ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ ΒΙΟΙΑΤΡΙΚΗ

ΔΙΑΣΦΑΛΙΣΗ ΙΔΙΩΤΙΚΟΤΗΤΑΣ ΣΕ ΠΛΑΤΦΟΡΜΑ ΕΠΙΚΕΡΔΟΥΣ MOBILE CROWD SENSING

ΛΟΥΚΙΔΗΣ ΑΛΕΞΑΝΔΡΟΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
Επιβλέπων
ΠΛΑΓΙΑΝΑΚΟΣ ΒΑΣΙΛΕΙΟΣ

Λαμία, Νοέμβριος έτος 2017

UNIVERCITY OF THESSALY

SCHOOL OF SCIENCE

INFORMATICS AND COMPUTATIONAL BIOMEDICINE

PRIVACY PRESERVING PLATFORM FOR PROFITABLE MOBILE CROWD SENSING

LOYKIDIS ALEXANDROS

Master thesis

PLAGIANAKOS VASSILIS

Lamia

2017

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΔΙΑΤΜΗΜΑΤΙΚΟ ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ ΒΙΟΙΑΤΡΙΚΗ ΚΑΤΕΥΘΥΝΣΗ

«ΥΠΟΛΟΓΙΣΤΙΚΗ ΙΑΤΡΙΚΗ ΚΑΙ ΒΙΟΛΟΓΙΑ»

ΔΙΑΣΦΑΛΙΣΗ ΙΔΙΩΤΙΚΟΤΗΤΑΣ ΣΕ ΠΛΑΤΦΟΡΜΑ ΕΠΙΚΕΡΔΟΥΣ MOBILE CROWD SENSING

ΛΟΥΚΙΔΗΣ ΑΛΕΞΑΝΔΡΟΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Επιβλέπων
ΠΛΑΓΙΑΝΑΚΟΣ ΒΑΣΙΛΕΙΟΣ

Λαμία, Νοέμβριος 2017

«Υπεύθυνη Δήλωση μη λογοκλοπής και ανάληψης προσωπικής ευθύνης»

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

Ο ΔΗΛΩΝ

Ημερομηνία

Υπογραφή

ΔΙΑΣΦΑΛΙΣΗ ΙΔΙΩΤΙΚΟΤΗΤΑΣ ΣΕ ΠΛΑΤΦΟΡΜΑ ΕΠΙΚΕΡΔΟΥΣ MOBILE CROWD SENSING

ΛΟΥΚΙΔΗΣ ΑΛΕΞΑΝΔΡΟΣ

Τριμελής Επιτροπή:

ΣΠΑΘΟΥΛΑΣ ΓΕΩΡΓΙΟΣ

ΠΛΑΓΙΑΝΑΚΟΣ ΒΑΣΙΛΕΙΟΣ

ΚΑΚΑΡΟΥΝΤΑΣ ΑΘΑΝΑΣΙΟΣ

Επιστημονικός Σύμβουλος:

ΣΠΑΘΟΥΛΑΣ ΓΕΩΡΓΙΟΣ

Περίληψη

Το Internet των πραγμάτων ή αλλιώς Internet of Things (IOT) αναμένεται να γεφυρώσει το χάσμα μεταξύ του φυσικού και ψηφιακού κόσμου με την βοήθεια δισεκατομμυρίων αισθητήρων. Στην σημερινή εποχή ο καθένας είναι κάτοχος ή χρήστης μιας ή περισσότερων φορητών συσκευών που είναι σε θέση να συλλέξουν ποικιλόμορφα δεδομένα μέσω των αισθητήρων που αυτές φέρουν. Τα δεδομένα αυτά μπορεί αθροιστικά να αποτελούν χρήσιμη πληροφορία για ερευνητές, κοινωφελείς οργανισμούς ή εταιρίες. Ως εκ τούτου γεννιέται η ανάγκη για την δημιουργία μιας πλατφόρμας η οποία θα επιτρέπει την συνεργασία των ενδιαφερομένων φορέων με τους χρήστες των φορητών συσκευών για την μαζική συλλογή αυτών των δεδομένων. Καθώς όμως τα συλλεγόμενα δεδομένα αποτελούν ευαίσθητη προσωπική πληροφορία για τους κατόχους των συσκευών, απαιτείται ειδική πρόνοια κατά την συλλογή και την διαχείρισή τους.

Στην παρούσα εργασία παρουσιάζεται μία πλατφόρμα Mobile Crowd Sensing που επιτρέπει την ασφαλή συλλογή δεδομένων από ένα μεγάλο αριθμό χρηστών διατηρώντας παράλληλα την ιδιωτικότητα όλων των εμπλεκομένων μελών αλλά και ένα σύστημα πληρωμής μέσω ενός ψηφιακού νομίσματος του Bitcoin. Στην περίπτωση μας το σύστημα προσφέρει ένα μικρό αντίτιμο (micropayments) στους πωλητές των δεδομένων.

Στο πρώτο κεφάλαιο παρουσιάζεται η έννοια του Mobile Crowd Sensing (MCS), οι πιθανές εφαρμογές και τα προβλήματα της ιδιωτικότητας.

Στη συνέχεια υπάρχει η παρουσίαση και η περιγραφή των βασικών εννοιών που είναι απαραίτητες να γνωρίζει κάποιος για την κατανόηση της λειτουργίας της προτεινόμενης πλατφόρμας.

Στο επόμενο κεφάλαιο περιγράφονται όλες οι τεχνολογίες και οι βιβλιοθήκες που χρησιμοποιήθηκαν κατά την δημιουργία αλλά και κατά την λειτουργία της πλατφόρμας.

Στο τρίτο κεφάλαιο γίνεται εκτενής αναφορά στο λειτουργικό σύστημα Android ξεκινώντας από την δημιουργία του μέχρι και την σημερινή του εξέλιξη.

Στο τέταρτο κεφάλαιο περιγράφεται το ψηφιακό κρυπτονόμισμα Bitcoin όπου αναλύεται η δημιουργία, η παραγωγή, και η χρήση του ενώ αναλύονται κάποια από τα πλεονεκτήματα και μειονεκτήματα του.

Στο πέμπτο κεφάλαιο περιγράφεται η έννοια της κρυπτογραφίας, τα είδη της και οι υπηρεσίες που παρέχονται καθώς επιχειρείται και μία εκτενέστερη ανάλυση του κρυπτοσυστήματος RSA.

Τέλος στο τελευταίο κεφάλαιο πραγματοποιείται εκτενής περιγραφή των λειτουργιών των δύο εφαρμογών (PHP Web app & Android application) της πλατφόρμας, καθώς και τα πιο σημαντικά μέρη του κώδικα που εκτελείται σε κάθε περίπτωση.

Ευχαριστίες

Πριν την παρουσίαση αυτής της εργασίας , αισθάνομαι την ανάγκη να ευχαριστήσω θερμά όλους όσους με βοήθησαν και έπαιξαν καθοριστικό ρόλο στην πραγματοποίηση της και συγκεκριμένα:

Τον κ. Γεώργιο Σπαθούλα , καθηγητή του τμήματος Πληροφορικής με Εφαρμογές στη Βιοϊατρική του Πανεπιστημίου Θεσσαλίας, επιβλέπων της εργασίας για την πολύτιμη βοήθεια και καθοδήγηση σε όλη τη διάρκεια της δουλειάς μου.

Την οικογένεια μου και τους κοντινούς μου φίλους για την υπομονή, τη συμπαράσταση και την υποστήριξη που μου παρείχαν σε όλη τη διάρκεια των σπουδών και γενικότερα σε όλα τα στάδια της ζωής μου.

Τέλος θα ήθελα να ευχαριστήσω την Κωνσταντίνα για την υπομονή, την συμπαράσταση και την υποστήριξη που μου παρείχε στην προσπάθεια ολοκλήρωσης αυτού του εγχειρήματος μου.

Πίνακας εικόνων

Περιεχόμενα

1. Εισαγωγή

Η πρόσφατη αύξηση της χρήσης των έξυπνων τηλεφώνων και άλλων κινητών συσκευών έχει δώσει την ευκαιρία στο διαμοιρασμό μεγάλου όγκου πληροφοριών μέσω των αισθητήρων τους. Ως όρος Mobile Crowd Sensing (MCS) ορίζεται ως ο μεγάλος αριθμός αισθητήρων που βρίσκονται στις κινητές συσκευές και είναι σε θέση να συλλέξουν και να συνεισφέρουν πολύτιμα δεδομένα σε διαφορετικές εφαρμογές. Το γεγονός ότι οι περισσότερες συσκευές είναι πάντα συνδεδεμένες στο διαδίκτυο μαζί με τις πρόσφατες εξελίξεις στο τομέα του cloud computing έχει καταστήσει το MCS σε ένα τομέα με πολλές προοπτικές. Ωστόσο η κύρια ανησυχία σχετικά με αυτή την προσέγγιση είναι η ιδιωτικότα της πληροφορίας. Ενώ η διάθεση της πληροφορίας είναι εύκολη η συμμετοχή είναι μικρότερη από την αναμενόμενη λόγο της ανησυχίας περί προσωπικών δεδομένων. Ο συνδυασμός των παρεχόμενων δεδομένων μαζί με τις πληροφορίες που μπορεί να εξάγει κανείς από τα κοινωνικά δίκτυα αυξάνει σημαντικά τους κινδύνους της ιδιωτικής ζωής. Σε αυτή την εργασία, παρουσιάζεται μια πλατφόρμα διατήρησης της ιδιωτικής ζωής για τους χρήστες. Χρησιμοποιείται κρυπτογράφηση για την ανταλλαγή των δεδομένων και οι συναλλαγές μεταξύ των εμπλεκομένων γίνονται μέσω ενός δημοφιλούς κρυπτονομίσματος του Bitcoin. Με αυτό τον τρόπο προστατεύεται η ιδιωτικότητα των συμμετεχόντων [1] [2] [54]

1.1. Mobile Crowd Sensing

Η ενσωμάτωση των αισθητήρων στις καθημερινής χρήσης συσκευών οι οποίες είναι συνδεδεμένες στο διαδίκτυο έχει οδηγήσει στην εξέλιξη του Διαδικτύου ή αλλιώς όπως ονομάζεται Internet of things (IoT). Μια αναδυόμενη κατηγορία των IoT είναι εστιασμένη στους καταναλωτές. Αποτελούνται από υπολογιστικές συσκευές με αισθητήρες , οι οποίες είναι συνδεδεμένες με το διαδίκτυο. Αυτές είναι τα smartphones (IPhone, Google Nexus), συσκευές αναπαραγωγής μουσικής (IPods), ενσωματωμένα συστήματα παιχνιδιών (Wii,XboX Kinect) και συσκευές εντός του οχήματος(GPS, OBD-II). Έχουν γίνει εξαιρετικά δημοφιλής και είναι δυνητικά σημαντικές πηγές δεδομένων μέσω των αισθητήρων τους. Είναι εξοπλισμένες με διάφορες δυνατότητες όπως η ασύρματη σύνδεση που τους επιτρέπει να παράγουν δεδομένα στο διαδίκτυο. Αντικείμενα (π.χ. μηχανές καφέ) που παραδοσιακά δεν διέθεταν κάποια υπολογιστική, σήμερα διαθέτουν μια ποικιλία αισθητήρων, υπολογιστών και επικοινωνίας και μπορούν να χρησιμεύσουν ως γέφυρα σε άλλα αντικείμενα καθημερινής χρήσης ή δημιουργίας πληροφορίας για το περιβάλλον. Οι συσκευές είναι σε θέση να συλλέξουν και να συνεισφέρουν πολύτιμα δεδομένα για διαφορετικές εφαρμογές. Στο MCS, ένας συμμετέχων ή ένας παροχέας δεδομένων είναι ένα άτομο που συλλέγει και συνεισφέρει δεδομένα χρησιμοποιώντας μια κινητή συσκευή (Π.χ. ένα έξυπνο τηλέφωνο) που μεταφέρει ή συλλέγει δεδομένα και καταναλώνονται απευθείας από τους τελικούς χρήστες ή μετά από κάποια επεξεργασία τους.[1][2][5][54]

1.2. Πιθανές Εφαρμογές

Σε αυτή την ενότητα, συζητούμε εν συντομία τις υπάρχουσες εφαρμογές που προσφέρουν μια βάση για διάφορες ερευνητικές προκλήσεις. Κατατάσσουμε τις εφαρμογές MCS σε τρεις κατηγορίες με βάση το είδος του φαινομένου όπου μετριέται ή χαρτογραφείται. Αυτές μπορεί να κατατάσσονται σε εφαρμογές

περιβαλλοντικές που σε αυτές που αφορά τις υποδομές αλλά και τις κοινωνικές. Στις περιβαλλοντικές εφαρμογές MCS, τα φαινόμενα είναι εκείνα του φυσικού περιβάλλοντος. Παραδείγματα περιλαμβάνουν τη μέτρηση των επιπέδων ρύπανσης σε μια πόλη, σε ύδατα σε κολπίσκους και την παρακολούθηση περιοχών άγριας ζωής. Τέτοιες εφαρμογές δίνουν τη δυνατότητα χαρτογράφησης διαφόρων περιβαλλοντικών φαινομένων μεγάλης κλίμακας με τη συμμετοχή πολλών ατόμων. Παράδειγμα ανάπτυξης πρωτοτύπου για τη παρακολούθηση της ρύπανσης, Common Sense [10].Το Common Sense χρησιμοποιεί χειροκίνητη συσκευή ανίχνευσης της ποιότητας του αέρας που επικοινωνεί με κινητά (με χρήση Bluetooth) για τη μέτρηση διαφόρων τύπων αέρα ρύπους (π.χ. CO2, NOχ). Αυτές οι συσκευές, όταν αυξηθούν σε αριθμό θα μπορούν συλλογικά να μετρήσουν την ποιότητα του αέρα μιας κοινότητας ή μιας μεγάλης περιοχή. Ομοίως, μπορούν να χρησιμοποιηθούν μικρόφωνα σε κινητά τηλέφωνα για να παρακολουθούν τα επίπεδα θορύβου σε συγκεκριμένες περιοχές. Ένα άλλο παράδειγμα είναι το CreekWatch που αναπτύχθηκε από το Κέντρο Ερευνών Almaden της IBM. Η εφαρμογή είναι σε θέση να παρακολουθεί τα επίπεδα και την ποιότητα των υδάτων σε κολπίσκους συγκεντρώνοντας αναφορές από άτομα, όπως εικόνες που λαμβάνονται σε διάφορες τοποθεσίες κατά μήκος του ποταμού ή μηνύματα κειμένου σχετικά με την ποσότητα των απορριμμάτων. Τέτοιες πληροφορίες μπορούν να χρησιμοποιηθούν από την όποια αρμόδια υπηρεσία έλεγχου του νερού για την παρακολούθηση των επιπέδων ρύπανσης. Οι εφαρμογές υποδομής περιλαμβάνουν τη μέτρηση των φαινομένων μεγάλης κλίμακας που σχετίζονται με την δημόσια υποδομή. Τα παραδείγματα περιλαμβάνουν τη μέτρηση κυκλοφοριακής συμφόρηση, οδικές συνθήκες, διαθεσιμότητα στάθμευσης, διακοπές λειτουργίας δημοσίων έργων (π.χ. δυσλειτουργία πυροσβεστικούς κρουνού, σπασμένα φανάρια) σε πραγματικό χρόνο. Άλλο παράδειγμα εφαρμογής σε MCS που αναπτύσσεται σε σχέση με την κυκλοφοριακή συμφόρηση των πόλεων είναι το Car-Tel [14] και το Nericell [15] του MIT και της Microsoft αντίστοιχα. Το CarTel χρησιμοποιεί εξειδικευμένες συσκευές εγκατεστημένες σε αυτοκίνητα για τη μέτρηση της θέσης και της ταχύτητας των αυτοκινήτων και μεταδίδει τις μετρήσεις χρησιμοποιώντας τα κοινά WiFi hotspots προς το κεντρικό διακομιστή. Ο κεντρικός διακομιστής μπορεί να ερωτηθεί για την παροχή πληροφοριών όπως για τις διαδρομές με την λιγότερη καθυστέρηση ή τα σημεία με τα μεγαλύτερα προβλήματα της κυκλοφορίας. Από την άλλη πλευρά, το Nericell χρησιμοποιεί τα άτομα, τα κινητά τηλέφωνα των χρηστών όχι μόνο για τον καθορισμό του μέσου όρου της ταχύτητας ή τις κυκλοφοριακές καθυστερήσεις, αλλά επίσης για την ανιχνεύσει του θορύβου(ιδίως σε χώρες όπως η Ινδία) και για τις λακκούβες στους δρόμους. Ένα άλλο παράδειγμα είναι το ParkNet [16], μια εφαρμογή που εντοπίζει διαθέσιμα σημεία στάθμευσης σε πόλεις χρησιμοποιώντας συσκευές ανίχνευσης υπερήχων εγκατεστημένες σε αυτοκίνητα σε συνδυασμό με τα έξυπνα τηλέφωνα. Μια άλλη κατηγορία είναι οι κοινωνικές εφαρμογές, όπου τα άτομα μοιράζονται πληροφορίες μεταξύ τους. Για παράδειγμα, άτομα μπορούν να μοιράζονται τα δεδομένα άσκησής τους (π.χ. πόσος είναι ο χρόνος που ασκείται σε μια μέρα) και συγκρίνονται με τα επίπεδα άσκησης της υπόλοιπης κοινότητας. Μπορεί να χρησιμοποιηθεί αυτή η σύγκριση για να βοηθήσει στη βελτίωση της καθημερινές άσκησης. Παράδειγμα ανάπτυξης εφαρμογής είναι το BikeNet [17] και το DietSense [18]. Στο BikeNet, ο χρήστης διαθέτει την τοποθεσία του με το ποδήλατο καθώς και το είδος της διαδρομής (π.χ., περιεχόμενο CO2 στη διαδρομή, δυσκολία της βόλτας) και συγκεντρώνονται όλα τα δεδομένα που πρέπει να ληφθούν στις ποδηλατικές διαδρομές. Στο DietSense, τα άτομα αποστέλλουν φωτογραφίες για αυτά που τρώνε και τα μοιράζονται μέσα σε μια κοινότητα για να συγκριθεί το φαγητό τους με τις συνήθειες τους. Μια τυπική περίπτωση χρήσης για

αυτό είναι για μια κοινότητα διαβητικών ώστε να παρακολουθούν τι τρώνε και να ελέγχουν τη διατροφή τους ή να παρέχουν προτάσεις σε άλλους.

Οι εφαρμογές μπορούν επίσης να ταξινομηθούν και σε δύο επιπλέον κατηγορίες. Αυτή της ατομικής αλλά και της συλλογική ανίχνευσης. Στις εφαρμογές ατομικής ανίχνευσης, η κατηγορία αναφέρεται σε ένα άτομο για παράδειγμα, η παρακολούθηση της δραστηριότητας της κίνησης (π.χ.,τρέξιμο, περπάτημα, άσκηση) ενός ατόμου για την προσωπική τήρηση αρχείων ή για λόγους υγειονομικής περίθαλψης. Ένα άλλο παράδειγμα προσωπικής χρήσης είναι αυτό της παρακολούθησης του τρόπου μεταφοράς ενός ατόμου ώστε να καθοριστεί η κατανάλωση της ενέργειας του. Από την άλλη πλευρά οι συλλογικές εφαρμογές σχετίζονται με την παρακολούθηση των φαινομένων μεγάλης κλίμακας που δεν μπορούν εύκολα να μετρηθούν από ένα μόνο άτομο. Για παράδειγμα, ένα συστήματα ενδέχεται να απαιτεί παρακολούθηση της κυκλοφοριακής συμφόρησης ή παρακολούθηση της ατμοσφαιρικής ρύπανσης. Αυτά τα φαινόμενα μπορούν να μετρηθούν με ακρίβεια μόνο όταν πολλά άτομα παρέχουν με ταχύτητα αυτές τις ποιοτικές πληροφορίες από τις ημερήσιες μετακινήσεις τους. Αυτά τα δεδομένα στη συνέχεια συγκεντρώνονται χωρο-χρονικά και αναλύονται σε επίπεδα συμφόρησης και ρύπανσης. Μια άλλη συλλογική χρήση είναι με την συμμετοχή ατόμων όπου οι χρήστες συμβάλουν ενεργά στέλνοντας δεδομένα (π.χ. λήψη φωτογραφιών, αναγγελία κλεισίματος δρόμου) που σχετίζονται με φαινόμενα μεγάλης κλίμακας.

Ορισμένες εφαρμογές μπορεί να κατηγοριοποιηθούν με βάση τη συμμετοχή των συμμετεχόντων. Ως συμμετοχική ή περιστασιακή. Σε μια συμμετοχική περίπτωση οι συμμετέχοντες συμφωνούν να εκπληρώσουν τις ζητούμενες δραστηριότητες ανίχνευσης. Έτσι εμπλέκονται ενεργά στην δράση ανίχνευσης (π.χ.λήψη φωτογραφίας ή εισαγωγή δεδομένων). Στη περιστασιακή περίπτωση, τα δεδομένα συλλέγονται με ελάχιστη ή μηδενική συμμετοχή του συμμετέχοντα (π.χ. αναφορά ταχύτητας κατά την οδήγηση, συνεχής δειγματοληψία θέσης χωρίς τη ρητή ενέργεια του χρήστη).Επίσης μπορεί να τρέξει ως διαδικασία στο παρασκήνιο, συνεπώς για τη συλλογή των δεδομένων δεν απαιτείται αλληλεπίδραση με τα άτομα που μεταφέρουν την πληροφορία.[1][4][6][54]

1.3. Προβλήματα ιδιωτικότητας

Μια σημαντική πτυχή των εφαρμογών MCS είναι ότι ενδέχεται να συλλέξουν ευαίσθητα δεδομένα αισθητήρων σε ιδιώτες. Για παράδειγμα, με τον αισθητήρα του GPS οι ενδιαφερόμενοι μπορούν να χρησιμοποιήσουν τις πληροφορίες για την εξαγωγή προσωπικών πληροφοριών για το άτομο, όπως οι διαδρομές της καθημερινές μετακινήσεις του ή την κατοικία του αλλά και την εργασία του [19]. Από την άλλη πλευρά, αυτές οι μετρήσεις του αισθητήρα GPS (από ημερήσιες μετακινήσεις) που μοιράζονται μέσα σε μια ευρύτερη κοινότητα μπορεί να χρησιμοποιηθούν ως δεδομένα για την κατανομή της κυκλοφοριακής συμφόρησης σε μία πόλη [14]. Έτσι, είναι σημαντικό να διατηρηθεί η ασφάλεια και η ιδιωτική ζωή ενός ατόμου, αλλά να επιτυγχάνεται και η ταυτόχρονη χρήση των εφαρμογών MCS. Είναι επίσης απαραίτητο να εξασφαλιστεί ότι τα δεδομένα δεν αποκαλύπτονται σε αναξιόπιστους, δηλαδή σε τρίτους οι οποίοι δεν έχουν σχέση με την εφαρμογή. Ένα πρόβλημα που προκύπτει από την χρήση των εφαρμογών με crowdsensing είναι όταν υπάρχουν κακόβουλα άτομα που συνεισφέρουν με λανθασμένα δεδομένα των αισθητήρων τους (π.χ. ψευδείς αναγνώσεις GPS) ως εκ τούτου, η διατήρηση της ακεραιότητας των συλλεγόμενων δεδομένων από τους αισθητήρες είναι ένα σημαντικό πρόβλημα. Μια

δημοφιλής προσέγγιση για τη διατήρηση της ιδιωτικής ζωής των δεδομένων είναι η ανωνυμοποίηση [20], η οποία καταργεί οποιαδήποτε αναγνώριση πληροφοριών από τα δεδομένα των αισθητήρων πριν τη διαθέσει σε ένα τρίτο μέρος. Το μειονέκτημα σε μια τέτοια προσέγγιση είναι ότι οι ανώνυμες πλέον μετρήσεις του αισθητήρα GPS (ή της τοποθεσίας) μπορούν να παραμείνουν ακόμα και να χρησιμοποιηθούν για την εξαγωγή των τοποθεσιών που επισκέπτεστε συχνά ένα άτομο και να αντλήσει τα προσωπικά του στοιχεία. Μια άλλη προσέγγιση για τη διατήρηση της ασφαλής ιδιωτικής ζωής είναι με την χρήση της τεχνικής τα κρυπτογράφησης [21], όπου μετατρέπονται τα δεδομένων προκειμένου να διατηρηθεί η ιδιωτικότητα. Η τεχνική της κρυπτογράφησης είναι πολύπλοκη και απαιτεί την παραγωγή και την συντήρηση πολλαπλών κλειδιών, η οποία οδηγεί επίσης σε υψηλή κατανάλωση ενέργειας αλλά και πόρων. Μία άλλη προσέγγιση είναι αυτή η οποία βασίζεται στην προσθήκη κοινής πληροφορίας στους αισθητήρες για την διατήρηση της ιδιωτικότητας. Αυτό επιτυγχάνεται με την προσθήκη θορύβου [22, 23] με τέτοιο τρόπο που να διατηρείται η ιδιωτική ζωή ενός ατόμου, αλλά ταυτόχρονα να είναι εφικτός ο υπολογισμός των στατιστικών στοιχείων του ενδιαφέροντος με μεγάλη ακρίβεια (λόγω της φύσης του θορύβου προστέθηκε). Για παράδειγμα, σε μια εφαρμογή παρακολούθησης βάρους, είναι σημαντικό να υπολογίσετε τον μέσο όρο βάρος του πληθυσμού. Κάθε άτομο θεωρεί ευαίσθητη πληροφορία την αποκάλυψη του βάρους του και το διαταράσσει προσθέτοντας έναν τυχαίο αριθμό, που προέρχεται από μια γνωστή διανομή . Παρόλο που τα μεμονωμένα βάρη διαταράσσονται και εμφανίζονται τυχαία, όταν αυτές οι τιμές υπολογίζονται κατά μέσο όρο, το τυχαιοποιημένο συστατικό εξαφανίζεται (δεδομένου επαρκούς αριθμού ατόμων), και το μέσο βάρος της κοινότητας μπορεί να υπολογιστεί με υψηλό βαθμό ακρίβειας. Η ακεραιότητα των δεδομένων πρέπει να διασφαλίζεται από τα παραγόμενα δεδομένα των αισθητήρων και αποτελεί ένα πρόβλημα που πρέπει να αντιμετωπίζεται από τις εφαρμογές των MCS. Μερικές προσεγγίσεις έχουν προταθεί στην υπάρχουσα βιβλιογραφία [24]. Μια τέτοια προσέγγιση είναι με την υπογραφή του αισθητήρα δεδομένων (με αξιόπιστο υλικό εγκατεστημένο στο κινητό τηλέφωνα). Ωστόσο, αυτή η προσέγγιση έχει δυνητικά προβλήματα, καθώς πρέπει να γίνει η επαλήθευση ακόμα και στο λογισμικό. Για παράδειγμα, μια θέση GPS μπορεί να διαταραχθεί με την προσθήκη θορύβου για την προστασία της ιδιωτικότητας. Παρατηρείται ότι η ιδιωτικότητα είναι υποκειμενική για κάθε χρήστη, δηλαδή κάθε άτομο αντιλαμβάνεται διαφορετικό την ιδιωτικότητα του. Για παράδειγμα, ένα άτομο μπορεί να είναι πρόθυμο να μοιραστεί τη θέση του συνεχώς, ενώ ένα άλλο μπορεί να μην το επιθυμεί. Οι εφαρμογές MCS πρέπει να αντιμετωπίσουν το πρόβλημα της ακεραιότητας δεδομένων, ώστε να εξάγουν ουσιαστικής σημασίας συμπεράσματα μέσω των αθροιστικών αισθητήρων δεδομένων.

