγραμμα δραστηριότητας που εξηγεί την περίπτωση αίτησης συναλλαγής κάποιου χρήστη μέσω της εφαρμογής που αναπτύχθηκε.

Έπειτα στο τέταρτο κεφάλαιο παρουσιάζεται ένα εγχειρίδιο χρήσης της πλατφόρμας που αναπτύχθηκε και πως η εφαρμογή συνδέεται με τις προδιαγραφές που θέτει η βιβλιογραφία. Τέλος, διατυπώνονται κάποιες συμπερασματικές σκέψεις για την τεχνολογία Blockchain και τις peer-to-peer lending εφαρμογές καθώς παρατίθενται και οι βιβλιογραφικές πηγές που αξιοποιήθηκαν.

1. Peer-to-peer Συναλλαγές

Ο θεσμός του Peer-to-peer lending (P2P lending) παρουσιάζει σημαντική ανάπτυξη στην Ευρώπη και στις ΗΠΑ τα τελευταία χρόνια, στο πλαίσιο του οποίου ιδιώτες αλλά και επιχειρήσεις εξασφαλίζουν δανεισμό απευθυνόμενοι όχι στα τραπεζικά σχήματα, αλλά απευθείας σε ιδιώτες επενδυτές, που συμβάλλουν σε μία εναλλακτική οικονομία.

Στην πράξη, το peer-to-peer lending λειτουργεί μέσω ελεγχόμενων διαδικτυακών εφαρμογών, στις οποίες δημιουργείται μια αγορά και η προσφορά συναντά τη ζήτηση.

Το P2P lending δουλεύει απλά. Άτομα που θέλουν να επενδύσουν τα χρήματά τους, τα δανείζονται σε επιχειρήσεις και ιδιώτες που τα έχουν ανάγκη μέσω μια διαδικτυακής εφαρμογής που τους φέρνει σε επαφή. Ο δανειστής κερδίζει χρήματα από το επιτόκιο του δανείου.

Οι τιμές των επιτοκίων έχουν συμφωνηθεί από την αρχή μεταξύ των μερών της συναλλαγής. Η διαδικτυακή εφαρμογή λαμβάνει προμήθεια από το επιτόκιο. Ο δανειολήπτης λαμβάνει το δάνειό του και αποπληρώνει τα ποσά που δανείστηκε, συν τους τόκους, σε προκαθορισμένες ημερομηνίες.

Παρότι οι διαδικασίες δεν είναι απολύτως ίδιες ανάμεσα στις διάφορες πλατφόρμες, βασικό στοιχείο τους είναι η ύπαρξη ενός διαμεσολαβητή. Του ιδιοκτήτη συνήθως της πλατφόρμας, που ελέγχει την πιστοληπτική ικανότητα των υποψηφίων δανειοληπτών.

Το όφελος για αμφότερες τις πλευρές είναι σημαντικό. Οι δανειστές εξασφαλίζουν κατά κανόνα υψηλότερα επιτόκια από αυτά που προσφέρουν οι τράπεζες. Οι δανειζόμενοι είτε διασφαλίζουν χαμηλότερο επιτόκιο σε σχέση με αυτά που προσφέρουν οι τράπεζες, είτε βρίσκουν χρήματα που δεν θα μπορούσαν να βρουν διαφορετικά.

Η ανάπτυξη αυτού του θεσμού (γνωστό και ως Crowdfunding) υπήρξε τεράστια τα τελευταία χρόνια, με κέρδη που αυξάνονται θεαματικά.

Το Peer-to-peer lending δεν εντάσσεται ξεκάθαρα σε κάποια από τις έως τώρα γνωστές κατηγορίες οικονομικών συναλλαγών, έτσι συχνά χαρακτηρίζεται ως εναλλακτικός τρόπος χρηματοδότησης.

Οι περισσότερες πλατφόρμες Peer-to-peer lending προσφέρουν τις εξής υπηρεσίες:

- Διαδικτυακή πλατφορμα που επιτρέπει σε δανειστές και δανειολήπτες να έρχονται σε επικοινωνία.
- Επικύρωση στοιχείων δανειστών, τραπεζικών λογαριασμών και έγκριση δανείων.
- Εκτέλεση τακτικών ελέγχων δανειστών και φιλτράρισμα ανεπιθύμητων χρηστών.
- Νομοθετική ενημέρωση χρηστών.

1.1 Peer-to-peer lending σε Ευρώπη και Αμερική

1.1.1 Ηνωμένο Βασίλειο

Η πρώτη επιχείρηση που προσέφερε peer-to-peer συναλλαγές ήταν η Zopa. Από την ίδρυσή της το Φλεβάρη το 2005 η Zopa έχει συνάψει δάνεια πάνω από £1,99 δισεκατομμύρια. Το 2010 μια ακόμη εταιρεία έκανε της εμφάνισή της στις Peer-to-peer συναλλαγές, η Funding Circle, η οποία εξειδικεύεται στο να προσφέρει μικρά επιχειρησιακά δάνεια, μέσω της πλατφόρμας της. Έως τον Ιανουάριο του 2017, η Funding Circle είχε συνάψει δάνεια πάνω από £1,93 δισεκατομμύρια. Τόσο η Zopa, όσο και η Funding Circle είναι μέλη της Peer-to-peer Finance Association (P2PFA), δηλαδή του ομίλου peer-to-peer οργανισμών του Ηνωμένου Βασιλείου. Το 2012, η κυβέρνηση του Ηνωμένου Βασιλείου επένδυσε £20 εκατομμύρια σε Βρετανικές επιχειρήσεις μέσω peer-to-peer συναλλαγών. Μια δεύτερη επένδυση ύψους £40 εκατομμυρίων ανακοινώθηκε το 2014. Σκοπός των επενδύσεων αυτών ήταν η παράκαμψη των μεγάλων τραπεζών, οι οποίες δεν ήταν και πρόθυμες να δανείσουν σε μικρές επιχειρήσεις.

1.1.2 Ηνωμένες Πολιτείες

Στην Αμερική, οι επιχειρηματικές δραστηριότητες στον τομέα του peer-to-peer lending ξεκίνησαν το Φλεβάρη του 2006, με τις εταιρείες Prosper και Lending Club. Και οι δύο εταιρείες εδρεύουν στο Σαν Φρανσίσκο στην Καλιφόρνια.
Το 2008, η Επιτροπή Κεφαλαιαγοράς (Securities and Exchange Commission - SEC) απαίτησε από τις εταιρείες peer-to-peer να καταχωρήσουν τις υπηρεσίες τους ως χρεόγραφα, σύμφωνα με τον νόμο περί κινητών αξιών του 1933. Οι όμιλοι Prosper και Lending Club αναγκάστηκαν να αναστείλουν προσωρινά την έγκριση νέων δανείων, ενώ άλλοι, όπως η Zopa Ltd. με έδρα την Ουκρανία, αποχώρησαν εξ ολοκλήρου από την αγορά των ΗΠΑ. Τόσο η Lending Club όσο και η Prosper εγκρίθηκαν από την Επιτροπή Κεφαλαιαγοράς και σχημάτισαν εταιρικές σχέσεις με την FOLIO Investing για να δημιουργήσουν μια δευτερογενή αγορά για τα νομίσματά τους, παρέχοντας έτσι ρευστότητα στους επενδυτές. Έτσι λύθηκε και το πρόβλημα της διαφάνειας των δανείων. Οι Αγοραστές πλέον έχουν πρόσβαση στις λεπτομερείς πληροφορίες για κάθε μεμονωμένο δάνειο πριν ακόμη επενδύσουν σε αυτά. Οι peer-to-peer επιχειρήσεις πρέπει επίσης να αναφέρουν λεπτομερώς τις προσφορές τους σε ενημερωτικό δελτίο και η SEC εκδίδει τις εκθέσεις στο κοινό μέσω του συστήματος EDGAR (Electronic Data-Gathering, Analysis, and Retrieval).
Όλο και περισσότεροι άνθρωποι στράφηκαν σε εταιρείες peer-to-peer lending μετά τη χρηματοπιστωτική κρίση στα τέλη της δεκαετίας του 2000, επειδή οι τράπεζες αρνήθηκαν να αυξήσουν τα χαρτοφυλάκια δανείων τους.
Έως το 2012, η Lending Club ήταν η μεγαλύτερη εταιρεία peer-to-peer lending τις Ηνωμένες Πολιτείες βάσει των όγκων των δανείων. Δεύτερη ήταν η Prosper. Οι δύο εταιρείες έχουν εγκρίνει πάνω από 180,000 δάνεια συνολικού ύψους πάνω από $2 δισεκατομμυρίων δολαρίων. Έως το Μάρτη του 2012 η Lending Club είχε εγκρίνει 117,412 δάνεια ύψους $1,512,560,075 ενώ η Prosper 63,023 δάνεια ύψους $433,570,651 δολαρίων. Πράγμα που σηματοδότησε ετήσια αύξηση μεγαλύτερη του 100%, ορίζοντας έτσι το peer-to-peer lending μία από τις πιο γρήγορα αναπτυσσόμενες επενδύσεις. Τα επιτόκια κυμαίνονται από 5,6% έως και 35,8%, ανάλογα με τις αξιολογήσεις του επενδυτή και του δανείου.

1.2 Νομοθετικό πλαίσιο

Σε πολλές χώρες, η προσέλκυση επενδυτών από το ευρύ κοινό θεωρείται παράνομη. Crowdsourcing συμφωνίες, στις οποίες καλούνται χρήστες να συνεισφέρουν χρήματα σε αντάλλαγμα με πιθανά κέρδη, που βασίζονται στο έργο άλλων, θεωρούνται χρεόγραφα. Η αντιμετώπιση των χρηματοοικονομικών εγγυήσεων συνδέεται με το πρόβλημα της κυριότητας. Στην περίπτωση δανείων peer-to-peer, εμφανίζεται το πρόβλημα του ποιος κατέχει τα δάνεια και πώς μεταβιβάζεται η κυριότητα μεταξύ του δανειολήπτη και του μεμονωμένου δανειστή. Το ερώτημα αυτό τίθεται κυρίως όταν μια εταιρεία δανειοδότησης από ομοτίμους δεν συνδέει μόνο τους δανειστές και τους δανειολήπτες, αλλά δανείζεται χρήματα από τους χρήστες και στη συνέχεια το δανείζει ξανά. Η δραστηριότητα αυτή ερμηνεύεται ως πώληση χρεογράφων και απαιτείται άδεια διαμεσολάβησης, αντιπροσώπου και συμβόλαιο επένδυσης για τη νόμιμη διαδικασία. Η άδεια και η εγγραφή μπορούν να ληφθούν από έναν ρυθμιστικό φορέα, όπως η Επιτροπή Κεφαλαιαγοράς των Η.Π.Α. (SEC) στις ΗΠΑ, η Επιτροπή Κεφαλαιαγοράς του Οντάριο στο Οντάριο του Καναδά, οι χρηματοπιστωτικοί οργανισμοί της Γαλλίας και του Κεμπέκ του Καναδά ή Αρχή Χρηματοπιστωτικών Υπηρεσιών στο Ηνωμένο Βασίλειο.

1.2.1 Νομοθετικό πλαίσιο στην Ελλάδα

Αρμόδιες αρχές για την αδειοδότηση και την εποπτεία πιστωτικών ιδρυμάτων, ιδρυμάτων πληρωμών και ιδρυμάτων ηλεκτρονικού χρήματος είναι η Τράπεζα της Ελλάδος και η ΕΚΤ. Συγκεκριμένα, οι προϋποθέσεις για την παροχή άδειας λειτουργίας καθώς και οι κανόνες εποπτείας που διέπουν τη λειτουργία των Ιδρυμάτων Ηλεκτρονικού Χρήματος καθορίζονται από τον ν. 3862/2010, τον ν. 4021/2011,21 την ΠΕΕ 33/19.12.2013 και την ΠΕΕ 22/12.7.2013. Σύμφωνα με το άρθρο 12, παρ. 1 του ν. 4021/2011, η Τράπεζα της Ελλάδος είναι αρμόδια για την αδειοδότηση και την προληπτική εποπτεία Ιδρυμάτων Ηλεκτρονικού Χρήματος. Δυνάμει της εν λόγω εξουσιοδότησης, η Τράπεζα της Ελλάδος εξέδωσε την ΠΕΕ 33/19.12.2013. Για τη χορήγηση άδειας λειτουργίας Ιδρύματος Ηλεκτρονικού Χρήματος απαιτείται αρχικό κεφάλαιο ποσού τουλάχιστον τριακοσίων πενήντα χιλιάδων (350.000) ευρώ. Περαιτέρω, με τον εν λόγω νόμο καθορίζονται οι κεφαλαιακές απαιτήσεις και ο τρόπος υπολογισμού των ιδίων κεφαλαίων τους, οι υποχρεώσεις διασφάλισης του ενεργητικού, τα εξαιρούμενα ιδρύματα και οι επιτρεπόμενες δραστηριότητες που μπορούν τα ιδρύματα αυτά να ασκούν εκτός της έκδοσης ηλεκτρονικού χρήματος, τόσο εντός Ελλάδος όσο και διασυνοριακά. Θέματα που αφορούν την ενημέρωση από τα Ιδρύματα Ηλεκτρονικού Χρήματος των κατόχων ηλεκτρονικού χρήματος για τα δικαιώματά τους, καθώς και τη διερεύνηση καταγγελιών των τελευταίων δεν εμπίπτουν στην αρμοδιότητα της Τράπεζας της Ελλάδος, αλλά σε αυτή της Γενικής Γραμματείας Καταναλωτή, σύμφωνα με το άρθρο 25, παρ. 1 του ν. 4021/2011. Στο πλαίσιο της ενημέρωσης, η Τράπεζα της Ελλάδος διαθέτει στο δικτυακό της τόπο Πίνακες Μητρώου με όλα τα πιστωτικά ή χρηματοδοτικά ιδρύματα που παρέχουν τραπεζικές υπηρεσίες. Σύμφωνα με τους Πίνακες Μητρώου, το 2016 δραστηριοποιείται μόνο ένα ίδρυμα ηλεκτρονικού χρήματος και εννέα ιδρύματα πληρωμών. Στους εν λόγω πίνακες αναφέρονται και τα ιδρύματα πληρωμών και ηλεκτρονικού χρήματος με έδρα σε άλλα κράτη-μέλη που γνωστοποίησαν την πρόθεση παροχής υπηρεσιών στην Ελλάδα χωρίς εγκατάσταση.

1.2.2 Νομοθετικό πλαίσιο στην Γερμανία

Στη Γερμανία υπάρχει ένα ευρύ τοπίο σχετικά με τις fintech επιχειρήσεις. Μία πρόσφατη μελέτη του Υπουργείου Οικονομικών της Γερμανίας (Νοέμβρη 2016), δημοσίευσε πως 346 χρηματοοικονομικές επιχειρήσεις ήταν ενεργές και λειτουργούσαν στην Γερμανία. Κύριες αρμοδιότητες των επιχειρήσεων αυτών ήταν οι επενδύσεις και οι δανεισμοί χρημάτων (crowdinvestment, crowdlending/ P2P lending), καθώς και η διαχείριση περιουσιακών στοιχείων και η παροχή πληρωμών.

Το Υπουργείο Οικονομικών της Γερμανίας προσφέρει το πρόγραμμα παροχής "INVEST" στο οποίο επενδυτές αποζημιώνονται το 20% των επενδύσεων τους σε περίπτωση που έχουν επενδύσει €10.000 και άνω. Επίσης, υπάρχουν πολλά υποστηρικτικά προγράμματα τα οποία διαχειρίζονται συνήθως από την ομοσπονδιακή τράπεζα "Kreditanstalt fuer Wiederaufbau". Δεν υπάρχει κάποιο συγκεκριμένο νομοθετικό πλαίσιο σχετικά με τις fintech επιχειρήσεις. Ανάλογα με τον τομέα στον οποίο δραστηριοποιείται η κάθε επιχείρηση, συμμορφώνεται με τους αντίστοιχους κανόνες του οικονομικού δικαίου.
Ο Γερμανικός νόμος περί πιστωτικών ιδρυμάτων (KWG) καθορίζει τις απαραίτητες προϋποθέσεις για τη χορήγηση άδειας λειτουργίας Ιδρύματος Ηλεκτρονικού Χρήματος.

1.2.3 Νομοθετικό πλαίσιο το Ηνωμένο Βασίλειο

Στο Ηνωμένο Βασίλειο, τον Οκτώβρη του 2014, η Χρηματοοικονομική Αρχή FCA (Financial Conduct Authority) ξεκίνησε ένα project το οποίο είχε σκοπό την υποστήριξη νεοσύστατων και καινοτόμων επιχειρήσεων μέσω ενός κόμβου καινοτομίας (Innovation Hub). Το Μάη του 2016 η FCA ήταν η πρώτη ρυθμιστική αρχή που ξεκίνησε ένα sandbox περιβάλλον, επιτρέποντας έτσι σε νέες επιχειρήσεις να δοκιμάσουν νέες και καινοτόμες χρηματοοικονομικές υπηρεσίες, χωρίς να τους επιβαρύνει κάποιο νομοθετικό πλαίσιο
Τον Απρίλη του 2017 το υπουργείο οικονομικών της Αγγλίας (HM Treasury), εξέδωσε ένα ρυθμιστικό σχέδιο καινοτομίας για χρηματοοικονομικές υπηρεσίες. Το σχέδιο προβλέπει την συνεργασία διαφόρων αρχών και τραπεζών (FCA, the Prudential Regulation Authority, the Payment Systems Regulator and the Bank of England), με απώτερο σκοπό την υποστήριξη και προώθηση των καινοτόμων χρηματοοικονομικών επιχειρήσεων.

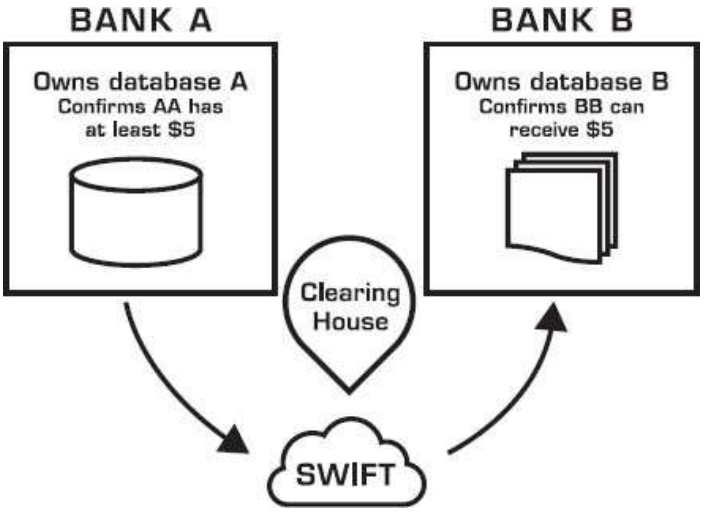
1.3 Ρίσκα του Peer-to-peer Lending

Το peer-to-peer lending προσελκύει δανειολήπτες οι οποίοι, εξαιτίας της πιστωτικής τους κατάστασης, είναι ακατάλληλοι για τα παραδοσιακά τραπεζικά δάνεια. Για αυτό οι μεσολαβητές του peer-to-peer lending άρχισαν να μειώνουν τον αριθμό υποψηφίων και να χρεώνουν υψηλότερα επιτόκια σε πιο "επικίνδυνους" δανειολήπτες που έχουν εγκριθεί. Ορισμένες εταιρείες μεσιτών δημιούργησαν επίσης δάνεια στα οποία οι δανειστές αποζημιώνονται εάν ο οφειλέτης δεν είναι σε θέση να εξοφλήσει το ποσό.
Αρχικά ένα από τα ελκυστικά χαρακτηριστικά του δανεισμού peer-to-peer για τους επενδυτές ήταν τα χαμηλά ποσοστά αθέτησης, π.χ. Το ποσοστό καθυστερήσεων του Prosper το 2007 ήταν μόλις 2,7%.
Επειδή, σε αντίθεση με τους καταθέτες στις τράπεζες, οι δανειστές peer-to-peer μπορούν να επιλέξουν οι ίδιοι εάν θα δανείσουν τα χρήματά τους σε "ασφαλέστερους" δανειολήπτες με χαμηλότερα επιτόκια ή σε πιο επικίνδυνους δανειολήπτες με υψηλότερες αποδόσεις, στον αμερικανικό όμιλο peer-to-peer lending αντιμετωπίζονται νόμιμα ως επενδύσεις και η αποπληρωμή σε περίπτωση αθέτησης του δανειολήπτη δεν διασφαλίζεται από την ομοσπονδιακή κυβέρνηση (U.S. Federal Deposit Insurance Corporation).

1.4 Γιατί εξετάζουμε τις Peer-to-peer συναλλαγές;

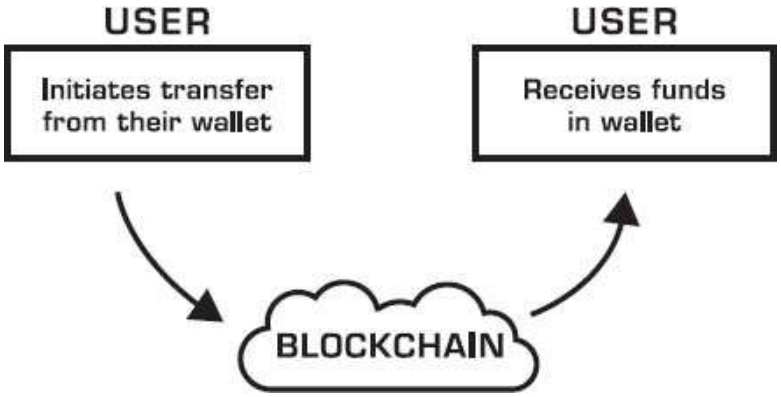
Οι αγορά για blockchain εφαρμογές είναι τεράστια και δεν περιορίζεται στον οικονομικό τομέα. Πλέον, έμπιστες συναλλαγές μεταξύ αγνώστων και σε μεγάλες αποστάσεις είναι πραγματικότητα και αυτό χάρη στο Bitcoin, το οποίο ήταν το πρώτο κρυπτονόμισμα που ανέδειξε τις ικανότητες μιας τέτοια peer-to-peer lending εφαρμογής.

Ας συγκρίνουμε λοιπόν ένα παραδοσιακό κεντροποιημένο τραπεζικό σύστημα, το οποίο χρησιμοποιεί μια συμβατική βάση δεδομένων, με ένα κατανεμημένο σύστημα που βασίζεται στην τεχνολογία των blockchains. Στο παραδοσιακό τραπεζικό σύστημα, κάθε φορά που κάποιος χρήστης επιθυμεί να κάνει μια συναλλαγή, απαιτείται η αυθεντικοποίηση των στοιχείων και η έγκρισή της τράπεζας, η οποία είναι και υπεύθυνη για τα στοιχεία που αποθηκεύονται στην κεντρική της βάση δεδομένων. Έπειτα η τράπεζα είτε εγκρίνει, είτε απορρίπτει την συναλλαγή των χρημάτων.



Εικόνα 1: Παραδοσιακός τρόπος τραπεζικών συναλλαγών - *William Mougayar, Vitalik Buterin-The Business Blockchain*

Η ίδια διαδικασία θα ήταν πολύ πιο εύκολη και γρήγορη, εάν είχε χρησιμοποιηθεί ένα σύστημα peer-to-peer συναλλαγών, βασισμένο στην τεχνολογία των blockchains. Οι χρήστες μπορούν να στέλνουν και να λαμβάνουν χρήματα ή value tokens μέσω των προσωπικών τους wallets, και το blockchain εγγυάται για την εγκυρότητα της συναλλαγής. Μια τέτοια συναλλαγή θα μπορούσε να επιτευχθεί ακόμη και χωρίς την χρήση κάποιου κρυπτονομίσματος.



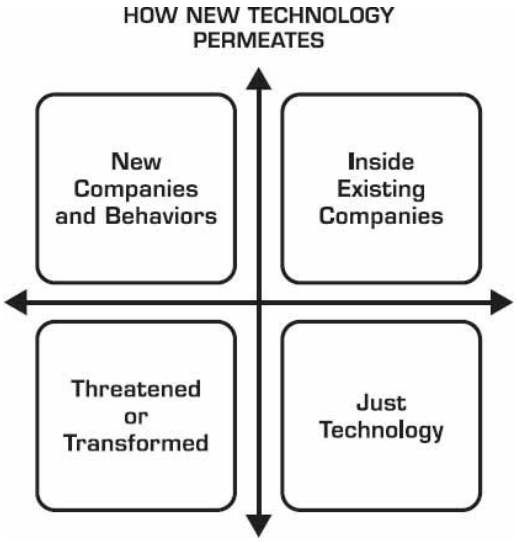
Εικόνα 2: Συναλλαγή μέσω της τεχνολογίας των Blockchains - *William Mougayar, Vitalik Buterin-The Business Blockchain*

Αυτός είναι και ένας από τους κύριους λόγους για τον οποίο οι peer-to-peer lending εφαρμογές θα πρέπει να χρησιμοποιούν εξατομικευμένα και ιδιωτικά blockchains.
Η Ethereum, η οποία είναι η δεύτερη μεγαλύτερη blockchain πλατφόρμα βάσει κεφαλαιοποίηση αγοράς, επιτρέπει στους χρήστες της να δημιουργούν έξυπνα συμβόλαια τα οποία οποία αποθηκεύονται στο ιδιωτικό της δίκτυο - blockchain.
Η τεχνολογία των blockchains μπορεί να χρησιμοποιηθεί και εκτός του οικονομικού τομέα, παραδείγματος χάρη, στο εμπόριο, στις εκλογές, σε περιπτώσεις κατοχύρωσης πνευματικής ιδιοκτησίας και λοιπά.

Το μέλλον των κατανεμημένων εφαρμογών (distributed applications) θα ξεδιπλωθεί σε παρόμοιο τρόπο με το διαδίκτυο.

Αν το κρίνουμε σήμερα, τα τελευταία 20 χρόνια, το διαδίκτυο έχει επηρεάσει τις εξής διαστάσεις:

1. Νέες επιχειρήσεις οι οποίες λειτουργούν εξ' ολοκλήρου διδικτυακά και προκάλεσαν νέες συμπεριφορές πελατών.
2. Ήδη υπάρχουσες επιχειρήσεις, οργανισμοί και κυβερνήσεις, ενστερνίστηκαν το διαδίκτυο στις εφαρμογές τους.
3. Επιχειρήσεις οι οποίες απειλήθηκαν από το διαδίκτυο, αναδιοργάνωσαν τον τρόπο λειτουργίας τους.
4. Η ανάπτυξη διαδικτυακού λογισμικού έγινε βασικό χαρακτηριστικό στον κόσμο του προγραμματισμού.

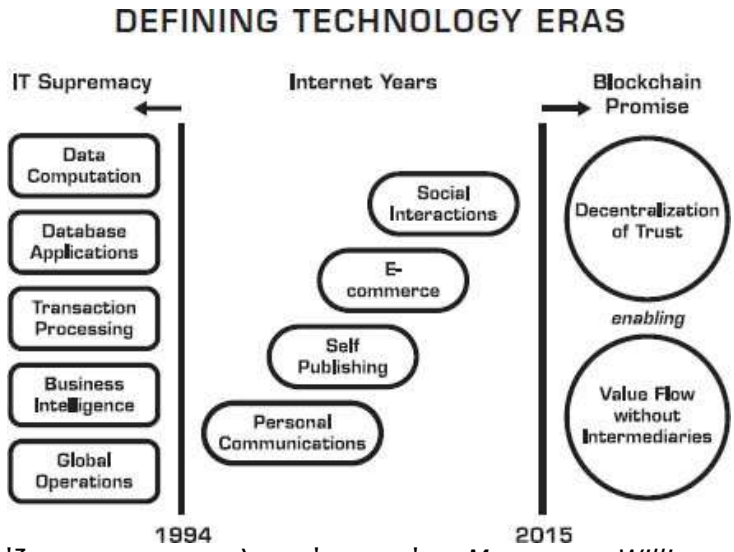


Εικόνα 3: Οι τέσσερις άξονες τους οποίος επηρέασε το διαδίκτυο τα τελευταία 20 χρόνια - *The Meaning of Decentralization – Vitalik Buterin*

Όπως είπε και ο Vitalik Buterin στο *The meaning of Decentralization "Εάν θα μπορούσαμε να προχωρήσουμε 10 χρόνια από σήμερα στο μέλλον, θα μπορούσαμε να αντικαταστήσουμε την λέξη Internet με Crypto-Tech*".
Όπως και με το διαδίκτυο, έτσι και με τις κατανεμημένες εφαρμογές και τα blockchain το τελικό αποτέλεσμα θα είναι το ίδιο. [1]

1. Νέες καινοτόμες επιχειρήσεις θα αναδειχθούν στον χώρο.
2. Οργανισμοί και κυβερνήσεις θα υιοθετήσουν την νέα τεχνολογία.
3. Παλιές επιχειρήσεις θα απειληθούν από την νέα τεχνολογία και είτε θα προσαρμοστούν, είτε θα καταρρεύσουν.
4. Η ανάπτυξη τέτοιων εφαρμογών θα γίνει βασική για τον κλάδο της πληροφορικής.

Ένας άλλος τρόπος να παρατηρήσουμε την συνεχή εξέλιξη της τεχνολογίας θα ήταν να απεικονίσουμε τις διαφορετικές φάσεις της εξέλιξης του διαδικτύου. Τοποθετώντας την τεχνολογία του Blockchain στην θέση μιας καινούριας φάσης που εστιάζει στις έννοιες peer-to-peer, εμπιστοσύνη και συναλλαγές αξιών.



Εικόνα 4: Ορίζοντας της τεχνολογικές εποχές - *Mougayar, William. The business blockchain*

2. Σχεδιασμός P2P Lending εφαρμογών

Οι peer-to-peer lending εφαρμογές είναι ηλεκτρονικές πλατφόρμες που επιτρέπουν στους χρήστες να ζητούν δημόσια δάνεια από ιδιώτες επενδυτές για τη χρηματοδότησή τους. Η ιδέα του δανεισμού peer-to-peer δεν είναι ένα νέο επιχειρηματικό μοντέλο, αφού αυτός είναι ο τρόπος με τον οποίο οι άνθρωποι συνήθιζαν να δανείζονται χρήματα ιδιωτικά χωρίς οποιαδήποτε διαμεσολάβηση. Η καινοτομία βρίσκεται στη χρήση του διαδικτύου για την οργάνωση αυτών των συναλλαγών. Οι διαδικασίες δανεισμού διαφέρουν από πλατφόρμα σε πλατφόρμα. Κάποιοι συνδέουν τους δανειστές με τους δανειολήπτες άμεσα και άλλοι περιλαμβάνουν ένα μεσολαβητή που συνήθως είναι μια τράπεζα. Η ίδια η πλατφόρμα δημιουργεί έσοδα από τέλη συναλλαγών [2].

Μία από τις κύριες ιδιότητες των P2P Lending εφαρμογών είναι η κατανεμημένη "φύση" των συστημάτων. Επομένως, θα μπορούσαμε να ενισχύσουμε την ιδιότητα αυτή, αυξάνοντας ταυτόχρονα και την ασφάλεια, αποθηκεύοντας τις συναλλαγές των χρηστών σε ένα κατανεμημένο δίκτυο blockchain. Σύμφωνα με τον *Mougayar, William στο The business blockchain*, οι κατανεμημένες εφαρμογές ξεκινούν από το να δημιουργούν τους δικούς τους νόμους σχετικά με την κυριαρχία αγαθών, τους δικούς τους περιορισμούς συναλλαγών και τη δική τους λογική.

Όπως αναφέρει και ο *Crosby, Michael* στο *Blockchain technology: Beyond bitcoin* [23] οι εφαρμογές που αναπτύσσονται βασισμένες σε blockchain δεν ανήκουν μόνο στον οικονομικό τομέα. Τα blockchains είναι μια τεχνολογία που ανθίζει όπου υπάρχει ανάγκη για μια έμπιστη διαδικτυακή αρχή. Μάλιστα αναφέρεται στη χρήση περισσότερων ειδών blockchains. Ξεχωρίζει τα εξής είδη:

1) Alternative Blockchains: Τα εναλλακτικά blockchains χρησιμοποιούνται από διάφορα συστήματα για να δημιουργήσουν κοινή συναίνεση μεταξύ κόμβων σε ένα δίκτυο.
2) Colored Coins: Τα Colored Coins είναι ένα πρωτόκολλο ανοιχτού κώδικα που περιγράφει ένα σύνολο μεθόδων για προγραμματιστές προκειμένου να αναπτυχθούν εφαρμογές που θα δημιουργούν αξία, κάνοντας χρήση ενός δικτύου blockchain.
3) Sidechains: Τα sidechains είναι εναλλακτικά blockchains που βασίζονται/ υποστηρίζονται από το Bitcoin, μέσω Bitcoin συμβολαίων, όπως το ευρώ υποστηρίζεται από το χρυσό. Τα sidechains κάνουν χρήση του blockchain του Bitcoin.

Επίσης αναφέρεται στο γεγονός ότι μεγάλες εταιρείες όπως η IBM, η Samsung, η Amazon, η UBS και πολλές άλλες, έχουν αρχίσει και ανακαλύπτουν την τεχνολογία των blockchains για δικούς τους σκοπούς. Ο Michael Crosby εξηγεί ακόμη, πως η χρήση των Smart Contracts έπαιξε μεγάλη σημασία στην καθίδρυση των κρυπτονομισμάτων και ιδιαίτερα των προγραμματισμένων συναλλαγών. Ακόμη, γίνεται λόγος για την συνεργασία των δύο αυτών εφαρμογών, των έξυπνων συμβολαίων και των blockchains. Πολλές από τις εφαρμογές και επιχειρήσεις που έχουν δημιουργηθεί βασισμένες σε blockchains και smart contracts, δεν θα μπορούσαν να λειτουργήσουν χωρίς την ύπαρξη ενός από των δύο, βλέπε Ethereum.