Στην παρούσα πλατφόρμα υπάρχει ένας αξιόπιστος εξυπηρετητής που εκτελεί τις κατάλληλες ενέργειες χωρίς να αλλάξει τα δεδομένα, αλλά και να μην παραβιάσει την προστασία της ιδιωτικότητας των συμμετεχόντων παρακολουθώντας τα δεδομένα που ανταλλάσσονται. Η πλατφόρμα προστατεύει το απόρρητο των χρηστών κάνοντας κρυπτογράφηση με το δημόσιο κλειδί του καταναλωτή δεδομένων που περιλαμβάνεται στο αρχικό αίτημα δεδομένων. Επομένως είναι αδύνατο για το διακομιστή για να αποκρυπτογραφήσει τις τιμές των δεδομένων που ανταλλάσσονται. Παρόλα αυτά ο διακομιστής μπορεί να συσχετίσει τις διευθύνσεις bitcoin που χρησιμοποιούνται και να βγάλει χρήσιμα συμπεράσματα για τους παραγωγούς των δεδομένων όπως την τοποθεσία. Ωστόσο κάθε χρήστης μπορεί να χρησιμοποιεί περισσότερα από μια bitcoin διευθύνση. Μπορεί πρακτικά να χρησιμοποιήσει μια νέα διεύθυνση bitcoin για κάθε μία νέα αίτηση που κάνει, προκειμένου να αυξήσει την

ιδιωτικότητα του. Ο καταναλωτής δεδομένων λαμβάνει τα δεδομένα που ζητά και δεν γνωρίζει την ταυτότητα των παραγωγών δεδομένων. Ο διακομιστής γνωρίζει μονάχα τις διευθύνσεις bitcoin οι οποίες χρησιμοποιούνται για τις συναλλαγές μεταξύ του παραγωγού και του καταναλωτή δεδομένων. Τέλος, οι παραγωγοί δεδομένων δεν μπορούν να προσδιορίσουν τον καταναλωτή στον οποίο δημοσιεύονται τα δεδομένα τους γιατί ο καθένας πληρώνεται από το διακομιστή και έτσι η διεύθυνση του bitcoin είναι άγνωστη.[3][4][5][7][54]

2. Εργαλεία Βασική ορολογία

2.1. Ιστοσελίδα

Ιστοσελίδα (website) είναι ένα είδος εγγράφου του παγκόσμιου ιστού που περιλαμβάνει πληροφορίες με τη μορφή κειμένου, υπερκειμένου και πολυμέσων. Πολλές ιστοσελίδες μαζί αποτελούν έναν ιστότοπο και εμφανίζονται κάτω από το ίδιο όνομα χώρου (domain name πχ www.google.gr). Οι ιστοσελίδες μπορεί να συνδέονται μεταξύ τους μέσω συνδέσμων (links) πάνω στα οποία ο χρήστης μπορεί να κάνει "click" και να οδηγηθεί από τη μία στην άλλη σελίδα.[24][44]

2.2. Web browser

2.2.1. Περιγραφή

Ένας φυλλομετρητής (web browser) είναι ένα λογισμικό το οποίο έχει σκοπό την ανάκτηση, την παρουσίαση και τη μετατροπή ιστοσελίδων από τον παγκόσμιο ιστό (www). Δίνοντας στον browser μία διεύθυνση URL, μπορεί να παρουσιάσει τα δεδομένα που απαιτούνται, τα οποία μπορεί να είναι μια ιστοσελίδα, εικόνες, βίντεο ή άλλα πολυμέσα. Με χρήση συνδέσμων (links) ο χρήστης μπορεί να οδηγήσει τον browser να παρουσιάσει κάποια άλλη ιστοσελίδα ή δεδομένα. [11][44]

Εικόνα 1 Λογότυπα Web browsers (φυλλομετρητών)

Οι κυριότεροι web browsers με τους περισσότερους χρήστες είναι:

- Google Chrome
- Mozilla Firefox
- Opera
- Microsoft Internet Explorer
- Safari

2.3. Web Server

2.3.1. Περιγραφή

Ο web server είναι ένα σύστημα υπολογιστή το οποίο επεξεργάζεται αιτήματα μέσω HTTP. Ο όρος αυτός μπορεί να αναφέρεται είτε σε ολόκληρο το σύστημα είτε σε συγκεκριμένα στο λογισμικό το οποίο αναλαμβάνει να δέχεται και να χειρίζεται αιτήματα HTTP(HTTP requests).
Η πιο κοινή χρήση του web server είναι να φιλοξενούν ιστοσελίδες, αλλά έχουν και πολλές άλλες χρήσεις όπως αποθήκευση δεδομένων εκτέλεση εφαρμογών, διαχείριση mail, FTP και αλλά. Η επικοινωνία μεταξύ του server και του client γίνεται με χρήση του πρωτοκόλλου HTTP. [25][44]

2.3.2.Κύρια χαρακτηριστικά

Τα κυριότερα χαρακτηριστικά των web servers είναι τα εξής:

- **virtual hosting** για τη φιλοξενία πολλών διαφορετικών websites υπό την ίδια διεύθυνση IP
- **Υποστήριξη μεγάλων αρχείων** ώστε να υπάρχει δυνατότητα εξυπηρέτησης αιτήσεων για μεγάλου όγκου αρχεία.
- **Bandwidth throttling** είναι ένα μέτρο που χρησιμοποιούνται σε δίκτυα επικοινωνίας για τη ρύθμιση της κυκλοφορίας του δικτύου και την ελαχιστοποίηση εύρους ζώνης κυκλοφοριακής συμφόρησης
- **Server Side Scripting** για τη δημιουργία δυναμικών ιστοσελίδων όπωςγια παράδειγμα με PHP.[25]

2.4. Άδειες λογισμικού (Software License)

2.4.1. Περιγραφή

Οι πιο συνηθισμένες άδειες λογισμικού είναι οι εξής :

- **Apache license 2.0** Είναι μία open source άδεια όπου το λογισμικό που διατίθεται με αυτή μπορεί να χρησιμοποιηθεί για εμπορική χρήση να γίνουν μετατροπές σε αυτό, να διανεμηθεί, να διανεμηθεί με ‘υπό άδεια’, να χρησιμοποιηθεί για ιδιωτική χρήση, να χρησιμοποιηθεί για αιτήσεις πατέντας

καθώς και να περιλαμβάνει εγγύηση. Δεν μπορεί να κατηγορηθεί ο ιδιοκτήτης του λογισμικού για ευθύνες από ζημίες ούτε να χρησιμοποιηθούν τα Trademarks αυτού. Πρέπει να περιλαμβάνεται το Copyright, η άδεια, να δηλώνονται τυχόν αλλαγές καθώς και το αρχείο Notice που περιλαμβάνεται.

- **MIT license.** Πρόκειται για μία μικρή, ανεκτική άδεια όπου γενικά ο προγραμματιστής μπορεί να κάνει ότι θέλει αρκεί να περιλαμβάνει το αρχικό Copyright και την άδεια. Γενικά μπορεί να χρησιμοποιηθεί για εμπορική χρήση, μπορούν να γίνουν μετατροπές στον αρχικό κώδικα, να διανεμηθεί, να διανεμηθεί 'υπό άδεια' και να χρησιμοποιηθεί για ιδιωτική χρήση. Δεν μπορεί να κατηγορηθεί ο ιδιοκτήτης του για ευθύνες από ζημίες.
- **GNU General Public License v3/v2**. Ο προγραμματιστής μπορεί να αντιγράψει διανείμει και να κάνει μετατροπές το λογισμικό αρκεί να γίνεται καταγραφή των αλλαγών στα αρχεία και οι αλλαγές να συνεχίζουν να είναι υπό την ίδια άδεια. Μπορεί να διανεμηθεί η εφαρμογή εμπορικά αλλά πρέπει να είναι open source υπό την GPLv3. Δεν μπορεί να κατηγορηθεί ο ιδιοκτήτης λογισμικού για ευθύνες από ζημίες ούτε να διανεμηθεί υπό άλλη άδεια. Πρέπει να περιλαμβάνεται η αρχική έκδοση της βιβλιοθήκης, να ορίζονται οι αλλαγές, να περιλαμβάνεται η άδεια και το copyright καθώς και το παραγόμενο λογισμικό που χρησιμοποιεί βιβλιοθήκη με αυτή την άδεια να είναι open source .[11]

2.5. Αντικειμενοστραφής προγραμματισμός (Object Oriented Programming)

Ο αντικειμενοστραφής προγραμματισμός (Object Oriented Programming - OPP) είναι ένα μοντέλο προγραμματισμού που αντιπροσωπεύει την έννοια των «αντικειμένων» τα οποία έχουν πεδία δεδομένων και σχετισμένες διαδικασίες γνωστές ως μεθόδους. Οι πιο γνωστές γλώσσες προγραμματισμού οι οποίες βασίζονται στον αντικειμενοστραφή προγραμματισμό είναι οι Java, C++, Objective – C, C#, Javascript, Perl και άλλες.

Ένα αντικείμενο είναι μία αφηρημένη μορφή δεδομένων με επιπλέον τον πολυμορφισμό και την κληρονομικότητα. Ένα αντικείμενο επίσης έχει «κατάσταση» (δεδομένα) και «συμπεριφορά» (κώδικας). Τα αντικείμενα συνήθως αναπαριστούν πράγματα του πραγματικού κόσμου, όπως π.χ. ένας «πελάτης» κ.τ.λ [12][43][47]

2.6. Model – View – Controller (MVC)

Το MVC είναι ένα μοντέλο αρχιτεκτονικής λογισμικού για δημιουργία διεπαφών χρήστη. Χωρίζει μια εφαρμογή σε 3 διασυνδεδεμένα κομμάτια έτσι ώστε να ξεχωρίσει τον τρόπο με τον οποίο ορίζονται τα δεδομένα εσωτερικά στην εφαρμογή σε σχέση με το πως παρουσιάζονται στο χρήστη.

Το σημαντικότερο μέρος του MVC, το **M**odel αναλαμβάνει να διαχειριστεί τη συμπεριφορά της εφαρμογής, τα δεδομένα, τη λογική και τους κανόνες. Ένα **V**iew μπορεί να είναι οποιαδήποτε μορφή παρουσίασης δεδομένων και πληροφοριών. Στο τρίτο μέρος, ο **C**ontroller δέχεται την είσοδο και τη μετατρέπει σε εντολές για το model και το view.

Η αρχιτεκτονική MVC χρησιμοποιείται ευρέως σε web εφαρμογές. Επίσης οι πλατφόρμες mobile εφαρμογών (όπως στην περίπτωση μας το Android) είναι έτσι σχεδιασμένες ώστε ο προγραμματισμός να βασίζεται στην αρχιτεκτονική MVC.[25][43]

.

3. Ανάλυση τεχνολογιών και βιβλιοθηκών

3.1. HTML

3.1.1. Περιγραφή

Κάθε σελίδα που εμφανίζεται στο Internet είναι ένα αρχείο γραμμένο με τη γλώσσα HTML (HyperText Markup Language, Γλώσσα Χαρακτηρισμού Υπερ-Κειμένου), που περιλαμβάνει το κείμενο της σελίδας, τη δομή της και τους συνδέσμους προς άλλα έγγραφα, εικόνες ή άλλα μέσα.

Ο φυλλομετρητής (Web browser) παίρνει τις πληροφορίες από τον web server, τις μορφοποιεί και τις εμφανίζει κατάλληλα για το σύστημά στο οποίο είναι εγκατεστημένος. Διαφορετικά προγράμματα φυλλο-μετρητή μπορεί να μορφοποιούν και να εμφανίζουν το ίδιο αρχείο με διαφορετικό τρόπο, ανάλογα με τις δυνατότητες του συστήματος στο οποίο τρέχουν και τις επιλογές διαμόρφωσης του προγράμματος του φυλλομετρητή. Μια web σελίδα ή ιστοσελίδα (web page) είναι ένα μεμονωμένο στοιχείο μιας παρουσίασης για το Web και περιέχεται σ' ένα αρχείο στον δίσκο, το οποίο ανακτάται από έναν web server και μορφοποιείται μέσω ενός φυλλομετρητή.
Η αρχική σελίδα (home page) είναι η πρώτη ή κορυφαία σελίδα μιας παρουσίασης για το Web, είναι δηλαδή το σημείο εισόδου ή εκκίνησης για τις υπόλοιπες σελίδες της παρουσίασης μας. Η αρχική σελίδα που περιέχει συνήθως μία σύνοψη του περιεχομένου της παρουσίασης με τη μορφή ενός πίνακα περιεχομένων ή μιας ομάδας εικονιδίων.

Η HTML είναι το ακρωνύμιο των λέξεων **H**yper**T**ext **M**arkup **L**anguage δηλαδή Γλώσσα Χαρακτηρισμού Υπερ-Κείμενου και βασίζεται στη γλώσσα SGML Standard Generalized Markup Language που είναι ένα πολύ μεγαλύτερο σύστημα επεξεργασίας εγγράφων. Η HTML ορίζει ένα σύνολο κοινών στυλ για τις web σελίδες όπως, τίτλοι (titles), επικεφαλίδες (headings), παράγραφοι (paragraphs), λίστες (lists) και πίνακες (tables). Ορίζει επίσης στυλ χαρακτήρων όπως, η έντονη γραφή (bold) και οι ενότητες κώδικα.

Κάθε στοιχείο έχει ένα όνομα και περιέχεται μέσα στα σύμβολα <> που αποκαλούνται tags (ετικέτες). Όταν γράφουμε μία Web σελίδα με την HTML στην ουσία δίνουμε τίτλους στα διάφορα στοιχεία της σελίδας με αυτά τα tags. Οι

φυλλομετρητές μαζί με τη δυνατότητά τους να ανακτούν σελίδες από το web λειτουργούν επίσης και σαν μορφοποιητές για την HTML. Όταν ένας φυλλομετρητής διαβάζει μία σελίδα γραμμένη με την HTML ανακτά τα tags της HTML και μορφοποιεί το κείμενο και τις εικόνες στην οθόνη. Διαφορετικοί φυλλομετρητές οι οποίοι τρέχουν σε διαφορετικούς υπολογιστές μπορεί να αντιστοιχίζουν διαφορετικά στυλ σε κάθε στοιχείο μιας σελίδας. Αυτό σημαίνει ότι οι σελίδες που δημιουργούνται με την HTML μπορεί να δείχνουν διαφορετικές από σύστημα σε σύστημα και από φυλλομετρητή σε φυλλομετρητή. Οι πραγματικές πληροφορίες και οι σύνδεσμοι που περιέχουν οι σελίδες θα είναι πάντα παρούσες, αλλά η εμφάνιση τους στην οθόνη θα είναι διαφορετική.[11][44]

3.1.2. Δομή μιας HTML σελίδας

Οι σελίδες που γράφουμε με την HTML είναι απλά αρχεία που σημαίνει ότι δεν περιέχουν πληροφορίες για κάποιο λειτουργικό σύστημα ή πρόγραμμα, αλλά μπορούν να διαβαστούν από οποιονδήποτε συντάκτη υποστηρίζει απλό κείμενο. Τα αρχεία της HTML περιέχουν το παρακάτω:

- το κείμενο της σελίδας.
- τα tags (ετικέτες) της HTML τα οποία υποδεικνύουν τα στοιχεία, τη δομή και τη μορφοποίηση των σελίδων, καθώς επίσης και τους συνδέσμους υπερκείμενου προς άλλες σελίδες ή προς αρχεία άλλων μορφών (π.χ. πολυμέσα)

Τα περισσότερα tags έχουν την εξής μορφή:

<ΌνομαTag> Κείμενο…</ ΌνομαTag >

Τα tags της HTML έχουν γενικά μία ετικέτα αρχής (ανοίγματος) και μία τέλους (κλεισίματος), τα οποία περικλείουν το κείμενο που επηρεάζουν. Το tag αρχής ενεργοποιεί μία λειτουργία ή ένα χαρακτηριστικό, όπως για παράδειγμα την έντονη γραφή, ενώ το tag τέλους την απενεργοποιεί. Τα tags τέλους έχουν το ίδιο όνομα με το tag αρχής αλλά με πρόθεμα το χαρακτήρα ‘/’.

Δεν αποτελούν ζευγάρι όλα τα tags της HTML καθώς ορισμένα είναι μονομελή, ενώ άλλα περιέχουν επιπλέον πληροφορίες και κείμενο ανάμεσα στα

σύμβολα < > . Επίσης όλα τα tags στην HTML δεν κάνουν διάκριση μεταξύ κεφαλαίων και πεζών γραμμάτων δηλαδή μπορούμε να γράφουμε είτε με κεφαλαία είτε με μικρά ακόμα και με οποιοδήποτε συνδυασμό τους.

Η HTML χρησιμοποιεί τρία tags για την περιγραφή της συνολικής δομής μιας σελίδας, τα οποία περιέχουν ορισμένες απλές πληροφορίες κεφαλίδας. Αυτά προσδιορίζουν τη σελίδα στους φυλλομετρητές και παρέχουν απλές πληροφορίες για αυτήν, όπως τον τίτλο ,τον συγγραφέα της, πριν από τη φόρτωση ολόκληρης της σελίδας.

Το tag < HTML >: η πρώτη ετικέτα που ελέγχει τη δομή μία σελίδες γραμμένης σε κώδικα HTML και υποδεικνύει ότι το περιεχόμενο του αρχείου περιέχει κώδικα μένω σε αυτή τη γλώσσα. Όλο το κείμενο και οι εντολές σε μία HTML σελίδα θα πρέπει να τοποθετούνται ανάμεσα σε tags αρχής και τέλους, ως εξής:

```
< HTML >
        ...Περιεχόμενα σελίδας...
< / HTML >
```

Το tag < HEAD> : προσδιορίζει ότι οι γραμμές που περιέχονται ανάμεσα στην αρχή και στο τέλος του είναι ο πρόλογος για το υπόλοιπο του αρχείου. Σε αυτή την ενότητα συνήθως τοποθετείται ο τίτλος της σελίδας καθώς και η εισαγωγή εξωτερικών βιβλιοθηκών και CSS αρχείων.

```
< HTML >
        < HEAD >
                < TITLE > Τίτλος < / TITLE>
        < / HEAD >
        .....
< / HTML >
```

Το tag < BODY>: το υπόλοιπο της HTML σελίδας, δηλαδή όλο το κείμενο και οποιοδήποτε άλλο περιεχόμενο όπως σύνδεσμοι, εικόνες κτλ περικλείεται σε μία τέτοια ετικέτα.

```
< HTML >
        < HEAD >
                < TITLE > Τίτλος < / TITLE>
        < / HEAD >
        <BODY>
                .....Περιεχόμενα...
        < /BODY>
< / HTML >
```

Όλες οι ετικέτες της HTML σχηματίζουν ξεχωριστές, ένθετες ενότητες κειμένου και θα πρέπει να δίνεται ιδιαίτερη προσοχή ώστε να μην υπάρχει επικάλυψη μεταξύ των tags.

Το tag < TITLE >: Κάθε σελίδα χρειάζεται ένα τίτλο, ο οποίος αποδεικνύει το περιεχόμενό της, και θα εμφανίζεται στη γραμμή τίτλου των δημοφιλέστερων. Οι ετικέτες < TITLE > τοποθετούνται πάντα μέσα στο ζευγάρι των < HEAD > όπως φαίνεται στο προηγούμενο τμήμα κώδικα HTML.

Τα tags επικεφαλίδων: Οι επικεφαλίδες χρησιμοποιούνται για το διαχωρισμό των ενοτήτων κειμένου όπως ακριβώς και σε ένα βιβλίο. Έχουν τη μορφή

<H1> τίτλος κειμένου </H1>

και ορίζονται με ετικέτες από H1 έως H6. Οι επικεφαλίδες δεν αριθμούνται όταν εμφανίζονται στην οθόνη, αλλά έχουν κάποιο χαρακτηριστικό που ξεχωρίζει από το υπόλοιπο κείμενο όπως μεγαλύτερο μέγεθος, εντονότερο κείμενο ή υπογράμμιση. Συνήθως το κείμενο του ελαττώνεται πηγαίνοντας από το H1 έως H6.

Τα tags παραγράφων: είναι τα <p> και η αλλαγή παραγράφου σημαίνει το ξεκίνημα μιας νέας γραμμής και επιπλέον κατακόρυφη απόσταση μεταξύ των παραγράφων. Το
 είναι κενό στοιχείο, δεν έχει περιεχόμενο, ούτε χρειάζεται ετικέτα τερματισμού.

Παρουσιάστηκαν εν συντομία κάποια βασικά στοιχεία της γλώσσας HTML. Επιπλέον αισθητικές παρεμβάσεις στη σελίδα μπορούν να επιτευχθούν χρησιμοποιώντας κάποια tags, όπως το <style> ή με την χρήση CSS (Cascade Style Sheets) αρχείων.[11][24][44]

3.2. CSS(Cascade Style Sheets)

3.2.1. Περιγραφή

Όπως αναφέρθηκε και σε προηγούμενη παράγραφο, η γλώσσα προγραμματισμού ιστοσελίδων και εφαρμογών HTML, χρησιμοποιείται για τη δημιουργία και την περιγραφή στοιχείων σε μία ιστοσελίδα. Όταν ο browser διαβάζει ένα κείμενο σε μία σελίδα λαμβάνει υπόψη του τα html tags, τα οποία χρησιμοποιούνται για να περιγράψουν αντικείμενα. Ότι βρίσκεται ανάμεσα σε tags, παίρνει τις αντίστοιχες ιδιότητες. Μέσω των επιλογέων (selectors) μπορούμε να

επικεντρωθούμε σε συγκεκριμένα στοιχεία στην HTML σελίδα και να τους διαμορφώσουμε στυλ.[11][55]

3.2.2. Σύνταξη

Η σύνταξη μιας CSS εντολής απεικονίζεται στο παρακάτω Σχήμα (**Εικόνα 2**):

Εικόνα 2 Σύνταξη εντολής CSS

Επειδή τα tags στην HTML είναι περιορισμένα, τα CSS δίνουν τη δυνατότητα να δηλωθούν νέα μοναδικά ή μη στοιχεία, και να διαμορφωθεί το στυλ τους εξαρχής. Αυτό επιτυγχάνεται με τη χρήση των attributes των tags της HTML, id και class. Τα ids και classes είναι διαφορετικά στοιχεία και επιτρέπεται να έχουν το ίδιο όνομα. Τα ids δηλώνουν μοναδικά στοιχεία και είναι ορθότερο να μη εμφανίζεται το ίδιο πάνω από μία φορά στη σελίδα. Αντίθετα δεν υπάρχει κάποιος περιορισμός για τις κλάσεις μέσα σε μία σελίδα. Το cascading αναφέρεται στο φαινόμενο του καταρράκτη και ουσιαστικά υποδεικνύει την κληρονομικότητα που έχουν τα στοιχεία των CSS. Με τα CSS μπορούν να οριστούν πολλά χαρακτηριστικά της εμφάνισης της ιστοσελίδας. Για παράδειγμα μπορεί να οριστεί το μέγεθος της γραμματοσειράς, το χρώμα της, χαρακτηριστικά τυπογραφίας να αλλαχθεί το background σε κάποιο element κ.α. [11][55]

3.2.3. Κληρονομικότητα

Όπως αναφέρθηκε προηγουμένως, στη CSS χρησιμοποιείται κληρονομικότητα μεταξύ των στυλ όπου ένα HTML element ‘‘παιδί’’ κάποιου άλλου HTML element κληρονομεί το στυλ του ‘‘πατέρα’’ element κτλ. Οι κανόνες που ορίζουν ποιο element κληρονομεί τι και με ποια σειρά υπερισχύει κάποιο στυλ σε σχέση με κάποιο άλλο αντίστοιχο είναι η ακόλουθη (από τα ισχυρότερα στα λιγότερα ισχυρά)[24][55]

Προτεραιότητα	*CSS τύπος*	*Περιγραφή*
1	User defined	Το CSS που ορίζει απευθείας ο χρήστης
2	Inline	Ένα στυλ ορισμένο σε ένα HTML element μέσω του “style” property
3	Media Type	Ο ορισμός ενός property σε όλα τα είδη media, εκτός αν ορίζεται κάποιο συγκεκριμένο CSS
4	Importance	Το ‘!important’ διαγράφει τα προηγούμενα είδη προτεραιότητας
5	Selector specificity	Ένας συγκεκριμένος selector υπερισχύει της γενικού ορισμού
6	Rule order	Ο πιο πρόσφατος κανόνας είναι αυτός που υπερισχύει
7	Parent inheritance	Αν κάποιο property δεν είναι ορισμένο, κληρονομείται από το element “πατέρας”
8	CSS property definition in HTML document	Ένας CSS κανόνας ή CSS inline style διαγράφει την προεπιλεγμένη τιμή του browser
9	Browser default	Η χαμηλότερη προτεραιότητα: Η αρχική τιμή του browser ορίζεται από τα specifications του W3C

Εικόνα 3 Πίνακας κανόνων κληρονομικότητας

3.3. JavaScript

3.3.1. Περιγραφή

Η JavaScript είναι μία δυναμική γλώσσα προγραμματισμού η οποία χρησιμοποιείται κυρίως στους web browsers. Η υλοποίηση της JavaScript σε ένα browser επιτρέπει την εκτέλεση client-side κώδικα (κώδικα δηλαδή που τρέχει στον υπολογιστή του χρήστη και όχι στον server) για αλληλεπίδραση με το χρήστη,

διαχείριση του browser, ασύγχρονη επικοινωνία με τον server και δυναμική αλλαγή της HTML που παρουσιάζεται στο χρήστη.

Πρόκειται για μία Dynamic typing γλώσσα όπου, δηλαδή οι μεταβλητές μπορούν να έχουν οποιοδήποτε τύπο (είτε strings, είτε αριθμοί) και να αλλάζουν δυναμικά χωρίς να δεις τα αυστηρά κάποιος συγκεκριμένος τύπος τους.

Τα τελευταία χρόνια η JavaScript χρησιμοποιείται και εκτός των ιστοσελίδων όπως σε αρχεία pdf, widgets ή ακόμα και σε server side εφαρμογές μέσω ανερχόμενων frameworks όπως το Node.Js και το Angular.Js.[11][24][48]

3.3.2. Σύνταξη

Οι μεταβλητές της Javascript ορίζονται χρησιμοποιώντας τη λέξη κλειδί ‘’var’’, για παράδειγμα:

var name; //Ορίζει μια μεταβλητή με το όνομα ‘‘name’’ χωρίς αρχική τιμή
var timi=5; // Ορίζει μία μεταβλητή με το όνομα ‘‘timi’’ με αρχική τιμή 5

Επίσης στη Javascript ο κώδικάς μπορεί να οργανωθεί σε functions, όπως παρακάτω :

```
Function Myfunction (var timi){
        timi = timi + 5;
        return timi;
}
```

3.3.3. Χρήση σε ιστοσελίδες

Η πιο συνηθισμένη χρήση της JavaScript είναι η χρήση της σε Ιστοσελίδες για εκτέλεση client-side κώδικα. Διάφορα JavaScripts περιλαμβάνονται στις ιστοσελίδες για να επηρεάζουν το DOM της σελίδας. Μερικά παραδείγματα είναι τα ακόλουθα:

- Φόρτωση νέων σελίδων ή μερών αυτών καθώς και υποβολή δεδομένων μέσω ασύγχρονων (AJAX) request χωρίς να απαιτείται η επαναφόρτωση της σελίδας.
- Κινήσεις στοιχείων της σελίδας
- Διαδραστικό περιεχόμενο

- Validation της εισόδου του χρήστη
- Αποστολή στοιχείων χρήσης του χρήστη πάνω στις διάφορες ιστοσελίδες

Το ρόλο της ερμηνείας του κώδικα JavaScript τον αναλαμβάνει ο εκάστοτε web browser. Η JavaScript μπορεί να τρέχει τοπικά στον υπολογιστή του χρήστη δίνοντας τη δυνατότητα στη σελίδα να ανταποκρίνεται γρήγορα στις πράξεις του χρήστη χωρίς να επιβαρύνει τον server. [24][26][48]

3.3.4. Χρήση εκτός ιστοσελίδων

Το τελευταίο χρονικό διάστημα η JavaScript αποκτά χρήση και εκτός ιστοσελίδων. Πολλές desktop εφαρμογές χρησιμοποιούν την εν λόγω γλώσσα για να τρέχουν διαφορά scripts για τις εσωτερικές λειτουργίες. Επίσης διάφορα developing frameworks κυκλοφορούν τα οποία βασίζονται στην JavaScript για τη δημιουργία εφαρμογών όπως το Node.Js και το Angular.Js. Στην παρούσα εργασία η JavaScript χρησιμοποιήθηκε μόνο εντός του web browser για χρήση στην ιστοσελίδα, οπότε δεν θα γίνει περαιτέρω αναφορά στη χρήση της εκτός των ιστοσελίδων.[24][48]

3.4. JQuery

3.4.1. Περιγραφή

Το jQuery είναι μια μικρή σε μέγεθος, απλή, γρήγορη και περιεκτική βιβλιοθήκη Javascript ανοιχτού κώδικα και πρωτοεμφανίστηκε το 2006. Συνδυάζοντας την απλότητα, την ευελιξία και την επεκτασιμότητα, άλλαξε τον τρόπο με τον οποίο οι προγραμματιστές γράφουν Javascript.

Αυτή η βιβλιοθήκη εντολών παρέχει δυνατότητες επιλογής και διαμόρφωσης HTML στοιχείων, διαμόρφωσης CSS στοιχείων, διεργασιών HTML γεγονότων, εφέ και animations, χρήσης AJAX και πληθώρα άλλων εφαρμογών.

Σε γενικές γραμμές το jQuery έχει αντικαταστήσει τον κώδικα παραδοσιακής Javascript. Οι περισσότεροι developers χρησιμοποιούν απευθείας αυτή τη βιβλιοθήκη καθώς προσφέρει πολλές επιπλέον δυνατότητες σε σχέση με την παραδοσιακή Javascript. Σα κυριότερα πλεονεκτήματα της είναι τα εξής:

- **Διαχωρισμός της Javascript και της HTML.** Αντί να γίνεται χρήση των HTML attributes για να καλούνται οι Javascript functions, το jQuery επιτρέπει

όλες τις functions για διαχείριση γεγονότων (event handling) να γίνονται πλήρως σε Javascript έτσι ώστε να μην χρειάζεται να εμπλέκονται με την HTML.

- **Συντομία και σαφήνεια.** Το jQuery παρέχει διάφορα είδη σύνταξης ώστε να είναι πιο σύντομο και πιο σαφές, πχ. chaining εφέ και πράξεις, συντόμευσης μεθόδων.
- **Εξαλείφει τις ασυμβατότητες μεταξύ των browsers.** Οι διάφοροι μηχανισμοί Javascript στους browsers μπορεί να έχουν διαφορές και Javascript κώδικα που λειτουργεί σε έναν browser να μη λειτουργεί σε κάποιον άλλο. Σο jQuery διαχειρίζεται όλες αυτές τις ασυμβατότητες και παρέχει ένα συνεπές interface που λειτουργεί σε όλους τους browsers που υποστηρίζει.
- **Επεκτάσιμο.** Το jQuery δίνει τη δυνατότητα επέκτασης του framework. Νέα γεγονότα, elements και μέθοδοι μπορούν να προστεθούν και να χρησιμοποιηθούν σε ένα plugin.[24][26][56]

3.4.2. Χαρακτηριστικά

Το jQuery περιλαμβάνει τα εξής χαρακτηριστικά:

- Επιλογή DOM elements με χρήση της open source μηχανής Sizzle
- Μετατροπές στο DOM
- Διαχείριση του DOM βασισμένο σε CSS selectors που χρησιμοποιεί το όνομα του element , τα attributes id και class ως κριτήρια επιλογής
- Events (γεγονότα)
- Εφέ και animations
- AJAX (asynchronous requests)
- JSON parsing
- Επεκτασιμότητα μέσω plugins
- Utilities (όπως πληροφορίες του user-agent και εντοπισμός features)
- Μέθοδοι συμβατότητας οι οποίες είναι παρούσες στους νέους browsers αλλά χρειάζονται fall-backs σε παλαιότερους
- Υποστήριξη πολλαπλών browsers

3.5. SQL (Structured Query Language)

3.5.1. Περιγραφή

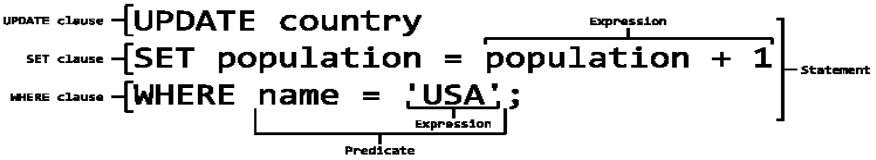
Η SQL (Structural Query Language) είναι μία γλώσσα υπολογιστών, που σχεδιάστηκε για τη διαχείριση δεδομένων, σε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (Relational Database Management System, RDBMS) η οποία αρχικά βασίστηκε στη σχεσιακή άλγεβρα. Η γλώσσα περιλαμβάνει δυνατότητες ανάκτησης και ενημέρωσης δεδομένων, δημιουργίας και τροποποίησης σχημάτων και σχεσιακών πινάκων, αλλά και ελέγχου πρόσβασης στα δεδομένα.

Η SQL αναπτύχθηκε στην IBM στην αρχή της δεκαετίας του 1970. Αυτή η πρώτη έκδοση που ονομάζεται SEQUEL είχε ως σκοπό να χειριστεί και ανακτήσει στοιχεία που αποθηκεύτηκαν στο πρώτο RDBMS της IBM το system R. Την ίδια περίοδο αναπτύχθηκε το πρώτο σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων RDBMS στο MIT και η INGRES στο πανεπιστήμιο Berkeley. Στα τέλη της δεκαετίας η Relational Software εισήγαγε την πρώτη διαθέσιμη εμπορικά εφαρμογή του SQL, λίγες εβδομάδες νωρίτερα από την IBM.[24][25][46]

3.5.2. Γλωσσικά στοιχεία

Η γλώσσα SQL υποδιαιρείται σε διάφορα γλωσσικά στοιχεία που περιλαμβάνουν:

- **Clauses,** προαιρετικές σε ορισμένες περιπτώσεις, άλλα απαραίτητα συστατικά όλων των δηλώσεων και ερωτήσεων.
- **Expressions**, που παράγουν είτε κλιμακωτές τιμές, είτε πίνακες αποτελούμενος από στήλες και σειρές δεδομένων
- **Predicates**, που διευκρινίζουν τους όρους που μπορούν να αξιολογηθούν σαν σωστό ή λάθος
- **Queries**, που ανακτούν τα στοιχεία βασιζόμενες σε ειδικά κριτήρια.
- **Statements**, που μπορούν να επιδρούν στα σχήματα και τα στοιχεία, ή να ελέγξουν τη ροή του προγράμματος και τις συνδέσεις με άλλα προγράμματα.



Εικόνα 4 Γλωσσικά στοιχεία σε ένα statement

Οι εκφράσεις σε SQL περιλαμβάνουν και το χαρακτήρα τερματισμού «;». Το κενό γενικά αγνοείτε στα statements και τα queries, αλλά είναι απαραίτητο για να ξεχωρίζει τα statements όπως και στην κανονική γραφή κειμένου.

3.5.3. Operators

Operator	*Περιγραφή*
=	Ισούται με
<>	Δεν ισούται με (τα περισσότερα DBMS επίσης δέχονται το != αντί για το <>)
>	Μεγαλύτερο από
<	Μικρότερο από
>=	Μεγαλύτερο ή ίσο
<=	Μικρότερο ή ίσο
BETWEEN	Μεταξύ ενός καθορισμένου εύρους
LIKE	Ταιριάζει με ένα πρότυπο χαρακτήρα
IN	Ίσο με μια από τις πολλαπλές δυνατές τιμές
IS ή IS NOT	Σύγκριση με το null (κενό δεδομένων)
IS NOT DISTINCT FROM	Ίσο με τιμή ή και τα 2 είναι null (κενό δεδομένων)
AS	Χρησιμοποιείται για αλλαγή του ονόματος πεδίου κατά την εμφάνιση

Εικόνα 5 Βασικοί Operators

Κάποιες από τις σημαντικότερες SQL εντολές είναι:

- **SELECT** – επιλογή δεδομένων από τη βάση
- **UPDATE** – ενημέρωση δεδομένων στη βάση
- **DELETE** – διαγραφή δεδομένων
- **INSERT INTO** – εισαγωγή νέων δεδομένων
- **CREATE DATABASE** – δημιουργία νέας βάση
- **ALTER DATABASE** – τροποποίηση της βάσης
- **CREATE TABLE** – δημιουργία νέου πίνακα
- **ALTER TABLE** – τροποποίηση ενός πίνακα
- **DROP TABLE** – διαγραφή ενός πίνακα
- **CREATE COLUMN** – δημιουργία πεδίου πίνακα
- **ALTER COLUMN** – τροποποίηση πεδίου πίνακα
- **DROP COLUMN** – διαγραφή πεδίου πίνακα
- **CREATE INDEX** – δημιουργία δείκτη (κλειδί αναζήτησης)
- **DROP INDEX** – διαγραφή ενός δείκτη

Για τον καλύτερο σχεδιασμό και βελτιστοποίηση μιας βάσης δεδομένων συχνά γίνεται χρήση και των ακόλουθων εντολών:

- FROM - υποδεικνύει από ποιο πίνακα θα τραβήξουμε τα δεδομένα που μας ενδιαφέρουν
- JOIN - ορίζει τους κανόνες για την ένωση πινάκων
- WHERE - θέτει όρους σύγκρισης στα δεδομένα που επιστρέφονται από τη βάση, θέτει φίλτρα αληθείας.
- GROUP BY - ομαδοποιεί τα δεδομένα σύμφωνα με ορισμένα κριτήρια
- ORDER BY - ταξινομείται δεδομένα σύμφωνα με ορισμένα κριτήρια

Άλλες εντολές για τη μεταχείριση των δεδομένων, διαμόρφωση των πινάκων και τις συναλλαγές μεταξύ τους είναι INSERT, UPDATE, DELETE, MERGE κτλ.[24]

3.5.4. Τύποι Δεδομένων

Κάθε στήλη πίνακα SQL δηλώνει τι τύπο δεδομένων περιέχει. Η ANSI SQL περιλαμβάνει τους παρακάτω τύπους δεδομένων:

- String χαρακτήρων – χαρακτήρες, κενά, μέγιστου μήκους n
- Bit
- Αριθμούς-Ακέραιους, πραγματικούς, κινητής υποδιαστολής, διπλής υποδιαστολή, κ.τ.λ
- DATE : τιμές για ημερομηνία (2017/05/20)
- TIME: τιμές για χρόνο(15:30:22)
- TIME WITH TIMEZONE : ίδιο με το TIME αλλά περιέχει πληροφορίες για τη ζώνη ώρας
- TIMESTAMP:DATE και TIME σε μία μεταβλητή(2017/05/20 15:30:22)

Η SQL παρέχει διάφορες λειτουργίες για τη δημιουργία μιας μεταβλητής ημερομηνίας και ώρας καθώς και για το διαχωρισμό των επιμέρους στοιχείων της όπως λεπτά, δευτερόλεπτα κτλ.[24][46]

3.6. PHP

3.6.1. Περιγραφή

Η PHP είναι μία server side scripting γλώσσα προγραμματισμού με κύρια χρήση την δημιουργία web εφαρμογών.

Στον προγραμματισμό με PHP δίνεται η δυνατότητα ανάμειξης κώδικα PHP με κλασική HTML καθώς και διάφορα άλλα frameworks. Ο PHP κώδικας επεξεργάζεται από έναν διερμηνέα (interpreter) ο οποίος συνήθως υλοποιείται από κάποιο module ενός web server ή από κάποιο CGI εκτελέσιμο (Common Gateway Interface). Αφού ο κώδικας ερμηνευτεί και εκτελεστεί, ο web server στέλνει το αποτέλεσμα στον client (συνήθως με τη μορφή μιας ιστοσελίδας, εικόνας, ή άλλων δεδομένων). Στην περίπτωσή του προτεινόμενου συστήματος ως PHP interpreter χρησιμοποιήθηκε το μοντέλο του Apache HTTP, η διαχείριση του οποίου γίνεται από το panel του XAMPP.[24][25][29][43]

3.6.2. Σύνταξη

Παρακάτω απεικονίζεται το παράδειγμα ενός απλού ΄Hello World΄ προγράμματος σε PHP.

```
<!DOCTYPE html>
<html>
        <head>
                <title> Example PHP</title>
        </head>
        <body>
                <?php echo ΄Hello World΄;  ?>
        </body>
<html>
```

Ο κώδικας PHP περιλαμβάνεται μέσα στην κλασική HTML με την χρήση ειδικών tags. Περικλείεται μεταξύ του αρχικού ΄΄<?php΄΄ και του τελικού ΄΄?>΄΄. Η εντολή ΄΄echo΄΄ χρησιμοποιείται για να τυπώσει κάποιο string ή κάποια μεταβλητή στην έξοδο.

Στη συνέχεια παρουσιάζεται ένα παράδειγμα μίας PHP function. Οι συναρτήσεις στην PHP όπως και σε όλες τις γλώσσες προγραμματισμού χρησιμοποιούνται για την οργάνωση κοινών κομματιών κώδικα τα οποία χρησιμοποιούνται σε πολλαπλά σημεία στην ίδια εφαρμογή. Με τη χρήση των συναρτήσεων είναι εφικτή η οργάνωση του κώδικα, η μείωση του μεγέθους του και η ευκολότερη συντήρηση και βελτίωσή του. Η PHP περιλαμβάνει πληθώρα στοιχείων και βιβλιοθηκών για τις πιο συνήθεις λειτουργίες. Φυσικά ο κάθε προγραμματιστής έχει τη δυνατότητα να φτιάξει τις δικές συναρτήσεις ανάλογα με τις απαιτήσεις του.

```
public static function IsEmptyOrNull($question){
        return (!isset($question) | | trim($question)=== ΄ ΄);
}
```

Η παραπάνω συνάρτηση για παράδειγμα χρησιμοποιείται για τον έλεγχο ενός string όσον αφορά το αν είναι κενό ή όχι.

Η PHP μετά την έκδοση 4 μπορεί να χρησιμοποιηθεί και για Object Oriented προγραμματισμό. Παρακάτω ακολουθεί ένα παράδειγμα μίας PHP Class η οποία περιλαμβάνει κάποιες functions για διάβασμα παραμέτρων από το εκάστοτε HTTP Request.

```
class Praxeis {
        function prosthesh($x,$y){
                $z = $x +$y;
                return $z;
        }
                function pollaplasiamos($x,$y){
                $z = $x * $y;
                return $z;
        }
}
```

3.6.3. License

Η PHP είναι δωρεάν λογισμικό και διανέμεται με την PHP License η οποία ορίζει ότι τα "προϊόντα" που προέρχονται από αυτό το λογισμικό και δεν μπορούν να ονομάζονται "PHP", ούτε το "PHP" μπορεί να εμφανίζεται στο όνομά τους, χωρίς προηγούμενη γραπτή άδεια από group@php.net.

3.7. Slim

3.7.1. Περιγραφή

Εικόνα 6 Λογότυπο Slim Framework

Το Slim είναι ένα micro PHP framework. Πρόκειται για ένα εργαλείο που βοηθά στην σύνταξη απλού και γρήγορου κώδικα για εφαρμογές διαδικτύου αλλά και APIs. Το Slim περιλαμβάνει έναν αποστολέα ο οποίος λαμβάνει αιτήματα HTTP τα επεξεργάζεται και επιστρέφει μια HTTP απόκριση.[41]

3.7.2. Πως λειτουργεί

Κατ' αρχάς θα πρέπει να έχει στηθεί ένας Web Server όπως ο Apache. Θα πρέπει να ρυθμιστεί ο Web Server, έτσι ώστε να στέλνει όλα τα κατάλληλα αιτήματα σε ένα "front controller" PHP αρχείο.

Το Slim περιέχει κάποιες διαδρομές που ανταποκρίνονται σε συγκεκριμένα αιτήματα HTTP. Κάθε διαδρομή επικαλείται επανάκληση και επιστρέφει μια HTTP απόκριση. Για να γίνει αυτό θα πρέπει να διαμορφωθεί κατάλληλα το Slim framework. Στην συνέχει να οριστούν οι διαδρομές της εφαρμογής και τέλος μπορεί να εκτελεστεί η εφαρμογή.[41]

Παράδειγμα εφαρμογής

```php
<?php
// Create and configure Slim app
$config = ['settings' => [
    'addContentLengthHeader' => false,
]];
$app = new \Slim\App($config);

// Define app routes
$app->get('/hello/{name}', function ($request, $response, $args) {
    return $response->write("Hello " . $args['name']);
});

// Run app
$app->run();
```

Εικόνα 7 Παράδειγμα χρήσης του framework Slim

3.7.3. License

Η χρήση του Slim framework είναι ελεύθερα διαθέσιμη υπό απλή νομική άποψη, λόγω του CC (Creative Commons), ένα μη κερδοσκοπικό οργανισμό.

3.8. MySQL

3.8.1. Περιγραφή

Η MysSQL είναι το δεύτερο πιο διαδεδομένο σύστημα (RDBMS) βάσεων δεδομένων παγκόσμιος. Ακολουθεί την SQlite η οποία είναι εγκατεστημένη σε κάθε κινητό τηλέφωνο το οποίο τρέχει iOS ή Android κερδίζοντας έτσι τεράστια αριθμό χρηστών σε σχέση με την MySQL και τις υπόλοιπες βάσεις δεδομένων. Πρόκειται για πολύ δημοφιλή επιλογή για χρήση σε web εφαρμογές καθώς προσφέρει πάρα πολλές δυνατότητες, είναι ταχύτατη και δωρεάν. Περιέχεται στο πακέτο ΄΄LAMP΄΄ δωρεάν εφαρμογών. Πολλές γνωστές και ευρέως χρησιμοποιούμενες εφαρμογές είναι βασισμένες στη MySQL, όπως για παράδειγμα το Joomla, το Wordpress, το Drupal, το phBB καθώς και πολλά μεγάλα web sites όπως το Google, Facebook, Twitter, Youtube κ.α.

Τα αρχικά SQL που περιλαμβάνονται στο όνομα της MySQL αντιστοιχούν στη γλώσσα για βάσεις δεδομένων ΄΄Structured Querry Language΄΄. Κατά την εγκατάσταση της δεν περιλαμβάνεται κάποιο γραφικό περιβάλλον, υπάρχουν όμως αρκετά 3rd Party όπως το MySQL Workbench και το PHPMyAdmin τα οποία χρησιμοποιήθηκαν για το σχεδιασμό και τη διαχείριση της βάσης δεδομένων της προτεινόμενης πλατφόρμας.

Ένα πολύ σημαντικό πλεονέκτημα της MySQL είναι ότι μπορεί να τρέξει στα περισσότερα περιβάλλοντα εργασίας όπως Windows, Linux, Mac Os, Solaris κ.α δίνοντας τη δυνατότητα στον προγραμματιστή να την χρησιμοποιήσει σε διαφορετικές πλατφόρμες χωρίς να χρειάζεται να γράψει επιπλέον κώδικα. Για την εργασία αυτή η MySQL χρησιμοποιήθηκε σε Windows και σε Linux άκρως επιτυχημένα.[24][26][46]

3.8.2. License

Η χρήση της MySQL είναι δωρεάν κάτω από την "GNU General Public License v2".Ο προγραμματιστής έχει το δικαίωμα να χρησιμοποιήσει, να μελετήσει να αντιγράψει (διανείμει) και να επεξεργαστεί αυτό το software.

3.8.3. Χρήση με PHP

Στην τεκμηρίωση (documentation) της MySQL ο όρος "connector" αναφέρεται στο κομμάτι λογισμικού το οποίο επιτρέπει σε μία εφαρμογή να συνδεθεί με μία βάση δεδομένων MySQL. Η MySQL παρέχει connectors για πολλές γλώσσες προγραμματισμού συμπεριλαμβανομένης της PHP.

Αν μία εφαρμογή χρειάζεται να επικοινωνεί με μία βάση δεδομένων προγραμματιστής θα πρέπει να γράψει κώδικα για να καλύψει τις περιπτώσεις σύνδεσης στη βάση, καθώς και εκτέλεσης ερωτημάτων και functions σε αυτή. Συνήθως αυτό το λογισμικό βρίσκεται σε βιβλιοθήκες και παρέχεται στους προγραμματιστές και είναι γενικά γνωστό ως connector καθώς επιτρέπει τη σύνδεση με τη βάση δεδομένων.

Ο driver είναι ένα κομμάτι λογισμικού σχεδιασμένο για να επικοινωνεί με κάποιο συγκεκριμένο τύπο βάσης δεδομένων. Ο driver μπορεί επίσης να καλεί

βιβλιοθήκες, όπως στην MySQL Client Library ή το MySQL Native Driver. Αυτές οι βιβλιοθήκες υλοποιούν ένα low level πρωτόκολλο το οποίο χρησιμοποιείται για την επικοινωνία με την MySQL .

Πολλές φορές οι όροι connector και driver χρησιμοποιούνται αντίστροφα το οποίο μπερδεύει. Στην περίπτωση της MySQL, driver θεωρείται το λογισμικό που παρέχει το connector που είναι συγκεκριμένο για αυτή τη βάση. Η PHP παρέχει διαφορά APIs για την επικοινωνία με την MySQL τα κυριότερα είναι:

- PHP's MySQL Extension
- PHP's mysqli Extension
- PHP Data Objects (PDO)

Το κάθε APIs έχει τα πλεονεκτήματα και τα μειονεκτήματά του. Τα κύρια χαρακτηριστικά του καθενός είναι τα ακόλουθα.

- PHP's MySQL Extension

Πρόκειται για το αρχικό extension της PHP σχεδιασμένο για εφαρμογές που επικοινωνούν με μία MySQL βάση δεδομένων. Αυτό το extension είναι σχεδιασμένο για να λειτουργεί με εκδόσεις της MySQL ως την 4.1.3. Δεν παρέχει όλα τα χαρακτηριστικά των νέων εκδόσεων της MySQL.

- PHP's mysqli Extension

Το extension ΄΄mysqli΄΄ είναι βελτιωμένη έκδοση του MySQL extention (MySQL *improved* extension) και σχεδιάστηκε για να εκμεταλλεύεται τα νέα χαρακτηριστικά των βάσεων δεδομένων MySQL από την έκδοση 4.13 και μετά. Το extension αυτό είναι ενσωματωμένο από την PHP 5 και μετά. Οι κυριότερες βελτιώσεις του *improved* extension από το αρχικό είναι οι ακόλουθες:

- Object-oriented interface
- Υποστήριξη για Prepared Statements
- Υποστήριξη για Multiple Statements
- Υποστήριξη για Transactions
- Βελτιωμένες δυνατότητες debugging

Αν η βάση MySQL είναι η έκδοση 4.1.3 ή μεταγενέστερη, συνίσταται η χρήση αυτού του extension.

- PHP Data Objects (PDO)

Το “PHP Data Objects” ή “PDO” είναι ένα “*database abstraction layer*” για εφαρμογές PHP. Το PDO παρέχει ένα συνεπές API άσχετα με τον τύπο βάσης δεδομένων που θα συνδεθεί η εφαρμογή. Θεωρητικά με τη χρήση του PDO θα μπορούσε ο προγραμματιστής να αλλάξει τύπο βάσης δεδομένων αλλάζοντας ελάχιστα τον κώδικα του.

Παρόλο που το PDO έχει τα πλεονεκτήματα ενός καθαρού, απλό και φορητού API, έχει το μειονέκτημα ότι δεν επιτρέπει τη χρήση των προηγμένων χαρακτηριστικών που είναι διαθέσιμα στις τελευταίες εκδόσεις της MySQL.
Στην παρούσα εργασία χρησιμοποιήθηκε το mysqli για την επικοινωνία με τη βάση μας .[24][25][26][43][46]

3.9. Java

3.9.1. Περιγραφή

Η Java είναι μία αντικειμενοστραφής (object oriented) γλώσσα προγραμματισμού βασισμένη σε Classes. Έχει σχεδιαστεί ώστε να έχει όσο το δυνατόν λιγότερες εξαρτήσεις. Ο κώδικας δεν τρέχει απευθείας όπως συνηθίζεται σε γλώσσες όπως C, αλλά μετατρέπεται σε ”Bytecode” και τρέχει πάνω σε ένα Java Virtual Machine (JVM) ανεξάρτητα από την αρχιτεκτονική του υπολογιστή. Δίνει έτσι τη δυνατότητα στους προγραμματιστές να γράψουν τον κώδικα τους μία φορά και να μπορούν να τρέξουν παντού (“Write once, Run anywhere”). Η Java είναι μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού με πάνω από 9 εκατομμύρια προγραμματιστές.

Ένα μεγάλο πλεονέκτημα της μετατροπής σε Bytecode είναι η φορητότητα. Λόγου όμως του ότι ο κώδικας Java δεν μετατρέπεται σε κώδικα μηχανής που είναι εξαρτημένος από την εκάστοτε αρχιτεκτονική υπολογιστή παρουσιάζει και σημαντικά μειονεκτήματα Τα προγράμματα που είναι γραμμένα σε αυτήν είναι πιο αργά από αντίστοιχα που γίνονται compile σε γλώσσα μηχανής καθώς επίσης καταναλώνουν και περισσότερη μνήμη. Έχουν όμως δημιουργηθεί compilers οι οποίοι μετατρέπουν

το bytecode σε κώδικα μηχανής κατά τη διάρκεια της εκτέλεσης (Just-In-Time – JST compilers) επιτυγχάνοντας την βέλτιστη εκτέλεση του κώδικα.

Ένα άλλο πλεονέκτημα είναι η αυτόματη διαχείριση μνήμης. Η ίδια Java αναλαμβάνει να διαχειριστεί τον κύκλου ζωής των αντικειμένων. Ο προγραμματιστής δημιουργεί ένα αντικείμενο και το Java Runtime αναλαμβάνει να ελευθερώσει την μνήμη του όταν αυτό δεν χρησιμοποιείται πια. Όταν δεν υπάρχουν άλλες αναφορές ενός αντικειμένου, η μνήμη που καταναλώνει μπορεί να ελευθερωθεί αυτόματα από τον garbage collector. Με αυτό το μηχανισμό ο προγραμματιστής απαλλάσσεται από τη διαδικασία του να διαθέσει μνήμη για ένα νέο αντικείμενο καθώς και να την ελευθερώσει όταν δεν το χρειάζεται πια. Μειώνονται έτσι οι περιπτώσεις των memory-leaks.

Η Java χρησιμοποιείται σε πάρα πολλά συστήματα για πάρα πολλές εφαρμογές. Μπορεί να χρησιμοποιηθεί για command line executed εφαρμογές, σε standalone εφαρμογές με διεπαφή χρήστη, σε πληθώρα web εφαρμογών καθώς και σε πολλά embedded συστήματα. Τα τελευταία χρόνια η Java χρησιμοποιείται και σε κινητά τηλέφωνα κυρίως οι συσκευές που τρέχουν Android καθώς αυτό χρησιμοποιεί τη Java για όλες του τις λειτουργίες. Στην παρούσα εργασία η γλώσσα Java χρησιμοποιείται από το Android SDK για τη δημιουργία της native android εφαρμογής[12][13][47]

3.9.2. Σύνταξη

Η σύνταξη της Java έχει προέλθει κυρίως από τη C++ και είναι σχεδιασμένη εξ ολοκλήρου ως object oriented γλώσσα. Όλος ο κώδικας γράφεται μέσα σε Classes και όλα είναι ένα αντικείμενο με εξαίρεση κάποιους primitive τύπους δεδομένων (πχ. integers, booleans, strings κτλ). Αντίθετα με την C++, η Java δεν επιτρέπει το overloading operators ή την πολλαπλή κληρονομικότητα. Μία κλάση της μπορεί να κληρονομεί από μόνο μία κλάση.

Ένα πρόγραμμα Hello World γραμμένο σε Java έχει την ακόλουθη μορφή:

```
Class HelloWorldProgram{
        Public static void main(String[] args){
                System.out.println(“Hello World!’’);
        }

}
```

Στο παράδειγμα αυτό φαίνεται τη δήλωση της κλάσης “HelloWorldProgram” η οποία περιλαμβάνει την πιο συνηθισμένη μέθοδο σε μία Java class, τη ‘main’ με όρισμα έναν πίνακα από Strings και τύπο void (δεν επιστρέφει κάποιο δεδομένο δηλαδή).