Καταλαβαίνουμε λοιπόν, πως για να δημιουργηθεί μια νέα peer-to-peer lending εφαρμογή η οποία θα βασίζεται σε κάποιο blockchain, θα πρέπει να υπάρχει η συνεργασία του blockchain με τα έξυπνα συμβόλαια.

Ο *Mougayar, William* στο *The business blockchain* [1] αναφέρει πως για την δημιουργία αποκεντρωμένων εφαρμογών υπάρχουν πολλά επίπεδα πολυπλοκότητας. Χαρακτηριστικά αναφέρεται στα εξής έξι επίπεδα:

1) Η χρήση ενός κρυπτονομίσματος ως μέσω συναλλαγών.
2) Η χρήση ενός blockchain για την αυθεντικοποίηση των δεδομένων και των αγαθών.
3) Η χρήση έξυπνων συμβολαίων τα οποία βασίζονται στο blockchain, έτσι ώστε να μπορέσει η εφαρμογή να τελέσει κάποιες λειτουργίες υπό συγκεκριμένες συνθήκες.
4) Η χρήση ενός blockchain ως θεμελιώδη τρόπο επικοινωνίας μέσω ενός δικτύου peer-to-peer με κόμβους.
5) Η χρήση ενός ιδιωτικού blockchain χωρίς κάποιο κρυπτονόμισμα ή value token.
6) Η χρήση ενός ιδιωτικού blockchain συμπεριλαμβανομένου κρυπτονομίσματος για την δημιουργία ενός δικτύου αξίας.

Ο Mougayar, William ακόμη ορίζει 12 χαρακτηριστικά τα οποία ορίζουν τις blockchain εφαρμογές. Τα χαρακτηριστικά αυτά είναι:

1) Η δυνατότητα προγραμματισμού και ποιές γλώσσες χρησιμοποιούνται.
2) Επεκτασιμότητα. Πόσους κόμβους μπορεί να φιλοξενήσει το δίκτυο;
3) Αναβάθμιση. Πόσο συχνά γίνονται αναβαθμίσεις στις εφαρμογές;
4) Πόσο εύκολη είναι η διαχείριση των συναλλαγών;
5) Υπάρχει πλήρη ορατότητα του βιβλιαρίου;
6) Πόσο είναι το κόστος για την ανάπτυξη της εφαρμογής;
7) Ποιό είναι το επίπεδο ασφάλειας του blockchain;
8) Ποιά είναι η σχέση ταχύτητας/ επίδοσης στην επιβεβαίωση συναλλαγών;
9) Υπάρχει διαθεσιμότητα των κόμβων;
10) Μπορεί να προστεθεί περεταίρω λειτουργικότητα στην εφαρμογή;
11) Διαλειτουργικότητα. Μπορεί η εφαρμογή να επικοινωνήσει με άλλες εφαρμογές της ίδιας ή αντίστοιχης τεχνολογίας;
12) Είναι ο κώδικας ανοιχτού τύπου; Σε τι επίπεδο μπορούν άλλοι προγραμματιστές να συμβάλλουν στην εφαρμογή;

Στο *Making Sense of Blockchain Applications: A Typology for HCI* [24] ο *Elsden, Chris* αναφέρει ένα σύνολο από blockchain εφαρμογές οι οποίες δραστηριοποιούνται σε διάφορους τομείς:

A TYPOLOGY OF BLOCKCHAIN APPLICATIONS

Application	Description	Examples
Underlying Infrastructure	Underlying protocols, decentralized application ecosystems, IoT architecture.	*Ethereum Blockstack, IoTA*
Currency	Payment services, internal currencies and utility tokens.	*Bitcoin, Dash, Kin*
Financial Services	Asset management, investment trading, and crowdfunding.	*Ripple OpenLedger Swarmfund*
Proof-as-a-service	Notaries, registers and attestation, supply-chain management.	*Blocknotary Chronicled, Everledger*
Property and Ownership	Digital rights management, copyright and ticketing services.	*Creative Chain Blockphase Aventus*
Identity Management	Self-sovereign digital identity, and authentication.	*Civic, Blockchain Helix, Bitnation*
Governance	Voting services, distributed autonomous organisations (DAO's).	*Followmyvote, Backfeed Crowdjury*

Εικόνα 5: Blockchain εφαρμογές και το τομέας στον οποίο δραστηριοποιούνται - *Elsden, Chris, et al. "Making Sense of Blockchain Applications: A Typology for HCI."*

Αναλύοντας την παραπάνω βιβλιογραφία, τις περιπτώσεις των άλλων επιχειρήσεων και τις απόψεις που εκφράζονται, θα μπορούσαμε να καταλήξουμε σε τέσσερα βασικά κριτήρια που πρέπει να πληροί μία peer-to-peer lending εφαρμογή η οποία κάνει χρήση της τεχνολογίας blockchain.

1) Η χρήση ενός κρυπτονομίσματος ως μέσω συναλλαγών και πληρωμών.
2) Η χρήση ενός blockchain για την ασφαλή αποθήκευση των συναλλαγών.
3) Η χρήση των Smart Contracts βασισμένα σε ένα δίκτυο blockchain για την εκτέλεση συγκεκριμένων εντολών, όταν αληθεύσει κάποια συνθήκη.
4) Η χρήση ενός ιδιωτικού blockchain για τις επιχειρησιακές λειτουργίες και την δημιουργία ενός μοναδικού και ιδιωτικού κρυπτονομίσματος.

2.1 Blockchain

Η τεχνολογία των Blockchain έγινε γνωστή μέσα από το Bitcoin. Βέβαια η τεχνολογία αυτή, έχει πολύ μεγαλύτερες δυνατότητες οι οποίες θα μπορούσαν να αλλάξουν μέχρι και το έως τώρα γνωστό σε εμάς διαδίκτυο.

Το Blockchain είναι ένα αποκεντρωμένο και κατανεμημένο βιβλιάριο που χρησιμοποιείται για την αποθήκευση των δεδομένων συναλλαγών αξιών και αγαθών χρηστών σε ένα δίκτυο. Στο σημείο αυτό ας ξεχωρίσουμε κάποιους τύπους δικτύων για να μπορέσουμε να αναλύσουμε τα blockchains καλύτερα.

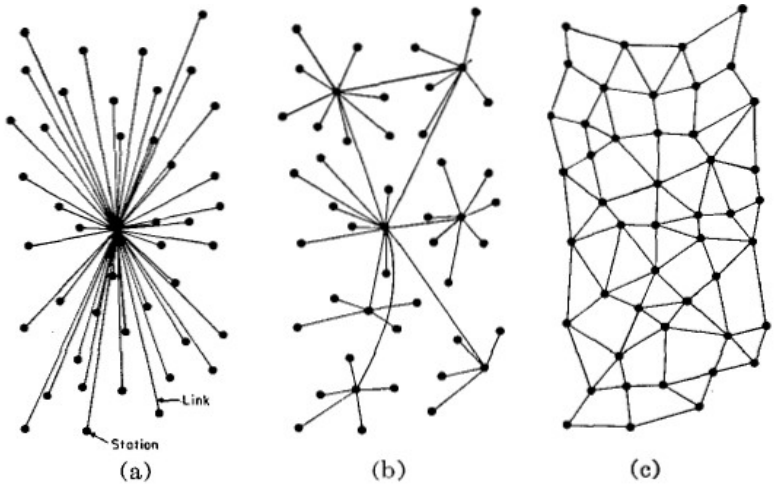


Fig. 1—(a) Centralized. (b) Decentralized. (c) Distributed networks.

Εικόνα 6: Παρουσίαση τριών διαφορετικών τύπων δικτύων - *The Meaning of Decentralization – Vitalik Buterin*

Στην παραπάνω εικόνα (εικόνα 6), οι τελείες αντιπροσωπεύουν τους κόμβους (stations) του δικτύου και οι γραμμές τις μεταξύ τους συνδέσεις (links).

A. Centralized network
Με τον όρο κεντρικό δίκτυο αναφερόμαστε σε ένα κεντρικό σύστημα στο οποίο όλοι οι κόμβοι είναι συνδεδεμένοι σε έναν κεντρικό υπολογιστή - διακομιστή. Ο κεντρικός υπολογιστής - διακομιστής είναι υπεύθυνος για τις μεταξύ των κόμβων επικοινωνίες και σε περίπτωση βλάβης του διακομιστή, καταρρέει όλο το σύστημα.

B. Decentralized network
Στο αποκεντρωμένο δίκτυο, παρουσιάζονται περισσότεροι από ένας κεντρικοί κόμβοι με τους οποίους συνδέονται οι υπόλοιποι κόμβοι. Εάν καταρρεύσει ένας κεντρικός κόμβος, το σύστημα θα συνεχίσει να λειτουργεί κανονικά.

C. Distributed network
Τα κατανεμημένα δίκτυα αποτελούνται από κόμβους όπου σχεδόν όλοι είναι συνδεδεμένοι μεταξύ τους και η επικοινωνία επιτυγχάνεται χωρίς την συμβολή κάποιου κεντρικού υπολογιστή - διακομιστή.

Είναι σημαντικό να κατανοήσουμε ότι τα δεδομένα που αποθηκεύονται σε ένα blockchain μπορεί να είναι τόσο φυσικά όσο και ψηφιακά. Ένα παράδειγμα φυσικού δεδομένου θα μπορούσε να είναι ο τίτλος ιδιοκτησίας ενός αυτοκινήτου, ενώ αντίστοιχα ένα παράδειγμα ψηφιακού δεδομένου θα μπορούσε να είναι η πνευματική ιδιοκτησία ενός τραγουδιού. Όπως προαναφέραμε, οτιδήποτε έχει αξία, μπορεί να καταχωρηθεί και να μοιραστεί σε ένα δίκτυο blockchain.

Αφού μια συναλλαγή ελεγχθεί με βάση τους κανόνες που διέπουν το εκάστοτε σύστημα, αποθηκεύεται χρονολογικά σε ομάδες που ονομάζονται μπλοκς, εξού και το blockchain, δηλαδή αλυσίδα από μπλοκ. Τα μπλοκ συνδέονται μεταξύ τους όπως μια αλυσίδα. Κάθε μπλοκ περιέχει την διεύθυνση - κλειδί του προηγούμενου μπλοκ και τις έγκυρες συναλλαγές. Η διεύθυνση - κλειδί του κάθε μπλοκ είναι μοναδική, ορίζει το συγκεκριμένο μπλοκ και έχει δημιουργηθεί μέσω της κρυπτογραφικής συνάρτησης SHA-256. Επειδή κάθε καινούργιο μπλοκ, περιέχει το κλειδί του προηγούμενου, δεν είναι δυνατή η εισαγωγή τρίτου/ κακόβουλου μπλοκ μεταξύ δύο συνδεδεμένων, έτσι ώστε να διαταραχθεί η αλληλουχία της αλυσίδας. Πρέπει λοιπόν να σημειώσουμε πώς κάθε εισαγωγή νέου μπλοκ καθιστά μια πολύ σημαντική διαδικασία για την ακεραιότητα και την ασφάλεια της αλυσίδας αφού επαληθεύει τα προηγούμενα.
Έτσι αποδεικνύεται ότι η αποθήκευση σε blockchain εξασφαλίζει την ασφάλεια των δεδομένων γιατί κανείς δεν μπορεί να τα παραποιήσει ή να τα διαγράψει. [3]

Ενώ κάθε μπλοκ έχει ένα μόνο "πρόγονο", μερικές φορές όταν δύο καινούργια μπλοκ αναρτηθούν ταυτόχρονα, μπορεί να δημιουργηθούν περισσότερα από ένας "διάδοχοι" μπλοκ. Η περίπτωση που δημιουργούνται περισσότερα μπλοκ - παιδία ονομάζεται fork και κάποια στιγμή το ένα από τα μπλοκ - παιδία θα απορροφηθεί και θα εισέλθει στο blockchain ενώ τα άλλα θα διαλυθούν.

Για την κρυπτογράφηση των περιεχομένων των μπλοκ, χρησιμοποιείται η κρυπτογράφησης δημοσίου κλειδιού. Δηλαδή, ο κάθε χρήστης έχει ένα προσωπικό δημόσιο κλειδί και ένα ιδιωτικό κλειδί. Έτσι, κάθε χρήστης του δικτύου μπορεί να διαχειρίζεται το τμήμα της πληροφορίας που του ανήκει ή που τον αφορά, με ασφαλή τρόπο και χωρίς να χρειάζεται η ύπαρξη κάποιας κεντρικής αρχής.

Η απαλοιφή της κεντρικής αρχής, αποτελεί σημαντικό στοιχείο σε ένα Blockchain. Αξιοσημείωτο και καινοτόμο στον κλάδο της πληροφορικής είναι το γεγονός ότι τα δεδομένα που έχουν αποθηκευτεί σε ένα blockchain είναι εξαιρετικά δύσκολο να αλλοιωθούν.

2.1.1 Η δομή ενός μπλοκ

Ένα μπλοκ είναι ένα σύνολο οργανωμένων δεδομένων, τα οποίο προορίζονται για το βιβλιάριο, το blockchain.

Size	Field	Description
4 bytes	Block Size	The size of the block, in bytes, following this field
80 bytes	Block Header	Several fields form the block header
1-9 bytes (VarInt)	Transaction Counter	How many transactions follow
Variable	Transactions	The transactions recorded in this block

Εικόνα 7: Η δομή ενός μπλοκ - *Antonopoulos, Andreas M. Mastering Bitcoin*

Ένα μπλοκ απαρτίζεται από το μέγεθος του μπλοκ σε bytes (Block Size), την κεφαλίδα του μπλοκ (Block Header) που έχει μέγεθος περίπου 80 bytes, τον συνολικό αριθμό των συναλλαγών στο μπλοκ (Transaction Counter) και τις συναλλαγές που έχουν καταχωρηθεί στο μπλοκ (Transactions).

2.1.2 Η κεφαλίδα του μπλοκ

Η κεφαλίδα του μπλοκ αποτελείται από τρία σετ μεταδεδομένων. Πρώτα υπάρχει η πληροφορία της τιμής hash του προηγούμενου μπλοκ. Έπειτα υπάρχουν τα εξής: η δυσκολία εξόρυξης, χρονοσφραγίδα και μια μεταβλητή ονόματη nonce.
Το τελευταίο σετ μεταδεδομένων είναι το λεγόμενο merkle tree, δηλαδή μία δομή δεδομένων για την βέλτιστη περιγραφή όλων των συναλλαγών στο μπλοκ.

Size	Field	Description
4 bytes	Version	A version number to track software/protocol upgrades
32 bytes	Previous Block Hash	A reference to the hash of the previous (parent) block in the chain
32 bytes	Merkle Root	A hash of the root of the merkle tree of this block's transactions
4 bytes	Timestamp	The approximate creation time of this block (seconds from Unix Epoch)
4 bytes	Difficulty Target	The proof-of-work algorithm difficulty target for this block
4 bytes	Nonce	A counter used for the proof-of-work algorithm

Εικόνα 8: Η δομή της κεφαλίδας ενός μπλοκ - *Antonopoulos, Andreas M. Mastering Bitcoin*

Το Previous Block Hash είναι αυτό που συνδέει το παρόν μπλοκ με τα προηγούμενο. Η τιμή hash αυτή δημιουργείται από τα δεδομένα του προηγούμενου μπλοκ. Όπως αναφέρει η Laurence, Tiana στο Blockchain for dummies "Το hash είναι η μαγική κόλλα που ενώνει τα μπλοκ μεταξύ τους και επιτρέπει εμπιστοσύνη με μαθηματική ακρίβεια". Για τη δημιουργία της τιμής hash χρησιμοποιείται η κρυπτογραφική συνάρτηση SHA-256, η οποία κρυπτογραφεί τα δεδομένα και επιστρέφει σχεδόν πάντα μια μοναδική αλφαριθμητική συμβολοσειρά 64 χαρακτήρων. Η λειτουργία αυτή θεωρείται μη αντιστρέψιμη και λειτουργεί σαν μοναδικό ψηφιακό αποτύπωμα που δεν μπορεί να αποκρυπτογραφηθεί. [3][4]

Στην παρακάτω εικόνα (εικόνα 9) παρουσιάζεται το αποτέλεσμα Hash που προκύπτει όταν εφαρμόσουμε τον αλγόριθμο SHA-256 σε τρεις διαφορετικές περιπτώσεις δεδομένων εισόδου.

input	61 62 63
hash	ba7816bf 8f01cfea 414140de 5dae2223 b00361a3 96177a9c b410ff61 f20015ad
input	61 62 63 64 62 63 64 65 63 64 65 66 64 65 66 67 65 66 67 68 66 67 68 69 67 68 69 6a 68 69 6a 6b 69 6a 6b 6c 6a 6b 6c 6d 6b 6c 6d 6e 6c 6d 6e 6f 6d 6e 6f 70 6e 6f 70 71
hash	248d6a61 d20638b8 e5c02693 0c3e6039 a33ce459 64ff2167 f6ecedd4 19db06c1
input	One million of 61
hash	cdc76e5c 9914fb92 81a1c7e2 84d73e67 f1809a48 a497200e 046d39cc c7112cd0

Εικόνα 9: Η κρυπτογραφική συνάρτηση SHA-256 - *ResearchGate*

Η μεταβλητή nonce είναι ένας τυχαίος αριθμός ο οποίος χρειάζεται από τον αλγόριθμο Proof-of-Work (θα αναλυθεί παρακάτω).

2.1.3 Η Αλυσίδα (Chain)

Ο όρος αλυσίδα προέρχεται από τον τρόπο που είναι ταξινομημένα τα μπλοκ σε ένα blockchain δίκτυο. Σύμφωνα με τον *Antonopoulos, Andreas M. Mastering Bitcoin* "Η δομή των δεδομένων στο Blockchain είναι μια ταξινομημένη λίστα από μπλοκ συνδεδεμένη προς τα πίσω". Μέσω αυτής της ταξινόμησης σχηματίζεται και η αλυσίδα.

2.1.4 Ψηφιακές Υπογραφές (Digital Signatures)

Η δημιουργία συναλλαγών στο Blockchain απαιτεί τις ψηφιακές υπογραφές για τον έλεγχο και την εγκυρότητα των συναλλαγών. Συγκεκριμένα, για την δημιουργία μιας ψηφιακής υπογραφής χρησιμοποιείται ένα ζεύγος κλειδιών, ένα δημόσιο και ένα ιδιωτικό.

Το δημόσιο κλειδί είναι ορατό σε όλους του χρήστες και χρησιμοποιείται για την επαλήθευση των δεδομένων, λειτουργεί σαν μια διεύθυνση που ορίζει έναν χρήστη.

Το ιδιωτικό κλειδί το γνωρίζει μόνο ο χρήστης και χρησιμοποιείται για την δημιουργία της ψηφιακής του υπογραφής.

2.2 Peer-to-Peer Δίκτυο

Οι χρήστες για να αλληλεπιδράσουν με το Blockchain χρησιμοποιούν ένα αποκεντρωμένο Peer-to-peer δίκτυο, στο οποίο κάθε χρήστης είναι και ένας κόμβος. Προκειμένου ένας χρήστης να μπορέσει θα θεωρηθεί κόμβος του δικτύου θα πρέπει να έχει το ηλεκτρονικό του πορτοφόλι και να προσφέρει αποθηκευτικό χώρο από τον υπολογιστή του. Επίσης, απαιτείται πρόσβαση στο διαδίκτυο. Όλοι οι ενεργοί χρήστες, ανταλλάσουν πληροφορίες μεταξύ τους, αφού δεν υπάρχει κάποιος κεντρικός κόμβος - εξυπηρετητής. Κάθε φορά που ένα κόμβος μεταδίδει δεδομένα σε έναν άλλο, ελέγχεται η ακεραιότητα των δεδομένων. Αφού τα δεδομένα κριθούν έγκυρα, μεταδίδονται στους γειτονικούς κόμβους και σύντομα όλο το δίκτυο έχει ενημερωθεί.

Το πλεονέκτημα της χρήσης ενός P2P δικτύου είναι ότι δεν χρειάζεται κάποιος ενδιάμεσος ή κάποια κεντρική αρχή για την επιβεβαίωση της ορθότητας, κάτι που το κάνει πιο ασφαλές από κακόβουλες επιθέσεις. [3]

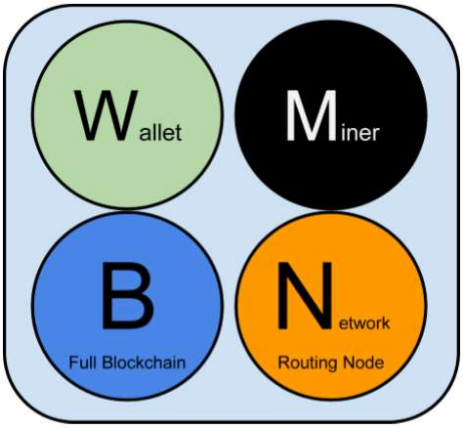
Εικόνα 10: Παρουσίαση ενός δικτύου με κεντρικό διακομιστή
και ενός δικτύου Peer-to-peer

Χαρακτηριστικό των peer-to-peer δικτύων είναι ότι σε περίπτωση που κάποιος από τους κόμβους σταματήσει να λειτουργεί το σύστημα δεν θα έχει κανένα απολύτως πρόβλημα. Σε αντίθεση με ένα κεντρικοποιημένο δίκτυο, όπου εάν καταρρεύσει ο διακομιστής, θα καταρρεύσει και ολόκληρο το σύστημα.

2.2.1 Τύποι κόμβων και ρόλοι

Όλοι οι κόμβοι σε ένα Blockchain P2P δίκτυο (π.χ. Bitcoin) είναι ισάξιοι, αυτό όμως δεν σημαίνει ότι δεν μπορούν να έχουν διαφορετικούς ρόλους ανάλογα με την λειτουργία που υποστηρίζουν. Ένας κόμβος του Bitcoin είναι ένα σύνολο από τις παρακάτω λειτουργίες:

1. Δρομολόγηση
2. Βάση δεδομένων του Blockchain
3. Εξόρυξη
4. Λειτουργίες Πορτοφολίου



Εικόνα 11: Ένας κόμβος στο δίκτυο του Bitcoin με όλες τις τέσσερις λειτουργίες - *Antonopoulos, Andreas M. Mastering Bitcoin*

Όλοι οι κόμβοι έχουν τη λειτουργία της δρομολόγησης για να μπορούν να συμμετέχουν στο δίκτυο. Επίσης, όλοι οι κόμβοι επικυρώνουν και διαδίδουν συναλλαγές και μπλοκ, καθώς επίσης επικοινωνούν με άλλους κόμβους. Στο παράδειγμα του πλήρες κόμβου (Full Node) στην Εικόνα 11, η λειτουργία της δρομολόγησης συμβολίζεται με το πορτοκαλί κυκλάκι. Κάποιοι κόμβοι (οι Full Nodes) διατηρούν και ένα πλήρες αρχείο της βάσης δεδομένων του Blockchain. Τέτοιοι κόμβοι μπορούν αυτόνομα και έγκυρα να επικυρώσουν οποιαδήποτε συναλλαγή, χωρίς εξωτερικές αναφορές.

Άλλοι κόμβοι διατηρούν μόνο ένα κομμάτι της βάσης δεδομένων του Blockchain και επικυρώνουν συναλλαγές μέσω μιας διαδικασίας που ονομάζεται simplified payment verification ή αλλοιώς SPV. Αυτοί οι κόμβοι είναι γνωστοί ως SPV κόμβοι ή lightweight nodes. Στην Εικόνα 11, η βάση δεδομένων (το βιβλιάριο) συμβολίζεται με το μπλε κυκλάκι.

Οι κόμβοι εξόρυξης (mining nodes) ανταγωνίζονται για να δημιουργήσουν νέα μπλοκ χρησιμοποιώντας ειδικά σχεδιασμένα μηχανήματα έτσι ώστε να λύνουν τις εξισώσεις proof-of-work πολύ πιο γρήγορα.

Πορτοφόλια χρηστών, μπορεί να είναι μέρος πλήρες κόμβου, συνήθως desktop πορτοφόλια, δηλαδή πορτοφόλια υπολογιστή που έχουν να διαθέσουν μεγάλο αποθηκευτικό χώρο. Σε φορητές συσκευές όπως κινητά τηλέφωνα και φορητούς υπολογιστές προτιμούνται τα SPV πορτοφόλια. Στην Εικόνα 11 η λειτουργία του πορτοφολίου συμβολίζεται με το πράσινο κυκλάκι.

Συμπληρωματικά των βασικών τύπων κόμβων του bitcoin P2P πρωτοκόλλου, υπάρχουν και διακομιστές που τρέχουν άλλου είδους πρωτόκολλα, όπως παραδείγματος χάρη πρωτόκολλα που επιτρέπουν την ευκολότερη πρόσβαση σε lightweight κόμβους.

2.2.2 Πρωτόκολλο κοινής συναίνεσης (Consensus Protocol)

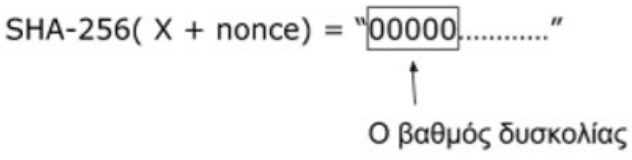
Για να μπορέσει να λειτουργήσει σωστά ένα P2P δίκτυο βασισμένο σε Blockchain, θα πρέπει να υπάρχει και η κοινή συναίνεση των χρηστών - κόμβων και αυτό επειδή δεν υπάρχει κάποια κεντρική αρχή για να παρέχει εμπιστοσύνη και ακεραιότητα στις πληροφορίες που ανταλλάσσονται. [3][4] Συναίνεση σύμφωνα με τον Antonopoulos, Andreas M. είναι "ένα αναδυόμενο δημιούργημα της ασύγχρονης αλληλεπίδρασης χιλιάδων ανεξαρτήτων κόμβων, ακολουθώντας απλούς κανόνες". [3]

Για να μπορέσουμε λοιπόν να αποφασίσουμε ποιά δεδομένα είναι έγκυρα και ποιά όχι, υπάρχει μια διαδικασία επικύρωσης που ονομάζεται εξόρυξη (γνωστή και ως mining). Η διαδικασία της εξόρυξης απαιτεί τη χρήση των αλγορίθμων Proof-of-work και Proof-of-stake. Μέσω των παραπάνω αλγορίθμων, επιτυγχάνεται και η προσθήκη νέων μπλοκ στην αλυσίδα, blockchain.

2.2.3 Proof-of-Work

Κύριος στόχος του αλγορίθμου proof of work (PoW) είναι η επικύρωση των δεδομένων και η αποτροπή επιθέσεων στο δίκτυο. Ενώ σαν ιδέα προϋπήρχε, πρώτη φορά υλοποιήθηκε από τον Satoshi Nakamoto στο κρυπτονόμισμα Bitcoin. Αρχικά, ο αλγόριθμος PoW δημοσιεύτηκε από τους Cynthia Dwork και Moni Naor το 1992 με τίτλο "Pricing via Processing or Combatting Junk Mail" [5]. Έπειτα, δημοσιεύτηκε μία παρόμοια πρόταση που ονομάζεται Hashcash το 1997 από τον Adam Back [6]. Ο όρος "Proof-of-Work" δημιουργήθηκε από τους Markus Jakobsson και Ari Juels σε έγγραφο με τίτλο "Proofs of Work and Bread Pudding Protocols" [7], που δημοσιεύτηκε το 1999. Πλέον ο αλγόριθμος αυτός χρησιμοποιείται ευρέως στην τεχνολογία Blockchain.
Για να κατανοήσουμε καλύτερα τη λειτουργία του αλγορίθμου proof of work θα πρέπει να έχουμε υπόψη μας πως κάθει φορά που πραγματοποιείται μια συναλλαγή, αυτή αποθηκεύεται σε ένα προσωρινό - πρόχειρο μπλοκ. Όταν το μπλοκ συμπληρώσει τον αριθμό των συναλλαγών που μπορεί να αποθηκεύσει, μεταδίδεται στους υπόλοιπους κόμβους. Τότε υλοποιείται ο αλγόριθμός Proof-of-Work όπου και ελέγχεται η εγκυρότητα ολόκληρου του μπλοκ. Στη συνέχεια, κάθε κόμβος προσθέτει την μεταβλητή nonce και σχηματίζει την δομή "Block + nonce". Το "Block + nonce" κρυπτογραφείται βάσει του αλγορίθμου SHA-256 και παράγεται μια αλφαριθμητική τιμή 64 χαρακτήρων που ονομάζεται Hash.

Στο πρωτόκολλο του Bitcoin τίθεται αυτομάτως ένας στόχος δυσκολίας (target Difficulty). Ο στόχος αυτός καθορίζει το πόσο γρήγορα και πόσο εύκολα η κρυπτογραφική συνάρτηση SHA-256 θα μπορέσει να υπολογίσει την τιμή Hash. Όλοι οι κόμβοι στο δίκτυο ανταγωνίζονται για το ποιός θα υπολογίσει πρώτος την τιμή Hash του νέου μπλοκ. Επειδή η συνάρτηση Hash για συγκεκριμένα δεδομένα εισόδου, έχει και συγκεκριμένα δεδομένα εξόδου, το μόνο που μπορούν να αλλάξουν οι κόμβοι προκειμένου να υπολογίσουν την τιμή hash, είναι η τιμή nonce. Ουσιαστικά η διαδικασία της εξόρυξης είναι η αναζήτηση της τιμής nonce έτσι ώστε να επιτυγχάνεται το επιθυμητό επίπεδο δυσκολίας, δηλαδή
αν ένα μπλοκ περιέχει δεδομένα X και επιθυμούμε δυσκολία ύψους 5 να ισχύει:



Η αλλαγή του nonce γίνεται αυτόματα και ο πρώτος κόμβος που θα βρει το σωστό αποτέλεσμα επικυρώνει τις αλλαγές και κερδίζει ένα ποσό κρυπτονομισμάτων.
Έπειτα, αποστέλλεται στους υπόλοιπους κόμβους το "Block + nonce + Hash" και αφού πιστοποιήσουν και οι υπόλοιποι κόμβοι την ορθότητα της τιμής Hash, προσκολλάται το μπλοκ στην αλυσίδα. Αμέσως μετά την ολοκλήρωση της παραπάνω διαδικασίας, οι κόμβοι ανανεώνουν το ιστορικό των συναλλαγών τους και ανταγωνίζονται πάλι από την αρχή για την επόμενη τιμή Hash και για να υπολογίσουν την επόμενη τιμή nonce. [3]

2.2.4 Proof-of-Stake

Ο αλγόριθμος Proof of Stake (PoS) έχει κοινό σκοπό με τον αλγόριθμο Proof of Work, η διαδικασία επικύρωσης συναλλαγών είναι όμως αρκετά διαφορετική. Το πρώτο κρυπτονόμισμα που υλοποίησε τον αλγόριθμο Proof-of-Stake ήταν το Peercoin το 2012, μαζί με το ShadowCash, το Nxt, το BlackCoin, το NuShares/NuBits, το Qora και το NavCoin.
Αντίθετα με το Proof of Work, στον αλγόριθμο Proof of Stake, ο κόμβος που θα επικυρώσει τις συναλλαγές επιλέγεται με ντετερμινιστικό τρόπο. Οι κόμβοι καταθέτουν στο δίκτυο ένα ποσό από τα κρυπτονομίσματα τους ως εγγύηση οτι θα επικυρώσουν τις συναλλαγές σε ένα μπλοκ.
Το ποιός κόμβος τελικά θα επιλεγεί, καθορίζεται από το ποσό που έχει καταθέσει. Έστω ένας κόμβος Α έχει θέσει 70 κρυπτονομίσματα και δύο άλλοι κόμβοι Β και Γ αντίστοιχα απο 15, τότε ο κόμβος Α θα έχει 70% πιθανότητα να επικυρώσει το μπλοκ ενώ οι άλλοι δύο από 15%. Στον αλγόριθμο PoS δεν υπάρχει ανταμοιβή για την επικύρωση του κάθε μπλοκ, αλλα ο κόμβος που θα επικυρώσει το μπλοκ, λαμβάνει ένα μέρος από την κάθε συναλλαγή.

2.3 Έξυπνα συμβόλαια (Smart Contracts)

Με την παραδοσιακή έννοια, ένα συμβόλαια δεσμεύει τα συμβαλλόμενα μέρη το ένα απέναντι στο άλλο.
Κάθε συμβαλλόμενο μέρος εμπιστεύεται την εκπλήρωση της συμφωνίας του συμβολαίου και από τις δύο συμμετέχουσες μεριές. Τα έξυπνα συμβόλαια (Smart Contracts) έχουν ακριβώς την ίδια λειτουργία με τα παραδοσιακά, με την μόνη διαφορά, να καταργούν την ανάγκη εμπιστοσύνης μεταξύ των μερών.