3.10. Apache HTTP Server

3.10.1. Περιγραφή

Ο **Apache HTTP Server** (apache) είναι ένα project με σκοπό τη δημιουργία ενός ισχυρού HTTP (Web) Server, επαγγελματικού επιπέδου, με πολλά χαρακτηριστικά αλλά στηριγμένου σε ελεύθερο κώδικα. Το project αυτό διαχειρίζεται από εθελοντές παγκοσμίως. Από το 1996 έως και σήμερα ο apache είναι ο κυρίαρχος web server παγκοσμίως. Χρησιμοποιείται σε πάρα πολλά λειτουργικά συστήματα όπως Windows, Linux, Mac Os , Solaris κ.α., αλλά συνηθέστερα συναντάται σε Unix based λειτουργικά συστήματα. Στην εργασία αυτή τρέχει μέσω του XAMPP.[26]

3.10.2. License

Ο apache παρέχεται με license Apache 2.0 License

3.11. XAMPP

3.11.1. Περιγραφή

Το **XAMPP** είναι το πιο δημοφιλές περιβάλλον προγραμματισμού για PHP. Πρόκειται για ένα Open Source πακέτο λογισμικού το οποίο έχει σχεδιαστεί και προετοιμαστεί έτσι ώστε να είναι πάρα πολύ εύκολο κατά την εγκατάσταση και τη χρήση. Είναι cross-platform και μπορεί να λειτουργήσει σε Linux, Microsoft Windows, Solaris και Mac Os. Τα αρχικά XAMPP είναι ακρωνύμιο των αρχικών γραμμάτων των βασικών τεχνολογιών που περιλαμβάνει και είναι τα εξής:

- **X** (από την λέξη “cross” με την έννοια του cross-platform)
- **A**pache HTTP Server
- **M**ySQL

- **P**HP
- **P**erl

3.11.2. Απαιτήσεις και χαρακτηριστικά

Το XAMPP απαιτεί ένα αρχείο Zip ή exe να εκτελεστεί. Καθόλου ή ελάχιστες ρυθμίσεις χρειάζονται για τη λειτουργία του. Είναι αυτόνομο και πολλαπλές παρουσίες του μπορούν να υπάρξουν ταυτόχρονα στον ίδιο υπολογιστή. Η κάθε εγκατάσταση μπορεί να αντιγραφεί εύκολα από έναν υπολογιστή σε άλλον.

3.11.3. Χρήση

Αρχικά το XAMPP είχε σκοπό τη χρήση μόνο ως εργαλείο προγραμματισμού, δίνοντας τη δυνατότητα στους προγραμματιστές να χρησιμοποιούν το PC τους για να δοκιμάσουν τις εφαρμογές τους χωρίς να είναι απαραίτητο να είναι συνδεδεμένοι στο internet. Για αυτό το λόγο έχει αρκετά χαρακτηριστικά των τεχνολογιών που περιλαμβάνει απενεργοποιημένα. Στην πραγματικότητα όμως χρησιμοποιείται ως server για πραγματικές ιστοσελίδες στο internet.

Το XAMPP περιλαμβάνει εργαλεία διαχείρισης βάσεων δεδομένων, όπως PHPMyAdmin. Επίσης περιλαμβάνονται εφαρμογές όπως FTP server και άλλα εργαλεία για την εγκατάσταση CMS εφαρμογών κ.α.[26][45]

3.11.4. License

Το XAMPP παρέχεται με license GNU GPL.

3.12. phpMyAdmin

3.12.1. Περιγραφή

Το phpMyAdmin είναι ένα δωρεάν και open source εργαλείο για τη διαχείριση βάσεων δεδομένων MySQL μέσω ενός Web browser. Είναι γραμμένο σε

PHP και MySQL είναι cross-platform, μπορεί να λειτουργήσει σε όλα τα συστήματα στα οποία μπορεί να τρέξει PHP και MySQL. Δίνει τη δυνατότητα δημιουργίας, επεξεργασίας και διαγραφής βάσεων δεδομένων, πινάκων, πεδίων και σειρών καθώς και εκτέλεσης απευθείας ερωτημάτων SQL. [45]

3.12.2. Χαρακτηριστικά

Τα κυριότερα χαρακτηριστικά που περιλαμβάνει η εφαρμογή είναι τα εξής:

- Παρέχεται ένα web interface
- Εγγραφή καινούριου χρήστη στην MySQL
- Επεξεργασία της βάσης μέσω αλλαγής κωδικού του χρήστη
- Εισαγωγή δεδομένων προς αναζήτηση
- Εξαγωγή δεδομένων σε μορφή πίνακα
- Πληρωμή με το ψηφιακό νόμισμα

3.12.3. License

Το phpMyAdmin παρέχεται με License GNU General Public License 2.

3.13. JSON

3.13.1. Περιγραφή

Το JSON είναι μία open standard μορφή που χρησιμοποιεί κείμενο το οποίο είναι δυνατόν να διαβάζεται από άνθρωπο για τη μεταφορά αντικειμένων δεδομένων τα οποία αποτελούνται από ζευγάρια “χαρακτηριστικών τιμών”. Χρησιμοποιείται κυρίως για τη μεταφορά δεδομένων μεταξύ των servers και των web εφαρμογών ως εναλλακτικό του XML. Στην παρούσα εργασία η μορφή JSON χρησιμοποιείται σε όλες τις επικοινωνίες της Android εφαρμογής με το web server.[57][70]

3.13.2. Τύποι δεδομένων και σύνταξη

Οι βασικοί τύποι δεδομένων είναι οι εξής:

- Αριθμοί
- String. Μία σειρά από κανένα ή από περισσότερους χαρακτήρες unicode.
- Boolean τιμές true ή false.
- Πίνακες. Ταξινομημένες λίστες από καμία οι περισσότερες τιμές οποιαδήποτε τύπου. Οι πίνακες περικλείονται ανάμεσα σε "square brackets" ("**[**" και "**]**") και τα στοιχεία διαχωρίζονται με κόμμα.
- Αντικείμενα. Μη ταξινομημένοι σχετιζόμενοι πίνακες (associative array) σε ζευγάρια ονόματος τιμής. Τα αντικείμενα χωρίζονται με "curly brackets" ("**{**" και "**}**") και χρησιμοποιούν κόμμα για να χωρίσουν το κάθε ζευγάρι. Το κάθε ζευγάρι ονόματος τιμής χωρίζεται με άνω-κάτω τέλεια (":") όπου αριστερά είναι το όνομα και δεξιά η τιμή. Όλα τα κλειδιά/ονόματα είναι strings και πρέπει να είναι μοναδικά σε ένα αντικείμενο.
- Null. Μία κενή τιμή με χρήση της λέξης null

Ένα παράδειγμα σύνταξης ενός αντικειμένου χρήστη σε μορφή JSON είναι το ακόλουθο:

```
{
        "name" : "Alexandros",
        "surname" : "Loukidis",
        "username" : "Loukis",
        "email" : "aloukis@hotmail.com",
        "phoneNumbers" : [
                {
                        "type" : "home",
                        "number" : '210-1112345"
                        },
                        {
                        "type" : "cellphone",
                        "number" : "691112345"
                        }
        ]
}
```

3.14. DataTables (jQuery plugin)

3.14.1. Περιγραφή

Show 10 entries Search:

Name	Position	Office	Age	Start date	Salary
Airi Satou	Accountant	Tokyo	33	2008/11/28	$162,700
Angelica Ramos	Chief Executive Officer (CEO)	London	47	2009/10/09	$1,200,000
Ashton Cox	Junior Technical Author	San Francisco	66	2009/01/12	$86,000
Bradley Greer	Software Engineer	London	41	2012/10/13	$132,000
Brenden Wagner	Software Engineer	San Francisco	28	2011/06/07	$206,850
Brielle Williamson	Integration Specialist	New York	61	2012/12/02	$372,000
Bruno Nash	Software Engineer	London	38	2011/05/03	$163,500
Caesar Vance	Pre-Sales Support	New York	21	2011/12/12	$106,450
Cara Stevens	Sales Assistant	New York	46	2011/12/06	$145,600
Cedric Kelly	Senior Javascript Developer	Edinburgh	22	2012/03/29	$433,060
Name	Position	Office	Age	Start date	Salary

Showing 1 to 10 of 57 entries Previous 1 2 3 4 5 6 Next

Εικόνα 8 plugin Datatable

Το DataTables είναι ένα plugin για την jQuery. Πρόκειται για ένα πολύ ευέλικτο εργαλείο το οποίο δίνει προχωρημένες λειτουργίες και ελέγχους σε πίνακες HTML. Στην παρούσα εργασία το plugin αυτό χρησιμοποιείται στους HTML πίνακες όπου απαιτείται σελιδοποίηση αναζήτηση και ταξινόμηση.

3.14.2. Χαρακτηριστικά

Τα πιο σημαντικά χαρακτηριστικά αυτού του plugin είναι τα εξής:

- Σελιδοποίηση. Χωρισμός των αποτελεσμάτων σε σελίδες για μείωση του απαιτούμενου χώρου
- Άμεση αναζήτηση ανάμεσα σε όλα τα στοιχεία που έχει ο πίνακας και φιλτράρισμα των γραμμών που περιλαμβάνουν τον όρο αναζήτησης.
- Ταξινόμηση. Δυνατότητα ταξινόμησης πολλαπλών πεδίων.
- Υποστήριξη DOM, Javascript, Ajax, server-side επεξεργασία.
- Πολλαπλά extensions.
- Πλήρες μεταγλωτίσιμο.

3.14.3. Χρήση

Για ενεργοποίηση το plugin πάνω σε κάποιο πίνακα προστίθεται η παρακάτω εντολή στο event “ready” του jQuery.

```
$(document).ready(function(){
        $('#myTable').DataTable();
});
```

3.14.4. License

Το plugin DataTables παρέχεται με License “MIT License”.

3.15. Uphold

3.15.1. Περιγραφή

Εικόνα 9 Λογότυπο πλατφόρμας Uphold

Το Uphold είναι μία πλατφόρμα που εξυπηρετεί περισσότερες από 184 χώρες σε περισσότερα από 20 διαφορετικά νομίσματα αλλά και μέταλλα σε ξένο συνάλλαγμα. Επιτρέπει το διασυνοριακό έμβασμα μέσω Virtual MasterCard για συναλλαγές από μέλη από όλο τον κόσμο και χρήση στο ηλεκτρονικό εμπόριο.

Το Uphold συνδυάζει ένα μοντέλο εφαρμογής πλατφόρμας με συνδεσιμότητα πληρωμών για να προσφέρει χρηματοπιστωτικές υπηρεσίες σε μια παγκόσμια αγορά. Το Uphold ενδυναμώνει την καινοτομία στις χρηματοπιστωτικές υπηρεσίες μέσω μιας πλατφόρμας προσέγγισης, όπου οι προγραμματιστές εφαρμογών και οι συνεργάτες της Fintech μπορούν να εκμεταλλευτούν την εμβέλεια της Uphold μέσω αδειοδοτημένων σχέσεων με τράπεζες και συνεργάτες χρηματοοικονομικών υπηρεσιών σε όλο τον κόσμο. [71]

3.15.2. Sandbox Uphold

Για την υλοποίηση της εφαρμογής χρησιμοποιείτε ένα αναπτυξιακό περιβάλλον για την κατασκευή και τον έλεγχο εφαρμογών που είναι ενσωματωμένες στην πλατφόρμα Uphold Platform. Το Sandbox δεν χρησιμοποιεί πραγματικά χρήματα αλλά εικονικά.

4. Εισαγωγή στο λειτουργικό σύστημα Android

Εικόνα 10 Λογότυπα εκδόσεων Android

4.1. Ιστορικά στοιχεία

Το Android είναι ένα λειτουργικό σύστημα ανοιχτού κώδικα, βασισμένο στο Linux, για φορητές κυρίως συσκευές όπως smartphones και tablet με οθόνη αφής. Αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance η οποία είναι μια κοινοπραξία εταιριών λογισμικού, κατασκευής hardware και τηλεπικοινωνιών, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις φορητές συσκευές. Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance. Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού.

Τον Ιούλιο του 2005, η Google εξαγόρασε την Android Inc, μια μικρή εταιρεία με έδρα το Palo Alto στην California των ΗΠΑ. Εκείνη την εποχή ελάχιστα ήταν γνωστά για τις λειτουργίες της Android Inc, εκτός του ότι ανέπτυσσαν λογισμικό για κινητά τηλέφωνα. Αυτή ήταν η αρχή της φημολογίας περί σχεδίων της Google για να διεισδύσει στην αγορά κινητής τηλεφωνίας. Στην Google, η ομάδα με επικεφαλής τον Andy Rubin ανέπτυξε μια κινητή πλατφόρμα που στηρίζεται στον πυρήνα του Linux, την οποία προώθησαν με την παροχή ενός ευέλικτου, αναβαθμίσιμου συστήματος. Έχει αναφερθεί ότι η Google είχε ήδη συγκεντρώσει μια σειρά από εταίρους hardware και software και επισήμανε στους παρόχους ότι ήταν ανοικτή σε διάφορους βαθμούς συνεργασίας εκ μέρους της. Έντυπα και ηλεκτρονικά μέσα ενημέρωσης σύντομα ανέφεραν φήμες ότι η Google ανέπτυσσε μια Google-branded συσκευή. Περισσότερες φήμες ακολούθησαν, αναφέροντας ότι η Google καθόριζε τις τεχνικές προδιαγραφές και έδειχνε πρωτότυπα στους κατασκευαστές κινητών τηλεφώνων και τους φορείς δικτύων. Τελικά η Google παρουσίασε το

smartphone της Nexus One που είναι η πρώτη συσκευή που βασίστηκε στο open source λειτουργικό σύστημα Android. Η συσκευή κατασκευάστηκε από την HTC, και έγινε διαθέσιμη στις 5 Ιανουαρίου 2010.[27][52]

4.2. Εξέλιξη

Πρόκειται για το πιο γρήγορα αναπτυσσόμενο λειτουργικό σύστημα στην κόσμο. Λίγο πριν την επίσημη κυκλοφορία του υπήρξαν τουλάχιστον δύο ανεπίσημες εκδόσεις. Στις 5 Νοεμβρίου 2007 παρουσιάστηκε η πρώτη έκδοση beta και τότε γεννήθηκε επίσημα. Το Android 1.0/1.1 κυκλοφόρησε στις 23 Σεπτεμβρίου του 2008 και ήταν το πρώτο κινητό με λειτουργικό Android ήταν το G1 της T-mobile.[27][52]

4.3. Εκδόσεις

Κωδικό όνομα	Νούμερο έκδοσης	Ημερομηνία αρχικής κυκλοφορίας	Επίπεδο API
N/A	1.0	23 Σεπτεμβρίου 2008	1
	1.1	9 Φεβρουάριου 2009	2
Cupcake	1.5	27 Απριλίου 2009	3
Donut	1.6	15 Σεπτεμβρίου 2009	4
Eclair	2.0 – 2.1	26 Οκτωβρίου 2009	5–7
Froyo	2.2 – 2.2.3	20 Μαΐου 2010	8
Gingerbread	2.3 – 2.3.7	6 Δεκεμβρίου 2010	9–10
Honeycomb	3.0 – 3.2.6	22 Φεβρουάριου 2011	11–13
Ice Cream Sandwich	4.0 – 4.0.4	18 Οκτωβρίου 2011	14–15
Jelly Bean	4.1 – 4.3.1	9 Ιουλίου 2012	16–18
KitKat	4.4 – 4.4.4	31 Οκτωβρίου 2013	19–20
Lollipop	5.0 – 5.1.1	12 Νοεμβρίου 2014	21–22
Marshmallow	6.0 – 6.0.1	5 Οκτωβρίου 2015	23
Nougat	**7.0 - 7.1.1**	**22 Αυγούστου 2016**	**24**

Εικόνα 11 Εκδόσεις Λειτουργικού Συστήματος Android

Από τον Απρίλιο του 2009, οι εκδόσεις του Android έχουν θέμα από την ζαχαροπλαστική στην κωδική ονομασία τους, και κυκλοφόρησαν σε αλφαβητική σειρά, εξαιρουμένων των εκδόσεων 1.0 και 1.1, που δεν τέθηκαν υπό συγκεκριμένα κώδικα ονόματα:

4.3.1. Alpha

Υπήρχαν τουλάχιστον δύο ανεπίσημες κυκλοφορίες στο εσωτερικό της Google και της (OHA) Open Handset Alliance πριν η Beta κυκλοφορήσει το Νοέμβριο του 2007. Οι ονομασίας που επιλέχθηκαν στις εσωτερικές εκδόσεις ήταν "Astro Boy", "Bender» και "R2-D2". Ο Dan Morrill δημιούργησε μερικά από τα πρώτα λογότυπα μασκότ, αλλά το τρέχον πράσινο λογότυπο του Android σχεδιάστηκε από την Irina Blok. Ο διαχειριστής του έργου, ο Ryan Gibson σχεδίασε το καθεστώς των ονομασιών γλυκισμάτων που έχει χρησιμοποιηθεί για την πλειονότητα των δημοσίων κυκλοφοριών του λειτουργικού, ξεκινώντας με το Android 1.5.

4.3.2. Beta

Η Beta κυκλοφόρησε στις 5 Νοεμβρίου του 2007, ενώ το σετ ανάπτυξης λογισμικού (SDK) κυκλοφόρησε στις 12 Νοεμβρίου, 2007. Η ημερομηνία της 5 Νοέμβρη είναι ευρέως γνωστή ως τα "γενέθλια" του Android.

4.3.3. Android 1.5 Cupcake

Εικόνα 12 Έκδοση λειτουργικού Συστήματος Android 1.5

Η έκδοση Android 1.5 Cupcake εμφανίστηκε της 27 Απριλίου 2009 και είναι μία από τις σημαντικότερες αναβαθμίσεις του καθώς σε αυτήν εμφανίζεται νέο ψηφιακό πληκτρολόγιο το οποίο έφερε ριζική αλλαγή στον σχεδιασμό των νέων smartphones αφαιρώντας το φυσικό πληκτρολόγιο QEWRTY από αυτά. Επίσης προστέθηκαν νέες λειτουργίες όπως η καταγραφή βίντεο από την κάμερα του κινητού, παρακολούθηση του αλλά και η μεταφόρτωση του στο Youtube.

4.3.4. Android 1.6 Donut

Εικόνα 13 Έκδοση λειτουργικού Συστήματος Android 1.6

Παρουσιάστηκε στις 15 Σεπτεμβρίου 2009 και από αυτή την έκδοση και μετά υποστηρίχθηκαν διαφορετικές αναλύσεις οθόνης ανεξαρτήτου πυκνότητας των pixels. Επιπλέον ξεκίνησε η υποστήριξη της επιλογή πολλαπλών αρχείων ταυτόχρονα κάτι που πλέον θεωρείται δεδομένο. Τέλος πρωτοεμφανίστηκε η μπάρα αναζήτησης της Google στην αρχική οθόνη του κινητού και επανασχεδιάστηκε το Android market.

4.3.5. Android 2.0-2.1 Eclair

Εικόνα 14 Έκδοση λειτουργικού Συστήματος Android 2

Παρουσιάστηκε στις 26 Οκτωβρίου 2009 και βελτίωσε σημαντικά τη λειτουργία της κάμερας με λειτουργίες όπως την ενσωμάτωση χαρακτηριστικών όπως led flash, χρωματικό εφέ, λειτουργίας σκηνών, ρυθμίσεων ισορροπίας λευκού και άλλα. Επίσης σημαντικές αλλαγές υπήρξαν και στους χάρτες Google Maps. Με δυνατότητα φωνητικής καθοδήγησης και οδηγιών. Τέλος βελτιώθηκε η δυνατότητα διαμόρφωσης της αρχικής οθόνης.

4.3.6. Android 2.2 – 2.2.3 Froyo

Εικόνα 15 Έκδοση λειτουργικού Συστήματος Android 2.2/2.3

Κυκλοφόρησε στις 20 Μαΐου 2010 με σημαντικές ενημερώσεις και βελτιώσεις σε απόδοση και ταχύτητα. Σύμφωνα με την Google έπειτα από αυτή την ενημέρωση οι συσκευές έγιναν δύο έως πέντε φορές ταχύτερες. Ενσωματώθηκε η λειτουργία wifi hotspot και ο έλεγχος χρήσης πακέτων

δεδομένων του παρόχου κινητής τηλεφωνίας από το χρήστη. Τέλος δώθηκε η δυνατότητα μεταφοράς των εφαρμογών από το χώρο αποθήκευσης της συσκευής στην κάρτα μνήμης.

4.3.7. Android 2.3 – 2.3.7 Gingerbread

Εικόνα 16 Έκδοση λειτουργικού Συστήματος Android 2.3

Εμφανίστηκε στις 6 Δεκεμβρίου 2010 και ήταν μία από τις δημοφιλέστερες εκδόσεις Android με σημαντικότερη αναβάθμιση την προσθήκη αισθητήρων (γυροσκόπιο, βαρόμετρο κ.α) κάτι που έδωσε τεράστιες δυνατότητες ανάπτυξης εφαρμογών και παιχνιδιών για το κινητό. Επίσης αναβαθμίστηκε ο έλεγχος της μπαταρίας και εμφανίστηκαν σημαντικές λειτουργίες όπως το NFC και η δυνατότητα κλήσεων μέσω internet (VoIP).

4.3.8. Android 3.0 - 3.2.6 Honeycomb

Εικόνα 17 Έκδοση λειτουργικού Συστήματος Android 3

Παρουσιάστηκε 22 Φεβρουαρίου 2011 και ήταν μία έκδοση η οποία δημιουργήθηκε κυρίως για tablets. Κύριες αλλαγές, η βελτίωση του περιβάλλοντος χρήστη και του multitasking καθώς και υποστήριξη επεξεργαστών δύο και τεσσάρων πυρήνων.

4.0 - 4.0.4 Ice Cream Sandwich

Εικόνα 18 Έκδοση λειτουργικού Συστήματος Android 4

Η έκδοση 4.0 που κυκλοφόρησε στις 18 Οκτωβρίου)01 και έκανε το android ένα ολοκληρωμένο του λειτουργικό)στημα, οπότε η αναβάθμιση του άρχισε σταδιακά να επιβραδύνεται. Βασικό κομμάτι σε αυτή την ενημέρωση είναι η αλλαγή του περιβάλλοντος χρήστη με τη σχεδίαση

Holo UI. Επίσης προστέθηκε η αναγνώριση προσώπου στην ασφάλεια της συσκευής. Τέλος βελτιώθηκε η λειτουργία NFC με τη χρήση του Android Beam καθώς και κάμερα με τη δυνατότητα εγγραφής βίντεο σε ανάλυση 1080p.

4.3.10. Android 4.1 - 4.3.1 Jelly Bean

Εικόνα 19 Έκδοση λειτουργικού Συστήματος Android 4.1

Η ανάπτυξη του συστήματος είχε αρχίσει ήδη να επιβραδύνεται και αυτό έγινε ακόμα πιο φανερό από την αμέσως επόμενη έκδοση που είναι το Jelly Bean η οποία έμεινε στην αγορά για περισσότερα από 15 μήνες μέχρι να έρθει η επόμενη έκδοση. Αυτή εμφανίστηκε 9 Ιουλίου 2012 και η τελευταία αναβάθμιση έγινε 24 Ιουλίου 2013. Συνοπτικά περιείχε αλλαγές στην απόδοση και την ταχύτητα του λογισμικού. Για πρώτη φορά εμφανίστηκε η δυνατότητα ύπαρξης πολλών διαφορετικών χρηστών σε μία συσκευή, αλλά και μία νέα δυνατότητα το Google Now όπου είναι ένας ψηφιακός βοηθός μέσω του οποίου η Google παρέχει στο χρήστη άμεσες πληροφορίες σε ότι ρωτάει μέσω του διαδικτύου όπως είναι ο καιρός ή οι ειδήσεις ανάλογα με τα ενδιαφέροντα του χρήστη ή την περιοχή στην οποία βρίσκεται.

4.3.11. Android 4.4 - 4.4.4 KitKat

Εικόνα 20 Έκδοση λειτουργικού Συστήματος Android 4.4

Κυκλοφόρησε στις 31 Οκτωβρίου του 2013 και έφερε τα πιο καινοτόμα, όμορφα και χρήσιμα χαρακτηριστικά του Android σε όλο και περισσότερες συσκευές σε όλο τον κόσμο. Η έκδοση αυτή σχεδιάστηκε για να τρέχει πιο γρήγορα, ομαλά και υπεύθυνα σε ένα ευρύτερο φάσμα συσκευών. Εξελίχθηκε το Google Now με την λειτουργία OK Google όπου επιτρέπει στον χρήστη να κάνει πράγματα στην συσκευή του χωρίς να την αγγίξει αλλά απλά με φωνητικές εντολές. Εμφανίστηκαν νέες NFC δυνατότητες μέσω Host

Card Emulation (HCE), για πληρωμές, πρόσβαση σε κάρτες αλλά και άλλες υπηρεσίες. Τέλος προστέθηκε και η λειτουργία ασύρματης εκτύπωσης.

4.3.12. Android 5.0 - 5.1.1 Lollipop

Εικόνα 21 Έκδοση λειτουργικού Συστήματος Android 5

Κυκλοφόρησε στις 12 Νοεμβρίου του 2014. Αυτή η έκδοση περιείχε πολλά νέα χαρακτηριστικά για τους χρήστες και χιλιάδες νέα API για τους προγραμματιστές. Επέκτεινε το λειτουργικό σε περισσότερα είδη συσκευών όπως ρολόγια, τηλεοράσεις και αυτοκίνητα. Προχώρησε σε μία ριζική αλλαγή στο περιβάλλον χρήστη και διαμόρφωσε το Material design, μία επίπεδη σχεδίαση που έδωσε απλότητα και περισσότερη ευκολία στη χρήση του Android. Επίσης προστέθηκε η δυνατότητα να συνδέονται άμεσα μεταξύ τους όλες οι συσκευές του χρήστη που έχουν Android Lollipop με τη λειτουργία Multi screen και να μοιράζονται από αρχεία ρύθμισης μέχρι και το ιστορικό αναζήτησης. Τέλος από αυτή την έκδοση ξεκίνησε να υποστηρίζονται επεξεργαστές με 64bit.

4.3.13. Android 6.0 - 6.0.1 Marshmallow

Εικόνα 22 Έκδοση λειτουργικού Συστήματος Android 6

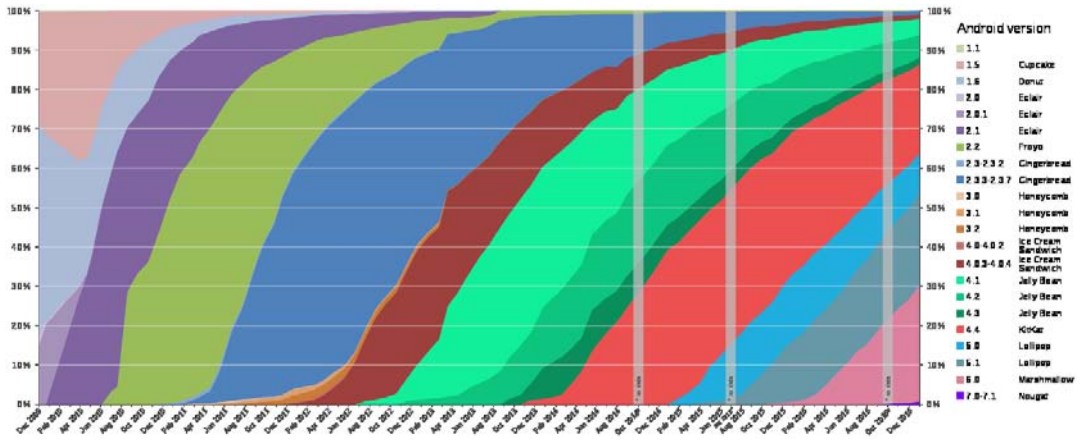
Το Marshmallow (με την κωδική ονομασία Android M κατά τη διάρκεια της ανάπτυξης του) είναι η έκτη κύρια έκδοση του λειτουργικού συστήματος. Κυκλοφόρησε επίσημα τον 5 Οκτωβρίου του 2015. Το Marshmallow επικεντρώνεται κυρίως στη βελτίωση της συνολικής εμπειρίας του χρήστη. Σε σύγκριση με τον προκάτοχο του, Lollipop, εισήγαγε μια νέα αρχιτεκτονική στα δικαιώματα των εφαρμογών, χρησιμοποιείται για πρώτη φορά ένα νέο χαρακτηριστικό το "Now on Tap" όπου επιτρέπει στους χρήστες να πραγματοποιούν αναζητήσεις στο πλαίσιο των πληροφοριών μιας εφαρμογής. Επίσης εισήγαγε ένα νέο σύστημα διαχείρισης ενέργειας που μειώνει την δραστηριότητα του

παρασκηνίου όταν μια συσκευή δεν χρησιμοποιείται, εγγενή υποστήριξη για την αναγνώριση δακτυλικών αποτυπωμάτων και USB τύπου-C, τη δυνατότητα μεταφοράς των δεδομένων και των εφαρμογές σε μια κάρτα microSD, και άλλες εσωτερικές αλλαγές.