Τα έξυπνα συμβόλαια είναι ένα κομμάτι κώδικας που αποθηκεύεται σε βιβλιάριο, χωρίς κανένας να είναι σε θέση να τον αλλοιώσει. Τα έξυπνα συμβόλαια είναι σε θέση να εκτελέσουν ή να επιβάλουν μια προκαθορισμένη συμφωνία μέσω του blockchain, όταν πληρούνται κάποιες συγκεκριμένες προϋποθέσεις. Δηλαδή, ο κώδικας περιέχει τη συμφωνία μεταξύ των διαφόρων μερών, η οποία δεν μπορεί να αλλάξει ή τα τροποποιηθεί παρά μόνο να εκτελεστεί όταν πληρούνται κάποιες προϋποθέσεις.

Τα βασικά στοιχεία που καθορίζουν ένα έξυπνο συμβόλαιο είναι τα εξής:

1. Αυτονομία: Μετά την δρομολόγηση και τη λειτουργία του, το συμβόλαιο και τα συμβαλλόμενα μέρη δεν χρειάζεται να βρίσκονται άλλο σε επαφή.
2. Αυτάρκεια: Τα έξυπνα συμβόλαια είναι σε θέση να λειτουργούν και να εκτελούν εσωτερικές λειτουργίες με τους πόρους που διαθέτουν.
3. Αποκεντροποίηση: Τα έξυπνα συμβόλαια δεν βρίσκονται αποθηκευμένα σε κάποιον κεντρικό υπολογιστή - διακομιστή, αλλα είναι αποθηκευμένα και εκτελούνται σε διάφορους κόμβους του δικτύου Blockchain.

Κύριος στόχος των έξυπνων συμβολαίων σε ένα Blockchain δίκτυο είναι να επιτρέψει μία αξιόπιστη συναλλαγή χωρίς την ανάγκη μεσαζόντων. Τα συμβόλαια ενεργοποιούνται αυτόματα όταν πληρούνται οι προϋποθέσεις που έχουν συμφωνήσει τα εμπλεκόμενα μέρη. Ένα παράδειγμα έξυπνου συμβολαίου θα μπορούσε να είναι η μεταβίβαση ενός ποσού όταν περάσει μια συγκεκριμένη ημερομηνία. Όταν ένα συμβόλαιο ενεργοποιηθεί και λειτουργήσει, δεν μπορεί να διακοπεί η λειτουργία του από τρίτα άτομα, ούτε να αλλάξει η εκάστοτε συμφωνία. Σημαντικό είναι να τονίσουμε ότι ένα μικρό λάθος στον κώδικα μπορεί να οδηγήσει στην απώλεια της αξίας του συμβολαίου. [8]
Τα έξυπνα συμβόλαια δεν είναι εφαρμογές blockchain, αλλα πολλές φορές μέρος τέτοιων εφαρμογών. [1] Μία ακόμη σημαντική διαφορά μεταξύ ενός έξυπνου συμβολαίου και ενός παραδοσιακού, είναι το χαμηλό κόστος υλοποίησης του Smart Contract. [9] Το πεδίο χρήση αυτών των έξυπνων συμβολαίων δεν περιορίζεται μόνο στον οικονομικό τομέα ή στον ψηφιακό κόσμο. Επειδή πρόκειται περί ενός κομματιού κώδικα, μπορεί να προγραμματιστεί έτσι ώστε να αλληλεπιδρά με οποιοδήποτε φυσικό περιουσιακό στοιχείο. Ένα έξυπνο συμβόλαιο μπορεί να εφαρμοστεί σε οτιδήποτε αλλάζει κατάσταση με την πάροδο του χρόνου και έχει καποιου είδους αξία. [1] Ένα παράδειγμα εφαρμογής Blockchain που κάνει χρήση των Smart Contracts είναι το Ethereum. Το Ethereum είναι μια εικονική μηχανή (virtual machine) βασισμένη σε Blockchain, που δίνει την δυνατότητα στους χρήστες της να δημιουργούν έξυπνα συμβόλαια. [10]

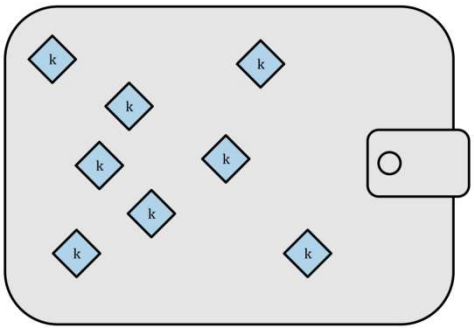
2.4 Πορτοφόλια (Wallets)

Η ιδιοκτησία κρυπτονομισμάτων αποδεικνύεται μέσω ψηφιακών κλειδιών, διευθύνσεων και ψηφιακών υπογραφών. Τα ψηφιακά κλειδιά δεν είναι αποθηκευμένα στο δίκτυο του blockchain, αλλα δημιουργούνται και αποθηκεύονται σε ένα αρχείο τοπικά στον υπολογιστή ή σε μία μικρή βάση δεδομένων, το λεγόμενο βιβλιάριο. Τα κλειδιά που δημιουργούνται εξαρτώνται από το πρωτόκολλο που ισχύει στο συγκεκριμένο βιβλιάριο και είναι τελείως ανεξάρτητα από το διαδίκτυο ή το blockchain δίκτυο. Μέσω των κλειδιών επιτρέπονται κάποιες βασικές λειτουργίες όπως είναι η αποκεντρωμένη εμπιστοσύνη, η απόδειξη ιδιοκτησίας και οι κρυπτογραφικοί αλγόριθμοι ασφάλειας.

Κάθε συναλλαγή απαιτεί μία έγκυρη υπογραφή για να συμπεριληφθεί στην αλυσίδα, η οποία μπορεί μόνο να δημιουργηθεί μέσω των κλειδιών. Έτσι, όποιος έχει πρόσβαση σε αυτά τα κλειδιά, έχει και πρόσβαση στον αντίστοιχο πορτοφόλι και συνεπώς στα κρυπτονομίσματα.
Το δημόσιο κλειδί είναι παρεμφερές με τον αριθμό ενός λογαριασμού, ενώ το ιδιωτικό κλειδί είναι κάτι σαν το κρυφό PIN της πιστωτικής κάρτας. Εξαιτίας αυτού, τα κλειδιά και ειδικότερα το ιδιωτικό κλειδί, δεν είναι φανερά στους χρήστες και παραμένουν αποθηκευμένα και ασφαλή στο προσωπικό πορτοφόλι των χρηστών.
Στην περίπτωση του Bitcoin, η διεύθυνση πληρωμής, δημιουργείται μέσω του δημοσίου κλειδιού ενός χρήστη. Αυτό δεν σημαίνει όμως ότι όλες οι διευθύνσεις πληρωμών έχουν δημιουργηθεί από δημόσια κλειδιά.

2.4.1 Μη ντετερμινιστικά (nondeterministic) Πορτοφόλια

Τα πρώτα πορτοφόλια - πελάτες (clients) του Bitcoin ήταν απλά ένα σύνολο τυχαία παραγμένων ιδιωτικών κλειδιών. Αυτό το είδος πορτοφολίου ονομάζεται Type-0 μη ντετερμινιστικά (nondeterministic) πορτοφόλια. Για παράδειγμα το Bitcoin Core Client προπαράγει 100 τυχαία ιδιωτικά κλειδιά και συνεχίζει να παράγει καθώς αυξάνεται και η απαίτηση για νέα. Αυτού του είδους τα πορτοφόλια ονομάζονται και "Just a Bunch Of Keys" ή απλά JBOK [3], τέτοια πορτοφόλια αντικαθίστανται από ντετερμινιστικά πορτοφόλια γιατί είναι πιο εύκολα στην διαχείριση. Η χρήση ενός Type-0 nondeterministic πορτοφολίου είναι μία κακή επιλογή, καθώς χρειάζεται συχνά backup έτσι ώστε να εξασφαλιστεί οτι δεν θα χρησιμοποιηθεί κάποιο κλειδί διπλά.



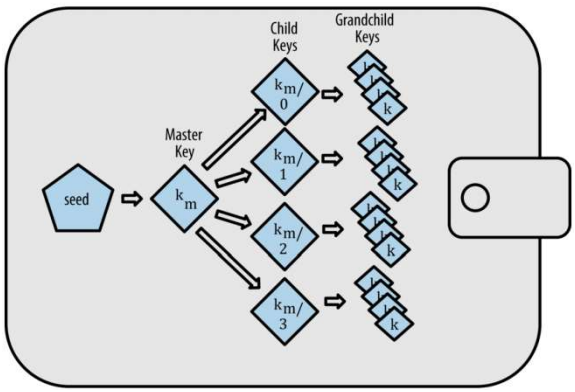
Εικόνα 12: Μη ντετερμινιστικό πορτοφόλι - *Antonopoulos, Andreas M. Mastering Bitcoin*

2.4.2 Ντετερμινιστικό (Deterministic) Πορτοφόλι

Ντετερμινιστικά (Deterministic) ή "seeded" πορτοφόλια είναι αυτά τα οποία περιέχουν ένα ιδιωτικό κλειδί το οποίο έχει παρθεί μέσω ενός "σπόρου" (seed), μέσα από μία μονόδρομης hash συνάρτησης. Αυτός ο "σπόρος" είναι ένας τυχαίος αριθμός ο οποίος είναι συνδυασμένος με επιπλέον δεδομένα για να παράξει ένα μοναδικό ιδιωτικό κλειδί.

2.4.3 Ιεραρχικό ντετερμινιστικό πορτοφόλι (BIP0032/ BIP0044)

Τα ντετερμινιστικά πορτοφόλια σχεδιάστηκαν για να διευκολυνθεί η δημιουργία πολλών κλειδιών από ένα μοναδικό "seed". Η πιό ανεπτυγμένη μορφή ενός ντετερμινιστικού πορτοφολίου είναι το ιεραρχικό ντετερμινιστικό πορτοφόλι (Hierarchical Deterministic Wallet) ή αλλοιώς HD Wallet, το οποίο ορίζεται από το πρότυπο BIP0032. Ιεραρχικά ντετερμινιστικά πορτοφόλια περιέχουν κλειδιά ιεραρχημενα σε δενδρική δομή, έτσι ώστε ένα κλειδί-γονέας να μπορεί να δημιουργήσει έναν αριθμό κλειδιών-παιδιά, τα οποία με τη σειρά τους μπορούν και αυτά να δημιουργήσουν περισσότερα κλειδιά-παιδιά. Η δενδρική δομή παρουσιάζεται από κάτω στην εικόνα 13.



Εικόνα 13: Ιεραρχικά ντετερμινιστικό πορτοφόλι - *Antonopoulos, Andreas M. Mastering Bitcoin*

Τα HD πορτοφόλια προσφέρουν δύο σημαντικά πλεονεκτήματα σε σχέση με τα μη ντετερμινιστικά:

1. Η δενδρική δομή των κλειδιών επιτρέπει την χρήση διαφορετικών κλειδιών για διαφορετικούς σκοπούς. Για παράδειγμά σε μια επιχείρηση τα διάφορα τμήματα θα μπορούσαν να έχουν διαφορετικά κλειδιά για να διεξάγουν τις πληρωμές τους.
2. Οι χρήστες των HD πορτοφολιών επιτρέπει την δημιουργία πολλών δημοσίων και ιδιωτικών κλειδιών, έτσι αν ένας χρήστης επιθυμεί να χρησιμοποιήσει ένα δημόσιο κλειδί σε μη ασφαλές περιβάλλον, δεν χρειάζεται να το ξαναχρησιμοποιήσει. [3]

2.5 Κρυπτονομίσματα

Τα κρυπτονομίσματα (cryptocurrencies) είναι ψηφιακές εναλλακτικές για το παραδοσιακό χάρτινο χρήμα. Τα κρυπτονομίσματα είναι μία από τις πολλές εφαρμογές των Blockchains και δημιουργήθηκαν για την ολοκλήρωση πληρωμών. Το Bitcoin για παράδειγμα είναι ένα αποκεντροποιημένο κρυπτονόμισμα, δηλαδή δεν υπάρχει κάποιο κεντρικό γραφείο συμψηφισμού. Όλες οι συναλλαγές επεξεργάζονται μέσω του δικτύου του Blockchain που εξασφαλίζει εγκυρότητα.
Ένα κρυπτονόμισμα λειτουργεί ως ένας οικονομικός διαμεσολαβητής για τις λειτουργίες ενός Blockchain. Μερικές φορές τα κρυπτονομίσματα αντικαθίστανται από value tokens, το οποίο σε ένα βαθμό υπονοεί την ύπαρξη κάποιου cryptocurrency.
Η κρυπτο - οικονομία δεν σχεδιάστηκε για να αναλάβει όλες τις οικονομικές υπηρεσίες και λειτουργίες που υπάρχουν. Θα αναπτύξει τον δικό της πλούτο και θα δημιουργήσει νέες επιχειρηματικές ευκαιρίες οι οποίες δεν θα περιορίζονται μόνο στον οικονομικό τομέα.
Επειδή τα κρυπτονομίσματα είναι προγραμματισμένες εφαρμογές που βασίζονται στα blockchains, μια συναλλαγή θα μπορούσε να σημαίνει περισσότερα από την μεταβίβαση χρημάτων. Η χρήση των κρυπτονομισμάτων θα μπορούσε να επεκταθεί και στην αποθήκευση τίτλων ιδιοκτησίας στο blockchain ή ακόμη να χρησιμοποιηθει για διαδικτυακές εκλογές. Υπάρχει μία μεγάλη διαφορά ανάμεσα στο χρήμα και την αξία όπως τόνισα ο Mougayar, William. "Τα χρήματα είναι μια μορφή αξίας, αλλά δεν είναι όλες οι μορφές αξίας χρήματα".

Η αξία δημιουργείται καθώς υπηρεσίες λειτουργούν πάνω στο Blockchain. Αυτές οι υπηρεσίες θα επιτύχουν δημιουργώντας ένα νέο οικοσύστημα βασισμένο στην εμπιστοσύνη, δημιουργώντας υπηρεσίες όπου η ταυτότητα, τα δικαιώματα, οι συνδρομές, οι ιδιοκτησίες ακόμη και οι ψηφοφορίες θα παίζουν ένα σημαντικό ρόλο. Όπως ακριβώς έγινε και με το διαδίκτυο, όταν το ηλεκτρονικό εμπόριο και αργότερα τα κοινωνικά δίκτυα δημιούργησαν τον δικό τους πλούτο και το δικό τους μερίδιο στην αγορά.

Ένα παράδειγμα είναι η εφαρμογή La’ Zooz, η οποία επιτρέπει στους χρήστες να μαζεύουν tokens μετακινώντας άλλους χρήστες με το αμάξι τους εντός και εκτός πόλης. Αυτοι οι πόντοι μπορούν να χρησιμοποιηθούν για άλλες μεταφορές με άλλους οδηγούς που συμμετέχουν στην εφαρμογή. Στη συγκεκριμένη περίπτωση δεν ανταλλάσσονται χρήματα, αλλα το δίκτυο ανταλλάσσει αξία.

Εννοείται πως τα κρυπτονομίσματα μπορούν να χρησιμοποιηθούν και για την πληρωμή υπηρεσιών και προϊόντων όπως περιγράφεται και στον βιβλίο The business blockchain του Mougayar, William και Vitalik Buterin.

Όλες οι υπηρεσίες Blockchain έχουν κάτι κοινό. Εκτός του γεγονότος ότι βασίζονται και λειτουργούν πάνω στο Blockchain, δημιουργούν το δικό τους κρυπτονόμισμα. Αυτό το κρυπτονόμισμα χρησιμοποιείται ως πόρος και υποστηρίζει τις εκάστοτε εφαρμογές. Με αυτόν τον τρόπο δημιουργείται μία αξία για το δίκτυο και την εφαρμογή, επομένως αναπτύσσεται και πλούτος.

3. Ανάπτυξη P2P lending εφαρμογής

3.1 Παρουσίαση προβλήματος

Στις παραπάνω ενότητες εξετάστηκε το Peer-to-peer lending, η τεχνολογία των Blockchains καθώς και πώς αυτές οι δύο έννοιες θα μπορούσαν να συνδυαστούν προκειμένου να δοθούν λύσεις σε πολλά προβλήματα, αλλά και να βελτιωθούν οι εμπειρίες χρήσης πολλών πληροφοριακών συστημάτων. Στην συνέχεια θα επιχειρηθεί η διατύπωση μιας νέας περίπτωσης χρήσης peer-to-peer lending μέσω της τεχνολογίας Blockchain.

Πιο αναλυτικά, θα εξεταστεί η περίπτωση χρήσης για συναλλαγές τόσο χρηματικές όσο και κυρπτονομισματικές (του κρυπτονομίσματος Obolus) μεταξύ χρηστών ενός δικτύου blockchain, καθώς και η περίπτωση μιας πλατφόρμας crowdfunding, δηλαδή μιας πλατφόρμας όπου οι χρήστες της θα μπορούν να διατυπώνουν τις επιθυμίες τους για δάνεια και να έρχονται σε επαφή με επενδυτές.

Στόχος είναι η δημιουργία μιας κατανεμημένης αγοράς που θα βασίζεται σε ένα ιδιωτικό Blockchain και στο πρότυπο ISO 20022 το οποίο είναι υπεύθυνο για την ανταλλαγή δεδομένων μεταξύ οικονομικών ιδρυμάτων. Οι χρήστες θα μπορούν ελεύθερα και εύκολα να συναλλάσσονται χρήματα και κρυπτονομίσματα χωρίς να περιορίζονται οι καταναλωτικές τους συνήθειες από έννοιες όπως capital controls.

3.2 Περιγραφή της εφαρμογής

Στα πλαίσια της πτυχιακής εργασίας αναπτύχθηκαν δύο συστήματα. Το πρώτο ήταν μια διαδικτυακή εφαρμογή στην οποία οι χρήστες μπορούν να εγγραφούν και να αιτηθούν κάποιο δάνειο και το δεύτερο σύστημα ήταν το ηλεκτρονικό πορτοφόλι το οποίο αναπτύχθηκε για λειτουργικά συστήματα windows, linux και macOS.

Για λόγους περιορισμένου χρόνου επιλέχθηκε η ανάπτυξη πλήρων κόμβων (Full Nodes), έτσι ώστε όλοι οι κόμβοι να εκτελούν και τις τέσσερις βασικές λειτουργίες όπως και στο Bitcoin. Επομένως το προσωπικό πορτοφόλι των χρηστών εκτελεί χρέη πορτοφολίου, εξόρυξης μπλοκ, αποθηκευτικού χώρου ολόκληρου του blockchain και δρομολόγησης των χρηστών στο δίκτυο. Επίσης, δημιουργήθηκε το κρυπτονόμισμα "Obolus" το οποίο οι χρήστες μπορούν να κερδίζουν μέσω της εξόρυξης νέων μπλοκ για το Blockchain.

3.2.1 Η διαδικτυακή Εφαρμογή

Στην διαδικτυακή εφαρμογή οι χρήστες μπορούν να εγγραφούν παρέχοντας κάποιες προσωπικές πληροφορίες. Έπειτα μπορούν να κάνουν χρήση της εφαρμογής η οποία τους παρέχει τις εξής λειτουργίες:

1. Αίτηση για δάνειο. Εάν κάποιος χρήστης επιθυμεί να αιτηθεί ένα συγκεκριμένο ποσό, μπορεί να το κάνει μέσω της εφαρμογής. Θα χρειαστεί να συμπληρώσει την αντίστοιχη φόρμα όπου και θα επιλέξει τον λόγο που επιθυμεί το συγκεκριμένο δάνειο, το ποσό, το επιτόκιο που προσφέρει σε όποιον επενδύσει στο δάνειό του, το χρόνο εξόφλησης του δανείου και το νόμισμα το οποίο επιθυμεί.

2. Επένδυση σε δάνεια άλλων χρηστών. Εάν κάποιος χρήστης το επιθυμεί, μπορεί εύκολα να αναζητήσει τα ανοιχτά δάνεια άλλων χρηστών. Σε μία καρτέλα εμφανίζονται όλα τα απαραίτητα στοιχεία και ο εκάστοτε χρήστης μπορεί να αποφασίσει σε ποιό δάνειο θέλει να επενδύσει τα χρήματά του. Δεν είναι απαραίτητο ένας χρήστης να καλύψει το πλήρες ποσό, μπορεί να επενδύσει όσο αυτός επιθυμεί.

3. Μεταφορά χρημάτων και κρυπτονομισμάτων Obolus. Στα πλαίσια της πτυχιακής εργασίας δημιουργήθηκε και το κρυπτονόμισμα "Obolus". Οι χρήστες της διαδικτυακής πλατφόρμας έχουν τη δυνατότητα να επιλέξουν την peer-to-peer συναλλαγή χρημάτων ή κρυπτονομισμάτων Obolus σε κάποιον άλλο χρήστη, γνωρίζοντας μόνο την δημόσια διεύθυνσή του. Η μεταφορά κρυπτονομισμάτων "Obolus" γίνεται μέσω του Blockchain όπως ακριβώς και στο Bitcoin, η μεταφορά χρηματικών ποσών γίνεται μέσω του προτύπου ISO 20022 στο οποίο θα αναφερθούμε αργότερα.

3.2.2 Το προσωπικό πορτοφόλι των χρηστών

Αφου οι χρήστες εγγραφούν στην διαδικτυακή πλατφόρμα θα πρέπει να κατεβάσουν στον υπολογιστή του το προσωπικό τους πορτοφόλι. Αφου το εγκαταστήσουν, κατά την διάρκεια της πρώτης σύνδεσης στο λογαριασμό τους, το πορτοφόλι θα τους αποδώσει ένα ζεύγος κλειδιών (ιδιωτικό και δημόσιο κλειδί), θα δημιουργήσει μία δημόσια διεύθυνση για το χρήστη στην οποία θα μπορεί να λαμβάνει χρήματα και θα ενημερώσει την βάση δεδομένων της διαδικτυακής εφαρμογής για την δημόσια διεύθυνση του χρήστη.
Οι λειτουργίες του πορτοφολίου είναι οι εξής:

1. Έγκριση συναλλαγών: Μέσω του προσωπικού τους πορτοφολίου οι χρήστες της εφαρμογής μπορούν να επιβεβαιώνουν τις συναλλαγές που έχουν αιτηθεί μέσω της διαδικτυακής εφαρμογής. Αυτό συμβαίνει γιατί το πορτοφόλι είναι υπεύθυνο για την επικοινωνία με τους άλλους κόμβους και την δρομολόγηση των συναλλαγών.

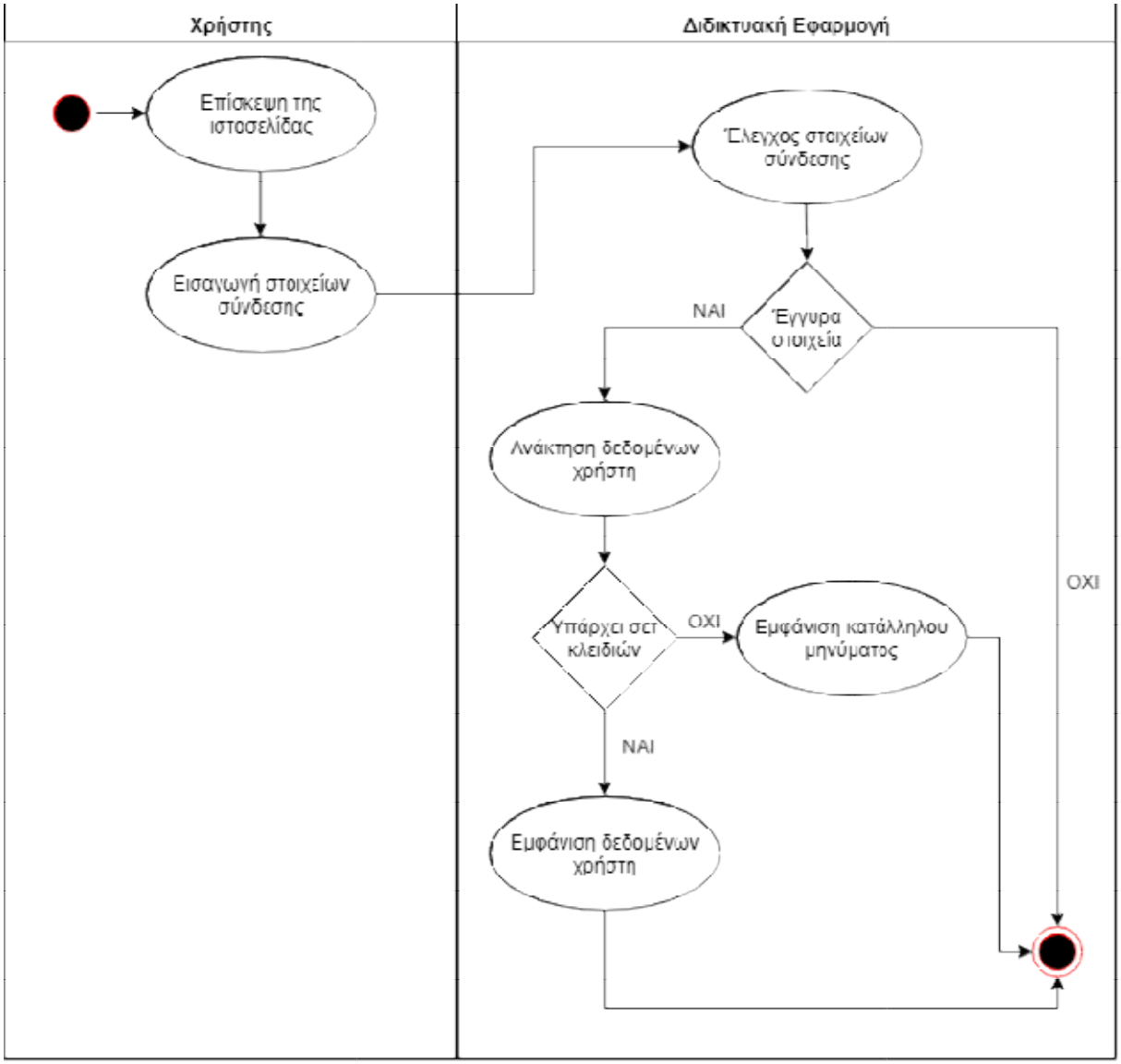
2. Αποθήκευση του Blockchain: Το προσωπικό πορτοφόλι των χρηστών είναι αρμόδιο για να τηρεί ένα πλήρες αντίγραφο του Blockchain. Όταν ένας χρήστης συνδέεται στο πορτοφόλι του, λαμβάνει από τους άλλους κόμβους του δικτύου την έκδοση της αλυσίδας και αντίστοιχα είτε ανανεώνει το Blockchain του είτε όχι.

3. Δημιουργία μπλοκ: Κάθε φορά που κάποιος χρήστης επιβεβαιώνει/ εγκρίνει μια συναλλαγή από το προσωπικό του πορτοφόλι, το πρόγραμμα δημιουργεί ένα καινούριο μπλοκ συναλλαγών το οποίο και προωθεί στο δίκτυο για να γίνει η εξόρυξη του και αργότερα να αναρτηθεί στο κύριο Blockchain.

4. Δρομολόγηση κόμβων: Όπως είδαμε και πιο πάνω το πορτοφόλι των χρηστών εκτελεί και χρέη Network Routing Node, δηλαδή είναι υπεύθυνο να επικοινωνήσει με τους υπόλοιπους κόμβους για την ανταλλαγή δεδομένων. Για το πετύχει αυτό υπάρχουν κάποιες διευθύνσεις παλαιών και σταθερών κόμβων προ-αποθηκευμένες μέσα στο πρόγραμμα, το πορτοφόλι επικοινωνεί με αυτές και ενημερώνει τους κόμβους για την δική του διεύθυνση η οποία εισέρχεται στην λίστα διαθέσιμων κόμβων. Έτσι η επικοινωνία παραμένει peer-to-peer χωρίς να παρεμβαίνει κάποιος κεντρικός υπολογιστής/ διακομιστής.

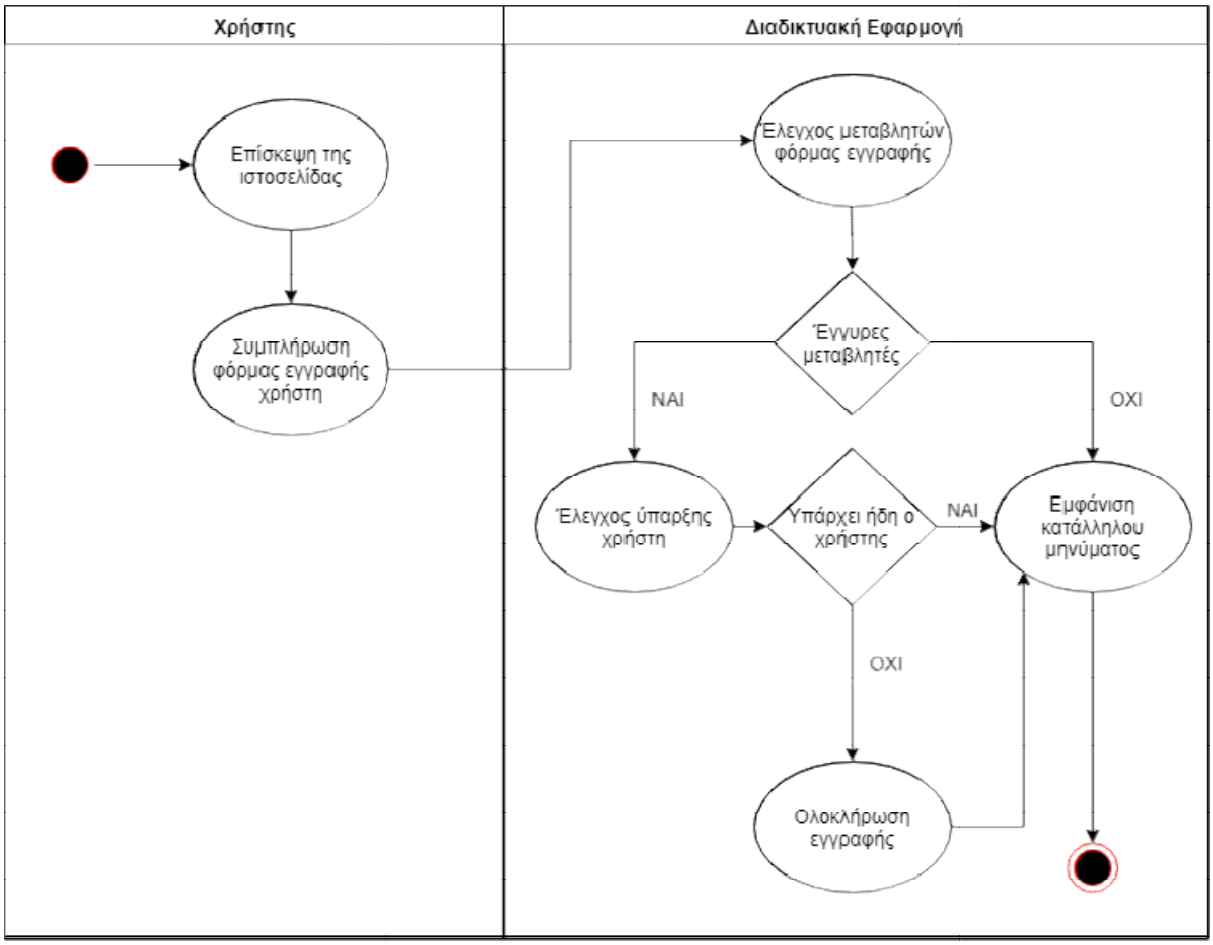
3.3 Διαγράμματα UML

Στην ενότητα αυτή παρουσιάζονται κάποια διαγράμματα UML που περιγράφουν κάποιες από τις βασικές λειτουργίες της εφαρμογής. Χαρακτηριστικά, παρατίθενται τρία διαγράμματα δραστηριοτήτων, καθώς και ένα διάγραμμα περίπτωσης χρήσης.
Στο πρώτο διάγραμμα δραστηριοτήτων απεικονίζεται ο τρόπος σύνδεσης χρήστη στη διαδικτυακή εφαρμογή.



Εικόνα 14: Διάγραμμα δραστηριοτήτων - Σύνδεση χρήστη στη διαδικτυακή εφαρμογή

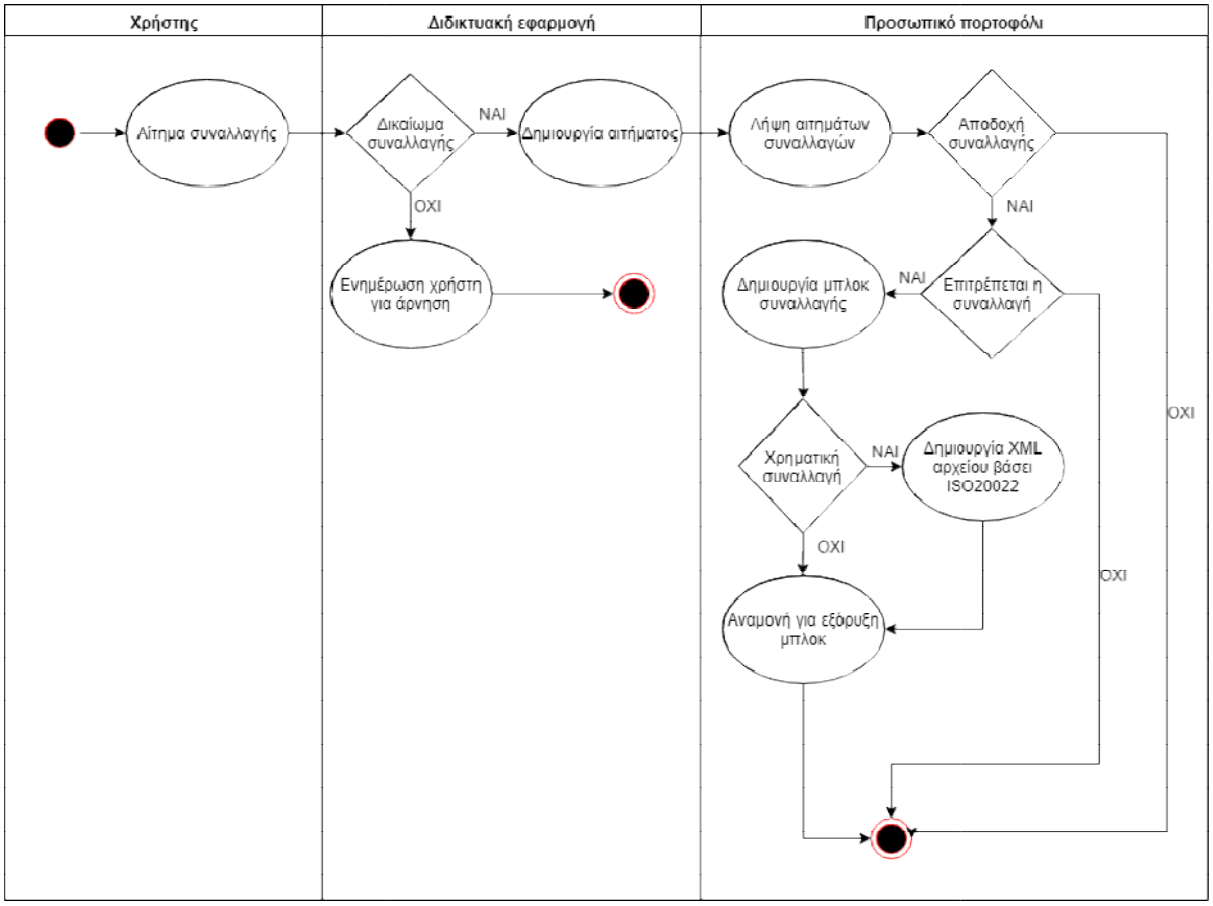
Στο δεύτερο διάγραμμα δραστηριοτήτων, φαίνεται η εγγραφή ενός νέου χρήστη στη διαδικτυακή εφαρμογή



Εικόνα 15: Διάγραμμα δραστηριοτήτων - Εγγραφή νέου χρήστη στην εφαρμογή

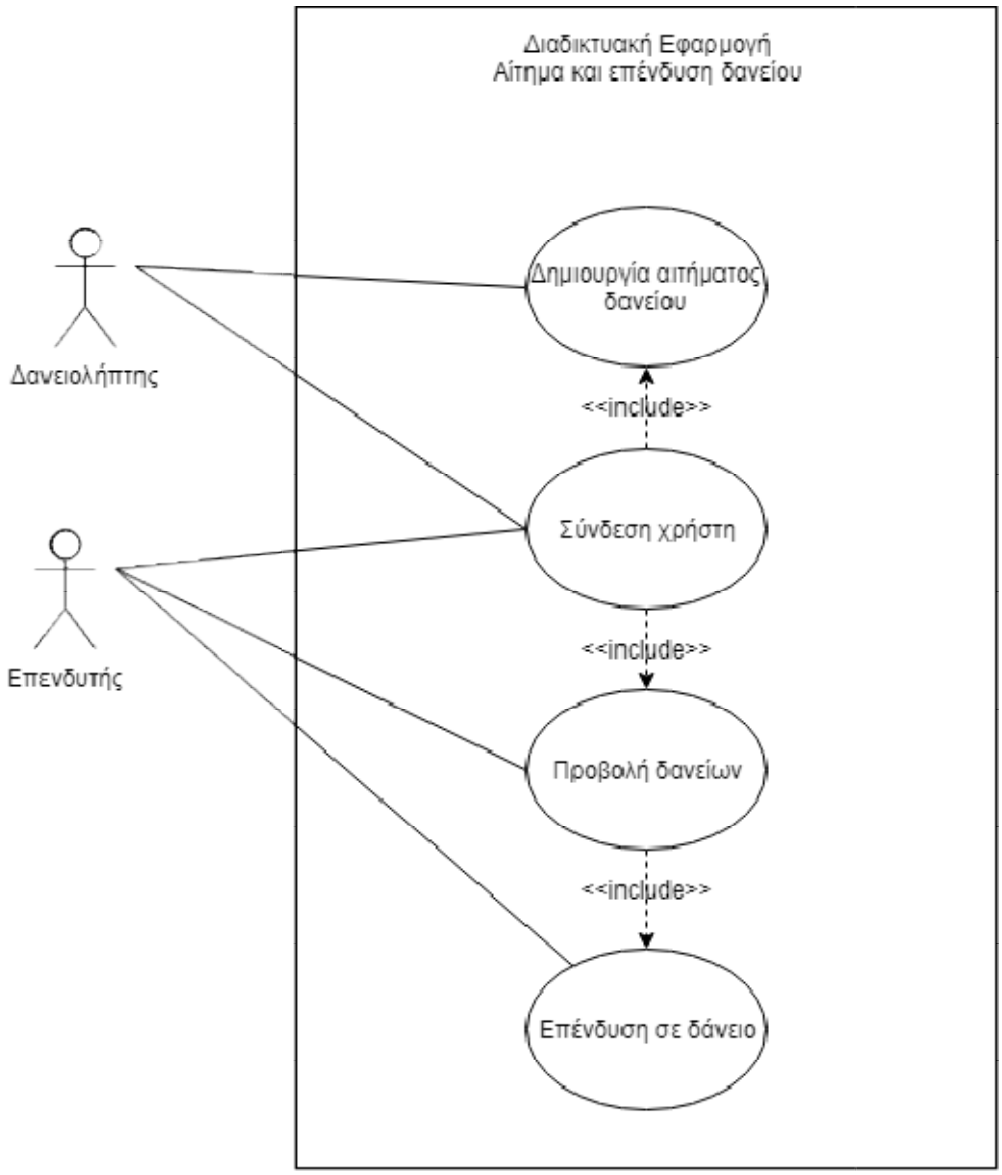
Στο παρακάτω διάγραμμα δραστηριοτήτων (activity diagram), θα εξετάσουμε την περίπτωση που κάποιος χρήστης επιθυμεί να κάνει μια συναλλαγή μέσω της διαδικτυακής εφαρμογής.

Το διάγραμμα δραστηριοτήτων δεν επηρεάζεται από τον τύπο της συναλλαγής (προσωπική συναλλαγή ή επένδυση σε δάνειο), αυτό οφείλεται στο γεγονός ότι η πλατφόρμα αντιμετωπίζει και τα δύο σαν απλές συναλλαγές.



Εικόνα 16: Διάγραμμα δραστηριοτήτων - Δημιουργία συναλλαγής

Στην επόμενη εικόνα απεικονίζεται το διάγραμμα περίπτωσης χρήσης. Στο συγκεκριμένο διάγραμμα βλέπουμε σαν δρώντες (actors) τους επενδυτές και τους δανειολήπτες. Παρατηρούμε ότι η εφαρμογή χαρακτηρίζεται από την σύνδεση των χρηστών και πως χωρίς αυτή, δεν μπορεί να ολοκληρωθεί καμία άλλη περίπτωση χρήσης. Η διαδικτυακή εφαρμογή και κατ επέκταση το πορτοφόλι των χρηστών εξαρτώνται τόσο πολύ από τη σύνδεση των χρηστών γιατί δημιουργούνται ή επιβεβαιώνονται τα ιδιωτικά/ δημόσια κλειδιά τους. Χωρίς το ζεύγος κλειδιών ένας χρήστης δεν μπορεί να κάνει χρήση της εφαρμογής.



Εικόνα 17: Διάγραμμα περίπτωσης χρήσης - Αίτημα για δάνειο και επένδυση σε δάνειο

3.4 Εργαλεία που χρησιμοποιήθηκαν

Το εργαλείο που χρησιμοποιήθηκε για την ανάπτυξη του προσωπικού πορτοφολίου ήταν το Electron[13]. Για την δικτύωση των κόμβων χρησιμοποιήθηκε η βιβλιοθήκη PeerJS [14] της JavaScript. Ακόμη έγινε χρήση της JavaScript βιβλιοθήκης jQuery [12].

Για την ανάπτυξη της διαδικτυακής εφαρμογής χρησιμοποιήθηκαν γνωστές τεχνικές web development. Η πλατφόρμα χρησιμοποιεί έναν web server, έναν file server και μια βάση δεδομένων τύπου MariaDB. Όσο για τον σχεδιασμό χρησιμοποιήθηκε HTML5, CSS, JavaScript, jQuery, η βιβλιοθήκη Bootstrap και PHP.

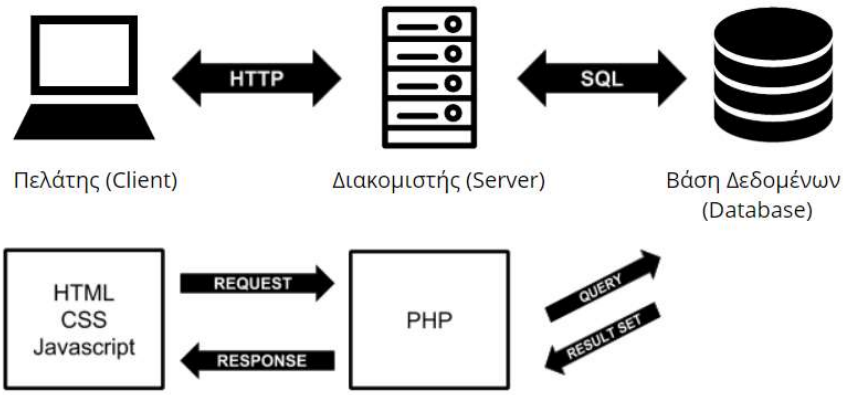
3.4.1 HTML5, CSS και PHP

Η HTML5 είναι γλώσσα σήμανσης για τον παγκόσμιο Ιστό. Είναι η επόμενη μεγάλη έκδοση της HTML. Ο σκοπός είναι η μείωση της ανάγκης για ιδιόκτητα plug-in και πλούσιες διαδικτυακές εφαρμογές όπως το Adobe Flash, το Microsoft Silverlight, το Apache Pivot, και η Sun JavaFX. Η PHP αποτελεί μια από τις πιο διαδεδομένες τεχνολογίες στο Παγκόσμιο Ιστό, καθώς χρησιμοποιείται από πληθώρα εφαρμογών και ιστότοπων. Είναι μια γλώσσα η οποία βρίσκεται σχεδόν σε κάθε διακομιστή. Διάσημες εφαρμογές που κάνουν εκτενή χρήση της PHP είναι το γνωστό Σύστημα Διαχείρισης Περιεχομένου (Content Management System, WordPress και το Drupal). Όπως θα δούμε και στο παρακάτω διάγραμμα, τα αρχεία τύπου *.html, *.css, *.js είναι αρχεία τα οποία επιβαρύνουν τον εκάστοτε χρήστη. Αντιθέτως τα αρχεία τύπου *.php είναι αρχεία τα οποία επιβαρύνουν τον διακομιστή. Επομένως, στην ανάπτυξη των διαδικτυακών εφαρμογών, οφείλουμε να επιβαρύνουμε όσο το δυνατόν λιγότερο το διακομιστή και να εκτελεί ο χρήστης τις περισσότερες λειτουργίες.

3.4.2 Μοντέλο πελάτη - διακομιστή

Στην επιστήμη των υπολογιστών το μοντέλο αρχιτεκτονικής λογισμικού πελάτη-διακομιστή (client - server) αποτελεί μία συνηθισμένη μέθοδο ανάπτυξης λογισμικού στην οποία ο πελάτης ζητά μια πληροφορία, ο διακομιστής, του την επιστρέφει. Ένας διακομιστής μπορεί να εξυπηρετήσει πολλαπλούς πελάτες.

Στο σημείο αυτό είναι σημαντικό να ξεχωρίσουμε λίγο ποια αρχεία τρέχουν στον πελάτη και ποια στον διακομιστή και πως αυτά επικοινωνούν.



Εικόνα 18: Σχέδιο αρχιτεκτονικής διαδικτυακής εφαρμογής

Όπως περιγράφει και το παραπάνω διάγραμμα, ο πελάτης μέσω του περιηγητή (web browser), όταν επιθυμεί να επισκεφθεί μία ιστοσελίδα (π.χ. di.ionio.gr) κάνει ένα HTTP αίτημα (HTTP Request) στο διακομιστή που φιλοξενεί την ιστοσελίδα.
Το Πρωτόκολλο Μεταφοράς Υπερκειμένου (HyperText Transfer Protocol, HTTP) είναι ένα πρωτόκολλο επικοινωνίας. Αποτελεί το κύριο πρωτόκολλο που χρησιμοποιείται στους φυλλομετρητές του παγκόσμιου ιστού για να μεταφέρει δεδομένα ανάμεσα σε έναν διακομιστή και έναν πελάτη. Εάν ο πελάτης ζητήσει κάποια δεδομένα τα οποία είναι αποθηκευμένα σε μία βάση δεδομένων, τότε ο διακομιστής θα τα ζητήσει από την βάση, μέσω της SQL και θα τα εμφανίσει στον client.

3.4.3 Bootstrap

Το Bootstrap είναι μια συλλογή εργαλείων ανοιχτού κώδικα (Ελεύθερο λογισμικό) για τη δημιουργία ιστοσελίδων και διαδικτυακών εφαρμογών. Περιέχει HTML και CSS για τις μορφές τυπογραφίας, κουμπιά πλοήγησης και άλλων στοιχείων του περιβάλλοντος, καθώς και προαιρετικές επεκτάσεις JavaScript. Είναι το πιο δημοφιλές πρόγραμμα στο GitHub και έχει χρησιμοποιηθεί από τη NASA και το MSNBC, μεταξύ άλλων.

Το Bootstrap έρχεται σταθερό με 940 pixel πλάτος. Εναλλακτικά, ο προγραμματιστής μπορεί να χρησιμοποιήσει μία μεταβλητού πλάτους διάταξη. Και για τις δύο περιπτώσεις, η εργαλειοθήκη έχει τέσσερις παραλλαγές για χρήση των διαφόρων ψηφισμάτων και τύπους συσκευών κινητά τηλέφωνα, πορτρέτα και το τοπία, tablets και υπολογιστές με χαμηλή και υψηλή ανάλυση. Κάθε παραλλαγή ρυθμίζει το πλάτος των στηλών.

Το Bootstrap παρέχει ένα σύνολο αρχείων τύπου css που παρέχουν βασικούς ορισμούς στυλ για όλα τα βασικά στοιχεία HTML. Αυτά παρέχουν ενιαία, σύγχρονη εμφάνιση για πίνακες, μορφοποίηση κειμένου, καθώς και για στοιχεία φορμών.

3.4.4 jQuery

Η jQuery είναι μια βιβλιοθήκη JavaScript σχεδιασμένη να απλοποιήσει την υλοποίηση σεναρίων (scripting) στη πλευρά του πελάτη (client-side) της HTML και υποστηρίζει πολλαπλούς φυλλομετρητές Ιστού [12]. Κυκλοφόρησε τον Ιανουάριο του 2006 από τον Τζον Ρέριγκ (John Resig). Χρησιμοποιείται σε πάνω από το 65% των ιστοτόπων με τη μεγαλύτερη επισκεψιμότητα.

Τα χαρακτηριστικά της jQuery είναι τα εξής:

1. DOM element επιλογές χρησιμοποιώντας την ανοιχτού κώδικα μηχανή επιλογής πολλαπλών φυλλομετρητών Sizzle.
2. DOM διάσχιση και τροποποίηση (υποστηρίζοντας CSS 1-3) χειρισμός DOM βασισμένος σε CSS επιλογείς που χρησιμοποιεί τα id και class σαν κριτήρια για να κατασκευάσει επιλογείς.
3. Events
4. Εφέ και κινητά στοιχεία
5. AJAX
6. Επεκτασιμότητα μέσω plug-ins
7. Μεθόδους συμβατότητας που είναι εγγενώς διαθέσιμα σε σύγχρονα προγράμματα περιήγησης.
8. Υποστήριξη πολλαπλών φυλλομετρητών.

3.4.5 Electron

Το Electron [13] (πρώην Atom Shell) είναι ένα framework ανοιχτού κώδικα (open source), αναπτύχθηκε και συντηρείται από το GitHub [15]. Το Electron επιτρέπει την ανάπτυξη desktop GUI εφαρμογών χρησιμοποιώντας front και back end στοιχεία. Για την ανάπτυξη back end στοιχείων χρησιμοποιείται το Node.js ενώ για το front end το Chromium. Το Electron είναι το framework με το οποίο αναπτύχθηκαν πολλές εφαρμογές ανοιχτού κώδικα όπως το GitHub Atom και το Visual Studio Code της Microsoft.

Το Electron επιλέχθηκε γιατί συνδυάζει την ευκολία του web development, κάνοντας χρήση HTML, CSS και JavaScript για το γραφικό περιβάλλον και την τεχνολογία του Node.js για την αλληλεπίδραση με τον σύστημα αρχείων (file system) του υπολογιστή.

Ακόμη, επειδή το Electron είναι βασισμένο στο Chromium, μπορούμε να κάνουμε χρήση κάποιον συγκεκριμένων στοιχείων που υπάρχουν στον φυλλομετρητή Google Chrome. Τα στοιχεία αυτά είναι οι μικροί αποθηκευτικοί χώροι session και local storage με τους οποίους διευκολύνεται η ανάπτυξη του πορτοφολίου μας, καθώς μπορούμε και αποθηκεύουμε κάποιες προσωρινές μεταβλητές. Ένας σημαντικός λόγος για τον οποίο επιλέχθηκε το Electron είναι επειδή το Chromium έχει τη δυνατότητα χρήσης του WebRTC [16]. Το WebRTC είναι ένα ανοιχτό project το οποίο παρέχει σε φυλλομετρητές και κινητές εφαρμογές Real-Time επικοινωνία. Έτσι μπορέσαμε και υλοποιήσαμε τις Peer-to-peer συνδέσεις μεταξύ των κόμβων του δικτύου Blockchain.

PeerJS

Η JavaScript βιβλιοθήκη PeerJS ουσιαστικά προσφέρει ένα εύχρηστο API για τις peer-to-peer συνδέσεις που απαιτούνται από την εφαρμογή μας. Κάνει χρήση του WebRTC και με την βοήθεια ID μπορεί και δρομολογεί τις επικοινωνίες μεταξύ των κόμβων ενός δικτύου.

Το PeerJS χρησιμοποιεί ένα διακομιστή για να μεσολαβήσει στις συνδέσεις, αλλά κανένα από τα δεδομένα που ανταλλάσσονται δεν διέρχονται από τον server. Η σύνδεση των κόμβων είναι Peer-to-peer.

3.5 ISO 20022

Το ISO 20022 [17] είναι ένα πρότυπο που αφορά την ηλεκτρονική ανταλλαγή δεδομένων μεταξύ χρηματοοικονομικών ιδρυμάτων. Περιγράφει μία αποθήκη μεταδεδομένων που περιλαμβάνει περιγραφές μηνυμάτων και επιχειρηματικών διαδικασιών. Το πρότυπο συμπεριλαμβάνει πληροφορίες που ανταλλάσσονται μεταξύ οικονομικών ιδρυμάτων, όπως χρηματικές συναλλαγές, πληροφορίες διακανονισμών, συναλλαγές χρεογράφων, συναλλαγές πιστωτικών και χρεωστικών καρτών, καθώς και άλλες οικονομικές πληροφορίες.
Το πρότυπο καλύπτει ένα μεγάλο εύρος οικονομικών μεταδεδομένων και διαδικασιών που έχουν τυποποιηθεί στον κλάδο. Τα μεταδεδομένα αυτά, αποθηκεύονται σε UML μοντέλα τα οποία συμμορφώνονται βάσει του ISO 20022 UML Profile. Το ISO 20022 χρησιμοποιείται ευρέως σε οικονομικά ιδρύματα, κάποια από αυτά είναι τα ISDA, SWIFT και VISA.

Ουσιαστικά πρόκειται για ένα γενικό framework ανταλλαγής μηνυμάτων χρηματοοικονομικών ιδρυμάτων και είναι και η προτεινόμενη κατά ISO πλατφόρμα για την ανάπτυξη όλων των οικονομικών μηνυμάτων. Το ISO 20022 αναπτύχθηκε και συντηρείται από το ISO Technical Committee, που είναι υπεύθυνο για την γενίκευση των μηνυμάτων που ανταλλάσσονται στο χώρο των τραπεζών και γενικότερα στην αγορά των χρηματοοικονομικών επιχειρήσεων και οργανισμών. Για να το πετύχει αυτό χρησιμοποιεί:

1. Μια μεθοδολογία μοντελοποίησης που βασίζεται στην UML, για να συλλάβει τις επιχειρηματικές συναλλαγές που ανταλλάσσονται ανάμεσα σε οικονομικά ιδρύματα.
2. Ένα κεντρικό λεξικό με επιχειρηματικές ορολογίες οι οποίες χρησιμοποιούνται συχνά στις επικοινωνίες μεταξύ οικονομικών επιχειρήσεων και οργανισμών.
3. Ένα σετ από προκαθορισμένους XML κανόνες που αφορούν τον σχεδιασμό, έτσι ώστε να μπορέσει να γίνει πετυχημένα η μετάφραση από XML σε UML. Να σημειωθεί πως δεν είναι απαραίτητη η χρήση της XML, αλλά είναι αυτή που προτιμάται τις περισσότερες φορές.

3.5.1 Μηνύματα πληρωμών κατά ISO 20022 [18]

Το πρότυπο παρέχει μηνύματα για συναλλαγές και πληρωμές τύπου:

1. Πελάτης προς τράπεζα (Customer-to-bank)
2. Τράπεζα προς τράπεζα (Bank-to-bank)
3. Δημιουργία αναφορών (Reporting)

Παρακάτω αναφέρονται οι αντίστοιχοι κωδική μηνυμάτων:

acmt – Account Management
admi – Administration
caaa – Acceptor to Acquirer Card Transactions
camt – Cash Management
catm – Terminal Management
pacs – Payments Clearing & Settlement
pain – Payments Initiation
reda – Reference Data
seev – Securities Events
semt – Securities Management
sese – Securities Settlement
setr – Securities Trade
trea – Treasury
tsmt – Trade Services Management
tsin – Trade Services Initiation

Παράδειγμα, μια οικονομική συναλλαγή όπου ένας χρήστης επιθυμεί να μεταφέρει χρήματα σε έναν άλλο χρήστη.

Πριν το πρότυπο 20022, ο χρήστης που επιθυμεί να πληρώσει, θα έπρεπε να ενημερώσει την τράπεζά του για την συναλλαγή και να καταθέσει τα χρήματά του στον τραπεζικό λογαριασμό του άλλου χρήστη. Η παραπάνω διαδικασία είναι χρονοβόρα, ειδικότερα εάν πρόκειται για δύο διαφορετικές τράπεζες.
Κάνοντας χρήση του προτύπου ISO 20022, δημιουργείται ένα μήνυμα έναρξης πληρωμής (pain.008.001.03) το οποίο αποστέλλεται από τον πρώτο χρήστη στον δεύτερο.
Στην περίπτωση ενός θετικού σεναρίου, τα μηνύματα που θα ανταλλάσσονταν θα ήταν τα παρακάτω:

1. CustomerDirectDebitInitiantion μήνυμα αποστέλλεται από τον πιστωτή
2. FItoFICustomerDirectDebit μήνυμα αποστέλλεται από το ένα οικονομικό ίδρυμα στο άλλο
3. Η θετική απάντηση CustomerPaymentStatusReport και FIToFIPaymentStatusReport μήνυμα αποστέλλονται για να ενημερωθεί ο παραλήπτης πως το αίτημα επεξεργάζεται.

Ένα παράδειγμα μηνύματος έναρξης πληρωμής θα είχε την ακόλουθη μορφή:

```
<?xml version="1.0" encoding="UTF-8"?>
<?valid true?>
<?description Sample Message?>
<Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.008.001.03">
        <CstmrDrctDbtInitn>
                <GrpHdr>
                <MsgId>CAVAY1234</MsgId>
                <CreDtTm>2013-06-02T14:25:00</CreDtTm>
                <NbOfTxs>1</NbOfTxs>
                <CtrlSum>985</CtrlSum>
                <InitgPty>
                <Nm>Smith</Nm>
                <PstlAdr>
                <StrtNm>Virginia Lane</StrtNm>
                <BldgNb>3</BldgNb>
                <PstCd>W12 8QT</PstCd>
                <TwnNm>London</TwnNm>
                <Ctry>GB</Ctry>
                </PstlAdr>
                <CtctDtls>
                <Nm>J. Thompson</Nm>
                <EmailAdr>Thompson@Smith.com</EmailAdr>
                </CtctDtls>
                </InitgPty>
                </GrpHdr>
               <PmtInf>
                <PmtInfId>CAVAY/88683</PmtInfId>
                <PmtMtd>DD</PmtMtd>
                <BtchBookg>false</BtchBookg>
                <ReqdColltnDt>2013-07-13</ReqdColltnDt>
                <Cdtr>
                <Nm>Smith</Nm>
                <PstlAdr>
                <StrtNm>Virginia Lane</StrtNm>
                <BldgNb>3</BldgNb>
                <PstCd>W12 8QT</PstCd>
                <TwnNm>London</TwnNm>
                <Ctry>GB</Ctry>
                </PstlAdr>
                </Cdtr>
                <CdtrAcct>
```

```
                <Id>
                <Othr>
                <Id>789123</Id>
                </Othr>
                </Id>
                </CdtrAcct>
                <CdtrAgt>
                <FinInstnId>
                <BICFI>AAAAUS29</BICFI>
                </FinInstnId>
                </CdtrAgt>
                <DrctDbtTxInf>
                <PmtId>
                <EndToEndId>AY090327/456</EndToEndId>
                </PmtId>
                <PmtTpInf>
                <InstrPrty>NORM</InstrPrty>
                <SvcLvl>
                <Prtry>VERPA-1</Prtry>
                </SvcLvl>
                <SeqTp>OOFF</SeqTp>
                </PmtTpInf>
                <InstdAmt Ccy="GBP">985</InstdAmt>
                <ChrgBr>SHAR</ChrgBr>
                <DrctDbtTx>
                <PreNtfctnId>Smith2435/2013</PreNtfctnId>
                <PreNtfctnDt>2013-06-08</PreNtfctnDt>
                </DrctDbtTx>
                <DbtrAgt>
                <FinInstnId>
                <BICFI>CCCCUS27</BICFI>
                </FinInstnId>
                </DbtrAgt>
                <Dbtr>
                <Nm>Lee</Nm>
                <PstlAdr>
                <StrtNm>Cross Road</StrtNm>
                <BldgNb>45</BldgNb>
                <PstCd>E56 7HY</PstCd>
                <TwnNm>London</TwnNm>
                <Ctry>GB</Ctry>
                </PstlAdr>
                </Dbtr>
                <DbtrAcct>
                <Id>
                <Othr>
                <Id>789101</Id>
                </Othr>
                </Id>
                </DbtrAcct>
                <RmtInf>
               <Ustrd>CAR INSURANCE PREMIUM</Ustrd>
                </RmtInf>
                </DrctDbtTxInf>
                </PmtInf>
        </CstmrDrctDbtInitn>
</Document>
```

Ενώ ένα μήνυμα αναφοράς θα είχε την εξής μορφή:

```
<?xml version="1.0" encoding="UTF-8"?>
<?valid true?>
<?description Sample CustomerPaymentStatusReport to acknowledge that the message passed
validation and was accepted based on the customer profile.?>
<Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.04">
        <CstmrPmtStsRpt>
                <GrpHdr>
                <MsgId>BBBB/120928-PSR/001</MsgId>
                <CreDtTm>2013-09-28T14:09:00</CreDtTm>
                <InitgPty>
                <Nm>Smith</Nm>
                <PstlAdr>
                <StrtNm>Virginia Lane</StrtNm>
                <BldgNb>3</BldgNb>
                <PstCd>W12 8QT</PstCd>
                <TwnNm>London</TwnNm>
                <Ctry>GB</Ctry>
                </PstlAdr>
                </InitgPty>
                <DbtrAgt>
                <FinInstnId>
                <BICFI>AAAAUS29</BICFI>
                </FinInstnId>
                </DbtrAgt>
                </GrpHdr>
                <OrgnlGrpInfAndSts>
                <OrgnlMsgId>CAVAY1234</OrgnlMsgId>
                <OrgnlMsgNmId>pain.008.001.03</OrgnlMsgNmId>
                <OrgnlCreDtTm>2013-09-28T14:07:00</OrgnlCreDtTm>
                <OrgnlNbOfTxs>1</OrgnlNbOfTxs>
                <OrgnlCtrlSum>985</OrgnlCtrlSum>
                <GrpSts>ACCP</GrpSts>
                </OrgnlGrpInfAndSts>
        </CstmrPmtStsRpt>
</Document>
```

3.6 Δυσκολίες & τεχνικές προκλήσεις

Όπως κάθε εκπόνηση εργασίας, έτσι και η συγκεκριμένη πτυχιακή είχε κάποιες δυσκολίες και ιδιαίτερα τεχνικές προκλήσεις. Μία από τις μεγαλύτερες προκλήσεις ήταν η δικτύωση των κόμβων και η αναγνώριση των κόμβων μεταξύ τους στο δίκτυο. Απαιτήθηκε λεπτομερή μελέτη και αναζήτηση αντίστοιχων υλοποιήσεων, εκτός του Bitcoin. Επόμενη τεχνική πρόκληση ήταν η επιλογή του εργαλείου ανάπτυξης του πορτοφολίου. Όταν άρχισε η ανάπτυξη της εφαρμογής, είχε επιλεγεί το πρόγραμμα Visual Studio και ως γλώσσα προγραμματισμού η C Sharp (C#). Αφού βρέθηκε η δυνατότητα υλοποίησης μέσω του Electron, έγινε αλλαγή και η ανάπτυξη της εφαρμογής άρχισε ξανά από την αρχή κάνοντας χρήση HTML,CSS και JavaScript.

Κύριος λόγος για την αλλαγή του τρόπου ανάπτυξης της εφαρμογής ήταν η υποστήριξη του Chromium από το Electron. Έτσι, επιλύθηκαν αρκετά από τα παραπάνω προβλήματα όπως η δικτύωση των κόμβων. Η τεχνολογία WebRTC επέτρεψε την επικοινωνία peer-to-peer.

Πριν σχεδιαστεί η υλοποίηση μέσω δύο διαφορετικών εφαρμογών (μια διαδικτυακή και μια προς εγκατάσταση στον υπολογιστή), υπήρχε έντονος προβληματισμός σχετικά με τον τρόπο υλοποίησης. Οι χρήστες έπρεπε να μπορούν να χρησιμοποιούν την εφαρμογή και χωρίς να βρίσκονται στο σπίτι, όμως η ανάπτυξη ενός lightweight πορτοφολίου απαιτεί την ύπαρξη ενός ενεργού Blockchain και μεγάλη προγραμματιστική εμπειρία.

3.7 Ευκαιρίες για βελτίωση

Όπως κάθε πληροφοριακό σύστημα, έτσι και αυτό της συγκεκριμένη πτυχιακής έχει περιθώρια για βελτίωση. Λόγω περιορισμένου χρόνου και περιορισμένων πόρων, δεν υπήρχε η δυνατότητα δοκιμής της εφαρμογής σε πραγματικό περιβάλλον. Τόσο η διαδικτυακή εφαρμογή, όσο και το πρόγραμμα του προσωπικού πορτοφολίου έχουν περιθώρια για βελτίωση, εικαστικά αλλα και λειτουργικά.