4.3.14. Android 7.0 - 7.1.1 Nougat

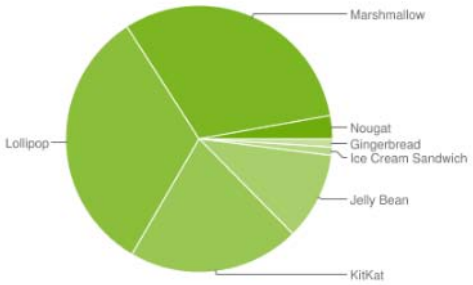
Εικόνα 23 Έκδοση λειτουργικού Συστήματος Android 7

Το "Nougat" με την κωδική ονομασία Android N κατά τη διάρκεια της ανάπτυξης του είναι η έβδομη σημαντική έκδοση του λειτουργικού συστήματος. Κυκλοφόρησε για πρώτη φορά ως beta build στις 9 Μαρτίου 2016 και επίσημα στις 22 Αυγούστου 2016.Το Nougat εισάγει σημαντικές αλλαγές στο λειτουργικό σύστημα και την πλατφόρμα ανάπτυξης του, συμπεριλαμβανομένης της δυνατότητας να εμφανίσει πολλαπλές εφαρμογές στην οθόνη με προβολή διαίρεσης της οθόνης, της υποστήριξη για την ενσωμάτωση απαντήσεων στις κοινοποιήσεις, και τις απρόσκοπτες ενημερώσεις συστήματος για υποστηριζόμενες συσκευές.



Εικόνα 24 Παγκόσμια κατανομή εκδόσεων Android από τον Δεκέμβριο του 2009

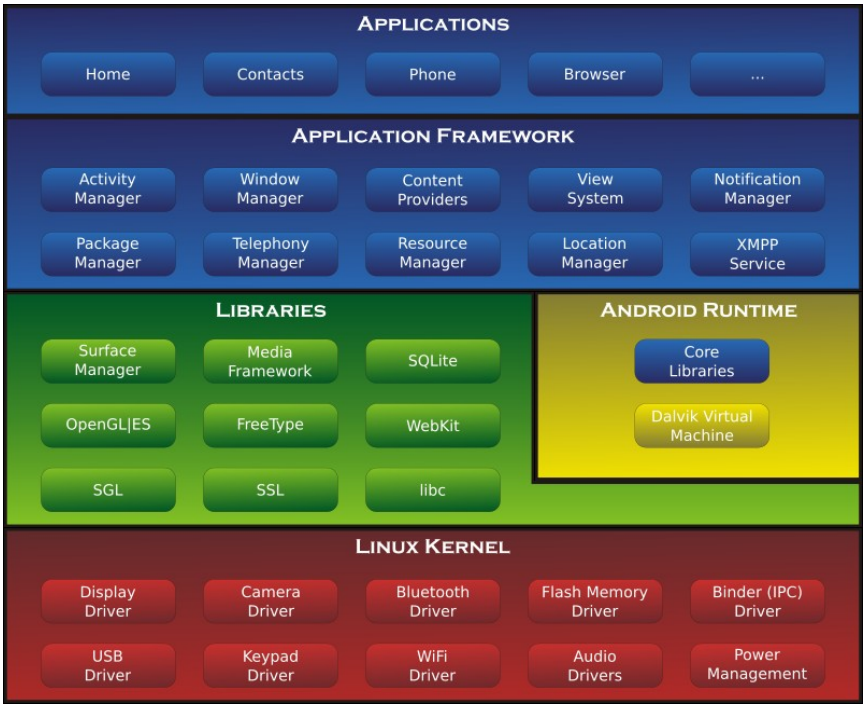
Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	1.0%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.0%
4.1.x	Jelly Bean	16	3.7%
4.2.x		17	5.4%
4.3		18	1.5%
4.4	KitKat	19	20.8%
5.0	Lollipop	21	9.4%
5.1		22	23.1%
6.0	Marshmallow	23	31.3%
7.0	Nougat	24	2.4%
7.1		25	0.4%



Εικόνα 25 Ποσοστό χρήσης εκδόσεων Android (Μάρτιος 2017)

4.4. Αρχιτεκτονική του Android

Η αρχιτεκτονική του Android περιλαμβάνει τα εξής επίπεδα-layers, (από το υψηλότερο προς το χαμηλότερο).



Εικόνα 26 Διάγραμμα Αρχιτεκτονικής Android

4.4.1. Επίπεδο Εφαρμογών (Application)

Το Android είναι εξαρχής εφοδιασμένο με ένα σύνολο από βασικές εφαρμογές που περιλαμβάνουν ένα email client, ένα πρόγραμμα για SMS μηνύματα, ημερολόγιο, την εφαρμογή χαρτών (Google Maps), τον περιηγητή ιστού, ένα πρόγραμμα για δομημένη αποθήκευση των επαφών και άλλα. Όλες οι εφαρμογές είναι γραμμένες στην γλώσσα προγραμματισμού Java.[27][33][38]

4.4.2. Επίπεδα Πλαισίου Εφαρμογών (Applications Framework)

Παρέχοντας μια ανοικτή πλατφόρμα ανάπτυξης, το Android προσφέρει στους προγραμματιστές την δυνατότητα να κατασκευάσουν πλούσιες και καινοτόμες εφαρμογές. Οι προγραμματιστές έχουν την δυνατότητα να εκμεταλλευτούν πλήρως το hardware της συσκευής, να έχουν πρόσβαση σε υπηρεσίες εντοπισμού θέσης, να εκτελούν υπηρεσίες στο background, να θέτουν χρονοδιακόπτες για εμφάνιση ειδοποιήσεων και πολλά άλλα. Επίσης έχουν πλήρη πρόσβαση στο ίδιο πλαίσιο από APIs που έχουν οι βασικές εφαρμογές του Android. Η αρχιτεκτονική είναι

διαμορφωμένη με τέτοιο τρόπο που κάθε εφαρμογή μπορεί να χρησιμοποιήσει τις δυνατότητες μιας άλλης και επίσης με τρόπο που δίνει την δυνατότητα στον χρήστη να αλλάξει τα συστατικά κάθε εφαρμογής. Κάτω από το πλαίσιο των εφαρμογών υπάρχει ένα σύστημα από υπηρεσίες και συστήματα τα οποία περιλαμβάνουν:

- **Διαχείριση Δραστηριότητας** (Activity Manager): Παρακολουθεί και ελέγχει κάθε λειτουργία της εφαρμογής.
- **Πάροχοι Περιεχομένου** (Content Providers): Επιτρέπει σε εφαρμογές να δημοσιοποιούν και να ανταλλάσσουν δεδομένα με άλλες εφαρμογές.
- **Διαχείριση Ειδοποιήσεων** (Notifications Manager): Επιτρέπει σε εφαρμογές να εμφανίζουν ειδοποιήσεις στο χρήστη.
- **Διαχείριση Πακέτου** (Package Manager): Επιτρέπει στις εφαρμογές να αντλούν πληροφορίες σχετικά με άλλες εφαρμογές, που είναι εγκατεστημένες στη συσκευή.
- **Διαχείριση Τηλεφωνίας** (Telephony Manager): Παρέχει πληροφορίες στην εφαρμογή για τις διαθέσιμες υπηρεσίες τηλεφωνίας, όπως και για την κατάσταση του συνδρομητή.
- **Διαχείριση Τοποθεσίας** (Location Manager): Παρέχει πρόσβαση στις υπηρεσίες εντοπισμού και επιτρέπει στην εφαρμογή να λαμβάνει ενημερώσεις σχετικά με τις αλλαγές θέσης. [27][33][38]

4.4.3. Επίπεδο Βιβλιοθηκών (Libraries)

Το επίπεδο βιβλιοθηκών περιλαμβάνει ένα σύνολο από βιβλιοθήκες οι οποίες χρησιμοποιούνται από διάφορα στοιχεία του συστήματος του Android. Οι δυνατότητες που προσφέρουν αυτές οι βιβλιοθήκες είναι προσβάσιμες στους προγραμματιστές δια μέσου του επιπέδου πλαισίου εφαρμογής.[13][38]

4.4.4. Επίπεδο Εκτέλεσης (Android Runtime)

Το τμήμα αυτό περιέχει ένα βασικό συστατικό που ονομάζεται Dalvik Virtual Machine το οποίο είναι ένα είδος Java Virtual Machine, ειδικά σχεδιασμένο και

βελτιστοποιημένο για κινητές συσκευές, που λειτουργούν με συσσωρευτή και διαθέτουν περιορισμένες δυνατότητες σε μνήμη και CPU. Η Dalvik VM κάνει χρήση των βασικών χαρακτηριστικών του Linux, όπως η διαχείριση της μνήμης και multi-threading. Η Dalvik VM ενεργοποιεί την κάθε εφαρμογή Android ώστε να τρέξει τη δική της διαδικασία. Το Android runtime παρέχει επίσης ένα σύνολο βασικών βιβλιοθηκών (Core Libraries) που επιτρέπουν σε προγραμματιστές να γράψουν εφαρμογές Android, χρησιμοποιώντας την γλώσσα Java.[13][38]

4.4.5. Πυρήνας Linux (Linux Kernel)

Το λειτουργικό Android έχει χτιστεί πάνω στον Linux Kernel ο οποίος διαχειρίζεται βασικές υπηρεσίες συστήματος όπως την ασφάλεια, την διαχείριση μνήμης, την διαχείριση διεργασιών, την στοίβα δικτύου, και τους οδηγούς συσκευών. Ο Linux Kernel λειτουργεί επίσης ως ένα ενδιάμεσο επίπεδο αφαίρεσης μεταξύ της στοίβας λογισμικού και του υλικού.[13][34]

4.5. Βοηθητικά εργαλεία για την ανάπτυξη εφαρμογών για το λειτουργικό σύστημα Android

4.5.1. Android Emulator

Ένα από τα βασικότερα εργαλεία ειδικά κατά τα πρώτα στάδια ανάπτυξης της εφαρμογής είναι ο προσομοιωτής που βασίζεται στον QEMU, ένα προσομοιωτή επεξεργαστών. Προσομοιώνει σχεδόν όλες τις λειτουργίες μιας πραγματικής συσκευής, όπως πλήρης λειτουργία του λειτουργικού συστήματος, πρόσβαση στις βασικές εφαρμογές του Android, σύνδεση με το διαδίκτυο και χρήση των Google Maps. Ο προσομοιωτής αρχίζει να χάνει την αξία του όταν η εφαρμογή απαιτεί δεδομένα από τους αισθητήρες. Ο προγραμματιστής μπορεί να δώσει εικονικά δεδομένα για τη θέση της συσκευής χρησιμοποιώντας το εργαλείο DDMS, όμως δεν συμβαίνει το ίδιο με τη λειτουργία της κάμερας και της πυξίδας.[13][40]

4.5.2. Android Virtual Devices (AVDs)

Ένα AVD περιγράφει μια εικονική συσκευή και χρησιμοποιείται από τον Android Emulator. Η δημιουργία μιας εικονικής συσκευής συμπεριλαμβάνει τον προσδιορισμό της έκδοσης του Android API, τη χρήση ή όχι Google Maps και τον καθορισμό του μεγέθους της μνήμης. Κάθε εικονική συσκευή είναι ανεξάρτητη από τις άλλες. Διαθέτει τη δική της μνήμη, τις δικές ρυθμίσεις και τις δικές εφαρμογές. Είναι σαν μια πλήρως ξεχωριστή συσκευή.[13][40]

4.5.3. Android Debug Bridge (adb)

Το Android Debug Bridge επιτρέπει τη διαχείριση εικονικών και πραγματικών συσκευών.

4.5.4. Android SDK

Εκτός από το λειτουργικό, το Android προσφέρει και κάποια εργαλεία που βοηθούν την ανάπτυξη εφαρμογών που εκτελούνται στο περιβάλλον του. Το πιο βασικό από αυτά είναι το Android SDK (Software Development Kit).

Το Android SDK παρέχει μια σειρά από εργαλεία με τη μορφή προγραμμάτων, που μπορούν να εγκατασταθούν σαν ένα πακέτο σε όλα τα ευρέως χρησιμοποιούμενα λειτουργικά συστήματα (Windows, Mac OS, Linux) και τα οποία παρέχουν όλα όσα χρειάζεται ένα προγραμματιστής για να αρχίσει να προγραμματίζει για το Android. Στο πακέτο περιλαμβάνονται εργαλεία για επικοινωνία με συσκευές, δημιουργία εικονικών συσκευών, παραδείγματα εφαρμογών, απαραίτητες βιβλιοθήκες κ.α..

Ο προγραμματισμός σε αυτό γίνεται με τη γλώσσα JAVA. Το Android SDK, καθώς και άλλα βοηθητικά εργαλεία, που χρησιμοποιήθηκαν κατά την ανάπτυξη της εφαρμογής, θα αναλυθούν εκτενέστερα σε επόμενη ενότητα.

Σε επόμενη ενότητα θα αναλύσουμε τις λειτουργικές απαιτήσεις της εφαρμογής που θα κατασκευαστεί στα πλαίσια της παρούσας πτυχιακής εργασίας.[13][40]

4.6. Mobile εφαρμογή

Η εφαρμογή για κινητό τηλέφωνο είναι μία native εφαρμογή για android γραμμένη σε γλώσσα προγραμματισμού Java. Το βασικό δομικό στοιχείο κάθε τέτοιας εφαρμογής είναι η δραστηριότητα (Activity).

Η δραστηριότητα είναι το στοιχείο αυτό που προσφέρει διεπαφή του χρήστη με την εφαρμογή. Οι περισσότερες εφαρμογές αποτελούνται από την αρχική δραστηριότητα, μέσα από την οποία μπορούμε να ξεκινήσουμε και άλλες προς εκτέλεση ξεχωριστές εργασίες.

Κάθε δραστηριότητα έχει ένα κύκλο ζωής με διακριτά στάδια και συγκεκριμένες επιτρεπτές μεταβάσεις όπως γίνεται φανερό στο παρακάτω σχήμα. Αξίζει να αναφέρουμε πως όταν η δραστηριότητα αλλάζει κατάσταση τότε το σύστημα καλεί και την ανάλογη μέθοδο γεγονός που μας επιτρέπει να χειριστούμε αποτελεσματικά τις αλλαγές που επέρχονται στην κατάσταση της δραστηριότητας Οι μεταβάσεις είναι απρόβλεπτες κατά το χρόνο εκτέλεσης και οφείλονται στο χρήστη ή στο λειτουργικό σύστημα. Για παράδειγμα αν δούμε ότι η δραστηριότητα είναι προς καταστροφή από το λειτουργικό σύστημα λόγω περιορισμένης μνήμης ή λόγω κλεισίματος από το χρήστη έχουμε τη δυνατότητα να αποθηκεύσουμε τα απαραίτητα δεδομένα κάνοντας τις ανάλογες ενέργειες στη μέθοδο on Destroy.

Σε κάθε δραστηριότητα μπορούμε να προσθέσουμε και να αφαιρέσουμε τεμάχια (fragments). Τα fragments αποτελούν κομμάτια κώδικα που μπορούν να ενσωματώσουν διεπαφή χρήστη και λογική εφαρμογής. Τα fragments μπορούν να προσθαφαιρεθούν σε μία δραστηριότητα δυναμικά κατά τη διάρκεια εκτέλεσης της εφαρμογής. Υπάρχει επίσης η δυνατότητα να συνδυαστούν πολλά fragments σε μία δραστηριότητα για τη δημιουργία ενός πολύπλοκου interface. Η χρήση των fragments συνεπάγεται πολλά πλεονεκτήματα, εκ των οποίων τα κυριότερα είναι η επαναχρησιμοποίηση κώδικα, η ευελιξία, η ταχύτητα και η δυνατότητα δόμησης της εφαρμογής μας με ένα τμηματικό τρόπο. Όπως και οι δραστηριότητες έτσι και κάθε fragment έχει ένα κύκλο ζωής που είναι έντονα εξαρτώμενος από τον κύκλο ζωής της δραστηριότητας μέσα στην οποία εκείνος ζει. Στο παρακάτω σχήμα βλέπουμε όλα τα πιθανά στάδια του κύκλου ζωής ενός fragment καθώς και όλες οι πιθανές μεταβάσεις μεταξύ τους.[13][36]

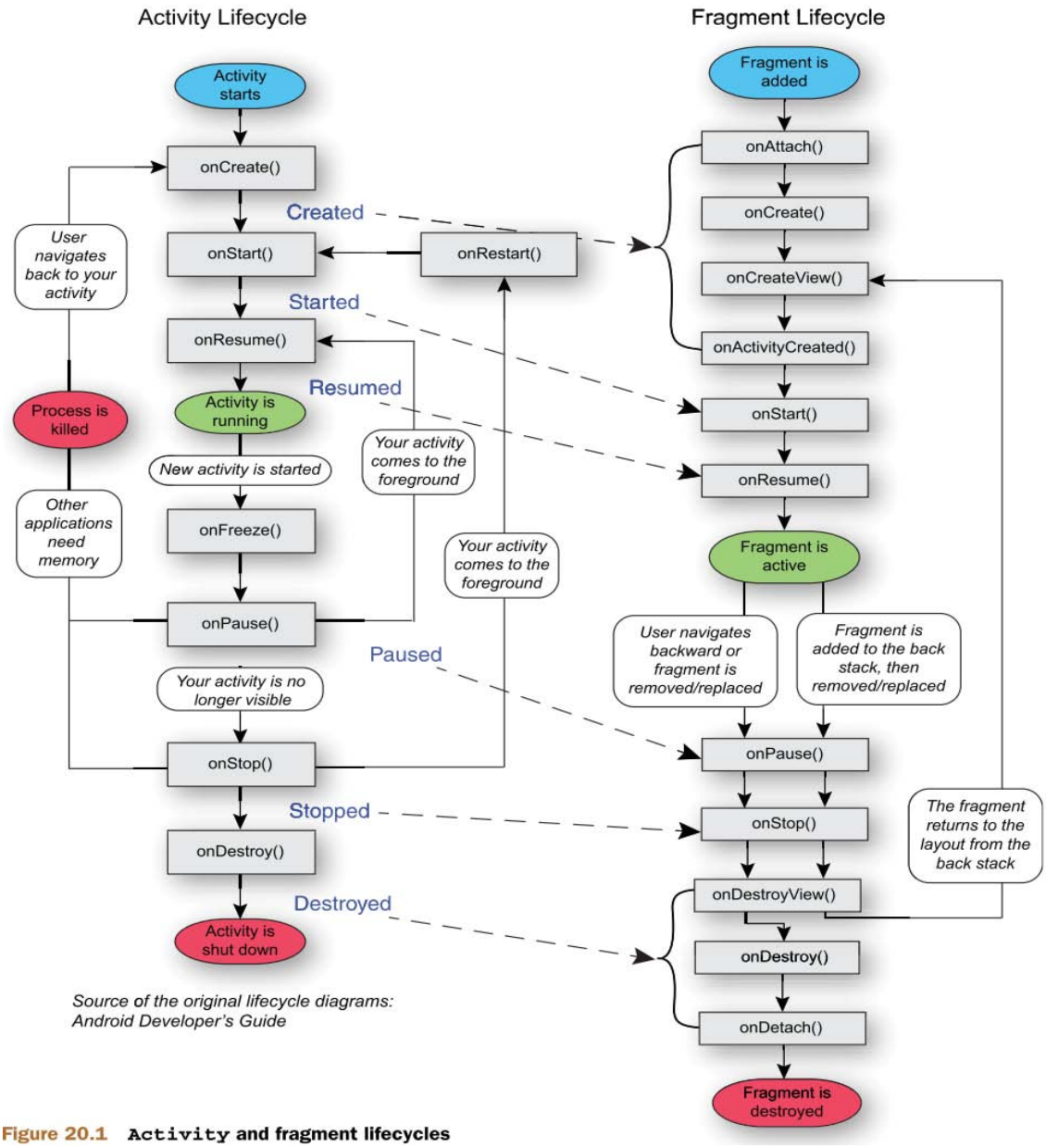


Figure 20.1 **`Activity` and fragment lifecycles**

Εικόνα 27 Δομή του κύκλου "ζωής" μια δραστηριότητας

Σε περίπτωση που απαιτούμε ένα κομμάτι κώδικα να τρέχει συνεχώς στο παρασκήνιο ανεξάρτητα από τη δραστηριότητα και το fragment που είναι στο προσκήνιο εκείνη τη στιγμή, για παράδειγμα να καταγράφεται η θέση του χρήστη ακόμα και αν το τηλέφωνο έχει κλειδωθεί και όλες οι δραστηριότητες είναι σταματημένες. Το SDK του Android μας παρέχει ένα πολύτιμο εργαλείο τις υπηρεσίες(services). Οι υπηρεσίες αποτελούν ένα στοιχείο εφαρμογής που δεν παρέχει διεπαφή χρήστη και είναι σχεδιασμένο να τρέχει μακροπρόθεσμες εργασίας στο παρασκήνιο.[13][36]

5. Bitcoin

5.1. Δημιουργία

Το 2008 δημοσιεύτηκε το πρώτο άρθρο που περιγράφει την ιδέα του ψηφιακού νομίσματος Bitcoin. Δημιουργήθηκε τον Ιανουάριο του 2009. Εμπνευστής και δημιουργός του ένα άγνωστο πρόσωπο με ψευδώνυμο Shatoshi Nakamoto. Περιέγραψε την εκτέλεση συναλλαγών με το ψηφιακό νόμισμα χωρίς την ανάγκη ύπαρξης ενδιάμεσων χρηματοπιστωτικών ιδρυμάτων και την μεταφορά χρημάτων απευθείας από τον έναν συναλλασσόμενο στον άλλο. Περιέγραψε επίσης τον τρόπο με τον οποίο θα μπορεί να διασφαλιστεί η ασφάλεια των συναλλαγών με τη χρήση της ηλεκτρονικής σφραγίδας.

Το Bitcoin δεν υπάρχει επισήμως σε καμία φυσική μορφή, κερμάτων ή χαρτονομισμάτων. Δεν παράγεται από καμιά συγκεκριμένη χώρα δεν ελέγχεται από καμία συγκεκριμένη τράπεζα. Η παραγωγή του η αποθήκευση του και διακίνηση του και όλες οι συναλλαγές με αυτό γίνεται αποκλειστικά σε ηλεκτρονική μορφή. Ουσιαστικά ανήκει στην κατηγορία cryptocurrency καθώς χρησιμοποιεί μεθόδους κρυπτογραφίας για τη δημιουργία και διαχείριση των χρημάτων και για την επιβεβαίωση της εγκυρότητας των συναλλαγών.

Αιτία της δημιουργίας του σύμφωνα με τον μελετητή του, ήταν η ανάγκη μείωσης του κόστους των συναλλαγών που πραγματοποιούνται λόγω της ανάπτυξης του ηλεκτρονικού εμπορίου. Οι συναλλαγές συνήθως επιβαρύνονται από τα κόστη των ενδιάμεσων (χρηματοπιστωτικά ιδρύματα) για την πίστη και την εξασφάλιση όλων των μερών που λαμβάνουν μέρος σε αυτές. Δεν είναι αμελητέο και το επιπλέον κόστους λόγω δόλου και εξαπάτησης που αναπόφευκτα λαμβάνει χώρα σε μερίδα των συναλλαγών αυτών. [28][29][31][53][54]

5.2. Παραγωγή

Κάθε συναλλαγή που γίνεται στο δίκτυο του Bitcoin ελέγχεται για την εγκυρότητα της και στη συνέχεια τοποθετείται σε ένα block μαζί με άλλες συναλλαγές που έχουν ελεγχθεί. Κάθε 10 λεπτά περίπου, δημιουργείται και ένα νέο block για να φιλοξενήσει αυτές τις συναλλαγές το οποίο σχετίζεται με το αμέσως

προηγούμενο αλλά και όλα τα υπόλοιπα block που έχουν δημιουργηθεί πριν από αυτό σχηματίζοντας έτσι μία αλυσίδα (block chain). Ένας μαθηματικός αλγόριθμος χρησιμοποιείται για να γίνει ο συσχετισμός του νέου block με τα προηγούμενα. Μόλις βρεθεί η λύση του αλγόριθμου τότε θα δημιουργηθεί το νέο block και μαζί με αυτό θα δημιουργηθεί κι ένας συγκεκριμένος αριθμός νέων bitcoins, τα οποία θα αποδοθούν σε αυτόν ή αυτούς που βρήκαν την λύση. Αυτή η διαδικασία ονομάζεται mining.

Για να μπορεί η λύση του αλγόριθμου (γρίφου) να προκύπτει κάθε 10 λεπτά περίπου ανεξάρτητα από το πόσοι χρήστες προσπαθούν ταυτόχρονα την βρουν είναι προφανές πώς θα πρέπει να τροποποιηθεί η δυσκολία του γρίφου. Αυτό γίνεται αυτόματα από το δίκτυο σε σχέση με το πόσοι χρήστες προσπαθούν να βρουν τη λύση. Συνεπώς όσο περισσότεροι χρήστες προσπαθούν να λύσουν τον γρίφο δηλαδή όσο μεγαλύτερη είναι η επεξεργαστική ισχύς τόσο μεγαλώνει η δυσκολία του και αντίστροφα. Κάθε λειτουργικός υπολογιστής ή συσκευή που αναζητά τη λύση του γρίφου συμμετέχει παράλληλα και την προστασία του δικτύου από επιθέσεις καθώς και στον έλεγχο και προώθηση των συναλλαγών στο δίκτυο, όπως αναφέρθηκε και παραπάνω.

Τέλος να αναφέρουμε πως το ποσό των bitcoins που αποδίδεται σε όποιον βρίσκει τη λύση του γρίφου υποδιπλασιάζεται κάθε τέσσερα χρόνια περίπου. Ο πρώτος υποδιπλασιασμός της αμοιβής από 50 σε 25 bitcoins έγινε στις 28 Δεκεμβρίου του 2012 και ο δεύτερος περί τα τέλη του 2014. Η δημιουργία των bitcoins γίνεται με ένα εξαιρετικά προβλέψιμο τρόπο. Με πιο απλά λόγια η δημιουργία τους δεν εξαρτάται από την προσφορά και ζήτηση ούτε από τη βούληση κάποιου κράτους ή οργανισμού για δημιουργία περισσότερων ή λιγότερων νομισμάτων. Το δίκτυο από μόνο του καθορίζει το ρυθμό δημιουργίας αυτών. Αυτό είναι ένα εξαιρετικό χαρακτηριστικό του Bitcoin που το καθιστά ανθεκτικό σε κρίσεις, πληθωρισμό και άλλα συχνά προβλήματα των παραδοσιακών νομισμάτων.[28][29][54]

5.3. Η χρήση του και το ψηφιακό πορτοφόλι-Bitcoin Wallet

Για να εκτελέσει κάποιος μία συναλλαγή μέσω Bitcoin πρέπει να είναι κάτοχος ενός ηλεκτρονικού πορτοφολιού. Η μορφή του ηλεκτρονικού πορτοφολιού

ποικίλλει. Συνηθίζεται η τήρηση του ηλεκτρονικού πορτοφολιού μέσω προγράμματος ηλεκτρονικού υπολογιστή ή η χρήση του μέσω πλατφόρμας στον παγκόσμιο ιστό (web wallet) και η τήρηση του στο Smartphone του χρήστη. Την τελευταία περίοδο κάνουν την εμφάνισή τους και μηχανήματα αυτόματων συναλλαγών ATM για αγορές μέσω μετρητών, το πρώτο ATM για Bitcoin εγκαταστάθηκε στο Βανκούβερ του Καναδά.

Μέσα στο ηλεκτρονικό πορτοφόλι μπορεί ο κάτοχός του να τηρεί άπειρο αριθμό διευθύνσεων. Ένα ψηφιακό πορτοφόλι μπορεί να περιλαμβάνει περισσότερες τέτοιες διευθύνσεις ακριβώς όπως ένας πελάτης μιας τράπεζας μπορεί να φέρει περισσότερους από ένα τραπεζικό τραπεζικούς λογαριασμούς. Η διεύθυνση αυτή είναι ένα αναγνωριστικό στοιχείο 27 έως 34 χαρακτήρων σε μήκος (εκτός των χαρακτήρων I,1,0 και O για αποφυγή οπτικών ασαφειών), η οποία ξεκινά με τον αριθμό 1 ή 3 (οι νέες διευθύνσεις Bitcoin ξεκινάνε με τον αριθμό 3 και έχουν μήκος 24 χαρακτήρων) δημιουργείται χωρίς κόστος μέσα από το ηλεκτρονικό πορτοφόλι του χρήστη και είναι ευαίσθητη και διακριτή σε κεφαλαίους και πεζούς χαρακτήρες. Με τη δημιουργία ενός πορτοφολιού για την εκτέλεση συναλλαγών μέσω Bitcoin, το πορτοφόλι του χρήστη συγχρονίζεται αυτόματα με τις συναλλαγές του δικτύου.

Παράδειγμα διεύθυνσης Bitcoin:14LyfYHA4rRxaMmFXseNEtJf9DANoYuhaw

Το δίκτυο συναλλαγών Bitcoin ξεκινάει την παρακολούθηση συναλλαγών μιας διεύθυνσης όταν αυτή κάνει την πρώτη της συναλλαγή μέσα στο δίκτυο, κάτι που έχει ως επακόλουθο να δημιουργούνται διευθύνσεις ακόμα και χωρίς την ανάγκη ύπαρξης δικτύου ή συναλλαγών.

Με τον τρόπο αυτό ο χρήστης μπορεί να δημιουργήσει τόσες διευθύνσεις όσες και οι συναλλαγές που θέλει να δημιουργήσει πριν από κάθε συναλλαγή ώστε με τον τρόπο αυτό να διατηρεί την ανωνυμία του σε συναλλαγές του στο δίκτυο. Για κάθε διεύθυνση που δημιουργείται, δημιουργείται και ένα **private key** που αφορά την κρυπτογράφηση των συναλλαγών που εκτελούνται από τη διεύθυνση αυτή. Το πορτοφόλι δημιουργεί ένα αρχείο με τα private keys των διευθύνσεων που έχουν αποθηκευτεί μέσα σε αυτό. Απώλεια του αρχείου αυτού σημαίνει απώλεια της δυνατότητας νέων συναλλαγών, με τελικό αποτέλεσμα την απώλεια των Bitcoin που έχουν αποθηκευτεί μέσα σε αυτές.[28][29][30][54]

5.4. Συναλλαγή

Για την αποστολή ενός ποσού σε bitcoin είναι απαραίτητη η γνώση της διεύθυνσης του παραλήπτη. Κάθε συναλλαγή ενδέχεται να περιλαμβάνει και μία αμοιβή συναλλαγής, το λεγόμενο fee. Αυτό συμβαίνει γιατί κάθε συναλλαγή για να γίνει τελικά αποδεκτή στο δίκτυο, πρέπει να ενσωματωθεί σε ένα block. Τα blocks δημιουργούνται από τους miners (οι οποίοι εισπράττουν το fee) που δαπανούν υπολογιστική ισχύ και έχουν το δικαίωμα να επιλέγουν τις προς ενσωμάτωση συναλλαγές για το επόμενο block. Είναι στη διακριτική ευχέρεια όσων δημιουργούν τα blocks να δεχτούν ή όχι την όποια συναλλαγή ανάλογα με την αμοιβή fee πού την συνοδεύει.

Για την επικύρωση μίας συναλλαγής που πραγματοποιείται από έναν λογαριασμό απαιτείται η ύπαρξη μίας ή περισσότερων συναλλαγών στο παρελθόν οι οποίες να συγκεντρώνουν σε αυτόν τον λογαριασμό το απαιτούμενο ποσό bitcoins. Οι παρελθούσες συναλλαγές πρακτικά χρησιμοποιούνται ως inputs στην παρούσα. Για να μπορεί κάποιος κόμβος να χρησιμοποιήσει μία παρελθούσα συναλλαγή ως input σε μία νέα, απαιτείται μία διαδικασία ξεκλειδώματος της πρώτης. Σε αυτή την διαδικασία απαιτείται το ιδιωτικό κλειδί που αντιστοιχεί στην διεύθυνση bitcoin στην οποία κατευθύνονται τα bitcoin στην παρελθούσα συναλλαγή. Θεωρητικά είναι αδύνατο κάποιος να χρησιμοποιήσει ως inputs συναλλαγές που απευθύνονται σε κάποιον άλλο.

Επιπλέον κατά την διαδικασία του mining, λαμβάνει χώρα μίας διαδικασία επικύρωσης της συναλλαγής, όσον αφορά το αν έχει χρησιμοποιηθεί ξανά στο παρελθόν. Σε περίπτωση που κάτι τέτοιο έχει συμβεί τότε το block που θα περιλαμβάνει μία τέτοια μη έγκυρη συναλλαγή, δεν γίνεται αποδεκτό από το δίκτυο και κατ' επέκταση ο miner δεν καρπώνεται την σχετική ανταμοιβή. Στην πράξη οι miners φροντίζουν να μην επικυρώνονται ποτέ οι συναλλαγές οι οποίες επιχειρούν να χρησιμοποιούν ήδη χρησιμοποιημένα inputs. Με αυτό τον τρόπο αποτρέπεται το φαινόμενο του double spending..[28][29][30][54]

5.5. Πλεονεκτήματα και μειονεκτήματα της χρήσης του Bitcoin ως μέσο συναλλαγών

Η χρήση του Bitcoin έχει σημαντικά πλεονεκτήματα, που το έχουν κάνει ανταγωνιστικό και αυτό φαίνεται από τον ολοένα και μεγαλύτερο αριθμό χρηστών που το προτιμούν για τις συναλλαγές τους. Παρακάτω παρατίθενται μερικά από τα πλεονεκτήματα που προσφέρει το ψηφιακό νόμισμα.

- **Ταχύτητα συναλλαγών:** Οι συναλλαγές γίνονται άμεσα και ανακοινώνονται σε όλους τους χρήστες σε όλο τον πλανήτη
- **Χαμηλό κόστος συναλλαγών:** Το κόστος είναι περίπου στα 5 Cents και πολλές φορές είναι προαιρετικό εάν δεν είναι απαραίτητη η άμεση επιβεβαίωση της συναλλαγής.
- **Έλεγχος από το χρήστη / Προστασία από υφαρπαγή:** Ο χρήστης είναι ο μοναδικός που μπορεί να κάνει συναλλαγές επειδή κατέχει τον ειδικό κωδικό πρόσβασης στο λογαριασμό του, και με αυτό τον τρόπο είναι πρακτικά αδύνατο να κλαπούν τα νομίσματα.
- **Φορητότητα:** Ανεξάρτητα από τα ποσά που έχει κάποιος αυτό που χρειάζεται είναι ο κωδικός πρόσβασης που είναι πολύ εύκολο να μεταφερθεί είτε σε χαρτί ή σε κάποιο USB.
- **Διαφάνεια στις συναλλαγές**: Όλες οι συναλλαγές που έχουν γίνει με κάποιο νόμισμα είναι καταγεγραμμένες και διαθέσιμες σε οποιονδήποτε θέλει να ελέγξει κάποια διεύθυνση και να δει τι συναλλαγές έχουν γίνει στο παρελθόν.
- **Συναινετική λύση χρήσης και αλλαγών:** Οποιαδήποτε αλλαγή γίνεται στο πρόγραμμα ή στους κανόνες συναλλαγών γίνεται μετά από συναίνεση της κοινότητας με αυτό τον τρόπο αποφεύγονται οποιαδήποτε αλλαγές θα μπορούσαν να γίνουν και να αλλάξουν την ουσία του λογισμικού.
- **Αποκεντρωμένο σύστημα**: Ο λόγος της επιτυχίας του Bitcoin είναι η αποκεντρωμένη φύση του, δηλαδή δεν χρειάζεται καμία αρχή ελέγχου που να επιβεβαιώνει την οποιαδήποτε συναλλαγή, κάθε κόμβος του δικτύου ενισχύει το σύστημα, και ο μόνος λόγος που θα επηρέαζε σε σημαντικό βαθμό τη λειτουργία του δικτύου θα ήταν να αποκοπούν όλοι οι υπολογιστές του δικτύου. Αλλά ακόμα και τότε στην επόμενη επανεκκίνηση θα συνέχιζε από κει που σταμάτησε.

- **Υποδιαιρέσεις του νομίσματος:** κάθε νόμισμα υποδιαιρείται έως 8 δεκαδικά ψηφία 0,00000001 με αυτό τον τρόπο επιτρέπει να γίνονται μικρό συναλλαγές που θα ήταν οικονομικά ασύμφορο να γίνουν με συμβατικά νομίσματα.
- **Μη αναστρέψιμη φύση συναλλαγών:** Οι συναλλαγές που γίνονται με Bitcoin είναι μη αναστρέψιμες, γεγονός που παρουσιάζει κάποια πλεονεκτήματα και κάποια μειονεκτήματα. Το πλεονέκτημα προς όσους διαθέτουν προϊόντα με bitcoin είναι ότι οι πληρωμές για αυτά δεν μπορούν να ανακληθούν. Αυτό δίνει επιπλέον κίνητρα σε επιχειρήσεις να πουλάνε σε χαμηλότερες τιμές. Από την άλλη οι χρήστες που εκτελούν συναλλαγές με bitcoin πρέπει να είναι προσεκτικοί και να κάνουν συναλλαγές με παρόχους προϊόντων που έχουν ένα σχετικά αξιόπιστο ιστορικό κινήσεων, όπως για παράδειγμα συμβαίνει και στις πωλήσεις μέσω του ebay.
- **Ιδιωτικότητα συναλλαγών**: κάθε χρήστης μπορεί να δημιουργήσει απεριόριστες διευθύνσεις και να κάνει συναλλαγές εφόσον έχει τα αντίστοιχα bitcoin. Δηλαδή οι διευθύνσεις δεν έχουν καμία σχέση με τα πραγματικά στοιχεία του χρήστη, αλλά παρόλα αυτά δεν συνεπάγεται και ανωνυμία συναλλαγών καθώς όλες οι συναλλαγές δημοσιεύονται στους άλλους χρήστες του δικτύου.

Η χρήση του bitcoin δεν έχει μόνο πλεονεκτήματα αλλά υπάρχουν και κάποια μειονεκτήματα που πολλές φορές κάνουν κάποιους χρήστες σκεπτικούς στη χρήση του.

- **Η απώλεια της ιδιωτικής ταυτότητας:** Ο μόνος τρόπος για να μπορέσει κάποιος να κλέψει τα bitcoin που έχει κάποιος υπό την κατοχή του είναι να καταφέρει να κλέψει την ιδιωτική ταυτότητα που τον καθορίζει μέσα στο δίκτυο, παρόλο που το ίδιο το πρόγραμμα έχει και άλλες δικλείδες ασφαλείας.
- **Ισοτιμίες:** Το bitcoin ως νόμισμα είναι ευαίσθητο σε μεγάλες διακυμάνσεις ισορροπίας τιμών. Αυτό οφείλεται κατά κύριο λόγο στο ότι δεν υπάρχει κάποια κεντρική αρχή που να παρεμβαίνει στην προσφορά και στη ζήτηση. Επίσης το μικρό βάθος της συγκεκριμένης αγοράς είναι ένας ακόμη παράγοντας που επηρεάζει την ισοτιμία, αφού όταν ανταλλάσσονται μεγάλα ποσά σε bitcoin αυτό έχει σαν αποτέλεσμα να επηρεάζονται οι ισπτιμίες στις χρηματαγορές.

- **Νομικό πλαίσιο:** Παρόλο που στην Ευρωπαϊκή Ένωση υπάρχει ένα πλαίσιο νόμου σε ότι αφορά τα κεντρικά ελεγχόμενα νομίσματα το γεγονός ότι το bitcoin είναι αποκεντρωμένο βάζει κάποιους νέους κανόνες και παραμέτρους που δεν έχουν εξεταστεί ακόμα από το νομικό περιβάλλον της κάθε χώρας. Στις ΗΠΑ αντίστοιχα γίνεται μια προσπάθεια να ελέγχου του μόνο όταν εμπλέκεται σε εγκληματικές προσπάθειες.
- **Ασφάλεια δικτύου:** Όλο το σύστημα του bitcoin έχει δείξει ότι μπορεί να αντιδράσει με γρήγορους ρυθμούς σε οποιαδήποτε επίθεση. Το νεαρό όμως της ηλικίας του αλλά και το μέγεθος που αρχίζει να παίρνει ενδεχομένως θα φέρουν προβλήματα που δεν έχουν προβλεφθεί. Από σενάρια που έχουν προσομοιωθεί έχουν αναδειχθεί κάποιοι κίνδυνοι που θα μπορούσαν να υπάρξουν, όπως είναι ο έλεγχος του μεγαλύτερου μέρους του δικτύου από ένα κακόβουλο πρόγραμμα που θα επηρέαζε τις συναλλαγές (διπλές συναλλαγές).
- **Παραβίαση των αλγορίθμων κρυπτογράφησης:** Αυτός ο κίνδυνος μέχρι σήμερα θεωρείται αμελητέος, διότι οι συγκεκριμένοι αλγόριθμοι υπάρχουν στην αγορά πάρα πολλά χρόνια και χρησιμοποιούνται ευρέως για την ασφάλεια των τραπεζικών συναλλαγών. Αυτό έχει σαν αποτέλεσμα να έχουν βελτιωθεί σε εξαιρετικό βαθμό και να θεωρούνται απόλυτα ασφαλείς. .[28][29][30][32][54]

6. Κρυπτογράφηση

6.1. Βασική ορολογία κρυπτογράφησης

Η κρυπτογραφία είναι επιστήμη που βασίζεται στα μαθηματικά για την κωδικοποίηση και την αποκωδικοποίηση των δεδομένων. Είναι δηλαδή το σύνολο των μαθηματικών τεχνικών που στοχεύουν στην εξασφάλιση θεμάτων που άπτονται της ασφάλειας μετάδοσης της πληροφορίας, όπως εμπιστευτικότητα, ακεραιότητα δεδομένων, πιστοποίηση ταυτότητας του αποστολέα και διασφάλιση του αδιάβλητου της πληροφορίας. Οι μέθοδοι κρυπτογράφησης καθιστούν τα ευαίσθητα προσωπικά δεδομένα προσβάσιμα μόνο σε όσους είναι κατάλληλα εξουσιοδοτημένοι. Εξασφαλίζουν έτσι το απόρρητο στις ψηφιακές επικοινωνίες αλλά και στην αποθήκευση ευαίσθητων πληροφοριών. Η αρχική μορφή του μηνύματος αποτελεί το απλό κείμενο (plain text) ενώ το κρυπτογραφημένο κείμενο αποτελεί το κρυπτοκείμενο (ciphetext). Ο μετασχηματισμός του απλού κειμένου σε

κρυπτοκείμενο ονομάζεται κρυπτογράφηση (encryption) ενώ ο μετασχηματισμός του κρυπτοκειμένου σε απλό κείμενου ονομάζεται αποκρυπτογράφηση (decryption). Οι διαδικασίες της κρυπτογράφησης και της αποκρυπτογράφησης υλοποιούνται με τους αλγορίθμους κρυπτογράφησης και αποκρυπτογράφησης αντίστοιχα. Οι δύο αυτοί οι αλγόριθμοι συνιστούν τον κρυπτοαλγόριθμο (cipher). Η διαδικασία κρυπτογράφησης και αποκρυπτογράφησης απαιτεί μία επιπλέον ποσότητα πληροφορίας που την ονομάζουμε κλειδί (key). Η ύπαρξη του κλειδιού είναι και η ειδοποιός διαφορά της κρυπτογράφησης με την κωδικοποίηση. Αναλυτικότερα η κρυπτογράφηση και αποκρυπτογράφηση ενός κειμένου μπορεί να πραγματοποιηθεί με επιτυχία μόνον από τον κάτοχο του σωστού κλειδιού. Το κλειδί είναι μία μυστική ποσότητα η γνώση της οποίας απαιτείται για την επιτυχή κρυπτογράφηση και αποκρυπτογράφηση. Όποιος έχει το κλειδί μπορεί χωρίς μεγάλη προσπάθεια να ανοίξει και να διαβάσει το μήνυμα. Η περιγραφή των διαδικασιών κρυπτογράφησης και αποκρυπτογράφησης αποτελούν το κρυπτοσύστημα. Η σωστή λειτουργία του κρυπτοσυστήματος έγκειται στην μυστικότητα του κλειδιού και όχι των αλγορίθμων.[43][51][53][61][62][69]

6.2. Είδη κρυπτογραφίας

6.2.1. Συμμετρική κρυπτογραφία (secret key cryptography)

Στη συμμετρική κρυπτογραφία χρησιμοποιείται το ίδιο κλειδί τόσο για την κρυπτογράφηση όσο και για την αποκρυπτογράφηση. Ο αποστολέας χρησιμοποιεί το μυστικό κλειδί για να κρυπτογραφήσει το μήνυμα και ο παραλήπτης για να το αποκρυπτογραφήσει. Η απαίτηση του κρυπτοσυστήματος να χρησιμοποιεί το ίδιο κλειδί στην κρυπτογράφηση και αποκρυπτογράφηση προϋποθέτει ότι αποστολέας και ο παραλήπτης έχουν κάποιο ασφαλή τρόπο να μοιραστούν αυτή την πληροφορία. Το κλειδί δημιουργείται και βρίσκεται αρχικά στον αποστολέα καθώς η κρυπτογράφηση προηγείται της αποκρυπτογράφησης, επομένως ο αποστολέας θα πρέπει να στείλει το κλειδί στον παραλήπτη χωρίς να πέσει στα χέρια κάποιου ενδιάμεσου αντιπάλου.

Η συμμετρική κρυπτογραφία χρησιμοποιείται όχι μόνο για κρυπτογράφηση αλλά και για πιστοποίηση ταυτότητας. Μία τεχνική είναι η MAC (Message Authendication Code). Τα πλεονεκτήματα της συμμετρικής κρυπτογραφίας συγκαταλέγονται οι υψηλές ταχύτητες κρυπτογράφησης αποκρυπτογράφησης καθώς επίσης και οι μικρές απαιτήσεις της μνήμης και υπολογιστικής ισχύς. Έτσι κάνει δυνατή την εφαρμογή της σε περιβάλλοντα όπως αυτά ενός κινητού τηλεφώνου ή μιας έξυπνης κάρτας. [43][51][53]

6.2.2. Ασύμμετρη Κρυπτογραφία ή Κρυπτογραφία Δημοσίου Κλειδιού (Public Key Cryptography)

Η τεχνική αυτή, γνωστή ως κρυπτογραφία δημοσίου κλειδιού η ασύμμετρη κρυπτογραφία βασίζεται στην ύπαρξη ενός ζεύγους κλειδιών. Το κλειδί για την κρυπτογράφηση ονομάζεται δημόσιο κλειδί γιατί μπορεί να διατεθεί ελεύθερα χωρίς να απαιτείται ασφαλές κανάλι για τη μετάδοση του. Το κλειδί που χρησιμοποιείται για την αποκρυπτογράφηση είναι το ιδιωτικό κλειδί και παραμένει υπό την κατοχή του παραλήπτη. Το δημόσιο κλειδί δεν μπορεί να χρησιμοποιηθεί για την αποκρυπτογράφηση διότι εάν χρησιμοποιηθεί το αποτέλεσμα δεν θα είναι το αρχικό απλό κείμενο. Το ιδιωτικό κλειδί είναι γνωστό μόνο στον παραλήπτη του μηνύματος. Έτσι σε αντίθεση με τα συμμετρικά κρυπτοσυστήματα, τα κλειδιά δημιουργούνται στον παραλήπτη, ο οποίος είναι ο μόνος που μπορεί να παράγει και να συσχετίσει ένα

ζευγάρι ασύμμετρων κλειδιών. Έτσι το μοντέλο επικοινωνίας ενός ασύμμετρου κρυπτοσυστήματος δεν περιλαμβάνει ασφαλές κανάλι αλλά η μετάδοση του μηνύματος περιλαμβάνει τα ακόλουθα στάδια:

1. ο αποστολέας ζητά από τον παραλήπτη το δημόσιο κλειδί K_{pub}
2. ο παραλήπτης στέλνει το δημόσιο κλειδί μέσω του μη ασφαλούς καναλιού επικοινωνίας
3. ο αποστολέας κρυπτογραφεί το μήνυμα P με το δημόσιο κλειδί του παραλήπτη και στέλνει το κρυπτοκείμενο C στον παραλήπτη
4. ο παραλήπτης αποκρυπτογραφεί το κρυπτοκείμενο χρησιμοποιώντας το ιδιωτικό κλειδί K_{pri}

Μία ενδιαφέρουσα και χρήσιμη ιδιότητα του ασύμμετρου κρυπτοσυστήματος είναι ότι ένα ζευγάρι ιδιωτικού/δημοσίου κλειδιού μπορεί να χρησιμοποιηθεί αντίστροφα. Δηλαδή το ιδιωτικό κλειδί μπορεί να κρυπτογραφήσει ένα απλό κείμενο και το δημόσιο κλειδί να αποκρυπτογραφήσει το αντίστοιχο κρυπτό κείμενο. Αυτή ιδιότητα αποτελεί την αρχή λειτουργίας της ψηφιακής υπογραφής. Ο κάτοχος του ιδιωτικού κλειδιού είναι ο μόνος που μπορεί να κρυπτογραφήσει ένα κείμενο με το ιδιωτικό κλειδί, ενώ οποιοσδήποτε μπορεί να το αποκρυπτογραφήσει. Εάν το κρυπτογραφημένο κείμενο συνοδεύεται από το απλό κείμενο τότε ο παραλήπτης μπορεί να συγκρίνει το απλό κείμενο με το αποτέλεσμα της αποκρυπτογράφησης και να επαληθεύσει ότι το κείμενο προέρχεται από τον κάτοχο του ιδιωτικού κλειδιού.

Το σημαντικότερο πλεονέκτημα της ασύμμετρης κρυπτογραφίας είναι ότι δεν απαιτείται ανταλλαγή μυστικού κλειδιού. Το δημόσιο κλειδί είναι ελεύθερα διαθέσιμο κάτι που κάνει τη διαχείριση των κλειδιών ευκολότερη. Επίσης το ιδιωτικό κλειδί είναι γνωστό μόνο στον ιδιοκτήτη του, καθιστώντας δυσκολότερη την παραποίηση του. Όμως η κρυπτογραφία δημοσίου κλειδιού έχει μεγάλες απαιτήσεις σε υπολογιστική ισχύ και είναι αρκετά αργή κυρίως στα μεγάλα μηνύματα. [43][51][53]

6.3. Κρυπτογραφικές υπηρεσίες

Οι κρυπτογραφικές υπηρεσίες είναι υπηρεσίες που χρησιμοποιώντας κρυπτογραφία, στοχεύουν στην αντιμετώπιση συγκεκριμένων απειλών. Οι κρυπτογραφικές υπηρεσίες είναι οι ακόλουθες:

- **Εμπιστευτικότητα** (Confidentiality). Είναι η προστασία από την αποκάλυψη της πληροφορίας. Η εμπιστευτικότητα θα πρέπει να προσφέρεται με τέτοιο τρόπο ώστε να είναι αδύνατη η αποκάλυψη και πολλές φορές η ίδια η διάθεση της πληροφορίας και σε μη εξουσιοδοτημένα άτομα.
- **Ακεραιότητα** (Intergity). Είναι η προστασία από τη μη εξουσιοδοτημένη τροποποίηση των δεδομένων. Η ακεραιότητα θα πρέπει να παρέχει στον παραλήπτη και γενικότερα στον κάτοχο ενός μηνύματος τη δυνατότητα να μπορεί να ανιχνεύσει πιθανές αλλαγές στο μήνυμα από μη εξουσιοδοτημένα άτομα. Στο χώρο των τηλεπικοινωνιών και της θεωρίας της πληροφορίας η ακεραιότητα είναι γνωστή ως ανίχνευση σφαλμάτων, όπου ένα μήνυμα μπορεί να υποστεί τροποποιήσεις λόγω του θορύβου του καναλιού επικοινωνίας.
- **Αυθεντικοποίηση** (Authentication). Είναι η εξασφάλιση του ότι γνωρίζουμε το χρήστη ή γενικότερα την οντότητα που επικοινωνούμε. Αυθεντικοποίηση δεδομένων είναι η εξασφάλιση ότι ένα μήνυμα προέρχεται πράγματι από τον αποστολέα που θεωρητικά το έχει στείλει.
- **Μη απάρνηση** (Non repudiation). Είναι η υπηρεσία κατά την οποία ο παραλήπτης δεν μπορεί να αρνηθεί ότι έλαβε το μήνυμα (μη απάρνηση προορισμού), ή η υπηρεσία κατά την οποία ο αποστολέας δεν μπορεί να απαρνηθεί ότι έστειλε το μήνυμα (μη απάρνηση προέλευσης) [43][51][53]

6.4. Κρυπτοσύστημα RSA

6.4.1. Το κρυπτοσύστημα RSA

Τα αρχικά του κρυπτοσυστήματος RSA προέρχονται από τα ονόματα των μελετητών R.L.Rivest, A.Shamir και L.Adleman οι οποίοι το δημοσίευσαν το 1978. Το κρυπτοσύστημα RSA είναι κρυπτοσύστημα δημοσίου κλειδιού. Ο αλγόριθμος αυτός είναι ένας από τους πιο διαδεδομένους και περισσότερο χρησιμοποιούμενος αλγόριθμους στην κρυπτογραφία δημοσίου κλειδιού. Είναι κατάλληλος για την κρυπτογράφηση και αποκρυπτογράφηση δεδομένων για τη δημιουργία ψηφιακών υπογραφών και την επαλήθευση τους καθώς και για την ασφαλή μεταφορά κλειδιών.[51][62]

6.4.2. Περιγραφή RSA

Το κρυπτοσύστημα αυτό χρησιμοποιεί υπολογισμούς στο σύνολο $Z_n = \{0,1,2 \dots n-1\}$, όπου n είναι το γινόμενο δύο μεγάλων, διακεκριμένων, πρώτων αριθμών p,q. Για ένα τέτοιο n υπολογίζεται και η συνάρτηση Euler $\varphi(n) = (p-1) * (q-1)$.

1) ***Αλγόριθμος παραγωγής κλειδιού***

Κάθε οντότητα δημιουργεί ένα δημόσιο κλειδί RSA και ένα αντίστοιχο ιδιωτικό κλειδί. ο Α ενεργεί ως εξής:

1. Παράγει δύο μεγάλους διακεκριμένους πρώτους αριθμούς p,q περίπου ίδιου μεγέθους.
2. Υπολογίζει τις ποσότητες n και $\varphi(n)$ ως εξής :

$$n = p * q$$

$$\varphi(n) = (p-1) * (q-1)$$

3. Επιλέγει ένα τυχαίο ακέραιο αριθμό e τέτοιο με 1<e<φ(n) τέτοιο ώστε :

$$\gcd(e, \varphi(n)) = 1$$

4. Χρησιμοποιώντας τον Εκτεταμένο Ευκλείδειο Αλγόριθμο υπολογίζει το μοναδικό ακέραιο d με 1<d<φ(n) τέτοιο ώστε :

$$e * d = 1 \, mod \, \varphi(n)$$

5. Το δημόσιο κλειδί του Α είναι το ζευγάρι (n,e) και το μυστικό κλειδί είναι το d.

Οι ακέραιοι e και d λέγονται εκθέτης κρυπτογράφησης και εκθέτης αποκρυπτογράφησης αντίστοιχα.

2) ***Αλγόριθμος κρυπτογράφησης***

Ο Β κρυπτογραφεί ένα μήνυμα για τον Α και ενεργεί ως εξής:

1. Αποκτά το αυθεντικό δημόσιο κλειδί (n,e) του Α.
2. Αναπαριστά το μήνυμα ως ένα ακέραιο m στο διάστημα [0,n-1].
3. Υπολογίζει $c = m^e mod n$
4. Στέλνει το κρυπτοκείμενο c στον Α.

3) ***Αλγόριθμος αποκρυπτογράφησης***

Ο Α για να ανακτήσει το απλό κείμενο m από το c ενεργεί ως εξής :

1. Χρησιμοποιεί το δημόσιο κλειδί d για να ανακτήσει το $c = m^e mod n$.

Παρατηρήσεις:

- Αφού το n είναι το γινόμενο των δύο πρώτων p,q θα ισχύει: $\varphi(n) = (p-1) * (q-1)$. Έτσι αν κάποιος γνωρίζει τους p,q που μπορεί εύκολα να υπολογίσει $\varphi(n)$ $και$ $άρα$ $το$ d.
- $Κ$αλή επιλογή για δημόσιο εκθέτη e είναι κάποιος πρώτος αριθμός μεγαλύτερος από το max{p,q}.
- Το πρόβλημα υπολογισμού του εκθέτη αποκρυπτογράφησης d στο RSA από το δημόσιο κλειδί (n,e) ανάγεται στο πρόβλημα παραγοντοποίησης του n.
- Όταν λοιπόν παράγονται τα κλειδιά, είναι πολύ σημαντικό οι πρώτοι p,q να επιλέγονται με τέτοιο τρόπο ώστε να επιτυγχάνεται το δυσεπίλυτο της παραγοντοποίησης του n=p*q.