3.7.1 Η διαδικτυακή εφαρμογή

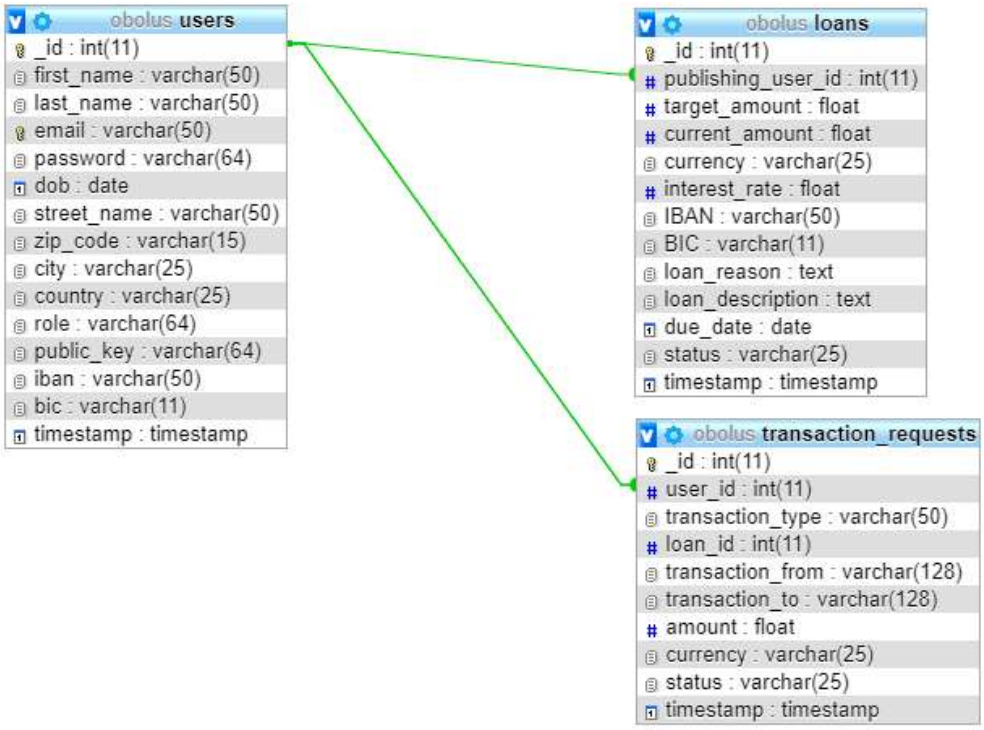
Οι βελτιώσεις της διαδικτυακής εφαρμογής θα πρέπει να εστιάσουν στην ασφάλεια. Η εφαρμογή κάνει χρήση της PHP εξαιτίας της πειραματικής της φύσης, σύγχρονα συστήματα χρηματοοικονομικών υπηρεσιών, χρησιμοποιούν πιο ασφαλές γλώσσες προγραμματισμού. Ακόμη, θα πρέπει να γίνει εκτενέστατη δοκιμή του συστήματος για τυχόν λάθη τα οποία δεν έχουν παρουσιαστεί έως τώρα. Μία ακόμη μεγάλη βελτίωση της διαδικτυακής εφαρμογής θα ήταν δημιουργία ενός εργαλείου ελέγχου της πιστοληπτικής ικανότητας των χρηστών. Έτσι, οι υπόλοιποι χρήστες που επιθυμούν να επενδύσουν τα χρήματά τους, θα είχαν μια καλύτερη εικόνα για τους δανειστές.

3.7.2 Το προσωπικό πορτοφόλι

Το προσωπικό πορτοφόλι μπορεί να βελτιωθεί με πολλούς τρόπους. Καταρχάς θα μπορούσε να γραφτεί ο κατάλληλος κώδικας και να δημιουργούνται παραπάνω από ένα σετ κλειδιών στο κάθε χρήστη. Ακόμη θα μπορούσε να γίνει βελτιστοποίηση της εφαρμογής σχετικά με την υλοποίησή της. Ένας τρόπος είναι η ανάθεσή των συναρτήσεων σε διαφορετικά renderer processes, με αποτέλεσμα να λειτουργεί πιο σταθερά η εφαρμογή. Τέλος, θα μπορούσε να αναπτυχθεί ένας καινούργιος τύπος πορτοφολίου, όπως παραδείγματος χάρη ένα lightweight πορτοφόλι.

3.8 Αρχιτεκτονική

Η αρχιτεκτονική που ακολουθήθηκε για την ανάπτυξη της διαδικτυακής εφαρμογής ήταν η γνωστή αρχιτεκτονική πελάτη - διακομιστή (βλ. υποενότητα "Μοντέλο πελάτη - διακομιστή"). Στο ακόλουθο σχήμα φαίνεται το σχεσιακό διάγραμμα της βάσης δεδομένων που χρησιμοποιείται στην διαδικτυακή εφαρμογή.

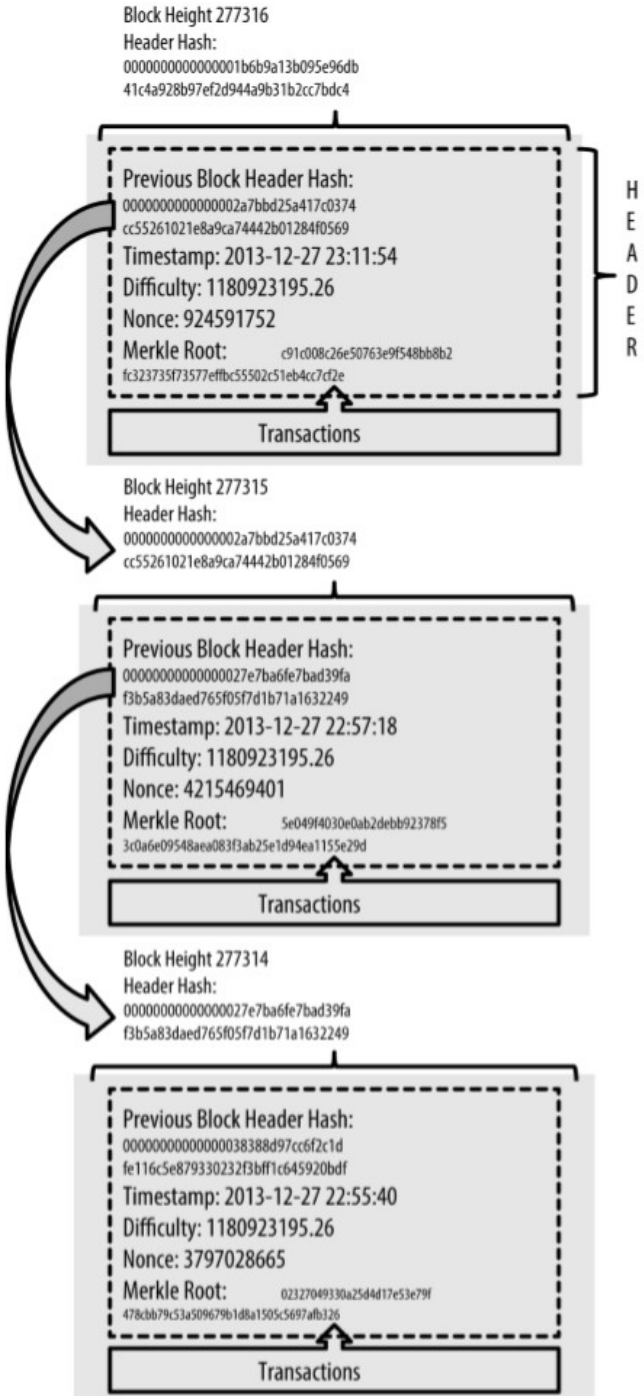


Εικόνα 19: Σχεσιακό διάγραμμα βάσης δεδομένων διαδικτυακής εφαρμογής

Όπως φαίνεται και από το παραπάνω διάγραμμα, τα δεδομένα που αποθηκεύονται είναι τα απολύτως απαραίτητα. Αυτό οφείλεται στο γεγονός ότι το ιστορικό των συναλλαγών βρίσκεται αποθηκευμένο στο Blockchain και δεν χρειάζεται να υπάρχουν κάπου αλλού.

Οι παραπάνω τρεις πίνακες αποθηκεύουν τους χρήστες της πλατφόρμας, τα δημόσια δάνεια που έχουν αιτηθεί οι χρήστες της πλατφόρμας, καθώς και τα αιτήματα συναλλαγών που έχουν κάνει οι χρήστες μέσω της διαδικτυακής εφαρμογή. Στο σημείο αυτό να τονίσουμε πως ο πίνακας transaction_requests δεν αποθηκεύει τις συναλλαγές, αλλα μόνο τα αιτήματα συναλλαγών, έτσι ώστε να γίνει η έγκρισή τους μέσω του πορτοφολίου αργότερα.

Όσο για την αρχιτεκτονική του Blockchain, θα πρέπει να αναφερθούμε στον *Antonopoulos Andreas* [3], ο οποίο περιγράφει την αρχιτεκτονική του Blockchain ως μια οργανωμένη δομή δεδομένων και συγκεκριμένα μια οπισθο-συνδεδεμένη λίστα η οποία αποτελείται από συναλλαγές.



Εικόνα 20: Αρχιτεκτονική του Blockchain - *Antonopoulos, Andreas M. Mastering Bitcoin*

3.9 Ανάπτυξη της εφαρμογής

Στα πλαίσια της πτυχιακής εργασίας αναπτύχθηκαν δύο εφαρμογές οι οποίες επικοινωνούν μέσω μιας βάσης δεδομένων. Η πρώτη εφαρμογή είναι μια διαδικτυακή εφαρμογή στην οποία οι χρήστες έχουν πρόσβαση από σταθερά αλλα και κινητά μέσα (όπως π.χ. κινητά τηλέφωνα), και μπορούν να διαχειρίζονται τον προσωπικό τους λογαριασμό και να κάνουν αιτήματα συναλλαγών. Η δεύτερη εφαρμογή αποτελεί το προσωπικό πορτοφόλι, που είναι ένα λογισμικό για λειτουργικά συστήματα windows, linux και macOS και ουσιαστικά αποτελεί έναν κόμβο του Blockchain δικτύου. Από το συγκεκριμένο λογισμικό, οι χρήστες, μπορούν και επιβεβαιώνουν τις συναλλαγές που έχουν αιτηθεί μέσω της διαδικτυακής πλατφόρμας.
Παρακάτω θα αναλύσουμε την διαδικασία ανάπτυξης και των δύο εφαρμογών, καθώς και τον τρόπο επικοινωνίας τους.

3.9.1 Η διαδικτυακή εφαρμογή

Η διαδικτυακή εφαρμογή αποτελείται από πολλά αρχεία τύπου HTML, CSS και PHP. Στην συγκεκριμένη ενότητα δεν μπορούμε να αναλύσουμε ολόκληρο τον κώδικα, όλων των αρχείων. Παρακάτω έχουν επιλεχθεί κάποιες ενότητες οι οποίες καλύπτουν τις βασικές λειτουργίες της εφαρμογής. Πολλές από τις λειτουργίες επαναλαμβάνονται συχνά, όπως π.χ. η σύνδεση με τη βάση δεδομένων, ή τα διάφορα ερωτήματα (queries) στη βάση. Ακόμη, γίνεται μια σύντομη αναπαράσταση της HTML χωρίς να παρουσιάζονται όλα τα αρχεία.

3.9.1.1 HTML και CSS

Κατα την ανάπτυξη της διαδικτυακής εφαρμογής έγινε χρήση των βιβλιοθηκών Bootstrap[11] και jQuery[12], οι οποίες συμπεριλαμβάνονται στον κώδικα με το εξής τρόπο:

```
<!-- Bootstrap core JavaScript
================================================== -->
<!-- Placed at the end of the document so the pages load faster -->
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/
bootstrap.min.css">
<!-- Latest compiled JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<!-- jQuery library -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
```

Εικόνα 21: Εισαγωγή βιβλιοθηκών Bootstrap και jQuery

Επίσης για κάθε αρχείο HTML, δημιουργήθηκε και το αντίστοιχο αρχείο CSS για εξατομικευμένες αλλαγές στην εμφάνιση. Για παράδειγμα στην αρχική σελίδα συμπεριλαμβάνεται και ο κώδικας του αρχείου index.css

```
<link rel="stylesheet" type="text/css" href="css/index.css">
```

Εικόνα 22: Εισαγωγή αρχείου τύπου Cascade Style Sheet (CSS)

Το γεγονός ότι συμπεριλήφθηκε η βιβλιοθήκη του Bootstrap, βοήθησε πολύ κατά την διαδικασία της ανάπτυξης αφού μας παρείχε ένα μεγάλο αριθμό προσχεδιασμένων στοιχείων της HTML. Χρήσιμη ήταν επίσης η δυνατότητα της HTML5 να περιορίζει τα πεδία εισαγωγής (input fields) σε συγκεκριμένο τύπο, παραδείγματος χάρη, το ακόλουθο πεδίο εισαγωγής περιορίζεται στον τύπο email:

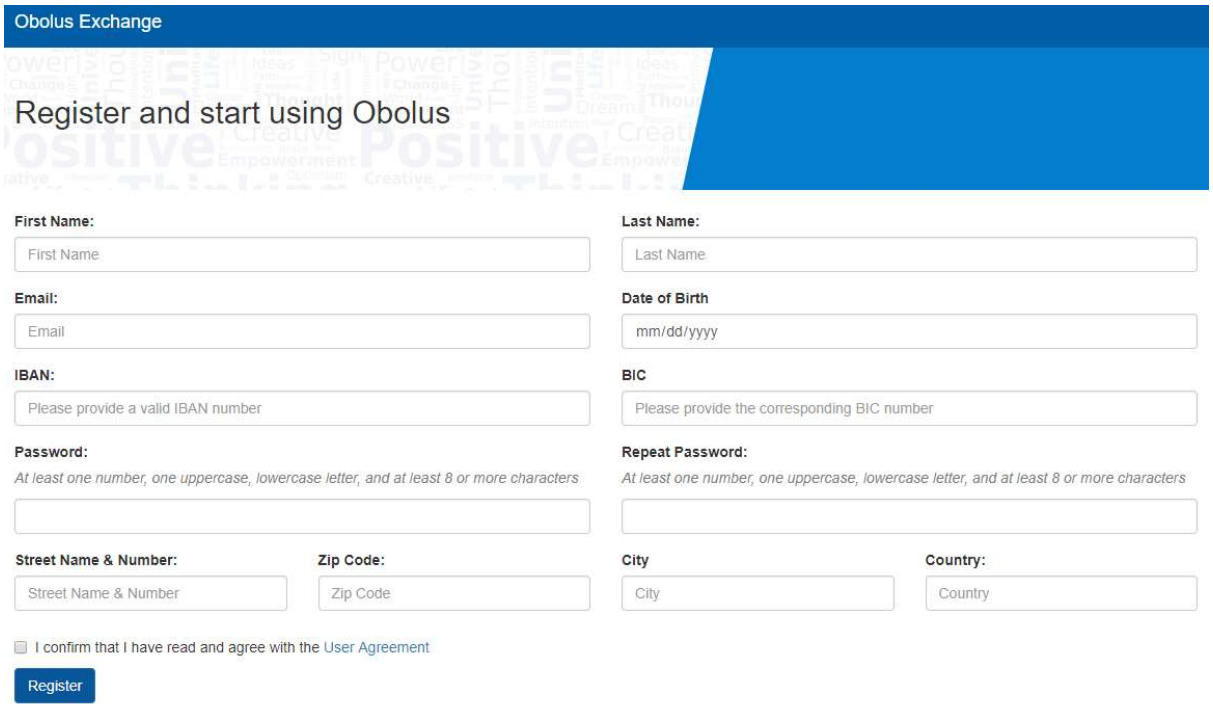
```
<div class="form-group">
  <input id="email" name="email" type="email" placeholder="Email" class="form-control"
  required="true">
</div>
```

Εικόνα 23: Τύπος εισαγωγής "email"

Πρέπει να τονίσουμε πως ο παραπάνω τρόπος διαχωρισμού των τύπων εισαγωγής δεν είναι ασφαλής, διότι ο χρήστης μπορεί εύκολα να τον αλλάξει μέσω του φυλλομετρητή του. Η ασφαλή επαλήθευση των στοιχείων θα ακολουθήσει στο υποκεφάλαιο "Διαχείριση φόρμας δεδομένων (Form Handling)".

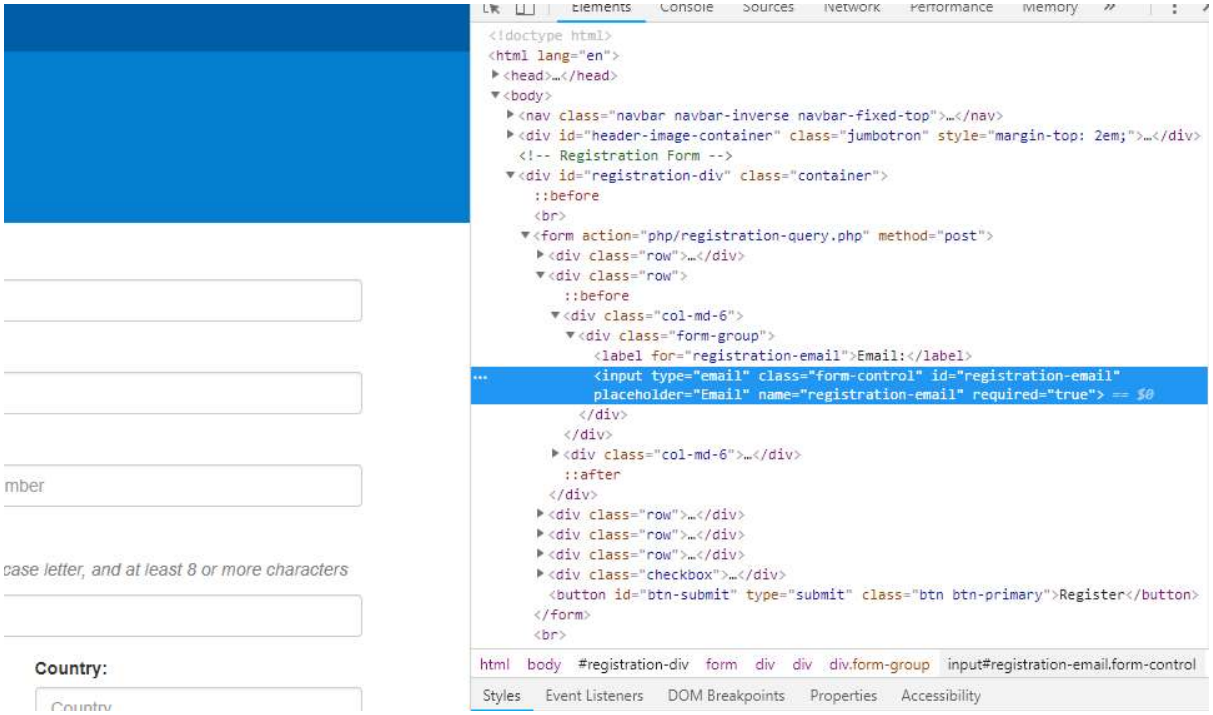
3.9.1.2 Διαχείριση φόρμας δεδομένων (Form Handling)

Ας παρατηρήσουμε λίγο την φόρμα εγγραφής νέων χρηστών (Εικόνα 24). Θα δούμε πως τα δεδομένα που ζητούνται είναι διαφόρων τύπων. Υπάρχουν πεδία αλφαριθμητικά, email, ημερομηνίας ακόμη και κωδικού πρόσβασης.



Εικόνα 24: Φόρμα εγγραφής νέων χρηστών

Όπως προαναφέρθηκε, η ιδιότητα της HTML5 να ξεχωρίζει τα πεδία εισόδου σε τύπους, δεν σημαίνει απαραίτητα πως τα δεδομένα που θα περάσουν στον διακομιστή θα είναι του ιδίου τύπου. Έχοντας κάποιες βασικές γνώσεις πληροφορικής, κάποιος κακόβουλος χρήστης θα μπορούσε να αλλάξει τον κώδικα HTML στο φυλλομετρητή έτσι ώστε η φόρμα να δέχεται και άλλους τύπους δεδομένων.



Εικόνα 25: Προεπισκόπηση και αλλαγή τύπου εισαγωγής μέσω του φυλλομετρητή

Για να μπορέσουμε να προστατέψουμε αποτελεσματικά την εφαρμογή μας από κακόβουλες αλλαγές στους τύπους των δεδομένων εισόδου και να αποτρέψουμε επιθέσεις τύπου SQL Injection, μπορούμε να ελέγχουμε τα δεδομένα που αποστέλλονται.
Ας εξετάσουμε το αρχείο registration-query.php. Το αρχείο αυτό δέχεται τα δεδομένα από την φόρμα εγγραφή νέων χρηστών του αρχείου registration.html μέσω της μεθόδου POST.

```php
// I introduced two functions strip_tags() and trim(). The strip_tags() function
strips all HTML and PHP tags from a variable,the trim() function just strips any white
space from beginning and end of the string.

$first_name = strip_tags(trim($_POST["first-name"]));
$last_name = strip_tags(trim($_POST["last-name"]));
if (filter_var($_POST["registration-email"], FILTER_VALIDATE_EMAIL)){
    $email = $_POST["registration-email"];
} else {
    $email = NULL;
}
$dob = strip_tags(trim($_POST["date-of-birth"]));
$iban = strip_tags(trim($_POST["registration-iban"]));
$bic = strip_tags(trim($_POST["registraion-bic"]));
$registration_password = strip_tags($_POST["registration-password"]);
// We hash the password to store it inside the database
$registration_password_hashed = hash('sha256', $registration_password);

$street_name = strip_tags(trim($_POST["street-name"]));
$zip_code = strip_tags(trim($_POST["zip-code"]));
$city = strip_tags(trim($_POST["city"]));
$country = strip_tags(trim($_POST["country"]));
```

Εικόνα 26: Έλεγχος δεδομένων εισόδου της φόρμας εγγραφής στο αρχείο registration-query.php

Ας επικεντρωθούμε στις δύο συναρτήσεις που αναφέρονται και στα σχόλια της PHP. Η συνάρτηση strip_tags(), αφαιρεί όλους τους χαρακτήρες-ετικέτες (tags) της HTML και PHP που ενδεχομένως κάποιος χρήστης έχει εισάγει. Η συνάρτηση trim(), αφαιρεί τους κενούς χαρακτήρες μέσα από τα δεδομένα εισόδου της φόρμας εγγραφής.
Στο όρισμα της συνάρτησης trim(), βλέπουμε τις μεταβλητές από την φόρμα εγγραφής.

Για να ελέγξουμε το πεδίο email, χρησιμοποιούμε μια συνάρτηση της PHP, τη filter_var() και ελέγχουμε αν το όρισμα είναι τύπου email:

```
if (filter_var($_POST["registration-email"], FILTER_VALIDATE_EMAIL)){
    $email = $_POST["registration-email"];
} else {
    $email = NULL;
}
```

Εικόνα 27: Έλεγχος μεταβλητής email

Ακόμη, την τιμή της μεταβλητής registration_password (κωδικού πρόσβασης), δεν χρειάζεται να ελέγξουμε για ειδικούς χαρακτήρες, μιας και αυτοί είναι επιθυμητοί σε κωδικούς πρόσβασης γιατί αυξάνουν την ασφάλειά τους. Αυτό που κάνουμε είναι να εφαρμόσουμε μια κρυπτογραφική συνάρτηση τύπου hash και συγκεκριμένα SHA-256, έτσι ώστε όταν αποθηκευτεί ο κωδικός στην βάση δεδομένων, να μην είναι φανερός. Ακόμη και ο διαχειριστής της βάσης δεδομένων δεν θα είναι σε θέση να διαβάζει τους κωδικούς πρόσβασης.

Στην σειρά 26 του αρχείου registration-query.php, βλέπουμε τη μεταβλητή role, αυτή η μεταβλητή ανατίθεται από το σύστημα, και ουσιαστικά ορίζει τον ρόλο του χρήστη που εγγράφεται στην πλατφόρμα. Η τιμή USR1001 είναι κωδικός της εφαρμογής για τον ρόλο/ ιδιότητα του απλού χρήστη (Εικόνα 28).

```
21    $street_name = strip_tags(trim($_POST["street-name"]));
22    $zip_code = strip_tags(trim($_POST["zip-code"]));
23    $city = strip_tags(trim($_POST["city"]));
24    $country = strip_tags(trim($_POST["country"]));
25
26    $role = 'USR1001';
```

Εικόνα 28: Η μεταβλητή role

Με τον τρόπο που είδαμε παραπάνω, έγινε ο έλεγχος των δεδομένων εισόδου της φόρμας εγγραφής.

3.9.1.3 Επικοινωνία με τη βάση δεδομένων

Η διαδικτυακή εφαρμογή, σε αντίθεση με τον κόμβο του Blockchain, χρησιμοποιεί μια βάση δεδομένων για την αποθήκευση των δεδομένων των χρηστών, των δανείων και των αιτημάτων συναλλαγών. Ακόμη, αποθηκεύονται τα αιτήματα συναλλαγών των χρηστών, χωρίς όμως το Blockchain να εξαρτάται από αυτά.

Αν εξετάσουμε το αρχείο registration-query.php, θα παρατηρήσουμε την εξής γραμμή κώδικα:

```
2    // Require the file that handles the database connection credentials.
3    require 'db_credentials.php';
```

Εικόνα 29: Εισαγωγή δεδομένων σύνδεσης με την βάση δεδομένων

Η εντολή require στη γραμμή 3, ουσιαστικά απαιτεί το αρχείο db_credentials.php και η εφαρμογή δεν θα ανταποκριθεί στο χρήστη σε περίπτωση που δεν είναι διαθέσιμο.
Το αρχείο db_credentials.php δεν περιέχει τίποτα παραπάνω από τα στοιχεία σύνδεσης του εξυπηρετητή με τη βάση δεδομένων. Το στοιχεία αυτά τα διατηρούμε σε ξεχωριστό αρχείο, γιατί σε περίπτωση που αλλάξει ένα από αυτά, έχουμε ευκολότερη πρόσβαση για να το διορθώσουμε.

```php
<?php
    $servername = "localhost";
    $db_username = "root";
    $db_password = "";
    $db_name = "obolus";
?>
```

Εικόνα 30: Το αρχείο db_credentials.php

Ας δούμε λοιπόν πώς γίνεται η επικοινωνία με τη βάση δεδομένων. Έστω ότι ελέγχουμε το αρχείο registration-query.php. Στις γραμμές 28 - 48 βλέπουμε την σύνδεση στη βάση δεδομένων και την αποθήκευση των δεδομένων εισόδου της φόρμας εγγραφής, στον αντίστοιχο πίνακα της βάσης μας (Εικόνα 31).

Στις σειρές 28 - 33, δημιουργείται η σύνδεση στη βάση δεδομένων και χρησιμοποιούνται οι μεταβλητές του αρχείου db_credentials.php. Σε περίπτωση που η σύνδεση δεν ήταν επιτυχής, επιστρέφεται ένα μήνυμα λάθους.

```php
// Create connection
$conn = new mysqli($servername, $db_username, $db_password, $db_name);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: ".$conn->connect_error);
}

$sql = "INSERT INTO users (first_name, last_name, email, password, dob, street_name,
    zip_code, city, country, role, iban, bic)
VALUES ('".$first_name."','".$last_name."','".$email."','".$
    registration_password_hashed."','".$dob."','".$street_name."','".$zip_code."','".$
    city."','".$country."','".$role."','".$iban."','".$bic."')";

if ($conn->query($sql) === TRUE) {
    // echo "New record created successfully";
    $conn->close();
    header("Location: ../php/message.php?msg=usrsuccess");
    die("User successfully created!");
} else {
    // echo "Error: ".$sql."<br>".$conn->error;
    $conn->close();
    header("Location: ../php/message.php?msg=usrerror");
    die("ERROR, user could not be created!");
}
```

Εικόνα 31: Σύνδεση στη βάση δεδομένων και αποθήκευση στοιχείων χρήστη

Στην αντίθετη περίπτωση, που η σύνδεση με τη βάση δεδομένων ήταν επιτυχής, δημιουργείται ένα ερώτημα (query), το οποίο δεν είναι τίποτα άλλο από μια αλφαριθμητική μεταβλητή. Έπειτα (στις γραμμές 38 - 48) γίνεται η αποστολή του query προς την βάση δεδομένων, για να αποθηκευτούν τα στοιχεία του νέου χρήστη. Στην περίπτωση που τα στοιχεία αποθηκεύτηκαν επιτυχώς, εκτελούνται οι γραμμές 39 - 42 του κώδικα, αλλοιώς εκτελούνται οι γραμμές 44 - 47 και επιστρέφονται τα αντίστοιχα μηνύματα.

3.9.1.4 Διαχείριση μηνυμάτων

Η αλληλεπίδραση της εφαρμογής με τους χρήστες είναι πολύ σημαντική. Αυτός είναι και ο λόγος για τον οποίο δεν μπορεί να λείπει η συνάρτηση διαχείρισης μηνυμάτων. Για να μπορέσουμε λοιπόν να προβάλουμε διάφορα μηνύματα στους χρήστες της πλατφόρμας, δημιουργήθηκε το αρχείο message.php. Για να προβάλουμε ένα μήνυμα από οποιοδήποτε αρχείο PHP το καλούμε ως εξής:

Έστω ότι η εγγραφή ενός νέου χρήστη ήταν επιτυχής:

```
if ($conn->query($sql) === TRUE) {
    // echo "New record created successfully";
    $conn->close();
    header("Location: ../php/message.php?msg=usrsuccess");
    die("User successfully created!");
```

Εικόνα 32: Εμφάνιση σελίδας μηνυμάτων

Η ουσιαστική εμφάνιση της σελίδα του μηνύματος γίνεται με την κλήση της συνάρτησης header(), όπου και επιλέγουμε το αρχείο message.php και το περιεχόμενό του μέσω της μεθόδου GET.

Το αρχείο message.php περιέχει κώδικα HTML με το θέμα της πλατφόρμας και επίσης μια συνάρτηση η οποία ορίζει το περιεχόμενο του μηνύματος προς προβολή.

Εικόνα 33: Σελίδα μηνυμάτων, χωρίς διαθέσιμο μήνυμα

Τα δεδομένα για το εκάστοτε μήνυμα, ορίζονται από την συνάρτηση messageContent ανάλογα με το όρισμα που θα της δοθεί.

```php
function messageContent($title, $message){
  echo '
    <!DOCTYPE html>
    <html lang="en">
      <head>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <!-- The above 3 meta tags *must* come first in the head; any other head content must come *after* these tags -->
        <meta name="description" content="Obolus Exchange">
        <meta name="author" content="Spyridon Georg Koustas">
        <link rel="icon" href="images/icon.png">

        <title>Obolus - Money Transfer</title>

      </head>

      <body>

        <nav class="navbar navbar-inverse navbar-fixed-top">
          <div class="container">
            <div class="navbar-header">
              <a class="navbar-brand" href="">Obolus Exchange</a>
            </div>
          </div>
        </nav>

        <div id="header-image-container" class="jumbotron" style="margin-top: 2em;">
          <div class="container">
            <h2>Message</h2>
          </div>
        </div>

        <div id="message-box" class="container">
          <div>
            <h4>'.$title.'</h4>
            <hr>
            <div>'.$message.'</div>
```

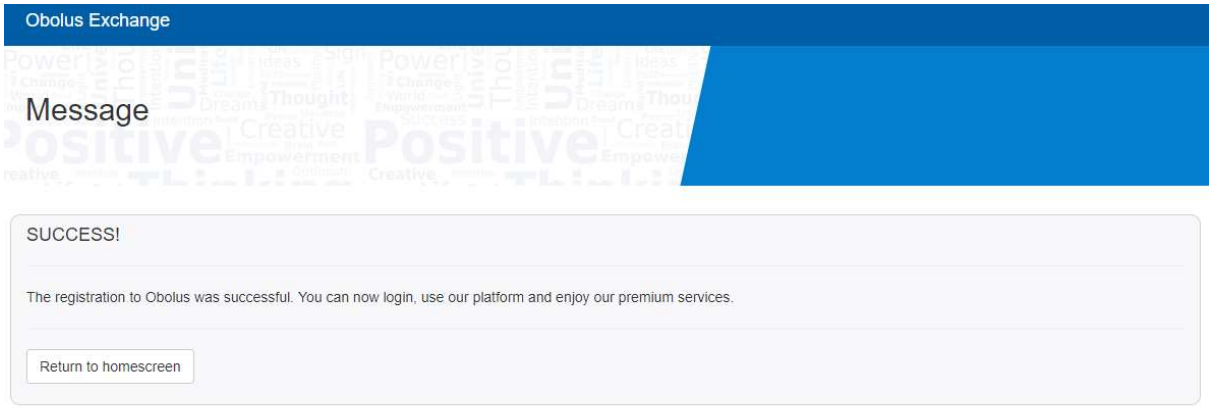
Εικόνα 34: Ο κώδικας HTML που περιέχει το θέμα χωρίς τα μηνύματα

```php
$msg = $_GET["msg"];

if ($msg == "usrsuccess") {
  messageContent("SUCCESS!","The registration to Obolus was successful. You can now login, use our platform and enjoy our premium services.");
} elseif ($msg == "usrerror") {
  messageContent("ERROR!","We have to apologize. There was an error with your registration at Obolus. Please try later again.");
} elseif ($msg == "lgnerror") {
  messageContent("ERROR!", "Something went wrong with your login attempt, please try again!</div><div>Should you need help with your account, please contact our <a href=''>support</a> team, we will be happy to help you.");
} elseif ($msg == "unauthatmpt") {
  messageContent("WARNING!", "Unauthorised activity has been spotted! Should you need help with your account, please contact our <a href=''>support</a> team, we will be happy to help you.");
} elseif ($msg == "lgout"){
  messageContent("SUCCESS!", "You have logged out successfully and Your session ended.");
} elseif ($msg == "updmailsuccess") {
  messageContent("SUCCESS", "Your email has been updated successfully! Now you need to login again.");
} elseif ($msg == "updpaswsuccess") {
  messageContent("SUCCESS", "Your password has been updated successfully! Now you need login again.");
} elseif ($msg == "addrsuccess") {
  messageContent("SUCCESS", "Your address has been updated successfully! Now you need login again.");
} elseif ($msg == "error") {
  messageContent("ERROR", "There has been an error with your request!");
}
```

Εικόνα 35: Ο ορισμός των μεταβλητών title και message ανάλογα με την τιμή της μεταβλητής msg.