- Στο RSA ισχύει: E(D(m))= m^e. Το κρυπτοσύστημα RSA βασίζεται στη συνάρτηση E(m)= $m^e mod n$ η οποία θεωρείται μη αντιστρέψιμη συνάρτηση. Δεν υπάρχει γνωστός αλγόριθμος που να αντιστρέφει αυτή τη συνάρτηση χωρίς τη γνώση του κρυφού εκθέτη αποκρυπτογράφησης d. Οι διαδικασίες κρυπτογράφησης και αποκρυπτογράφησης είναι αντίστροφες διαδικασίες.

6.5. Πλεονεκτήματα RSA

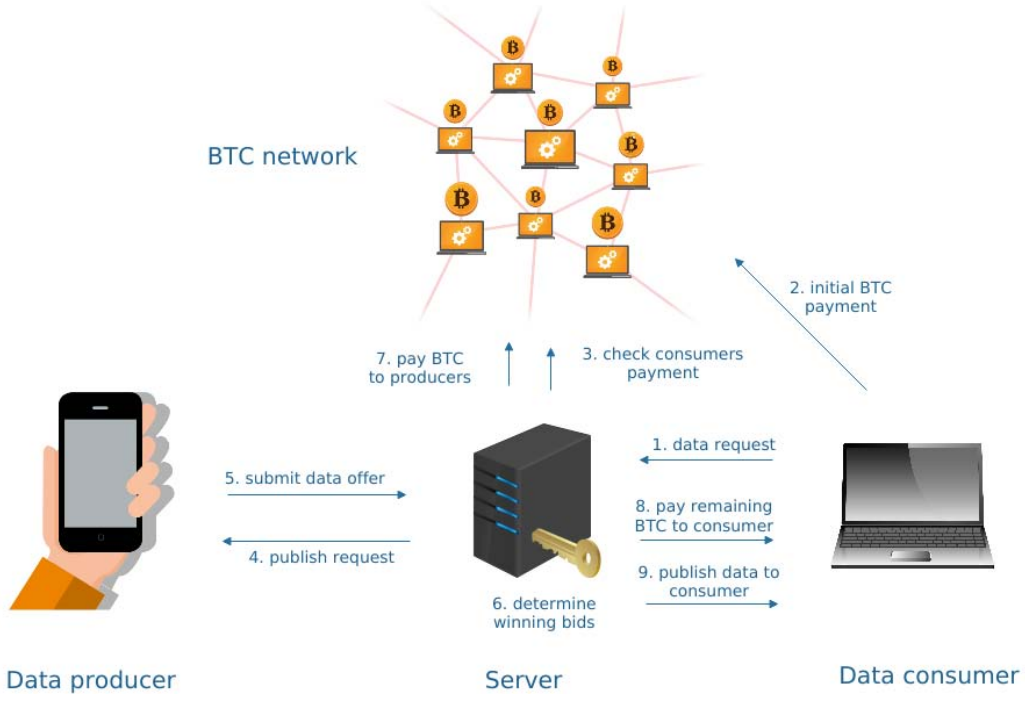
Ο RSA παρέχει μερικά πλεονεκτήματα τα οποία βοήθησαν στην υλοποίηση πιο ασφαλών και ευκολότερα διαχειρίσιμων συναλλαγών. Τα πλεονεκτήματα αυτά περιλαμβάνουν:

- Απλοποίηση του προβλήματος της διαχείρισης κλειδιών: στις συμμετρική κρυπτογραφία ο αριθμός των κλειδιών που απαιτείται για την επικοινωνία n οντοτήτων σε ένα κρυπτοσύστημα είναι ανάλογος του n^2. Στην ασύμμετρη κρυπτογραφία όμως κάθε χρήστης χρειάζεται δύο κλειδιά, έτσι ο απαιτούμενος αριθμός κλειδιών είναι απλά 2n. Άρα είναι κατανοητό ότι σε ένα τέτοιο σύστημα δημοσίου κλειδιού η σχέση που συνδέει τον αριθμό των χρηστών με τον αριθμό των κλειδιών είναι γραμμική. Για αυτό το λόγο η διαχείριση των κλειδιών είναι εύκολη ακόμα και όταν ο αριθμός των χρηστών είναι αρκετά μεγάλος.

- Ενισχυμένη ασφάλεια των συναλλαγών: κάθε χρήστης παράγει μόνος του για δική του χρήση ένα ζεύγος κλειδιών. Τα ιδιωτικό κλειδί θα πρέπει να μένει μυστικό και κρυφό από οποιαδήποτε μη εξουσιοδοτημένη οντότητα εξαλείφοντας έτσι όχι μόνο το πρόβλημα της μεταφοράς του αλλά και την απαίτηση για την εγκατάσταση ενός ασφαλούς διαύλου επικοινωνίας. Το δημόσιο κλειδί από την άλλη είναι ευρέως διαθέσιμο Και άρα μπορεί να μεταφερθεί με οποιοδήποτε προσφερόμενη μέθοδο σε ένα δίκτυο χωρίς να τίθεται θέμα για διατήρηση της μυστικότητας του. [51][62]

7. Αρχιτεκτονική

7.1. Γενική άποψη του συστήματος



Εικόνα 28 Γενική αρχιτεκτονική της πλατφόρμας

Ρόλοι

- Data consumer : Ενδιαφερόμενος για πληροφορία
- Data producer : Πάροχος πληροφορίας.
- Server : Υπεύθυνος για τον έλεγχο και για την σωστή λειτουργία όλων των ενεργειών της πλατφόρμας.

Ενέργειες

1. Data request : Το αίτημα αυτό περιέχει όλες τις απαραίτητες παραμέτρους όπως το δημόσιο κλειδί, το σημείο ενδιαφέροντος, τη ζητούμενη ποσότητα καθώς και την επιθυμητή τιμή που διαθέτει για την αγορά της πληροφορίας.
2. Initial BTC payment: Πληρωμή στην πλατφόρμα για κάθε αίτημα (data request)
3. Check consumer payment: Έλεγχος των πληρωμών.
4. Publish request : Δημοσίευση αιτήματος (data request)

5. Submit data offer : Δήλωση προσφοράς μαζί με την κρυπτογραφημένη πληροφορία.
6. Determine winning bids : Διενέργεια πλειστηριασμού και ανάδειξη των νικητών.
7. Pay BTC to producers : Πληρωμή των παρόχων πληροφορίας (data producers)
8. Pay remaining BTC to consumer: Επιστροφή του υπόλοιπου ποσού στους ενδιαφερόμενους πληροφορίας (data consumer)
9. Publish data to consumer : Δημοσίευση των πληροφοριών στους ενδιαφερόμενους (data consumer)

7.1.1. **Περιγραφή**

Στο παραπάνω Σχήμα (**Εικόνα 28**) απεικονίζεται η γενική εικόνα του συστήματος. Αποτελείται από τρία μέρη:

- Αυτό που αφορά τους Data Consumers που έχουν ανάγκη από δεδομένα και είναι διατεθειμένοι να πληρώσουν για να τα αποκτήσουν.
- Αυτό που αφορά τους Data producers οι οποίοι με την σειρά τους διαθέτουν τα δεδομένα τους κρυπτογραφημένα με σκοπό την πώλησή τους σε κάποιον ενδιαφερόμενο.
- Τέλος το ενδιάμεσο τμήμα διαχείρισης, τον Server που είναι υπεύθυνος για τον έλεγχο και την σωστή λειτουργία των περισσότερων λειτουργιών όλου του συστήματος.

Τυπικό σενάριο λειτουργίας της πλατφόρμας:

1. Ο καταναλωτής δεδομένων (data consumer) υποβάλει ένα αίτημα (data request) στο σύστημα. Το αίτημα αρχικά τίθεται σε κατάσταση αναμονής μέχρι να γίνει η απαραίτητη διαδικασία πληρωμής.
2. Στην συνέχεια πραγματοποιεί μια πληρωμή με την χρήση bitcoin στην διεύθυνση της πλατφόρμας. Το ποσό της πληρωμής ισούται με το μέγιστο

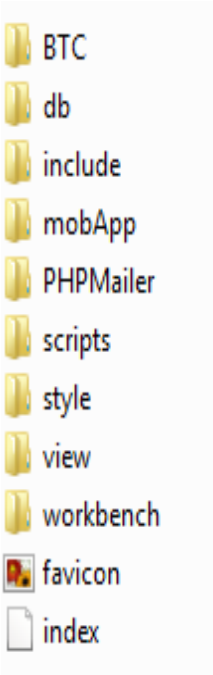
δυνατό ποσό που θα χρειαστεί να δαπανήσει για την αγορά των δεδομένων.

3. Η πλατφόρμα ελέγχει αν έχει γίνει η πληρωμή και όταν αυτό συμβεί τότε το αίτημα δεδομένων καθίσταται ενεργό.
4. Το ενεργό αίτημα δεδομένων δημοσιοποιείται προς τους εν δυνάμει παραγωγούς δεδομένων (data producers).
5. Κάθε χρήστης data producer που ενδιαφέρεται να προσφέρει τα δεδομένα του προς εκπλήρωση στου συγκεκριμένου αιτήματος, αποστέλλει μία σχετική προσφορά (data response). Σε αυτή ορίζει την τιμή στην οποία είναι διατεθειμένος να προσφέρει τα δεδομένα (είναι μικρότερη ή ίση από αυτή όπου ο data consumer έχει ορίσει κατά το data request). Τα δεδομένα του data producer κρυπτογραφούνται με το δημόσιο κλειδί του data consumer και αποστέλλονται στην πλατφόρμα μαζί με το ποσό της προσφοράς.
6. Όταν ολοκληρωθεί χρονικά το αίτημα τότε αρχίζει η διαδικασία της δημοπρασίας με νικητές τους data producers με τις μικρότερες τιμές προσφοράς.
7. Εκτελούνται οι πληρωμές στους επιλεγμένους data producers. Οι υπόλοιπες προσφορές απορρίπτονται από το σύστημα.
8. Επιστρέφεται στον data consumer το υπόλοιπο ποσό σε bitcoins που δεν έχει χρησιμοποιηθεί από την αρχική συναλλαγή του βήματος 2.
9. Τέλος η πλατφόρμα διαθέτει τα κρυπτογραφημένα δεδομένα στο data consumer.Ο data consumer αφού κατεβάσει τα δεδομένα θα πραγματοποιήσει την αποκρυπτογράφηση με την χρήση του ιδιωτικού κλειδιού, τοπικά και εκτός σύνδεσης(offline).

7.2. Ανάλυση λειτουργιών της πλατφόρμας

7.2.1.1. Γενική δομή της Web εφαρμογής

Η πρώτη εφαρμογή της παρούσας εργασίας είναι ο Web Server ο οποίος αποτελεί ταυτόχρονα και το Web Interface των χρηστών. Στην ακόλουθη εικόνα απεικονίζεται η δομή του κύριου directory της εφαρμογής.



Εικόνα 29 Κύρια δομή του directory της web εφαρμογής

Στο directory BTC περιλαμβάνονται τα αρχεία που έχουν σχέση με τις πληρωμές του συστήματος. Η παρούσα εργασία χρησιμοποιεί το API της Uphold και πιο συγκεκριμένα το Sandbox Uphold.

Το directory db περιέχει όλη την βάση του συστήματος που έχει δημιουργηθεί και αποθηκεύεται και εκεί ως αντίγραφο ασφαλείας αν τακτά χρονικά διαστήματα.

Το directory include είναι αυτός που περιέχει αρχεία με τις σημαντικότερες και περισσότερες λειτουργίες του συστήματος μας όπως το αρχείο fg_membersite.php με τις συναρτήσεις για την δημιουργία νέου χρήστη , την δημιουργία πινάκων και την εισαγωγή δεδομένων στους πίνακες.

Το directory MobApp περιέχει το Framework Slim που χρησιμοποιήθηκε για την επικοινωνία της ιστοσελίδας με το λειτουργικό σύστημα Android.

Το directory PHPMailer περιέχει αρχεία για την αποστολή email.

Το directory script περιέχει σε μορφή JS αρχεία όπως το αρχείο map.js.

Το directory style περιέχει αρχεία σε μορφή css για την διαμόρφωση της ιστοσελίδας.

Το directory view περιλαμβάνει όλες τις σελίδες που μπορεί να επισκεφτεί ένας χρήστης του συστήματος όπως login.php , register.php , history.php αλλά και κάποιες που δημιουργήθηκαν για έλεγχο όπως οι σελίδες testmap.php και testtime.php.

Στο directory workbench υπάρχει το Schema της βάσης διαθέσιμο μέσω του προγράμματος MySQL Workbench.

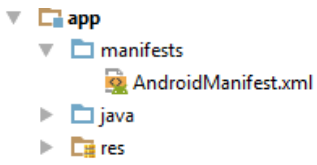
Τέλος το αρχείο favicon.png είναι μια εικόνα που χρησιμοποιείται ως λογότυπο στην ιστοσελίδα και το αρχείο index.html αποτελεί την αρχική σελίδα.

7.2.1.2. Γενική δομή της Android εφαρμογής

Με την κατασκευή μιας Native Android εφαρμογής πάνω στο Android SDK παρουσιάζονται στο χρήστη κάποια από τα δεδομένα της βάσης με όλες τις πληροφορίες που απαιτούνται χρησιμοποιώντας τα Android GUI Elements.

- Δίνεται η δυνατότητα στο χρήστη της εφαρμογής να κάνει αυτόματη εγγραφή ή είσοδο σε περίπτωση που έχει κάνει ήδη την εγγραφή του.
- Προβάλλεται χάρτης μέσω Google maps, όπου ο χρήστης επιλέγει σημεία ενδιαφέροντος
- Δίνεται η δυνατότητα δήλωσης προσφοράς για κάθε σημείο και αποστολή των δεδομένων από τους αισθητήρες.
- Κρυπτογραφούνται τα δεδομένα και αποστέλλονται προς τον server της πλατφόρμας.

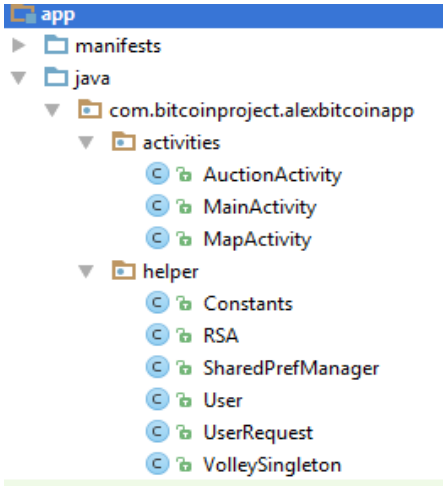
Η γενική δομή του project είναι η τυπική δομή ενός Android project.



Εικόνα 30 Βασική δομή Android Εφαρμογής

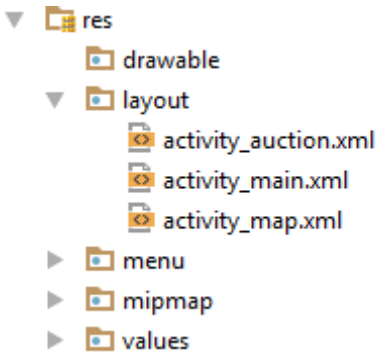
Στο root directory βρίσκεται το directory manifest με το σημαντικό αρχείο AndroidManifest.xml στο οποίο δηλώνονται τα Activities που χρησιμοποιεί η εφαρμογή καθώς και τις άδεις (Permissions) που χρειάζεται η εφαρμογή για να λειτουργήσει.

Στην συνέχεια κάτω από το directory manifest βρίσκεται το directory java και το πρώτο directory που αποτελεί το Company Domain που δηλώθηκε όταν δημιουργήθηκε το project και έχει την εξής μορφή com.example.application.name. Η δομή του πραγματικού κώδικα βρίσκεται στο directory com.bitcoinproject.alexbitcoinapp.



Εικόνα 31 Company Domain direcroty

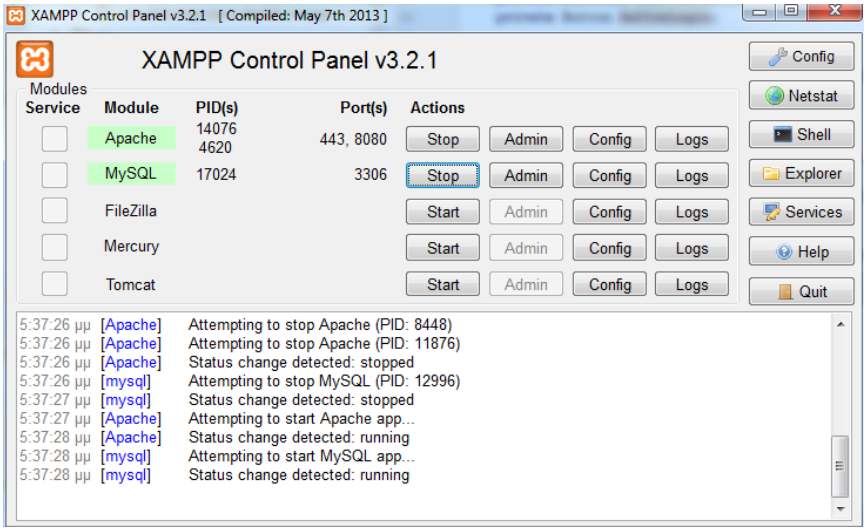
Το directory activities περιλαμβάνει όλες τις Java Classes που αποτελούν τις ίδιες τις Activitites, τους controllers δηλαδή του project. Το directory helper περιέχει βοηθητικές Classes για την χρήση του project.



Εικόνα 32 directory res

Έπειτα μέσα στο directory res βρίσκονται κάποια άλλα directories. Ενδεικτικά στον drawable είναι τοποθετημένα όλα τα εικονίδια και οι φωτογραφίες που χρησιμοποιούνται για τον σχεδιασμό του UI της εφαρμογής. Το directory layout περιλαμβάνει όλα τα xml αρχεία στα οποία σχεδιάζονται τα διάφορα views και components αυτών. Γενικά περιλαμβάνεται οτιδήποτε αφορά την σχεδίαση αλλά και την εμφάνιση ενός project.

7.2.2. Εκκίνηση εφαρμογής – Πρόσβαση στην Web εφαρμογή



Εικόνα 33 Πλατφόρμα XAMPP

Για να ξεκινήσει ο Sever και για να μπορούμε να έχουμε πρόσβαση στην εφαρμογή πρέπει αρχικά να εκκινήσουμε τον Apache και την MySQL από το XAMPP πατώντας το κουμπί Start στο κάθε προαναφερθέν component.

Στην συνέχεια ο χρήστης θα ανοίξει τον browser της επιλογής του και θα χρησιμοποιήσει τον σύνδεσμο http://localhost:8080/ για να μπει στην εφαρμογή του τοπικού υπολογιστή που τρέχει και το XAMPP.

Εναλλακτικά για είσοδο στην εφαρμογή από τρίτο υπολογιστή αντικαθιστάτε το localhost με την IP διεύθυνση στον οποίο τρέχει το XAMPP ή με κάποιο DNS name αν υπάρχει Πχ. http://192.168.1.3:8080.

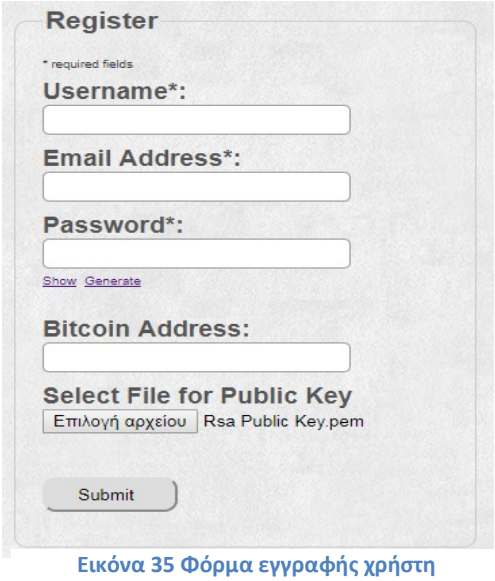
Επίσης υπάρχει η δυνατότητα πρόσβασης μέσω διαδικτύου καθώς βρίσκεται πλέον και σε Web Server φιλοξενίας ιστοσελίδων στην διεύθυνση http://bitcoinproject.website .

Εικόνα 34 Αρχική σελίδα

Η πρώτη σελίδα που εμφανίζεται στην web εφαρμογή είναι η αρχική σελίδα στην οποία ο χρήστης μπορεί να κάνει είτε είσοδο (login) ή στην περίπτωση που δεν έχει κάποιο ενεργό λογαριασμό να κάνει εγγραφή (register) χρήστη δίνοντας κάποια στοιχεία.

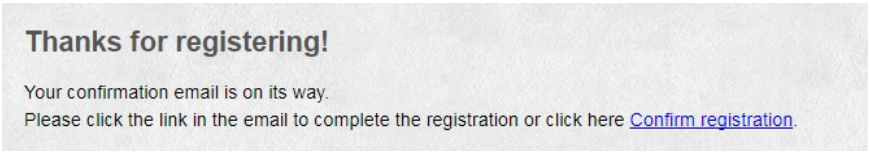
7.2.3. Register

Στην περίπτωση που κάνει εγγραφή ο χρήστης η σελίδα register.php έχει ως εξής.



Εικόνα 35 Φόρμα εγγραφής χρήστη

Πατώντας το κουμπί Submit θα μεταφερθεί αυτόματα στην παρακάτω σελίδα όπου ο χρήστης ενημερώνεται πως του έχει σταλεί στο email του κωδικός επιβεβαίωσης της εγγραφής του.



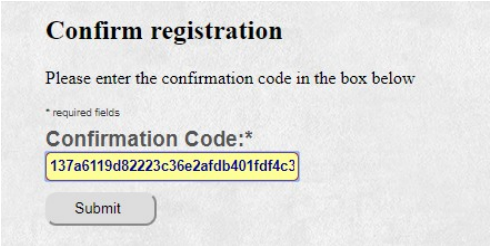
Εικόνα 36 Ενημέρωση αποστολής κωδικού επιβεβαίωσης στο email του χρήστη

Το μήνυμα που λαμβάνει ο χρήστης έχει ως εξής.

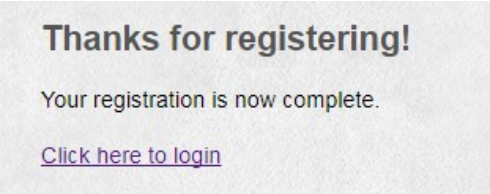
Hello pest
Thanks for your registration with Bitcoin Project
Please click the link below to confirm your registration.
http://192.168.1.3:8080/complete/source/confirmreg.php?code=137a6119d82223c36e2afdb401fdf4c3
Regards,
Loukis
Bitcoin Project

Εικόνα 37 Μήνυμα προς τον χρήστη

Στην σελίδα επιβεβαίωσης εισάγεται ο κωδικός που στάλθηκε στο προηγούμενο στάδιο.



Εικόνα 38 Σελίδα επιβεβαίωσης του κωδικού

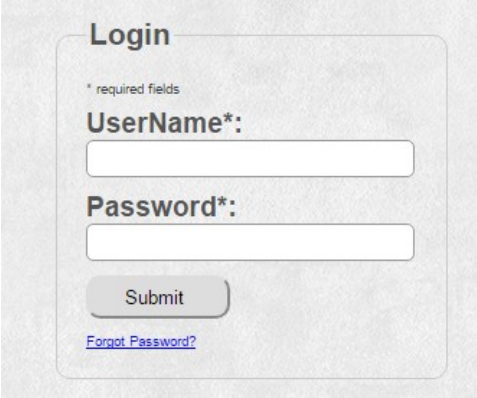


Εικόνα 39 Σελίδα ολοκλήρωσης της εγγραφής

Η ενεργοποίηση του λογαριασμού έχει ολοκληρωθεί.

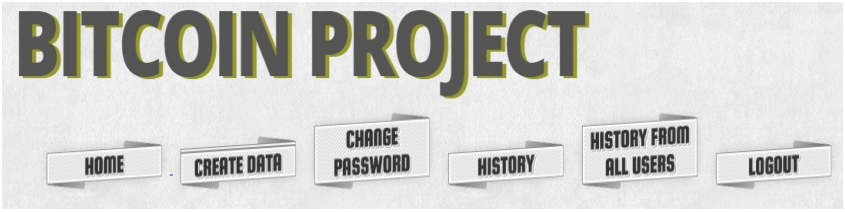
7.2.4. Login

Στην περίπτωση που ο χρήστης διαθέτει ενεργό λογαριασμό και πατήσει στο σύνδεσμο login θα του ζητηθούν τα παρακάτω στοιχεία.



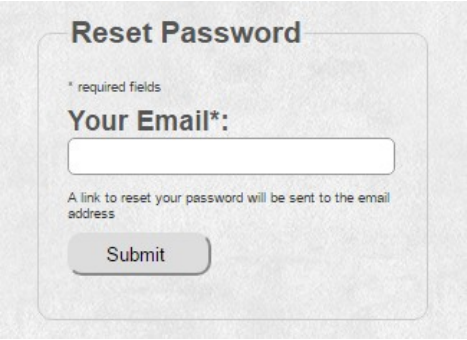
Εικόνα 40 Φόρμα εισόδου χρήστη

Σε αυτό το σημείο όταν ο χρήστης δώσει τα στοιχεία που απαιτούνται πατώντας το κουμπί Submit θα μεταφερθεί στην παρακάτω σελίδα.



Εικόνα 41 Αρχική σελίδα μετά από είσοδο του χρήστη

Στην περίπτωση που ο χρήστης δεν έχει στην διάθεση του τον κωδικό πρόσβασης πατώντας το κατάλληλο σύνδεσμο Forgot Password θα μεταφερθεί στη σελίδα για επαναφορά του κωδικού του με αποστολή στο email του χρήστη.



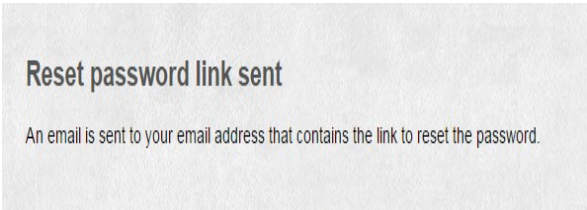
Εικόνα 42 Σελίδα επαναφοράς κωδικού πρόσβασης

Έπειτα αφού συμπληρωθεί το κατάλληλο email γίνεται αποστολή μηνύματος ότι έχει ζητηθεί αλλαγή κωδικού με το κατάλληλο link ώστε να γίνει η αλλαγή.

Hello alex There was a request to reset your password at Bitcoin Project Please click the link below to complete the request: http://192.168.1.3:8080/complete/source /resetpwd.php?email=aloukis%40hotmail.com& code=b96d869971 Regards, Webmaster Bitcoin Project

Εικόνα 43 Email για την αλλαγή κωδικού πρόσβασης

Όταν ο χρήστης πατήσει στον σύνδεσμο εμφανίζεται η σελίδα ενημερώνοντας τον χρήστη για το μήνυμα που θα λάβει για την αλλαγή του κωδικού πρόσβασης.



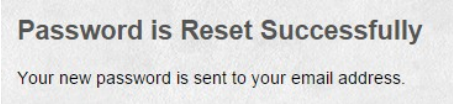
Εικόνα 44 Email με το κατάλληλο περιεχόμενο για την αλλαγή κωδικού πρόσβασης

Ο χρήστης λαμβάνει το email με την πληροφορία που απαιτείτε για την ολοκλήρωση της αλλαγής κωδικού πρόσβασης.