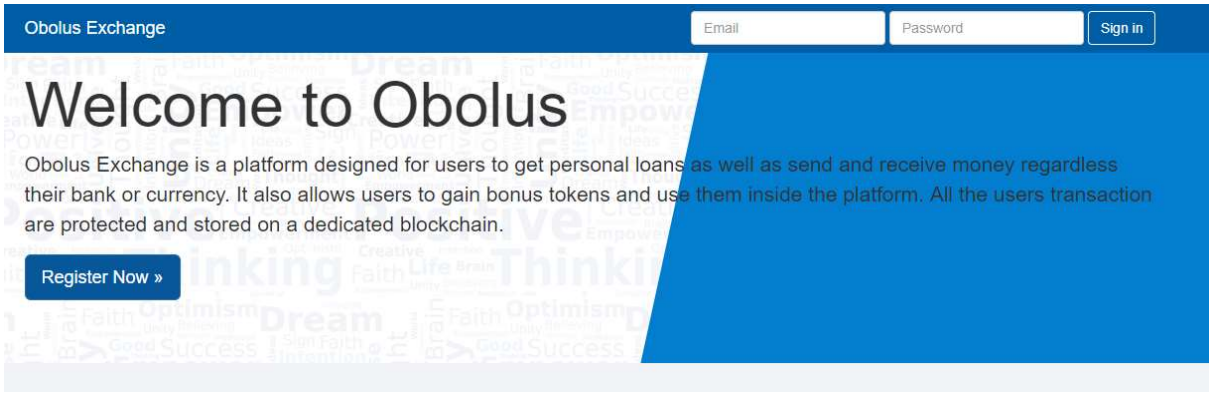
Με τον παραπάνω τρόπο μπορούμε και έχουμε εξατομικευμένα μηνύματα για οποιαδήποτε περίπτωση επιθυμούμε.



Εικόνα 36: Παράδειγμα μηνύματος επιτυχημένης εγγραφής νέου χρήστη

3.9.1.5 Διαχείριση λογαριασμών και σύνδεσης χρήστη

Μια από τις πιο σημαντικές λειτουργίες της εφαρμογής είναι η διαχείριση των λογαριασμών των χρηστών. Μετά την εγγραφή, ένας χρήστης μπορεί να συνδεθεί στην πλατφόρμα και να κάνει χρήση των υπηρεσιών που προσφέρει.



Εικόνα 37: Αρχική σελίδα - σελίδα σύνδεσης χρηστών

Αν επισκεφθούμε το αρχείο login-query.php θα συναντήσουμε το αρχείο που διαχειρίζεται τη σύνδεση χρηστών στην εφαρμογή. Όπως φαίνεται και στον κώδικα παρακάτω, η εφαρμογή ελέγχει τα δεδομένα εισόδου του χρήστη με τον τρόπο που αναφέραμε παραπάνω και έπειτα ελέγχει στη βάση δεδομένων αν υπάρχει ο αντίστοιχος χρήστης (Εικόνα 38).

Στην περίπτωση που δεν αντιστοιχεί χρήστης στα δεδομένα εισόδου ή τα δεδομένα αυτά παρουσιάζουν κάποιο πρόβλημα, η εφαρμογή θα εμφανίσει μήνυμα λάθους.

Στην περίπτωση που βρεθεί ο χρήστης και τα στοιχεία είναι ορθά, θα δημιουργηθεί μια συνεδρίαση (session) στον διακομιστή, στην οποία θα αποθηκευτούν τα δεδομένα του χρήστη έως να αποσυνδεθεί ή εως να λήξη η συνεδρίαση. Τα δεδομένα που αποθηκεύονται είναι τα στοιχεία του χρήστη που χρειάζεται η εφαρμογή για να μπορέσει να προσφέρει αποτελεσματικά τις υπηρεσίες της.

```php
if (filter_var($_POST["email"], FILTER_VALIDATE_EMAIL)){
  $email = $_POST["email"];
} else {
  $email = NULL;
}
$password = strip_tags($_POST["password"]);
// We hash the password to store it iside the database
$password_hashed = hash('sha256', $password);

// Create connection
$conn = new mysqli($servername, $db_username, $db_password, $db_name);
// Check connection
if ($conn->connect_error) {
  die("Connection failed: ".$conn->connect_error);
}

$sql = "SELECT _id, first_name, last_name, role, public_key FROM users WHERE email = '".$
  email."' AND password = '".$password_hashed."' LIMIT 1";

$result = $conn->query($sql);
```

Εικόνα 38: Έλεγχος δεδομένων εισόδου και αναζήτηση χρήστη

```php
if ($first_name != NULL && $last_name != NULL && $role != NULL) {
  // echo "SUCCESS! ".$role;
  $conn->close();
  session_start();
  // $_SESSION['previous_page'] = basename($_SERVER['PHP_SELF']);
  $_SESSION['user_id'] = $user_id;
  $_SESSION['first_name'] = $first_name;
  $_SESSION['last_name'] = $last_name;
  $_SESSION['role'] = $role;
  $_SESSION['email'] = $email;
  $_SESSION['password'] = $password_hashed;
  $_SESSION['public_key'] = $public_key;
  header("Location: ../php/dashboard.php");
  die("Login succeeded!");
} else {
  // echo "Error: ".$conn->error;
  $conn->close();
  header("Location: ../php/message.php?msg=lgnerror");
  die("ERROR, login terminated!");
}
```

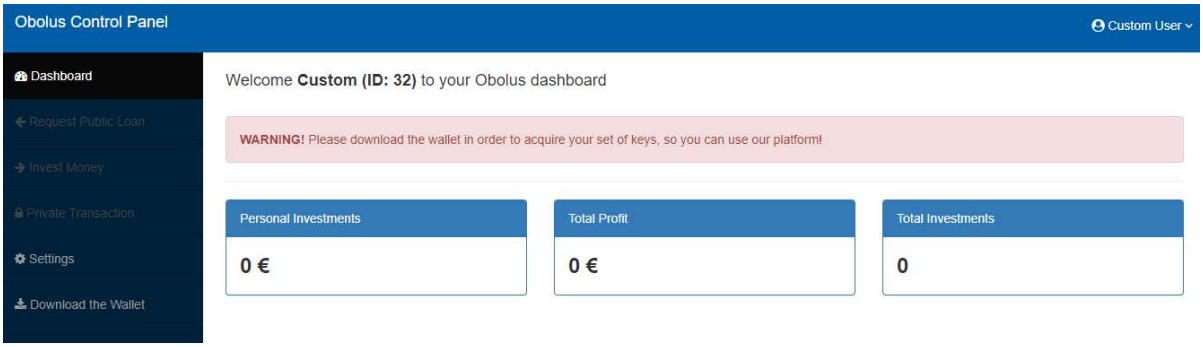
Εικόνα 39: Σε περίπτωση επιτυχημένης σύνδεσης, αποθηκεύονται τα παραπάνω δεδομένα του χρήστη

Τα δεδομένα που αποθηκεύονται στη συνεδρίαση είναι τα εξής:

1. Ο κωδικός του χρήστη (user_id)
2. Το όνομα του χρήστη (first_name)
3. Το επίθετο του χρήστη (last_name)
4. Ο ρόλος του χρήστη στο συγκεκριμένο πληροφοριακό σύστημα (role)
5. Η διεύθυνση ηλεκτρονικού ταχυδρομείου του χρήστη (email)
6. Ο κρυπτογραφημένος κωδικός πρόσβασης του χρήστη (password)
7. Το δημόσιο κλειδί του χρήστη (public_key)

Στα πλαίσια την πτυχιακής εργασία χρησιμοποιείται το δημόσιο κλειδί των χρηστών και ως διεύθυνση συναλλαγών. Οι διευθύνσεις αυτές είναι απλά μοναδικές συμβολοσειρές. Στις πρώτες εκδόσεις του Bitcoin χρησιμοποιούνταν οι διευθύνσεις IP ως διευθύνσεις πληρωμών. Όλες οι παραπάνω μεταβλητές χάνονται από τη μνήμη του διακομιστή όταν ο χρήστης αποσυνδέεται ή όταν λήξη η συνεδρίασή του.

Την πρώτη φορά που συνδέεται ένας χρήστης στη διαδικτυακή εφαρμογή και δεν έχει αποκτήσει ακόμη ζεύγος κλειδιών, η εφαρμογή δεν του δίνει πρόσβαση στις λειτουργίες, αν δεν κατεβάσει και αποκτήσει πρώτα ένα ζεύγος κλειδιών. Την στιγμή που θα κατεβάσει το πορτοφόλι του και θα συνδεθεί για πρώτη φορά, συνδυάζονται ο λογαριασμός του χρήστη με το συγκεκριμένο πορτοφόλι και δημοσιεύεται στην πλατφόρμα το δημόσιο κλειδί του χρήστη.



Εικόνα 40: Προειδοποιητικό μήνυμα της πλατφόρμας για την εγκατάσταση του πορτοφολίου και της δέσμευσης κλειδιών

3.9.1.6 PHP Συνεδρίες (Sessions)

Για να κατανοήσουμε τις συνεδρίες της PHP, πρέπει να φανταστούμε μία απλή εφαρμογή στον υπολογιστή μας. Την ανοίγουμε, δουλεύουμε σε αυτήν και την ξανα κλείνουμε. Κάτι τέτοιο είναι και οι συνεδρίες. Η διαφορά με την εγκατεστημένη εφαρμογή είναι ότι ο διακομιστής (web server), λόγω του HTTP δεν διατηρεί καταστάσεις (states).

Οι συνεδρίες στην PHP είναι ένας τρόπος αποθήκευσης δεδομένων (σε μορφή μεταβλητών), έτσι ώστε να μπορούν να χρησιμοποιούνται σε περισσότερες σελίδες.
Η διαφορά των μεταβλητών της PHP συνεδρίας σε σχέση με τα cookies είναι ότι η πληροφορία δεν αποθηκεύεται στον υπολογιστή του χρήστη.

Οι μεταβλητές των συνεδριών διατηρούνται μέχρι ο χρήστης να κλείσει τον φυλλομετρητή του ή να τερματιστεί η συνεδρία.

Για να ξεκινήσουμε μια συνεδρία στην PHP χρησιμοποιούμε τη συνάρτηση session_start() και για να ολοκληρώσουμε μια συνεδρία, τη συνάρτηση session_destroy(). Για να σβήσουμε όλες τις μεταβλητές μέσα από μια συνεδρία χωρίς να την κλείσουμε, χρησιμοποιούμε τη συνάρτηση session_unset().

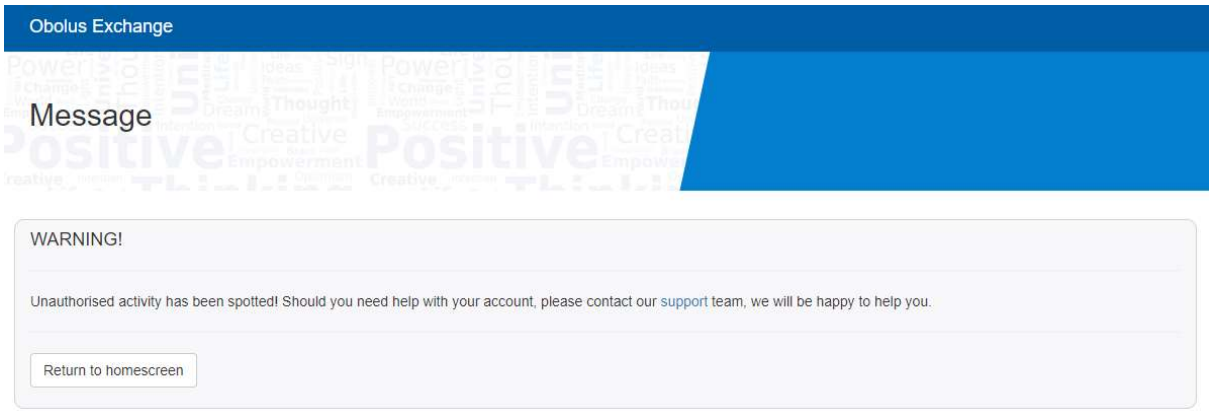
Στην παρούσα πτυχιακή εργασία, οι συνεδρίες της PHP έπαιξα ένα πολύ μεγάλο ρόλο, ειδικότερα στην επαλήθευση δεδομένων. Ένας ακόμη πολύ χρήσιμος τρόπος αξιοποίησης των συνεδριών είναι να ελέγχουμε εάν ένας χρήστης έχει πρόσβαση σε κάποια σελίδα ή όχι.

Στο ακόλουθο παράδειγμα γίνεται έλεγχος εάν κατά την προσπελαση της σελίδας, ο χρήστης ήταν συνδεδεμένος στο λογαριασμό του. Δηλαδή ελέγχουμε εάν του είχε ανατεθεί η μεταβλητή role. Στην περίπτωση που κάποιος χρήστης προσπαθεί να προσπελάσει την συγκεκριμένη σελίδα χωρίς να έχει συνδεθεί, θα λάβει μήνυμα λάθους.

```php
if (!$_SESSION['role']) {
  header("Location: ../php/message.php?msg=unauthatmpt");
  die("Unauthorized Attempt!");
} else {
  $role = $_SESSION['role'];
  $first_name = $_SESSION['first_name'];
  $last_name = $_SESSION['last_name'];
  $public_key = $_SESSION['public_key'];
}
```

Εικόνα 41: Έλεγχος πρόσβασης χρήστη μέσω της συνεδρίας της PHP

Το μήνυμα λάθους που θα λάβει είναι το ακόλουθο:



Εικόνα 42: Μήνυμα λάθους, μη συνδεδεμένος χρήστης

3.9.1.7 Επικοινωνία με το πορτοφόλι

Η διαδικτυακή εφαρμογή επιτρέπει στους χρήστες να έχουν πάντα πρόσβαση στα χρήματά τους και στις συναλλαγές τους. Η πραγματικές όμως συναλλαγές πραγματοποιούνται όταν εγκριθούν από το προσωπικό πορτοφόλι των χρηστών, το οποίο βρίσκεται εγκατεστημένο στους προσωπικούς τους υπολογιστές. Για να μπορέσει η διαδικτυακή εφαρμογή να επικοινωνήσει με τα πορτοφόλια των χρηστών, γίνεται χρήση της βάσης δεδομένων της εφαρμογής. Εκεί αποθηκεύονται τα αιτήματα των συναλλαγών των χρηστών.

Κάθε φορά που κάποιος χρήστης επενδύει σε κάποιο δάνειο ή μεταφέρει χρήματα μέσω της διαδικτυακής εφαρμογής, πραγματοποιείται μια εισαγωγή στην βάση δεδομένων.

```php
// Create connection
$conn = new mysqli($servername, $db_username, $db_password, $db_name);
// Check connection
if ($conn->connect_error) {
  die("Connection failed: ".$conn->connect_error);
}

$sql = "INSERT INTO transaction_requests (user_id, transaction_type,
  transaction_from, transaction_to, amount, currency) VALUES (".$user_id."
  ,'private-transaction', '".$public_key."', '".$recipient_address."', ".$
  amount.", '".$currency."')";

if ($conn->query($sql) === TRUE) {
    // echo "New record created successfully";
  $conn->close();
    header("Location: ../php/dashboard-messages.php?msg=prvttranssuccess");
    die("User successfully created!");
} else {
    // echo "Error: ".$sql."<br>".$conn->error;
  $conn->close();
    header("Location: ../php/dashboard-messages.php?msg=prvttranserror");
    die("ERROR, user could not be created!");
}
```

Εικόνα 43: Παράδειγμα αποθήκευσης αιτήματος συναλλαγής για προσωπική συναλλαγή

Στην παραπάνω εικόνα (Εικόνα 43), βλέπουμε τον κώδικα που είναι αρμόδιος να αποθηκεύσει τα αιτηματα συναλλαγών σε περίπτωση προσωπικής συναλλαγής. Ανάλογα με την επιτυχία ή την αποτυχία της ενημέρωσης, εκτυπώνονται και τα αντίστοιχα μηνύματα, όπως αναλύσαμε στην υποενότητα "διαχείριση μηνυμάτων".

Αντίστοιχα και όταν κάποιος χρήστης επιθυμεί να επενδύσει σε κάποιο δάνειο.

```php
$sql = "INSERT INTO transaction_requests (user_id, transaction_type,
  loan_id, transaction_from, transaction_to, amount, currency) VALUES (".$
  user_id.",'loan-investment', '".$loan_id."', '".$public_key."', '".$
  recipient_address."', ".$investmentAmount.", 'Euros')";
```

Εικόνα 44: Παράδειγμα αποθήκευσης αιτήματος συναλλαγής για επένδυση σε δάνειο

Στην παραπάνω περίπτωση (Εικόνα 44), αυτό που ουσιαστικά αλλάζει είναι η τιμή της μεταβλητής transaction_type (είδος συναλλαγής). Η συγκεκριμένη μεταβλητή είναι πολύ σημαντική, γιατί από αυτή κρίνεται το πως θα αποθηκευτεί η συναλλαγή στο Blockchain.

3.9.2 Το προσωπικό πορτοφόλι

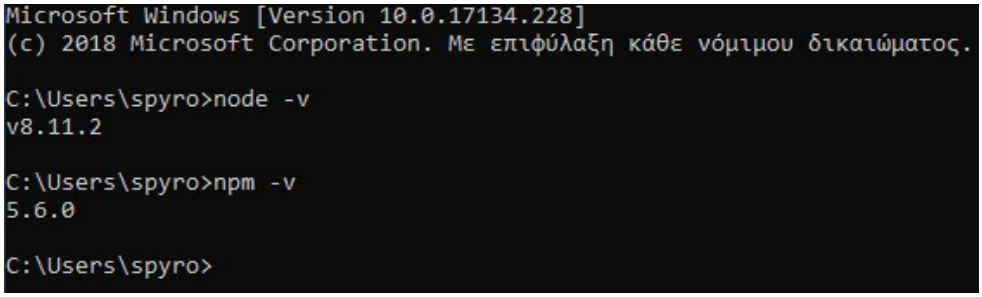
Το πρόγραμμα το οποίο εγκαθιστούν οι χρήστες στους υπολογιστές τους και χρησιμοποιείται για τις λειτουργίες του πλήρη κόμβου του δικτύου, αναπτύχθηκε μέσω του Electron. Το Electron μας επιτρέπει την ανάπτυξη cross-platform εφαρμογών κάνοντας χρήση τεχνολογιών ανάπτυξης διαδικτυακών εφαρμογών, δηλαδή, για το γραφικό περιβάλλον χρησιμοποιείται HTML, CSS και JavaScript (άρα και Bootstrap), ενώ για την αλληλεπίδραση με τα αρχεία του υπολογιστή χρησιμοποιείται η NodeJS.

3.9.2.1 Εγκατάσταση του Electron

Προκειμένου να εγκατασταθεί το electron [19], πρέπει πρώτα να εξασφαλίσουμε ότι ο υπολογιστής έχει εγκατεστημένο το NodeJS [20], το οποίο είναι ένα JavaScript runtime που αναπτύχθηκε βασισμένο στο JavaScript engine του Chrome V8. Έπειτα πρέπει να βεβαιωθούμε ότι έχουμε εγκατεστημένο το npm [21], το οποίο είναι package manager της JavaScript.

Για να ελέγξουμε τα παραπάνω μπορούμε απλά να πληκτρολογήσουμε στην γραμμή εντολών των Windows τις εξής εντολές:

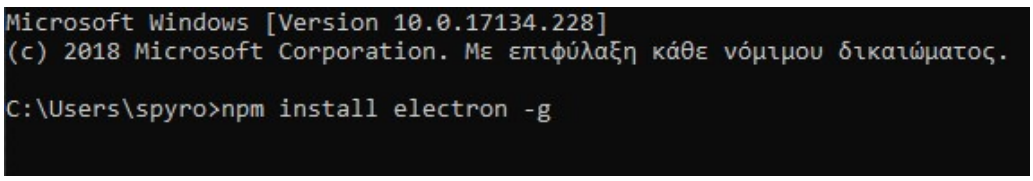
1. Για το node: node -v
2. Για το npm: npm -v



Εικόνα 45: Έλεγχος εκδόσεων μέσω της γραμμής εντολών των Windows

Αφού έχουμε εξασφαλίσει ότι υπάρχουν τα παραπάνω, η εγκατάσταση του Electron γίνεται μέσω μιας εντολής:

npm install electron -g



Εικόνα 46: Εγκατάσταση του Electron

3.9.2.2 Αρχικοποίηση της εφαρμογής

Μετά την εγκατάσταση, δημιουργείται ένα έγγραφο package.json. Στο αρχείο αυτό αποθηκεύονται τα στοιχεία της εφαρμογής που θα αναπτύξουμε (όνομα, έκδοση, περιγραφή, συγγραφέας, άδεια κτλ). Εκτός των άλλων αποθηκεύονται και τα dependencies της εφαρμογής. Ακόμη, ορίζεται μια τιμή main, αυτή καθορίζει ποιό αρχείο θα εκτελεστεί όταν θα ανοίγει η εφαρμογή.
Παρακάτω παρουσιάζεται το αρχείο package.json την εφαρμογής μας, το οποίο έχει την ακόλουθη μορφή:

```
{
  "name": "wallet",
  "version": "1.0.0",
  "description": "Obolus Desktop Blockcahin Wallet",
  "main": "main.js",
  "scripts": {
    "start": "electron ."
  },
  "author": "Spyridon Georg Koustas",
  "license": "ISC",
  "dependencies": {
    "electron": "^2.0.1",
    "file-system": "^2.2.2",
    "js-sha256": "^0.9.0",
    "mysql": "^2.15.0",
    "secp256k1": "^3.5.0"
  }
}
```

Εικόνα 47: Το αρχείο package.json

3.9.2.3 Το αρχείο main.js

Βλέπουμε ότι στην εφαρμογή μας έχουμε ορίσει ως main, το αρχείο main.js. Ας δούμε τί ακριβώς περιέχει αυτό το αρχείο.

```
1  const electron = require('electron');
2  const url = require('url');
3  const path = require('path');
4
5  const {app, BrowserWindow, Menu} = electron;
6
7  let mainWindow;
8
9  //Listen for app to be ready
10 app.on('ready', function(){
11     //Create window
12     mainWindow = new BrowserWindow({});
13     //Load HTML into window
14     mainWindow.loadURL(url.format({
15         pathname: path.join(__dirname, 'wallet.html'),
16         protocol: 'file',
17         slashes: true
18     }));
19 });
```

Εικόνα 48: Το αρχείο main.js

Αν παρατηρήσουμε τις γραμμές 13 - 18, βλέπουμε τον κώδικα που είναι υπεύθυνος για την δημιουργία ενός νέου παραθύρου και επίσης της φόρτωσης του αρχείου wallet.html. Οι υπόλοιπες συναρτήσεις αρχικοποιούν δεδομένα τα οποία χρειάζονται για την εκτέλεση του Electron.

```
21 // Quit when all windows are closed.
22 app.on('window-all-closed', () => {
23   // On macOS it is common for applications and their menu bar
24   // to stay active until the user quits explicitly with Cmd + Q
25   if (process.platform !== 'darwin') {
26     app.quit()
27   }
28 })
```

Εικόνα 49: Βελτιστοποίηση για λειτουργικά macOS

Στις σειρές 21 - 28, γίνεται βελτιστοποίηση της εφαρμογής για λειτουργικά συστήματα macOS και ειδικότερα της πλατφόρμας "darwin". Ουσιαστικά ορίζουμε την περάτωση της εφαρμογής, όταν κλείνουν όλα τα παράθυρά της.

```
30 //Create menu template
31 const loginMenuTemplate = [
32     {
33         label:'Exit'
34     },
35     {
36         label:'Help'
37     }
38 ];
```

Εικόνα 50: Ορισμός μπάρας μενού της εφαρμογής

Στις γραμμές 30 - 38, ορίζουμε την μορφή που θέλουμε να έχει η μπάρα μενού της εφαρμογής μας. Τα στοιχεία ορίζονται σε μορφή json. Εμείς επιθυμούμε μόνο τις δύο επιλογές Exit (Τερματισμός) και Help (Βοήθεια).

3.9.2.4 Το αρχείο wallet.html

Το αρχείο wallet.html περιέχει τον κώδικα HTML μέσα στον οποίο θα εμφανίσουμε τα περιεχόμενα του πορτοφολίου.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <!-- Latest compiled and minified CSS -->
5      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
6      <!-- jQuery library -->
7      <script> window.$ = window.jQuery = require('./scripts/jquery-3.3.1.min.js');</script>
8      <!-- Latest compiled JavaScript -->
9      <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
10     <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
11     <script src="scripts/myFunctions.js"></script>
12     <link rel="stylesheet" href="css/wallet.css">
13
14     <title>Obolus Desktop Wallet</title>
15 </head>
16 <body>
17   <div id="wallet-view">
18   </div>
19 </body>
20 <script type="text/javascript">
21   if (sessionStorage.session_id) {
22     showDashboard();
23   } else {
24     showLogin();
25   }
26
27   $("#dashboard-btn").click(function(){
28     showDashboard();
29   });
30
31 </script>
32 </html>
```

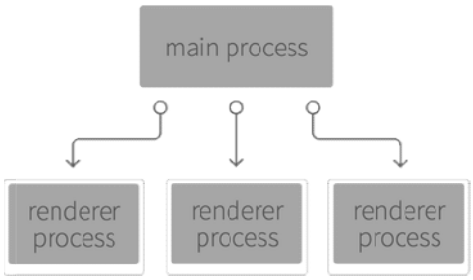
Εικόνα 51: Το αρχείο wallet.html

Παρατηρώντας τον κώδικα, βλέπουμε ουσιαστικά μια κενή (από άποψη περιεχομένου) σελίδα, η οποία όμως περιέχει δύο σημαντικές συναρτήσεις, καθώς και ένα αρχείο ονόματι myFunctions.js (στη γραμμή 11).

Στις γραμμές 21 - 25, ελέγχεται εάν έχει ανατεθεί στο χρήστη κάποιο session_id, δηλαδή αν έχει συνδεθεί και έχει ξεκινήσει κάποια συνεδρίαση ή όχι. Στην περίπτωση που έχει session_id, του εμφανίζεται το πορτοφόλι του μέσω της συνάρτησης showDashboard(). Αν δεν έχει συνδεθεί, το εμφανίζεται η οθόνη σύνδεσης χρήστη μέσω της συνάρτησης showLogin(). Οι δύο αυτές συναρτήσεις, όπως και πολλές άλλες, βρίσκονται στο αρχείο myFunctions.js.

3.9.2.5 Main & Renderer Processes

Προτού συνεχίσουμε στις συναρτήσεις, πρέπει να κατανοήσουμε πως ακριβώς δουλεύει το Electron.



Εικόνα 52: Ο τρόπος λειτουργίας του Electron - *Building a Desktop Application with Electron*

Το Electron λειτουργεί με μία κύρια διαδικασία (main process) και μία ή περισσότερες δευτερεύουσες διαδικασίες (renderer processes). Στην περίπτωση της εφαρμογής μας, η κύρια διαδικασία βρίσκεται μέσα στο αρχείο main.js. Οι δευτερεύουσες διαδικασίες βρίσκονται μέσα στα παράθυρα που δημιουργούνται από την κύρια διαδικασία, δηλαδή σε εμάς το mainWindow, που ανοίγει το wallet.html. Τα renderer processes είναι αρμόδια να εμφανίζουν το περιεχόμενο στον χρήστη μέσω της HTML.

```
9  let mainWindow;
10
11 //Listen for app to be ready
12 app.on('ready', function(){
13     //Create window
14     mainWindow = new BrowserWindow({});
15
16     //Load HTML into window
17     mainWindow.loadURL(url.format({
18         pathname: path.join(__dirname, 'wallet.html'),
19         protocol: 'file',
20         slashes: true
21     }));
```

Εικόνα 53: Διαχωρισμός διαδικασιών και κύριες και δευτερεύουσες

Στην περίπτωση που θέλουμε, τα διάφορα renderer processes μπορούν και επικοινωνούν μεταξύ τους μέσω του main process.

3.9.2.6 Το αρχείο myFunctions.js

Το αρχείο myFunctions.js περιέχει σχεδόν ολόκληρη τη λειτουργικότητα της εφαρμογής. Πρόκειται για ένα αρχείο JavaScript το οποίο αποτελείται περίπου από 1200 γραμμές κώδικα. Όλες οι λειτουργίες της εφαρμογής είναι χωρισμένες σε διάφορες συναρτήσεις. Μερικές συναρτήσεις εμφανίζουν δεδομένα στο χρήστη επεξεργάζοντας την HTML που προβάλλεται και άλλες πληρούν λειτουργικούς σκοπούς, όπως παραδείγματος χάρη οι κρυπτογραφικές συναρτήσεις.

Ας εξετάσουμε για παράδειγμα την συνάρτηση που εμφανίζει την σύνδεση του χρήστη κατά την εκκίνηση.

```
2  function showLogin () {
3    var inputHTML = `
4      <div class="container">…
62     </div>
63   `;
64   $("#wallet-view").html(inputHTML);
65   $("#login-error").hide();
66   $("#network-error").hide();
67
68   let fs = require('fs');
69
70   fs.exists('obolusKeys.json', function(exists) {
71     console.log("Does the Obolus File exist ? " + exists);
72
73     if (exists) {
74       let content = fs.readFileSync("obolusKeys.json");
75
76       var jsonContent = JSON.parse(content);
77       $("#email").val(jsonContent.email);
78       $("#email").prop('disabled', true);
79
80       console.log(jsonContent);
81     }
82   });
83
```

Εικόνα 54: Η συνάρτηση σύνδεσης χρήστη (1/3)

Στη συνάρτηση showLogin(), στις γραμμές 3 - 66, βρίσκονται στοιχεία της HTML για να επηρεάζουν την εμφάνιση της εφαρμογής. Στις γραμμές 68 - 82 παρουσιάζεται για πρώτη φορά η χρησιμότητα του electron, ενώ θεωρητικά μέχρι την γραμμή 66 ο κώδικας είχε την μορφή διαδικτυακής ανάπτυξης με jQuery, ξαφνικά στη γραμμή 70 μπορούμε και έχουμε πρόσβαση στο file system (σύστημα αρχείων) των windows, με μια εντολή και μία συνάρτηση που ανήκει στη NodeJS. Η συνάρτηση ελέγχει αν έχει οριστεί χρήστης για το συγκεκριμένο πορτοφόλι ή όχι. Σε περίπτωση που έχει οριστεί χρήστης, υπάρχει αποθηκευμένο το email του λογαριασμού σε ένα τοπικό αρχείο JSON με όνομα obolusKeys.json.

Παρακάτω (Εικόνα 55), βλέπουμε ότι στις γραμμές 85 - 109, η συνάρτηση κρυπτογραφεί τον κωδικό που εισήγαγε ο χρήστη και δημιουργεί μια σύνδεση με την βάση δεδομένων (πρόκειται για την ίδια βάση με την οποία επικοινωνεί και η διαδικτυακή εφαρμογή) προκειμένου να ολοκληρωθεί η σύνδεσή του. Σε περίπτωση επιτυχημένης σύνδεσης, η συνάρτηση εμφανίζει ένα μήνυμα στην κονσόλα.

```
85    $("#login-btn").click(function(){
86      var sha256 = require('js-sha256');
87
88      let email = $("#email").val();
89      let password = $("#password").val();
90
91      if (email && password) {
92        var passwordHashed = sha256(password);
93
94        var mysql = require("mysql");
95        var connection = mysql.createConnection({
96          host: "localhost",
97          user: "root",
98          password: "",
99          database: "obolus"
100       });
101
102       connection.connect((err) => {
103         if (err) {
104           return console.log(err.stack);
105         }
106
107         console.log("Connection successfully established");
108       });
109
```

Εικόνα 55: Η συνάρτηση σύνδεσης χρήστη (2/3)

```
connection.query($queryString, (err, rows, fields) => {
  if (err) {
    $("#network-error").show();
    $("#login-form")[0].reset();
    return console.log("An error ocurred with the query", err);
  }

  if (!rows[0]) {
    // console.warn("LOGIN-ERROR!");
    $("#login-error").show();
    $("#login-form")[0].reset();
    $("#email").css({"border": "1px solid red"});
    $("#password").css({"border": "1px solid red"});
  } else {
    sessionStorage.user_id = rows[0]._id;
    sessionStorage.first_name = rows[0].first_name;
    sessionStorage.last_name = rows[0].last_name;
    sessionStorage.email = email;
    sessionStorage.dob = rows[0].dob;
    sessionStorage.street_name = rows[0].street_name;
    sessionStorage.zip_code = rows[0].zip_code;
    sessionStorage.city = rows[0].city;
    sessionStorage.country = rows[0].country;
    sessionStorage.public_key = rows[0].public_key;
    sessionStorage.role = rows[0].role;
    sessionStorage.timestamp = rows[0].timestamp;

    showDashboard();
    sessionStorage.session_id = Math.floor(Math.random() * 1000) + 1;
  }
});
```

Εικόνα 56: Η συνάρτηση σύνδεσης χρήστη (3/3)

Σε περίπτωση επιτυχημένης σύνδεσης, το πρόγραμμα ποθηκεύει κάποιες μεταβλητές σχετικές με το χρήστη στη session μνήμη της εφαρμογής προκειμένου να χρησιμοποιηθούν αργότερα.

3.9.2.7 Η δημιουργία των κλειδιών

Μία από τις βασικότερες συναρτήσεις της πτυχιακής εργασίας είναι η συνάρτηση privateKeyCreation(), η οποία είναι υπεύθυνη για την δημιουργία του ιδιωτικού αλλα και του δημοσίου κλειδιού των χρηστών.

```
395 function privateKeyCreation() {
396   const crypto = require('crypto');
397   const { randomBytes } = require('crypto');
398   const secp256k1 = require('secp256k1');
399
400
401   const privateKey = crypto.randomBytes(32);
402   console.log("privateKey: ", privateKey.toString('hex'));
403   console.log(`${privateKey.length} bytes of random data: ${privateKey.toString('hex')}`);
404
405   var privateKeyBuffer = Buffer.from(privateKey);
406   console.log("privateKeyBuffer: ", privateKeyBuffer);
407
408   // get the public key in a compressed format
409   const publicKey = secp256k1.publicKeyCreate(privateKeyBuffer);
410   console.log("publicKey: ", publicKey.toString('hex'));
```

Εικόνα 57: Η συνάρτηση δημιουργία ζεύγος κλειδιών

Για την δημιουργία του ζεύγους κλειδιών έγινε χρήση της βιβλιοθήκης crypto, η οποία περιέχει ένα μεγάλο αριθμό κρυπτογραφικών συναρτήσεων, πολλές από τις οποίες χρησιμοποιούνται και σε γνωστά κρυπτονομίσματα. Επίσης έγινε χρήση της συνάρτησης secp256k1 η οποία αναφέρεται στις παραμέτρους της ελλειπτικής συνάρτησης που χρησιμοποιείται στην κρυπτογραφία δημοσίου κλειδιού του Bitcoin.

Το ιδιωτικό κλειδί των χρηστών δημιουργείται στη γραμμή 401 με τη βοήθεια της συνάρτησης crypto.randomBytes() η οποία είναι μια συνάρτηση υψηλής εντροπίας και απόλυτα ασφαλή για τα δεδομένα της πτυχιακής εργασίας. Έπειτα κάνοντας χρήσης της συνάρτηση secp256k1.publicKeyCreate() στη γραμμή 409, δημιουργείται το δημόσιο κλειδί του χρήστη μέσα από το ιδιωτικό.

```
412 // your content is into a variable named 'content'
413 let content = '{"privateKey":"' + privateKey.toString('hex') + '","publicKey":"'+ publicKey.toString('hex') +'","ema
414 sessionStorage.public_key = publicKey.toString('hex');
415 let fs = require('fs');
416 try {
417   fs.writeFileSync('obolusKeys.json', content, 'utf-8');
418   console.log("Obolus Key File Created");
419 }
420 catch(e) { alert('Failed to save the file !'); }
421
422 var mysql = require("mysql");
423 var connection = mysql.createConnection({
424   host: "localhost",
425   user: "root",
426   password: "",
427   database: "obolus"
428 });
429
430 connection.connect((err) => {
431   if (err) {
432     return console.log(err.stack);
433   }
434   console.log("Connection successfully established");
435 });
436
437 //Here are the queries that will get executed
438 $queryString = 'UPDATE users SET public_key = "'+publicKey.toString('hex')+'" WHERE _id = '+sessionStorage.user_id;
439
440 connection.query($queryString, (err, rows, fields) => {
441   if (err) {
442     return console.log("An error ocurred with the query", err);
```

Εικόνα 58: Αποθήκευση ζεύγος κλειδιών

Έπειτα, η εφαρμογή αποθηκεύει το ζεύγος κλειδιών τοπικά στον υπολογιστή του χρήστη, για να μπορεί να έχει πρόσβαση για μελλοντικές χρήσης. Ακόμη, μιας και χρησιμοποιείται το δημόσιο κλειδί των χρηστών για διεύθυνση πληρωμής, αποθηκεύεται το δημόσιο κλειδί στη διαδικτυακή βάση δεδομένων και επιστρέφεται το αντίστοιχο μήνυμα.

3.9.2.8 Το Blockchain

Ας δούμε τη δημιουργία του Blockchain μέσα από τον κώδικα της πτυχιακής εργασίας.

```
571  class Transaction{
572    constructor(fromAddress, toAddress, amount){
573      this.fromAddress = fromAddress;
574      this.toAddress = toAddress;
575      this.amount = amount;
576    }
577  }
```

Εικόνα 59: Η δημιουργία της συναλλαγής

Ξεκινώντας παρατηρούμε την δημιουργία ενός constructor για την κλάση Transaction, η οποία ουσιαστικά θα δημιουργεί τις συναλλαγές της εφαρμογής.

```
579  class Block {
580    constructor(timestamp, transactions, previousHash = '') {
581      this.previousHash = previousHash;
582      this.timestamp = timestamp;
583      this.transactions = transactions;
584      this.hash = this.calculateHash();
585      this.nonce = 0;
586    }
```

Εικόνα 60: Η δημιουργία του μπλοκ

Αμέσως μετά βλέπουμε το constructor για την κλάση Block, με την οποία θα δημιουργούμε τα μπλοκ μας για το Blockchain.

```
588    calculateHash() {
589      return SHA256(this.previousHash + this.timestamp + JSON.stringify(this.transactions) + this.nonce).toString();
590    }
591
592    mineBlock(difficulty) {
593      while (this.hash.substring(0, difficulty) !== Array(difficulty + 1).join("0")) {
594          this.nonce++;
595          this.hash = this.calculateHash();
596      }
597
598      console.log("BLOCK MINED: " + this.hash);
599    }
600  }
```

Εικόνα 61: Εξόρυξη μπλοκ

Στις σειρές 588 - 600 βλέπουμε τις συναρτήσεις calculateHash() και mineBlock(). Η συνάρτηση calculateHash() είναι υπεύθυνη για τον υπολογισμό της τιμή SHA256 του μπλοκ και η συνάρτηση mineBlock() είναι υπεύθυνη για την εξόρυξη του νεου μπλοκ. Η μεταβλητή difficulty είναι η δυσκολία εξόρυξης, στην οποία έχουμε προαναφερθεί στην υποενότητα Proof-of-Work.

```
603 class Blockchain{
604   constructor() {
605     this.chain = [this.createGenesisBlock()];
606     this.difficulty = 2;
607     this.pendingTransactions = [];
608     this.miningReward = 100;
609   }
610
611   createGenesisBlock() {
612     return new Block(Date.parse("2018-01-01"), [], "0");
613   }
614
615   getLatestBlock() {
616     return this.chain[this.chain.length - 1];
617   }
618
619   minePendingTransactions(miningRewardAddress){
620     let block = new Block(Date.now(), this.pendingTransactions, this.getLatestBlock().hash);
621     block.mineBlock(this.difficulty);
622
623     console.log('Block successfully mined!');
624     this.chain.push(block);
625
626     this.pendingTransactions = [
627         new Transaction(null, miningRewardAddress, this.miningReward)
628     ];
629   }
```

Εικόνα 62: Το Blockchain, η δημιουργία του genesis block και εξόρυξη συναλλαγών

Στην παραπάνω εικόνα (Εικόνα 62) βλέπουμε το constructor για την κλάση Blockchain. Η κλάση ορίζεται από την αλυσίδα, τη δυσκολία εξόρυξης, τις συναλλαγές σε αναμονή και την αμοιβή εξόρυξης. Η συνάρτηση createGenesisBlock() είναι υπεύθυνη για τη δημιουργία του πρώτου μπλοκ της αλυσίδας. Η συνάρτηση getLatestBlock() επιστρέφει το τελευταίο εξορυγμένο μπλοκ στην αλυσίδα. Η minePendingTransactions() είναι η συνάρτηση που εξορύσσει τα νέα μπλοκ που βρίσκονται σε αναμονή. Όταν τελειώσει η εξόρυξη, προστίθεται αυτομάτως άλλη μία συναλλαγή με την αμοιβή εξόρυξης του κόμβου.

```
635   getBalanceOfAddress(address){
636       let balance = 0;
637
638       for(const block of this.chain){
639           for(const trans of block.transactions){
640               if(trans.fromAddress === address){
641                   balance -= trans.amount;
642               }
643
644               if(trans.toAddress === address){
645                   balance += trans.amount;
646               }
647           }
648       }
649
650       return balance;
651   }
```

Εικόνα 63: Συνάρτηση υπολογισμού υπολοίπου

Μία ακόμη σημαντική συνάρτηση είναι η getBalanceOfAddress(), η οποία ανατρέχει ολόκληρο το Blockchain και υπολογίζει το ποσό που αντιστοιχεί στον εκάστοτε χρήστη. Βέβαια η παραπάνω συνάρτηση δεν θα ήταν σωστή χωρίς τη συνάρτηση που ελέγχει εάν η αλυσίδα μας είναι ακαίρεα. Η συνάρτηση που είναι υπεύθυνη για τον έλεγχο της αλυσίδας είναι η isChainValid() (Εικόνα 64).

```
653   isChainValid() {
654       for (let i = 1; i < this.chain.length; i++){
655           const currentBlock = this.chain[i];
656           const previousBlock = this.chain[i - 1];
657
658           if (currentBlock.hash !== currentBlock.calculateHash()) {
659               return false;
660           }
661
662           if (currentBlock.previousHash !== previousBlock.hash) {
663               return false;
664           }
665       }
666
667       return true;
668   }
669 }
```

Εικόνα 64: Συνάρτηση ελέγχου ακεραιότητας αλυσίδας

Πλέον, μπορούμε να δημιουργήσουμε το Blockchain μας και να εξορύξουμε συναλλαγές με την κλήση των παραπάνω συναρτήσεων. Έστω δύο χρήστες user1 και user2 με διευθύνσεις user1_public_key και user2_public_key αντίστοιχα, έχουμε:

```
675 let Obolus = new Blockchain();
676
677 Obolus.createTransaction(new Transaction('user1_public_key','user2_public_key', 100));
678
679 Obolus.createTransaction(new Transaction('user2_public_key','user1_public_key', 50));
680
681 console.log("Starting the miner...");
682 Obolus.minePendingTransactions('user1_public_key');
683
684 console.log("Balance of " + 'User 1' + " is: ", Obolus.getBalanceOfAddress('user1_public_key'));
```

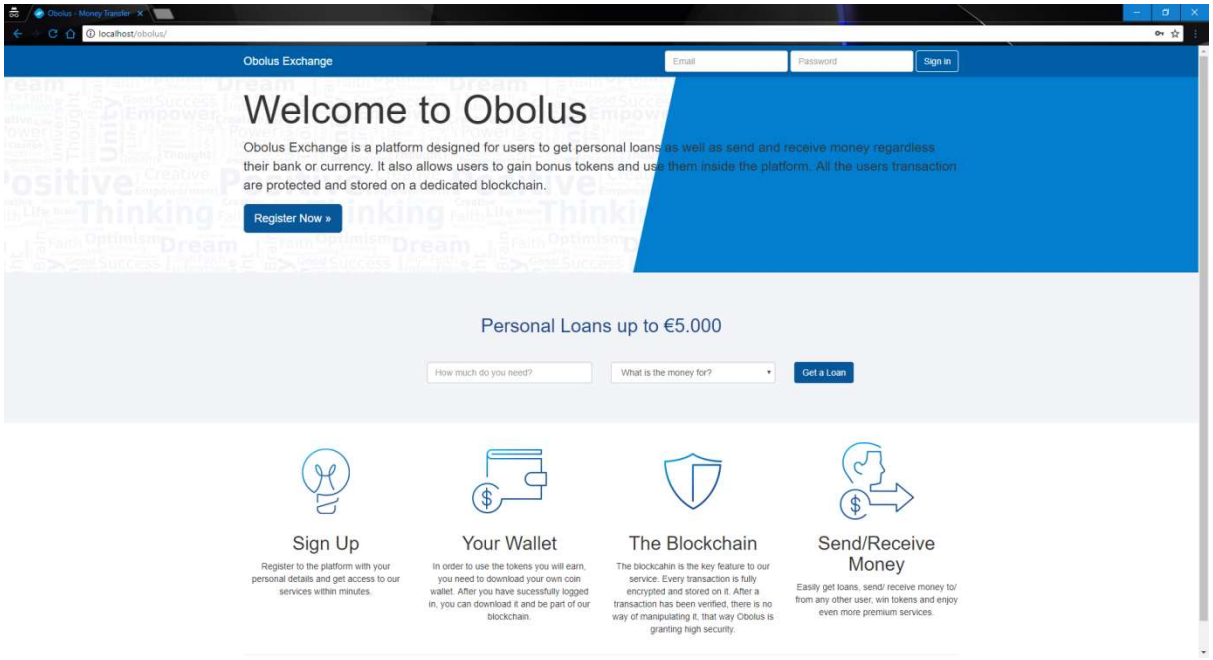
Εικόνα 65: Παράδειγμα εφαρμογής των λειτουργιών του Blockchain

4. Εγχειρίδιο χρήσης

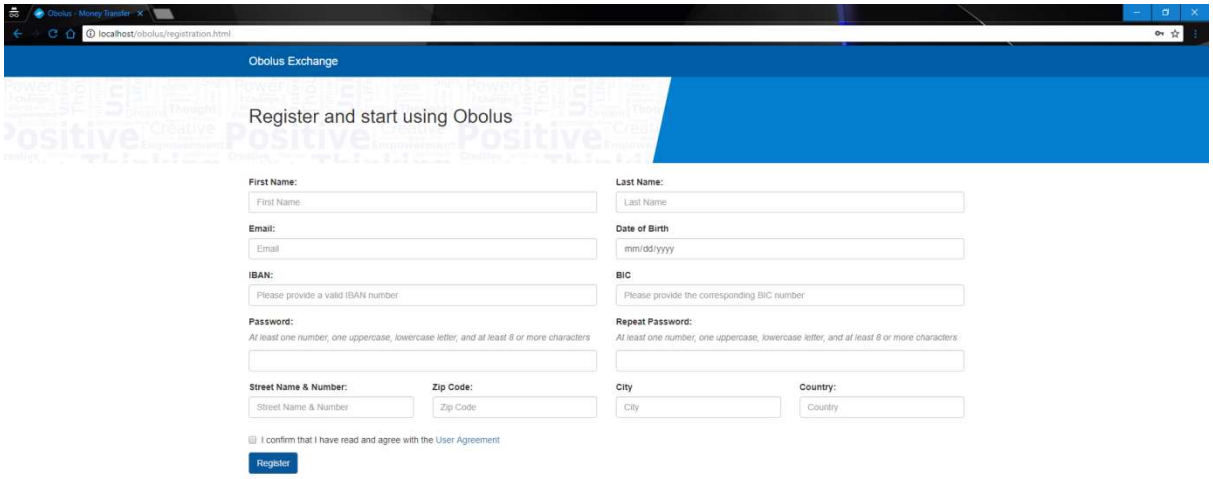
Στην ενότητα αυτή θα δούμε πώς ακριβώς ένας χρήστης μπορεί να χρησιμοποιήσεις τις δύο εφαρμογές που αναπτύχθηκαν στα πλαίσια αυτής της πτυχιακής εργασίας.

4.1 Η διαδικτυακή εφαρμογή

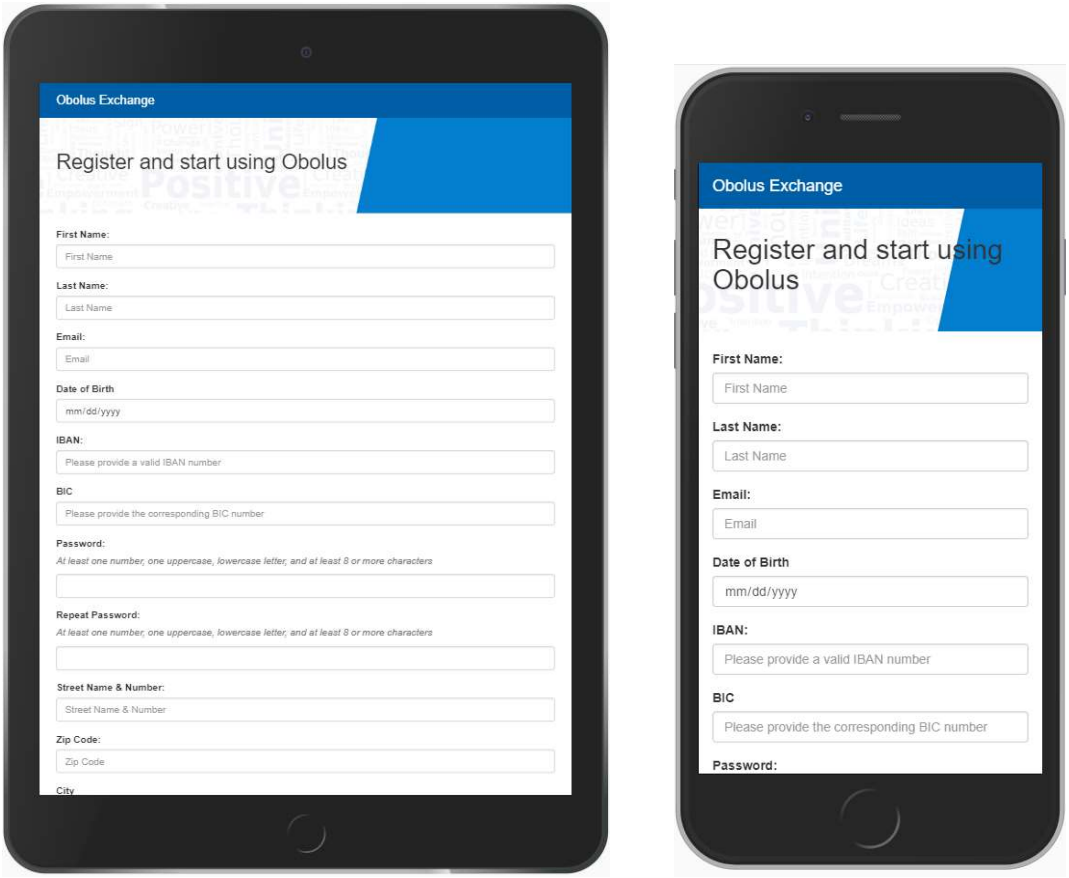
Εγγραφή νέου χρήστη. Η εγγραφή νέου χρήστη γίνεται μέσω της εφαρμογής. Ο νέος χρήστης απλά πρέπει να επιλέξει το Register Now και θα μεταφερθεί στην αντίστοιχη οθόνη με την φόρμα εγγραφής. Στο σημείο αυτό να τονίσουμε πως η ανάπτυξη της πλατφόρμας έγινε με εστίαση στο mobile responsiveness, δηλαδή στο να ανταποκρίνεται η εφαρμογή σε διάφορα μεγέθη κινητών συσκευών.



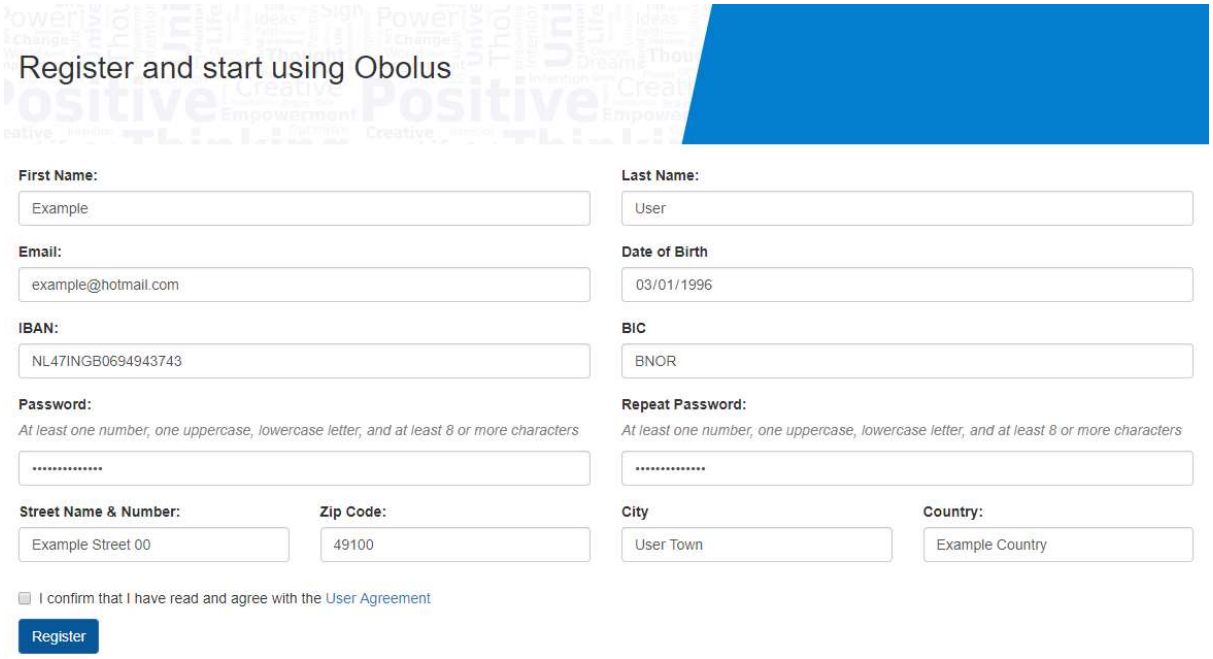
Εικόνα 66: Αρχική σελίδα της εφαρμογής Obolus Exchange



Εικόνα 67: Οθόνη εγγραφής σε υπολογιστή



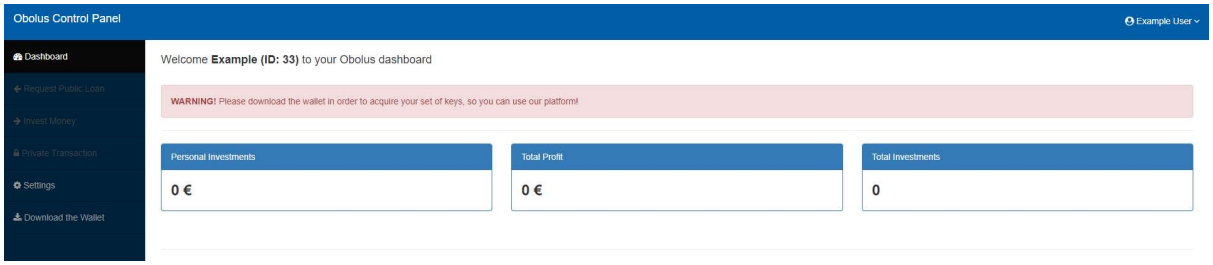
Εικόνα 68: Οθόνη εγγραφής σε tablet και κινητό



Εικόνα 69: Φόρμα εγγραφή με συμπληρωμένα στοιχεία

Αφού ο χρήστης συμπληρώσει τα απαραίτητα στοιχεία του και εγγραφεί επιτυχώς στην εφαρμογή, μπορεί πλέον να συνδεθεί και να κατεβάσει το προσωπικό του πορτοφόλι.

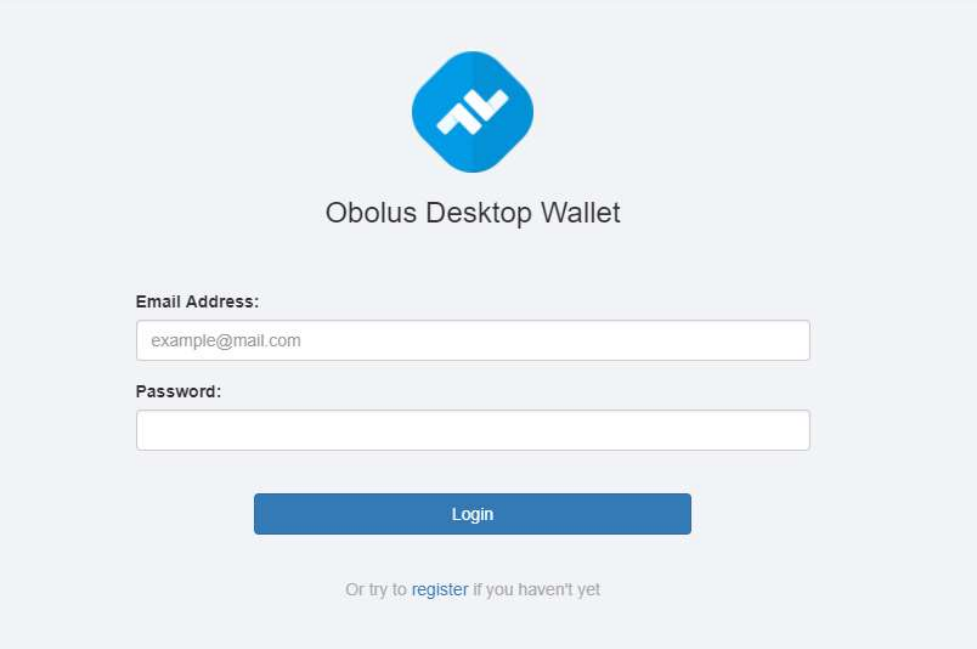
Την πρώτη φορά που ένα νέος χρήστης θα συνδεθεί στην διαδικτυακή εφαρμογή θα αντικρίσει την ακόλουθη οθόνη (Εικόνα 70), με αποτέλεσμα να μην μπορεί να κάνει πλήρη χρήση της πλατφόρμας. Το προειδοποιητικό μήνυμα παροτρύνει τον χρήστη να εγκαταστήσει το προσωπικό του πορτοφόλι για να του αποδοθούν τα απαραίτητα κλειδιά (ιδιωτικό και δημόσιο).



Εικόνα 70: Προειδοποιητικό μήνυμα χρήστη

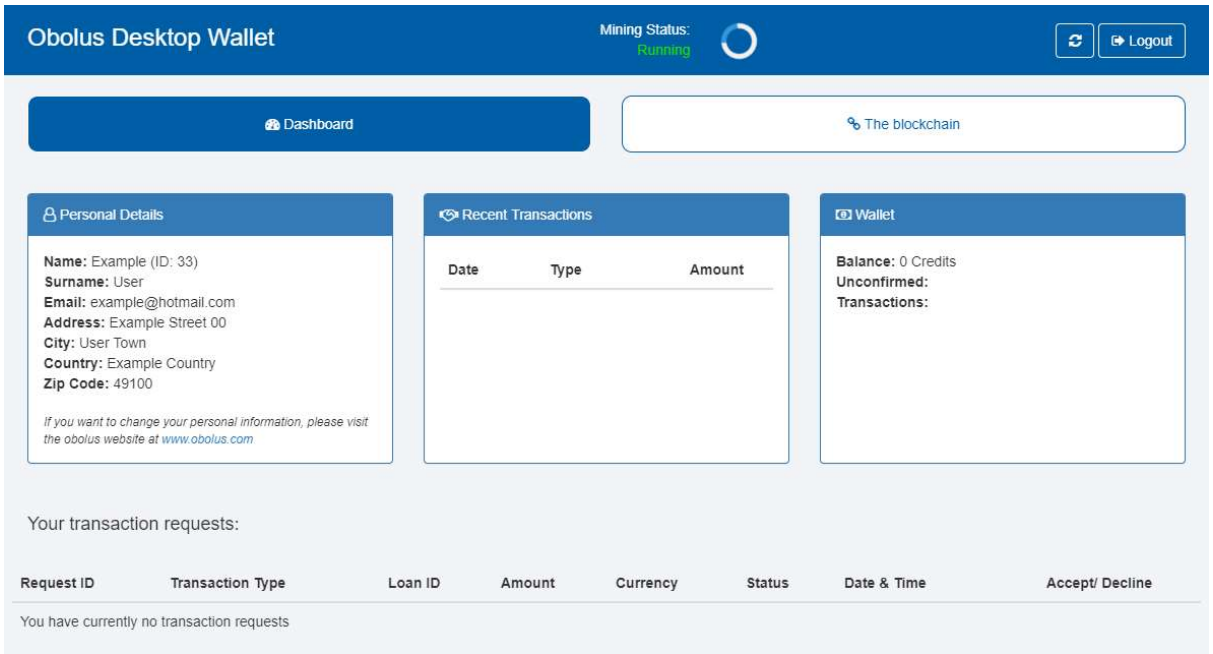
Εικόνα 71: Λήψη του προσωπικού πορτοφολίου.

Αφού ο χρήστης εγκαταστήσει το προσωπικό του πορτοφόλι για πρώτη φορά στον υπολογιστή του θα αντικρίσει την εξής οθόνη:



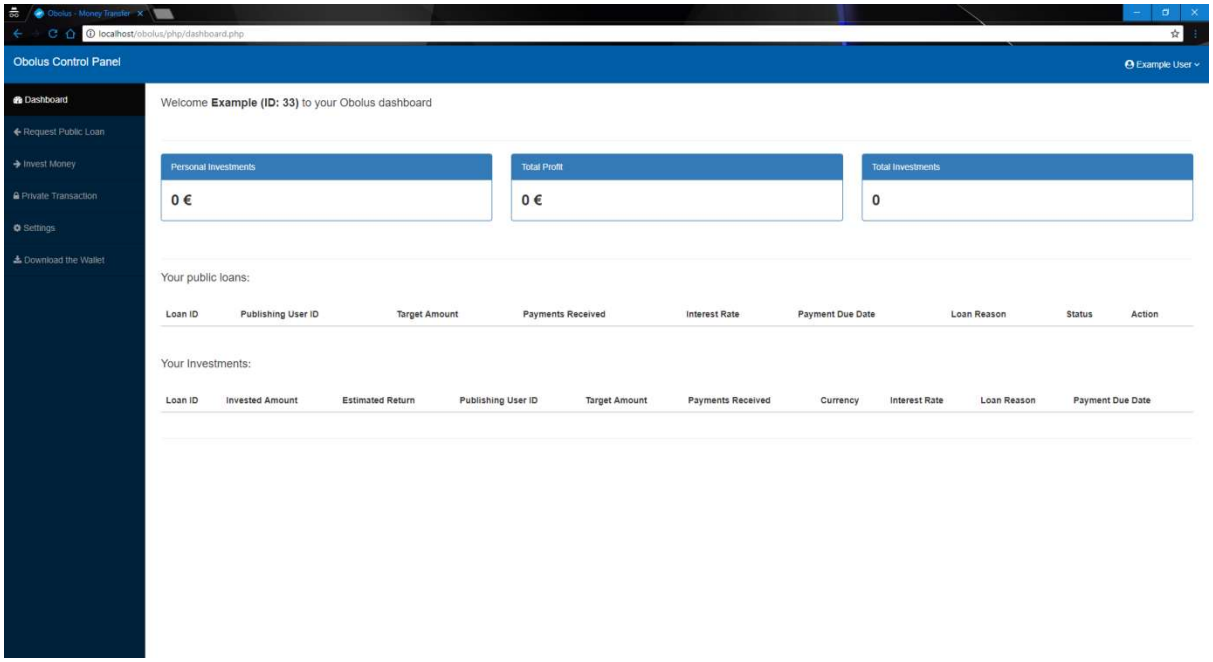
Εικόνα 72: Πρώτη σύνδεση στο προσωπικό πορτοφόλι του χρήστη

Κατά την πρώτη σύνδεση, το πορτοφόλι θα δημιουργήσει ένα ζεύγος κλειδιών και θα συνδέσει το πορτοφόλι με τον συγκεκριμένο χρήστη. Στις επόμενες συνδέσεις, δεν θα απαιτείται πια το email, αλλα μόνο ο κωδικός πρόσβασης για λόγους ασφαλείας.



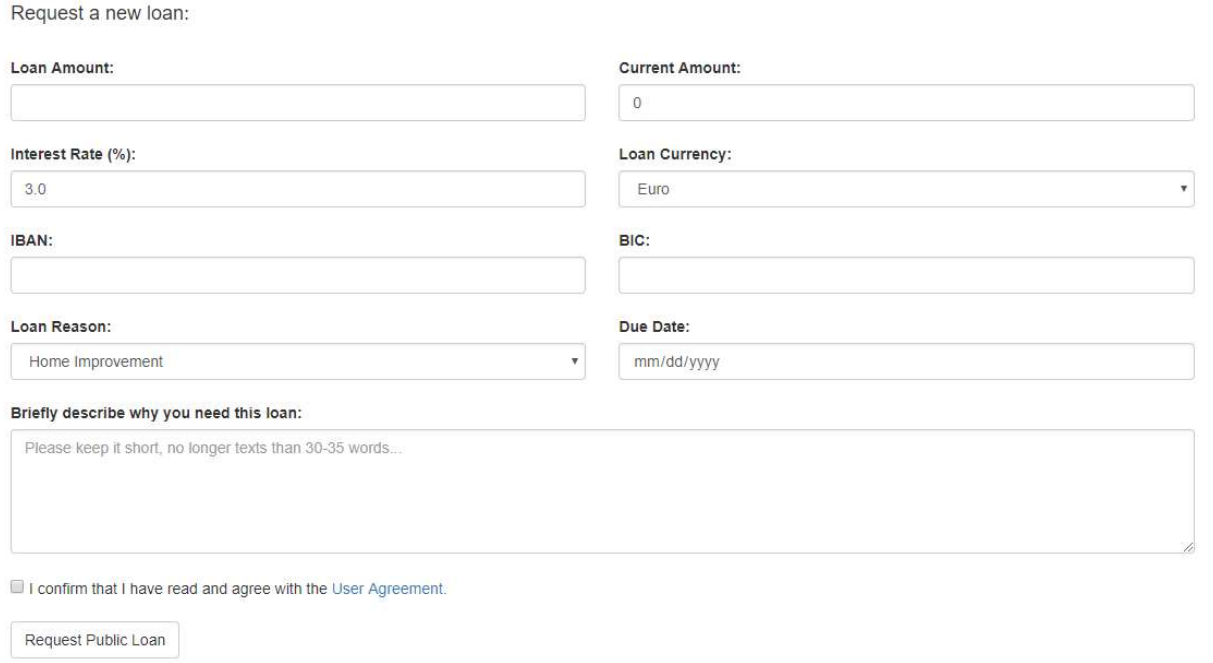
Εικόνα 73: Επιτυχής σύνδεση στο προσωπικό πορτοφόλι

Στην περίπτωση που ο χρήστη συνδεθεί με επιτυχία στο πορτοφόλι του θα αντικρίσει την παραπάνω εικόνα (Εικόνα 73). Εδώ θα μπορεί πλέον να διαχειρίζεται τις συναλλαγές του, να κάνει προεπισκόπηση των στοιχείων του και να ελέγχει και το υπόλοιπό του. Μετά την σύνδεση του χρήστη το προσωπικό πορτοφόλι, έχει ξεκλειδώσει και ο λογαριασμός του χρήστη στην διαδικτυακή πλατφόρμα.



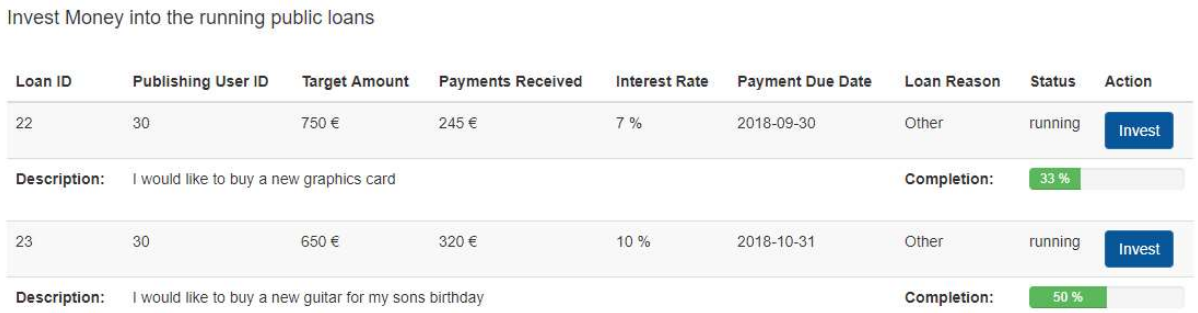
Εικόνα 74: Χρήστης με πλήρη πρόσβαση στην εφαρμογή

Πλέον ο χρήστης μπορεί να ζητήσει ένα δάνειο από τους υπόλοιπους χρήστες της εφαρμογής, να επενδύσει σε κάποιο ανοιχτό δάνειο, με μεταφέρει χρήματα ή κρυπτονομίσματα Obolus σε κάποιο άλλο χρήστη και να επεξεργαστεί τα δεδομένα του. Στην αρχική οθόνη Dashboard (πίνακας ελέγχου), εμφανίζονται συνοπτικά οι επενδύσεις, τα δάνεια και οι συναλλαγές του εκάστοτε χρήστη. Ακόμη εμφανίζεται και το συνολικό ποσό που έχει κερδίσει ο χρήστης από τις επενδύσεις του σε άλλα δάνεια.



Εικόνα 75: Δημιουργία αίτησης δανείου

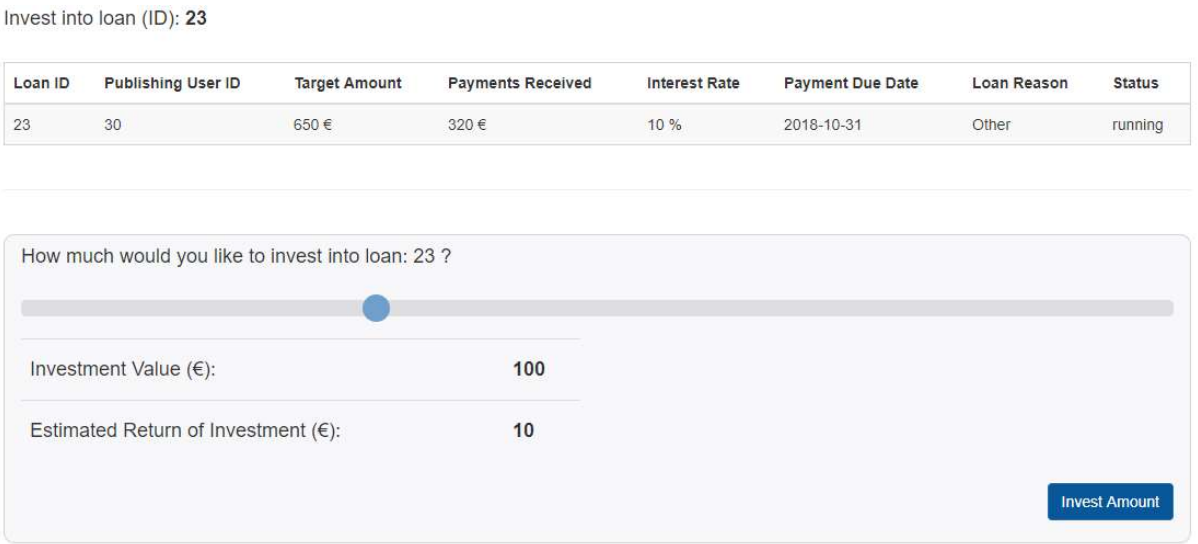
Παραπάνω (Εικόνα 75) φαίνεται η φόρμα που πρέπει να συμπληρώσει κάποιος χρήστης προκειμένου να ζητήσει ένα δημόσιο δάνειο. Τα πεδία που πρέπει να συμπληρωθούν είναι το ποσό του δανείου, αν υπάρχει ήδη κάποιο αρχικό κεφάλαιο, το επιτόκιο που προσφέρει, το νόμισμα, το IBAN και BIC στο οποίο θα κατατεθούν τα χρήματα, τον τίτλο του δανείου, την ημερομηνία λήξης, καθώς και μια μικρή περιγραφή για τον λόγο της αίτησης του δανείου.



Εικόνα 76: Επένδυση σε ενεργά δάνεια άλλων χρηστών

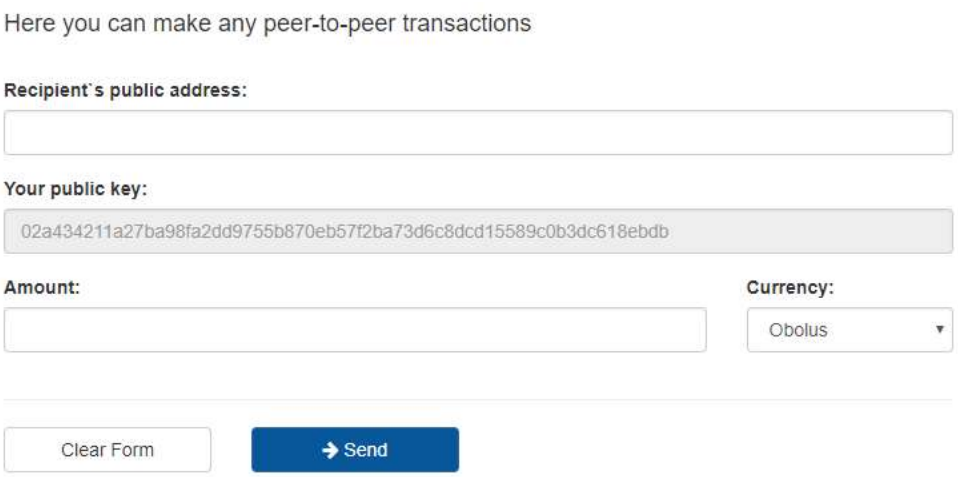
Στην Εικόνα 76 παρουσιάζεται η λειτουργία της εφαρμογής που επιτρέπει στους χρήστες να επενδύσουν σε δάνεια άλλων χρηστών. Εμφανίζονται όλες οι λεπτομέρειες του δανείου καθώς και το ποσοστό συμπλήρωσης του επιθυμητού ποσού. Οι χρήστες μπορούν να επιλέξουν κάποιο από αυτά και να επενδύσουν τα χρήματά τους. Να σημειωθεί ότι αν δεν επαληθευτεί η συναλλαγή από το προσωπικό πορτοφόλι, αυτή, δεν θα ολοκληρωθεί. Τότε το ποσό θα ξανα αποδεσμευτεί από το δάνειο.

Έστω ότι ένας χρήστης επιθυμεί να επενδύσει στο δάνειο με Loan ID 23, θα επιλέξει το κουμπί Invest και θα βρεθεί στην ακόλουθη οθόνη:



Εικόνα 77: Επένδυση σε δάνειο χρήστη

Εδώ ο χρήστης έχει την δυνατότητα να ξαναδεί τα στοιχεία του δανείου και να επιλέξει το ποσό που επιθυμεί να επενδύσει. Αν αποφασίσει να επενδύσει πρέπει απλά να επιλέξει το κουμπί Invest Amount και θα του επιστραφεί αντίστοιχο μήνυμα.
Στην περίπτωση που κάποιος χρήστη επιθυμεί να κάνει μια προσωπική συναλλαγή μέσω της επιλογής Private Transaction, θα του εμφανιστεί η εξής οθόνη:



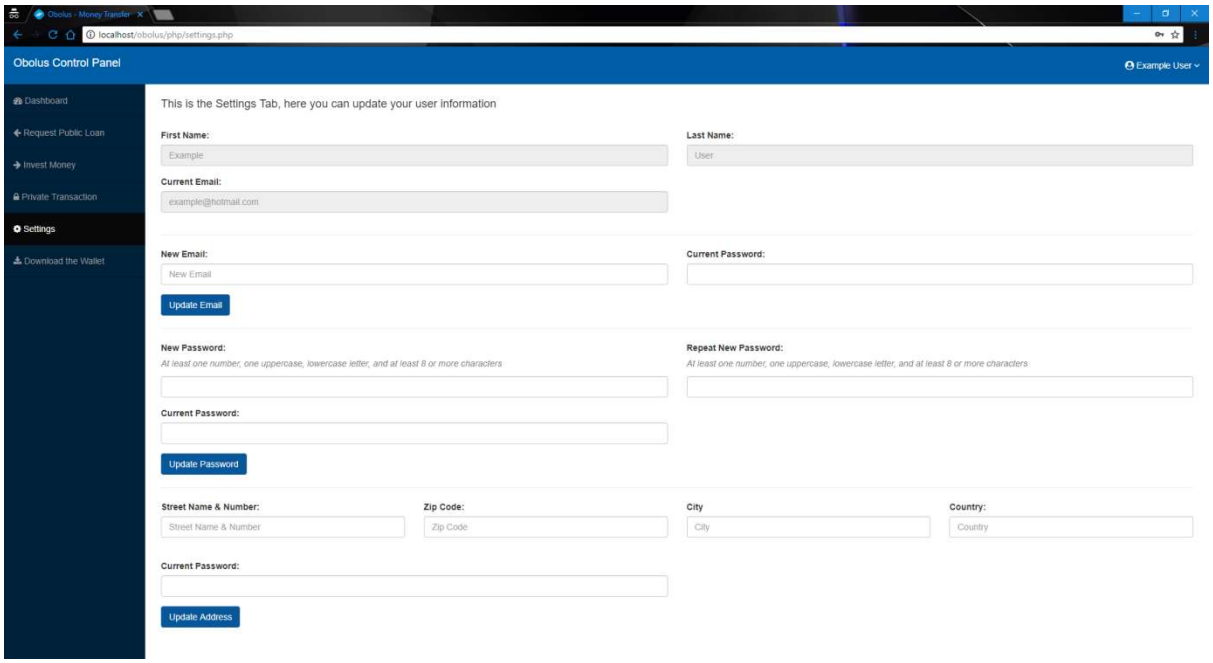
Εικόνα 78: Προσωπική συναλλαγή προς άλλο χρήστη της εφαρμογής

Στη φόρμα της προσωπικής συναλλαγής, ο χρήστης πρέπει να συμπληρώσει την διεύθυνση του παραλήπτη, το ποσό και το νόμισμα (ευρώ ή κρυπτονόμισμα).
Όταν τα συμπληρώσει αυτά και πατήσει Send, θα του εμφανιστεί το ακόλουθο μήνυμα:

Εικόνα 79: Μήνυμα επιτυχημένης συναλλαγής, επιβεβαίωση από το προσωπικό πορτοφόλι

Όπως αναφέρει και το μήνυμα, ο χρήστης πρέπει να εγκρίνει τις συναλλαγές που έκανε από το προσωπικό του πορτοφόλι.

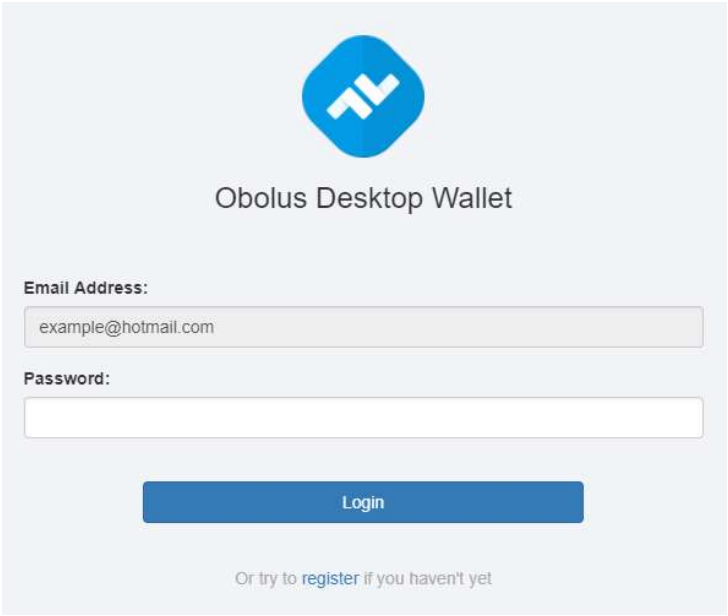
Στην επιλογή Settings, ο χρήστης μπορεί να επεξεργαστεί τα δεδομένα που έχει θέσει κατά την εγγραφή του στην πλατφόρμα.



Εικόνα 80: Επιλογή Settings για ενημέρωση στοιχείων του χρήστη

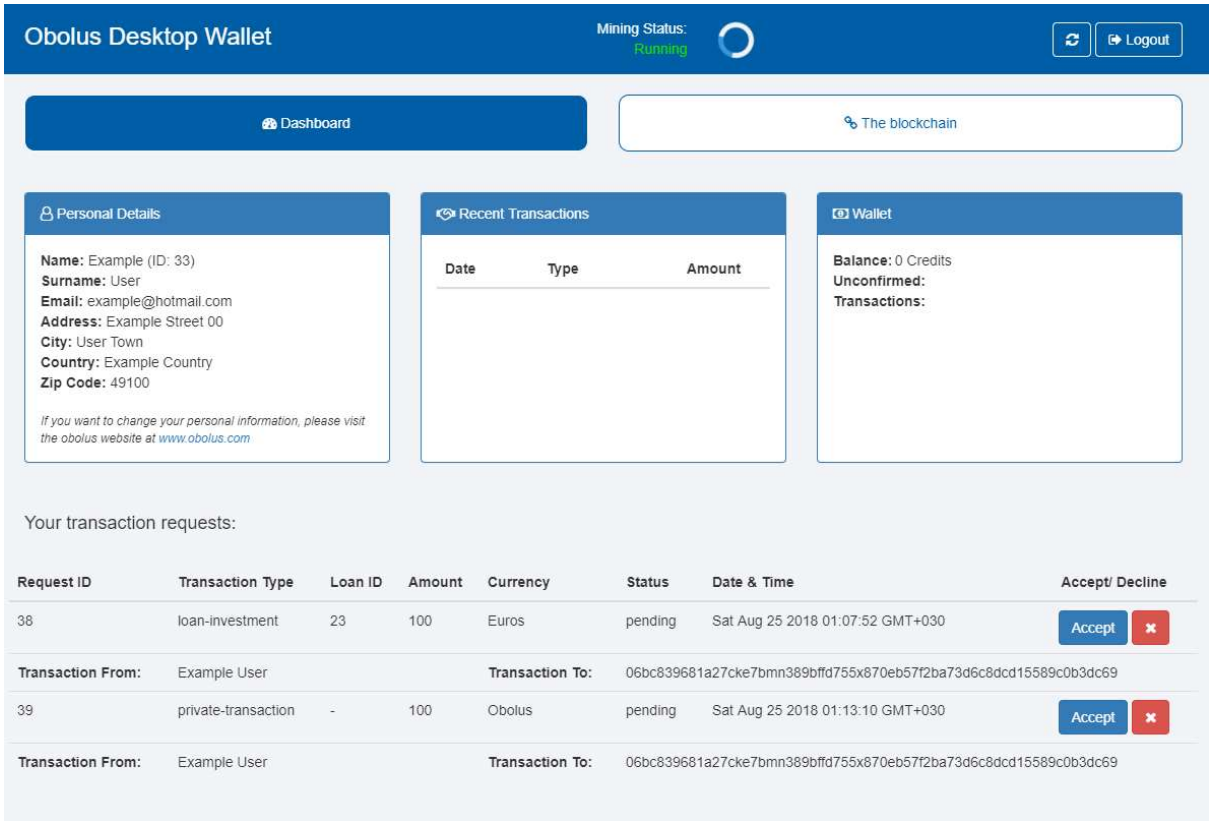
4.2 Το προσωπικό πορτοφόλι

Ας επιστρέψουμε στο πορτοφόλι του χρήστη για να δούμε τι έγινε με τις συναλλαγές που ξεκινήσαμε παραπάνω.



Εικόνα 81: Σύνδεση στο πορτοφόλι

Παρατηρούμε ότι πλέον δεν χρειάζεται να δηλώσουμε το email, αλλά μόνο τον κωδικό πρόσβασης.



Εικόνα 82: Προσωπικό πορτοφόλι με αιτήματα συναλλαγών

Πλέον έχουν εμφανιστεί οι δύο συναλλαγές που επιθυμούμε να κάνουμε, μία επένδυση σε δάνειο και άλλη μία ως προσωπική συναλλαγή. Για να τις αποδεχτούμε πρέπει απλά να πατήσουμε το πλήκτρο Accept, ή να τις αρνηθούμε επιλέγοντας το Χ.

Όσο το προσωπικό πορτοφόλι παραμένει ανοιχτό, η λειτουργία εξόρυξης είναι ενεργή, πράγμα που σημαίνει πως ο υπολογιστής του χρήστη ενεργεί ως κόμβος στο δίκτυο του Blockchain και επίσης υπάρχει πιθανότητα να εξορύξει και να κερδίσει κρυπτονομίσματα.

5. Συμπεράσματα

Η εκπόνηση της παρούσας πτυχιακής εργασίας είχε ως προαπαιτούμενο την έρευνα και ενδελεχή μελέτη διαφόρων Peer-to-peer εφαρμογών, εφαρμογών που κάνουν χρήση της τεχνολογίας των Blockchains και διαφόρων κρυπτονομισμάτων, με εστίαση στο κρυπτονόμισμα Bitcoin. Η ιδέα του Bitcoin ήρθε και διατάραξε την παγκόσμια οικονομία με αποτέλεσμα μεγάλες τράπεζες και κυβερνήσεις να απειλούνται από αυτό. Με τον καιρό, ακόμη και τα μεγάλα οικονομικά ιδρύματα άρχισαν να ανακαλύπτουν την δύναμη των κρυπτονομισμάτων και των Blockchains, αναπτύσσοντας δικές τους εφαρμογές βασισμένες στις συγκεκριμένες τεχνολογίες.

Το πληροφοριακό σύστημα που αναπτύχθηκε στα πλαίσια αυτής της πτυχιακής εργασίας και αποτελείται από τα δύο υποσυστήματα (την διαδικτυακή εφαρμογή και το προσωπικό πορτοφόλι), δεν είναι έτοιμο για να λειτουργήσει σε πραγματικό περιβάλλον. Δεν ήταν όμως και ο σκοπός του αυτός. Σκοπός του πληροφοριακού συστήματος ήταν η ανάδειξη μιας νέα τεχνολογίας και του τρόπου προσέγγισης ανάπτυξης τέτοιων συστημάτων. Το σύστημα του Obolus διαθέτει τις βασικές λειτουργίες μιας peer-to-peer lending εφαρμογής η οποία κάνει χρήση ενός ιδιωτικού Blockchain και υπό τις κατάλληλες προϋποθέσεις θα μπορούσε να ολοκληρωθεί σε εμπορική εφαρμογή. Οι προϋποθέσεις αυτές είναι ένας μεγάλος αριθμός χρηστών, αρκετός έτσι ώστε να μην παραβιάζεται η ακεραιότητα του βιβλιαρίου. Επίσης θα πρέπει να αντιμετωπιστούν κάποιες δυσκολίες στην διαχείριση του μεγάλου όγκου δεδομένων που θα δημιουργηθούν. Εκεί βρίσκεται και η δύναμη των blockchains, δεν χρειάζεται να υπάρχει μια παντοδύναμη πλατφόρμα η οποία θα εξυπηρετεί τους χρήστες της, αλλα το αντίθετο, οι χρήστες είναι αυτοί που θα δίνουν τη δυνατότητα στην πλατφόρμα να λειτουργεί. Ακόμη, προκειμένου η εφαρμογή να μπορέσει να λειτουργήσει σε εμπορικό επίπεδο, θα πρέπει να βρεθεί τρόπος επικοινωνίας της εφαρμογής με κάποιον πάροχο έξυπνων συμβολαίων, π.χ. Ethereum. Έτσι θα μπορέσουν οι συναλλαγές για επενδύσεις σε δάνεια χρηστών, να αποθηκεύονται σε ένα έξυπνο συμβόλαιο.

Κατά την διεξαγωγή της έρευνάς μου, συνειδητοποίησα πως μέχρι στιγμής οι εφαρμογές που υπάρχουν αξιοποιούν μόνο ένα πολύ μικρό δείγμα των δυνατοτήτων της τεχνολογίας του Blockchain. Πολλοί πιστεύουν πως η δημιουργία των Blockchains είναι αντίστοιχης σπουδαιότητας με την δημιουργία του παγκόσμιου ιστού. Η εμπιστοσύνη και η αποκεντροποίηση που προσφέρει η συγκεκριμένη τεχνολογία, μας δίνουν τη δυνατότητα να αναπτύσσουμε νέου τύπου εφαρμογές οι οποίες μέχρι τώρα δεν ήταν τόσο εύκολο να δημιουργηθούν. Ενδεχομένως, η νέα αυτή τεχνολογία να αλλάξει τόσο ριζικά τον τρόπο που ζούμε, επικοινωνούμε και που λειτουργεί ο κόσμος, όσο είχε κάνει και το διαδίκτυο το 1990.

Η εφαρμογή των blockchains είναι κάτι καινοτόμο. Τα blockchains είναι κάτι παραπάνω από απλά μια νέα τεχνολογία που θα αποφέρει κέρδος σε κάποιες επιχειρήσεις, είναι ένα όργανο για κοινωνική και πολιτική διαφοροποίηση. Η μία πτυχή της τεχνολογίας αυτής είναι η κρυπτοοικονομία, η οποία ανέδειξε κάποιες από τις δυνατότητες των blockchains στον τομέα αυτό. Δεν έχουν αναδειχθεί όμως όλες οι άλλες δυνατότητές τους. Ποιές θα είναι οι νέες εταιρείες Amazon, Google και Facebook μιας αποκεντροποιημένης τεχνολογίας; Το μήνυμα που μεταφέρει το blockchain είναι απλό, η καινοτομία πρέπει να μας καθοδηγήσει.

Τα blockchains δεν αναφέρονται σε ένα νέο και καλύτερο διαδίκτυο, μια νέα τράπεζα ή μια καλύτερη υπηρεσία, η καθίδρυση του blockchain εξαρτάται από το πως θα χρησιμοποιηθεί από την αγορά και όχι μόνο σε τεχνολογικό επίπεδο. Η εξοικείωση με την νέα αυτή τεχνολογία θα γίνει σταδιακά, ξεκινώντας από επιχειρήσεις που ασχολούνται με την τεχνολογία και ύστερα από διάφορους οργανισμούς που θα έχουν δει τα αποτελέσματα. Σίγουρα θα αντιμετωπιστούν κάποια προβλήματα, όπως και με κάθε νέα τεχνολογία. Νομοθετικά η Ευρώπη και συγκεκριμένα η Ελλάδα θα χρειαστεί να προσαρμοστεί σε νέα επιχειρηματικά μοντέλα. Θα πρέπει να διευκολυνθεί η λειτουργία τέτοιου τύπου χρηματοοικονομικών επιχειρήσεων. Το κράτος και η νομοθεσία θα χρειαστεί να αναγνωρίσουν τα έξυπνα συμβόλαια ως έγκυρα συμφωνητικά έτσι ώστε να μπορέσει να ανθίσει μια αγορά που θα βασίζεται σε αλγορίθμους κοινής συναίνεσης και έξυπνων συμβολαίων. Τα blockchains δεν μας περιορίζουν, αντιθέτως μας δίνουν νέα επίπεδα ελευθερίας και μας επιτρέπουν να προγραμματίσουμε και να σχεδιάζουμε τον κόσμο μας πάνω σε αυτά. Πάνω σε ένα δίκτυο που χαρακτηρίζεται από εμπιστοσύνη μεταξύ αγνώστων.

Στην Ελλάδα οι εφαρμογές των blockchains μπορούν να βελτιώσουν την καθημερινότητά μας κατά πολύ. Έννοιες όπως capital controls ή καταναλωτικά δάνεια με υψηλά επιτόκια δεν θα έχουν πια θέση στη ζωή μας. Ο κόσμος δεν θα προβληματίζεται για το πως να μεταφέρει τα χρήματά του, αφού θα μπορεί να μεταφέρει απευθείας αξία και αγαθά. Τα χρήματα και οι συναλλαγές θα αποθηκεύονται σε αποκεντροποιημένα βιβλιάρια και έτσι η επαλήθευση των δεδομένων θα γίνεται πολύ πιο γρήγορα και με μεγαλύτερη ασφάλεια. Θα αναπτυχθούν εφαρμογές οι οποίες θα βασίζονται στο σύνολο των χρηστών και όχι σε μια κεντρική αρχή, προωθώντας κατα μία έννοια την δημοκρατία, γιατί στη βάση τους τα blockchains είναι μια υλοποίηση της άμεσης δημοκρατίας.

Σίγουρα υπάρχουν και κάποιοι κινδύνοι, όπως παραδείγματος χάρη το ξέπλυμα χρήματος ή η αγοραπωλησία παράνομων αγαθών στη μαύρη αγορά. Εκεί θα χρειαστεί να επέμβει η νέα νομοθεσία, για να εξαλείψει τις παράνομες δραστηριότητες και να χρησιμοποιείται η εν λόγω τεχνολογία για το ευρύ καλό. Προκειμένου όμως να φτάσουμε σε αυτό το σημείο θα πρέπει να ενστερνιστούμε και να αναγνωρίσουμε τί πραγματικά μας προσφέρουν τα blockchains και πως μπορούμε να τα χρησιμοποιήσουμε για να δημιουργήσουμε ένα καλύτερο αύριο.

Βιβλιογραφία

[1] Mougayar, William. The business blockchain: promise, practice, and application of the next Internet technology. John Wiley & Sons, 2016.

[2] "Online Peer-to-Peer Lending--A Literature (PDF Free Download)." https://www.researchgate.net/publication/236735575_Online_Peer-to-Peer_Lending--A_Literature.

[3] Antonopoulos, Andreas M. Mastering Bitcoin: unlocking digital cryptocurrencies. " O'Reilly Media, Inc.", 2014.

[4] Laurence, Tiana. Blockchain for dummies. John Wiley & Sons, 2017.

[5] Dwork, Cynthia, and Moni Naor. "Pricing via processing or combatting junk mail." Annual International Cryptology Conference. Springer, Berlin, Heidelberg, 1992.

[6] Back, Adam. "Hashcash." (1997).

[7] Jakobsson, Markus, and Ari Juels. "Proofs of work and bread pudding protocols." Secure Information Networks. Springer, Boston, MA, 1999. 258-272.

[8] Swan, Melanie. Blockchain: Blueprint for a new economy. " O'Reilly Media, Inc.", 2015.

[9] Delmolino, Kevin, et al. "Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab." International Conference on Financial Cryptography and Data Security. Springer, Berlin, Heidelberg, 2016.

[10] Ethereum Project https://www.ethereum.org/

[11] Bootstrap https://getbootstrap.com/

[12] jQuery https://jquery.com/

[13] Electron | Build cross platform desktop apps with JavaScript, HTML, and CSS https://electronjs.org/

[14] The PeerJS Library https://peerjs.com/

[15] GitHub https://github.com/

[16] WebRTC https://webrtc.org/

[17] ISO 20022 https://www.iso20022.org/

[18] Implement ISO 20022 payment initiation messages in a payment processing solution, Edel Meehan, Oleg Tyschenko July 03, 2013 https://www.ibm.com/developerworks/data/library/techarticle/dm-1307isopayment/index.html

[19] Electron https://electronjs.org/

[20] NodeJS https://nodejs.org/en/

[21] NPM https://www.npmjs.com/

[22] Building a Desktop Application with Electron - https://dzone.com/articles/building-a-desktop-application-with-electron

[23] Crosby, Michael, et al. "Blockchain technology: Beyond bitcoin." Applied Innovation 2 (2016): 6-10.

[24] Elsden, Chris, et al. "Making Sense of Blockchain Applications: A Typology for HCI." Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. ACM, 2018.

Ευρετήριο

Στο σημείο αυτό κρίνεται αναγκαία η αποσαφήνιση ορισμένων εννοιών που θα εμφανίζονται συχνά στην παρούσα εργασία.

- **Βιβλιάριο**

  -Το βιβλιάριο (ledger) είναι ένα ψηφιακό αρχείο δεδομένων, το οποίο χρησιμεύει ως μέσο αποθήκευσης πληροφορίας.

  - Δημόσιο βιβλιάριο (public ledger) είναι το ψηφιακό αρχείο δεδομένων στο οποίο όλοι έχουν πρόσβαση και μπορούν να προσπελάσουν τα δεδομένα

  - Κατανεμημένο βιβλιάριο (distributed ledger) είναι το βιβλιάριο το οποίο είναι αποθηκευμένο και διαμοιρασμένο σε περισσότερους από ένα χρήστες. Τα διάφορα βιβλιάρια είναι ίδια μεταξύ τους.

- **Κρυπτογραφία**

  - Με τον όρο κρυπτογραφία εννοούμε την διαδικασία με την οποία μια πληροφορία γίνεται μη αναγνώσιμη στα μάτια τρίτων. Η αντίστροφη διαδικασία (αποκρυπτογράφηση), περιλαμβάνει την εφαρμογή ενός αλγορίθμου με αποτέλεσμα την επιστροφή στην αρχική πληροφορία.

- **Κόμβος (Node)**

  - Ως κόμβος (node) ενός δικτύου μπορεί να οριστεί οποιαδήποτε ηλεκτρονική συσκευή η οποία έχει πρόσβαση και μπορεί να ανταλλάσει πληροφορίες στο δίκτυο αυτό. Παράδειγμα συσκευών κόμβων σε ένα δίκτυο είναι οι υπολογιστές και οι εκτυπωτές σε ένα οικιακό δίκτυο.

- **Blockchain δίκτυο**

  - Με τον όρο Blockchain δίκτυο αναφερόμαστε στο σύνολο των κόμβων του συγκεκριμένου δικτύου, δηλαδή του Blockchain δικτύου.

- **Block**

  - Ένα block είναι ένα οργανωμένο σύνολο δεδομένων.

- **Hash (SHA 256)**

  - Η συνάρτηση Hash και συγκεκριμένα η συνάρτηση SHA-256 (Secure Hash Algorithm), είναι μια κρυπτογραφική συνάρτηση κατακερματισμού. Γενικά η συνάρτηση κατατεμαχισμού είναι μια μαθηματική συνάρτηση που έχει ως είσοδο μια αυθαίρετου μεγέθους ομάδα δεδομένων δίνει ως έξοδο μια καθορισμένου μεγέθους αλφαριθμητική στοιχειοσειρά.

- **Peer-to-peer (P2P)**

  - Ο όρος Peer-to-peer ή P2P σημαίνει ότι σε ένα δίκτυο, όλοι οι κόμβοι που συμμετέχουν είναι ομότιμοι μεταξύ τους, όλοι είναι ίσοι και όλοι οι κόμβοι μοιράζονται τις ίδιες υπηρεσίες του δικτύου. Όλοι οι κόμβοι σε ένα P2P δίκτυο αλληλοσυνδέονται σε ένα δίκτυο επίπεδης τοπολογίας και δεν υπάρχει κάποια κεντρική υπηρεσία. Στα P2P δίκτυα απουσιάζει τελείως η έννοια της ιεραρχίας.
  Ένα παράδειγμα Peer-to-peer δικτύου, είναι το δίκτυο του Bitcoin, όπου όλοι οι κόμβοι είναι ίσοι.

- **Κρυπτονόμισμα (Cryptocurrency)**

  - Τα κρυπτονομίσματα είναι ψηφιακά νομίσματα με αξία, τα οποία έχουν σχεδιαστεί ως μέσο συναλλαγής. Χαρακτηριστικό των κρυπτονομισμάτων είναι η χρήση κρυπτογραφικών αλγορίθμων για την εξασφάλιση της ακεραιότητας των δεδομένων των συναλλαγών.

- **Value Tokens ή Tokens**

  - Value Token ή token, μπορεί να χαρακτηριστεί ένα ψηφιακό αγαθό (digital asset). Τα tokens είναι κατα κάποιο τρόπο κλειδιά, τα οποία πιστοποιούν ότι ένας χρήστης είναι ιδιοκτήτης κάποιας αξίας. Τα value tokens μπορούν να αντιστοιχούν και σε φυσικά αγαθά τα οποία μπορούν να καταναλωθούν έναντι κάποιου προϊόντος ή υπηρεσίας μέσα στην πλατφόρμα. Τα tokens είναι κατα βάση αποθηκευμένα σε μία κατανεμημένη βάση τύπου blockchain, έτσι ώστε να εξασφαλίζεται η ακεραιότητα τους.

Περιεχόμενα