Hello alex Your password is reset successfully. Here is your updated login: username:alex password:8246471e98 Login here: http://192.168.1.3:8080/complete/source/login.php Regards, Webmaster Bitcoin Project

Εικόνα 45 Μήνυμα επιτυχημένης αλλαγής κωδικού πρόσβασης

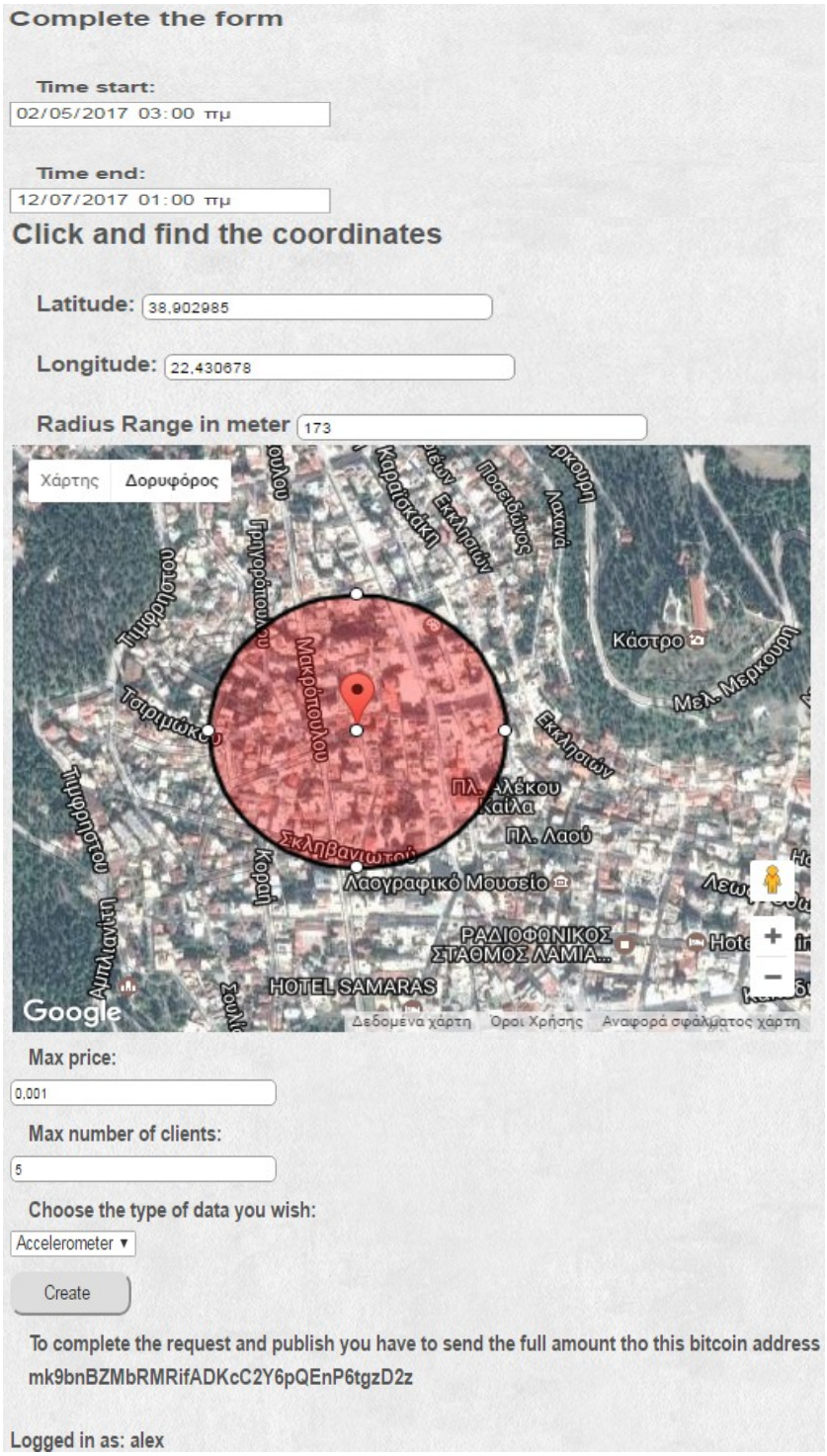
Όταν ο χρήστης εισάγει στον φυλλομετρητή του τον σύνδεσμο που έλαβε με email θα του εμφανιστεί η σελίδα για την επιτυχημένη αλλαγή του κωδικού πρόσβασης.



Εικόνα 46 Σελίδα επιτυχημένης αλλαγής κωδικού πρόσβασης

7.2.5. Create Data

Στην συνέχεια όταν ο χρήστης πατήσει το κατάλληλο σύνδεσμο δημιουργίας ερωτήματος για δεδομένα (Create Data) θα εμφανιστεί μία φόρμα προς συμπλήρωση.



Εικόνα 47 Δημιουργία ερωτήματος

Ο χρήστης καλείτε να συμπληρώσει κάποια πεδία για την δημιουργία του ερωτήματος.

- Ακριβής χρόνο – ημερομηνία εκκίνησης και τερματισμού.
- Γεωγραφικό μήκος και πλάτος καθώς και το εύρος της απόστασης που επιθυμεί συμπληρώνοντας τα κατάλληλα πεδία ή κάνοντας χρήση του χάρτη.
- Μέγιστη χρηματική τιμή διάθεσης ανά πληροφορία.
- Μέγιστος αριθμός χρηστών προς διάθεση.
- Τύπος δεδομένων προς αναζήτηση.

7.2.6. History και History from all Users

Από το μενού ο χρήστης μπορεί να πλοηγηθεί στις σελίδες HISTORY και HISTORY FROM ALL USERS

Εικόνα 48 Μέρος από το Menu της Σελίδας (HISTORY ,HISTORY FROM ALL USERS)

Στις σελίδες αυτές εμφανίζονται τα αιτήματα του χρήστη αλλά και συνολικά όλων των χρηστών

Show 10 entries Search:

Number of request	Time start	Time end	Location - Range	Type of data	Price	Status
2	2017-02-23 01:00:00	2018-01-01 01:00:00	38.902771 22.435799 -- 50	location	0.0001	1
3	2017-02-27 01:00:00	2017-03-08 01:00:00	38.901703 22.432709 -- 50	location	0.0003	2
4	2017-02-01 01:00:00	2017-02-17 01:00:00	38.900569 22.434254 -- 50	location	0.0001	2
5	2017-02-21 04:02:00	2017-03-16 12:58:00	38.902387 22.434061 -- 86	gyroscope	0.0001	2
6	2017-02-01 01:00:00	2017-02-24 02:00:00	38.898229 22.433846 -- 50	location	0.0001	2
7	2017-02-24 03:00:00	2017-02-21 19:33:00	38.901287 22.433395 -- 88	location	0.0007	2

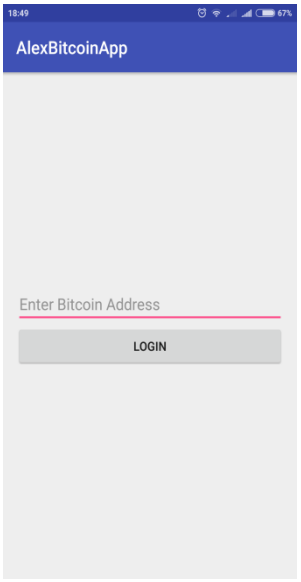
Εικόνα 49 Σελίδα με το ιστορικό όλων των χρηστών

7.2.7. Android εφαρμογή

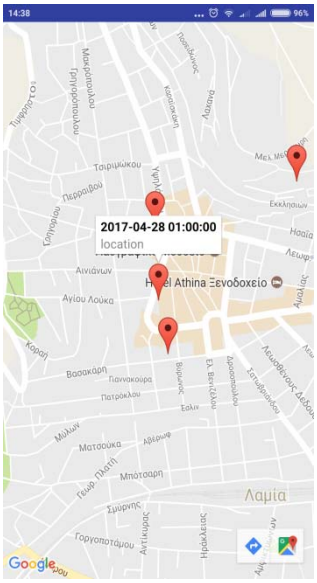
Οι χρήστες οι οποίοι θα διαθέσουν τα δεδομένα τους θα γίνει μέσω της Android εφαρμογής.

Η εφαρμογή αποτελείται από 3 κύριες οθόνες.

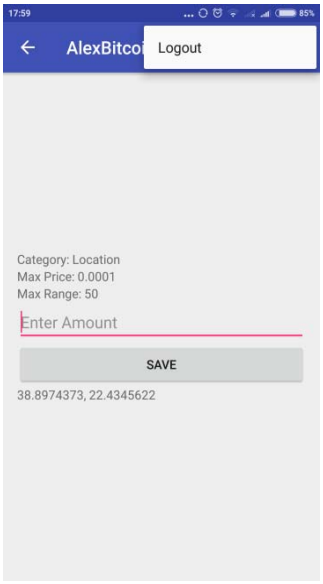
- Η αρχική στην οποία δηλώνεται η διεύθυνση Bitcoin που θα γίνει η πληρωμή.
- Στην συνέχεια μετά την είσοδο μεταφέρεται στην εικόνα όπου φαίνονται τα ενεργά χρονικά αιτήματα πάνω στον χάρτη.
- Τέλος όταν ο χρήστης επιλέξει κάποιο αίτημα θα μεταφερθεί στην παρακάτω εικόνα όπου έχει την δυνατότητα να βάλει το ποσό ως αντίτιμο που επιθυμεί να λάβει για να παρέχει τα δεδομένα του. Στην προκειμένη περίπτωση πρόκειται να διαθέσει δεδομένα τοποθεσίας τα οποία θα κρυπτογραφηθούν και στην συνέχεια θα αποθηκευτούν στην βάση . Επίσης υπάρχει και η δυνατότητα αποσύνδεσης του χρήστη μέσω του κουμπιού Logout



Εικόνα 51 Αρχική οθόνη της εφαρμογής σε Android όπου δηλώνεται η διεύθυνση Bitcoin



Εικόνα 52 Οθόνη για την αναζήτηση αιτήματος μέσω χάρτη



Εικόνα 50 Οθόνη δήλωσης προσφοράς για συγκεκριμένο αίτημα

Για την υλοποίηση της Android εφαρμογής έχει γίνει χρήση κάποιων βοηθητικών κλάσεων όπως αναφέρθηκε στην αρχή του κεφαλαίου. Αυτές είναι :

- Η κλάση Constants όπου δηλώνονται κάποιες σταθερές.
- Η κλάση RSA η οποία κάνει την κρυπτογράφηση.
- Η κλάση SharedPrefManager για την διαχείριση εισόδου, εξόδου του χρήστη από την εφαρμογή.

- Η κλάση User όπου δηλώνονται τα χαρακτηριστικά του κάθε χρήστη
- Η κλάση UserRequest όπου δηλώνονται οι μεταβλητές για τον πίνακα Request.
- Η κλάση VolleySingleton η οποία χρησιμοποιεί την Volley βιβλιοθήκη της HTTP για την επικοινωνία με το δίκτυο.

7.2.8. Πρόσβαση στην Android εφαρμογή.

Όταν ο χρήστης μπει για πρώτη φορά στην εφαρμογή (**Εικόνα 51**) θα του ζητηθεί να δηλώσει μια Bitcoin διεύθυνση για να πληρωθεί. Στην συνέχεια ο χρήστης θα μεταφερθεί στον χάρτη (**Εικόνα 52**) με όλες τις ενεργές τοποθεσίες όπου μπορεί να πάρει μέρος σε όποια επιθυμεί. Πατώντας την κάθε πινέζα εμφανίζεται η ημερομηνία που τελειώνει η προσφορά καθώς και το είδος αυτής. Κάνοντας είσοδο για συμμετοχή σε ένα αίτημα εμφανίζεται η (**Εικόνα 50**). Πέρα από το είδος που φαίνεται και σε προηγούμενη φάση απεικονίζεται και η μέγιστη τιμή που πρόκειται να διατεθεί αλλά και σε τι εύρος μπορεί να βρίσκεται ο χρήστης της κινητής εφαρμογής. Στο κάτω μέρος και πριν το κουμπί αποθήκευσης βρίσκονται τα δεδομένα που θα διατεθούν και στο συγκεκριμένο είναι γεωγραφικό μήκος και πλάτος του χρήστη. Αυτή είναι η πληροφορία η οποία θα κρυπτογραφηθεί και θα αποθηκευτεί στην βάση.

7.2.9. Sandbox uphold

Για την υλοποίηση του project χρησιμοποιήθηκε το Api του Sandbox Uphold.

Δημιουργήθηκε ένας λογαριασμός που λειτουργεί ως διαχειριστής του συστήματος Admin , ένας λογαριασμός ως Data consumer αυτός δηλαδή που διαθέτει ένα ποσό στον διαχειριστή ώστε να λάβει κάποια δεδομένα και τέλος ο Data producer που είναι ο χρήστης της κινητής συσκευής που θα διαθέσει τα δεδομένα και στο τέλος θα λάβει την αμοιβή του σε Bitcoins. Κάθε Requester που ζητάει δεδομένα κάνει και την πληρωμή στον διαχειριστή. Η πληρωμή εξακριβώνεται αν έχει γίνει μέσω ενός μοναδικού Reservechain Transaction ID.

7.2.10. Σελίδα πληρωμής μέσω Sandbox uphold

Id	Time Start	Time End	Latitude	Longitude	Location	Max Price	User Id	Status	Num Clients	Category	Winner
3	2017-02-27 01:00:00	2017-03-08 01:00:00	38.901703	22.432709	50	0.0003	1	2	4	location	Finalize Winner
4	2017-02-01 01:00:00	2017-02-17 01:00:00	38.900569	22.434254	50	0.0001	1	2	1	location	Finalize Winner
5	2017-02-21 04:02:00	2017-03-16 12:58:00	38.902387	22.434061	86	0.0001	1	2	1	gyroscope	Finalize Winner
6	2017-02-01 01:00:00	2017-02-24 02:00:00	38.898229	22.433846	50	0.0001	1	2	1	location	Finalize Winner

Εικόνα 53 Σελίδα πληρωμής αιτημάτων

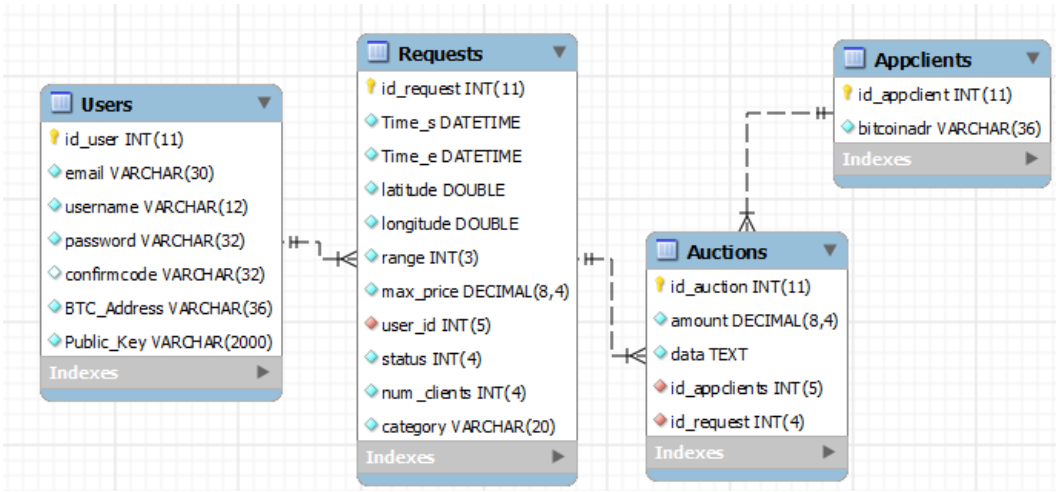
Στην σελίδα πληρωμών εμφανίζονται όλα τα αιτήματα που έχουν ολοκληρωθεί αλλά δεν έχει γίνει η πληρωμή τους από τον Διαχειριστή προς τους χρήστες κινητών συσκευών, Data producers. Στη κατάσταση αυτή (**Εικόνα 53**) βρίσκονται τα αιτήματα που έχουν πληρωθεί από τους Data consumers και έχουν ολοκληρωθεί χρονικά. Όταν ολοκληρωθεί η πληρωμή στην συνέχεια δημοσιοποιείται το αίτημα και είναι ορατό σε όλους τους χρήστες. Η πληρωμή του κάθε Data producer από τον διαχειριστή πραγματοποιείται όταν ολοκληρωθεί χρονικά το αίτημα και ο χρήστης έχει καταβάλει ποσό μικρότερο ή ίσο από αυτό που διαθέτει ο Data consumer. Επίσης προϋποθέτει ότι έχει δηλώσει καλύτερη πρόσφορα από άλλους χρήστες αλλά και την στιγμή που την κάνει βρίσκεται στο κατάλληλο σημείο δηλαδή εντός της ακτίνας ενδιαφέροντος.

7.3. Κατασκευή εφαρμογών

7.3.1. Βάση δεδομένων

7.3.1.1. Περιγραφή

Η βάση δεδομένων της παρούσας εργασίας είναι σχεδιασμένη σε MySQL και MySQL Workbench και τρέχει πάνω στο MySQL Server του XAMPP.



Εικόνα 54 Δομή Βάσης Δεδομένων

Η βάση δεδομένων αποτελείται από 4 πίνακες:

- Users
- Requests
- Auctions
- Appclients

7.3.1.2. Πίνακας Users

Στον πίνακα Users αποθηκεύονται οι χρήστες που κάνουν εγγραφή στο σύστημα και θέλουν να αγοράσουν δεδομένα έναντι κάποιου ποσού (data consumers). Στο Σχήμα (**Εικόνα 54**) απεικονίζονται τα πεδία του πίνακα. Πιο συγκεκριμένα αυτά είναι το πεδίο id_user που είναι Primary Key του πίνακα, τα πεδία email, username και password που χρησιμοποιούνται για την ταυτοποίηση του χρήστη. Το πεδίο confirmcode χρησιμοποιείται για την επιβεβαίωση του χρήστη κατά την εγγραφή του στο σύστημα στέλνοντας το κατάλληλο μήνυμα στο email του

χρήστη, το πεδίο BTC_Address όπου δηλώνεται η bitcoin διεύθυνση του και τέλος το Public_Key το οποίο χρησιμοποιείται για να γίνει η κρυπτογράφηση των δεδομένων.

7.3.1.3. Πίνακας Requests

Στον πίνακα Requests (**Εικόνα 54**) αποθηκεύονται τα αιτήματα των χρηστών που επιθυμούν να λάβουν δεδομένα. Αποτελείται από το πεδίο id_request όπου είναι το Primary Key του πίνακα, τα πεδία Time_s και Time_e όπου δηλώνεται η χρονική διάρκεια του αιτήματος, τα πεδία latitude, longitude και range για τον προσδιορισμό της θέσης κατά γεωγραφικό μήκος, πλάτος καθώς και το εύρος αυτού, το πεδίο max_price δηλώνει την μέγιστη τιμή που πρόκειται να διατεθεί για κάθε μία πληροφορία και το πεδίο num_clients περιγράφει τον αριθμό των χρηστών που θα λάβουν μέρος. Επιπλέον το πεδίο status περιγράφει την κατάσταση στην οποία βρίσκεται ένα αίτημα (ενεργό, μη ενεργό, ολοκληρωμένο και πληρωμένο) και το πεδίο category περιγράφει αντίστοιχα τον τύπο δεδομένων που ζητούνται (accelerometer, location, gyroscope και light). Στο πεδίο αυτό συναντά κανείς τους βασικούς αισθητήρες που βρίσκονται στην πλειονότητα των φορητών συσκευών. Μελλοντικά το εύρος των πιθανών τύπων δεδομένων μπορεί να διευρυνθεί, βάσει των αισθητήρων που θα ενσωματώνουν οι νέες συσκευές. Τέλος το πεδίο user_id αποτελεί τον Foreign Key του πίνακα Users.

7.3.1.4. Πίνακας Appclients

Στον πίνακα Appclients (**Εικόνα 54**) υπάρχουν τα πεδία id_appclients το οποίο είναι το Primary Key του πίνακα και το πεδίο bitcoinadr στο οποίο δηλώνεται η διεύθυνση που θα γίνει η πληρωμή.

7.3.1.5. Πίνακας Auctions

Ο πίνακας Auctions (**Εικόνα 54**) περιλαμβάνει τις προσφορές που κάνουν οι χρήστες data producers από κινητές συσκευές. Τα πεδία είναι το id_auction που αποτελεί το Primary Key του πίνακα, το πεδίο amount, όπου περιέχει το αντίτιμο που

ζητάει ο χρήστης για να παρέχει τα δεδομένα του και το πεδίο data όπου αποθηκεύονται τα κρυπτογραφημένα δεδομένα. Τέλος τα πεδία id_appclients και id_requests αποτελούν Foreign Keys των πινάκων Appclients και Requests αντίστοιχα.

7.3.2. Τεχνική ανάλυση των εφαρμογών

Η εφαρμογή αποτελείται κατά βάση από δύο κομμάτια αρκετά διαφορετικά μεταξύ τους, το mobile κομμάτι και την web πλατφόρμα. Για τη λειτουργία του συστήματος είναι απαραίτητο να υπάρχει αμφίδρομη επικοινωνία μεταξύ αυτών των δύο δομικών μονάδων η επικοινωνία αυτή επιτυγχάνεται μέσω του HTTP πρωτόκολλου.

7.3.2.1. Web εφαρμογή

Για την λειτουργία της κάθε σελίδας έχει παραχθεί κάποιος κώδικας. Ενδεικτικά παρουσιάζεται ο κώδικας για την κάθε περίπτωση.

Για την σελίδα της εγγραφής νέου χρήστη Register (**Εικόνα 35**)

```
require_once("./include/membersite_config.php");

if(isset($_POST['submitted']))
{
   if($fgmembersite->RegisterUser())
   {
        $fgmembersite->RedirectToURL("thank-you.html");
   }
}
```

Εικόνα 55 Κώδικας εγγραφής χρήστη

Συμπεριλαμβάνεται το απαραίτητο αρχείο membersite_config.php στο οποίο δηλώνονται κάποιες σταθερές.

```
<?PHP
require_once("./include/fg_membersite.php");

$fgmembersite = new FGMembersite();

//Provide your site name here
$fgmembersite->SetWebsiteName('Bitcoin Project');

//Provide the email address where you want to get notifications
$fgmembersite->SetAdminEmail('aloukis@hotmail.com');

//Provide your database login details here:
//hostname, user name, password, database name and table name
//note that the script will create the table (for example, fgusers in this case)
//by itself on submitting register.php for the first time
$fgmembersite->InitDB(/*hostname*/'localhost',
                      /*username*/'root',
                      /*password*/'12345',
                      /*database name*/'bitcoinproject',
                      /*table name*/'Users');
```

Εικόνα 56 περιεχόμενο αρχείου membersite_config.php

Όπως διακρίνεται γίνεται χρήση του αρχείου fg_memebersite.php όπου περιέχει τις περισσότερες συναρτήσεις που χρησιμοποιούνται στην Web εφαρμογή.

Για την εγγραφή γίνεται χρήση των παρακάτω συναρτήσεων.

```php
function RegisterUser()
{
    if(!isset($_POST['submitted']))
    {
       return false;
    }

    $formvars = array();

    if(!$this->ValidateRegistrationSubmission())
    {
        return false;
    }

    $this->CollectRegistrationSubmission($formvars);

    if(!$this->SaveToDatabase($formvars))
    {
        return false;
    }

    if(!$this->SendUserConfirmationEmail($formvars))
    {
        return false;
    }

    $this->SendAdminIntimationEmail($formvars);

    return true;
}
```

Εικόνα 58 Συναρτήσεις εγγραφής

Η συνάρτηση ValidateRegirtationSubmission χρησιμοποιείτε για να ελέγξει την εγκυρότητα των δεδομένων πριν εισαχθούν στην βάση.

```php
function ValidateRegistrationSubmission()
{
    //This is a hidden input field. Humans won't fill this field.
    if(!empty($_POST[$this->GetSpamTrapInputName()]) )
    {
        //The proper error is not given intentionally
        $this->HandleError("Automated submission prevention: case 2 failed");
        return false;
    }

    $validator = new FormValidator();
    $validator->addValidation("username","req","Please fill in User Name");
    $validator->addValidation("password","req","Please fill in Password");
    $validator->addValidation("email","email","The input for Email should be a valid email value");
    $validator->addValidation("email","req","Please fill in Email");
    $validator->addValidation("BTC_Adrdress","req","Please fill in Bitcoin Adrdress");
    $validator->addValidation("Public_Key","req","Please fill in Public Key");

    if(!$validator->ValidateForm())
    {
        $error='';
        $error_hash = $validator->GetErrors();
        foreach($error_hash as $inpname => $inp_err)
        {
            $error .= $inpname.':'.$inp_err."\n";
        }
        $this->HandleError($error);
        return false;
    }
    return true;
}
```

Εικόνα 57 Συνάρτηση Validate Registration

Η συνάρτηση CollectRegistrationSubmission αναλαμβάνει να στείλει τα δεδομένα στην βάση.

```
function CollectRegistrationSubmission(&$formvars)
{
    $formvars['email'] = $this->Sanitize($_POST['email']);
    $formvars['username'] = $this->Sanitize($_POST['username']);
    $formvars['password'] = $this->Sanitize($_POST['password']);
    $formvars['BTC_Adrdress'] = $this->Sanitize($_POST['BTC_Adrdress']);
    $formvars['Public_Key'] = $this->Sanitize($_POST['Public_Key']);

}
```

Εικόνα 59 Συνάρτηση Collect Registration

Η συνάρτηση SaveToDatabase αφού κάνει τους απαραίτητος ελέγχους εκτελεί την εισαγωγή των δεδομένων στην βάση.

```
function SaveToDatabase(&$formvars)
{
    if(!$this->DBLogin())
    {
        $this->HandleError("Database login failed!");
        return false;
    }
    if(!$this->Ensuretable())
    {
        return false;
    }
    if(!$this->IsFieldUnique($formvars,'email'))
    {
        $this->HandleError("This email is already registered");
        return false;
    }

    if(!$this->IsFieldUnique($formvars,'username'))
    {
        $this->HandleError("This UserName is already used. Please try another username");
        return false;
    }
    if(!$this->InsertIntoDB($formvars))
    {
        $this->HandleError("Inserting to Database failed!");
        return false;
    }
    return true;
}
```

Εικόνα 60 Συνάρτηση Save to Database

Η συνάρτηση DBLogin δημιουργεί την σύνδεση με την βάση δεδομένων.

```
function DBLogin()
{

    $this->connection = mysql_connect($this->db_host,$this->username,$this->pwd);

    if(!$this->connection)
    {
        $this->HandleDBError("Database Login failed! Please make sure that the DB login credentials
            provided are correct");
        return false;
    }
    if(!mysql_select_db($this->database, $this->connection))
    {
        $this->HandleDBError('Failed to select database: '.$this->database.' Please make sure that the
            database name provided is correct');
        return false;
    }
    if(!mysql_query("SET NAMES 'UTF8'",$this->connection))
    {
        $this->HandleDBError('Error setting utf8 encoding');
        return false;
    }
    return true;
}
```

Εικόνα 61 Συνάρτηση DB Login

Η συνάρτηση Ensuretable ελέγχει την ύπαρξη του πίνακα για την αποθήκευση των δεδομένων.

```
function Ensuretable()
{
    $result = mysql_query("SHOW COLUMNS FROM $this->tablename");
    if(!$result || mysql_num_rows($result) <= 0)
    {
        return $this->CreateTable();
    }
    return true;
}
```

Εικόνα 62 Συνάρτηση Ensure Table

Στην περίπτωση που δεν υπάρχει ο πίνακας τότε τρέχει η συνάρτηση CreateTable για την δημιουργία του πίνακα.

```
function CreateTable()
{
    $qry = "Create Table $this->tablename (".
            "id_user INT NOT NULL AUTO_INCREMENT ,".
            "email VARCHAR( 30 ) NOT NULL ,".
            "username VARCHAR( 12 ) NOT NULL ,".
            "password VARCHAR( 32 ) NOT NULL ,".
            "confirmcode VARCHAR(32) ,".
            "BTC_Adrdress VARCHAR( 36 ) NOT NULL ,".
            "Public_Key VARCHAR( 32 ) NOT NULL ,".
            "PRIMARY KEY ( id_user )".
            ")";

    if(!mysql_query($qry,$this->connection))
    {
        $this->HandleDBError("Error creating the table \nquery was\n $qry");
        return false;
    }
    return true;
}
```

Εικόνα 63 Συνάρτηση Create Table

Στην συνέχεια ελέγχεται το όνομα χρήστη και το email αν χρησιμοποιούνται ήδη στην βάση μέσω της συνάρτηση IsFieldUnique .

```php
function IsFieldUnique($formvars,$fieldname)
{
    $field_val = $this->SanitizeForSQL($formvars[$fieldname]);
    $qry = "select username from $this->tablename where $fieldname='".$field_val."'";
    $result = mysql_query($qry,$this->connection);
    if($result && mysql_num_rows($result) > 0)
    {
        return false;
    }
    return true;
}
```

Εικόνα 64 Συνάρτηση Is field unique

Τέλος έχουμε την συνάρτηση εισαγωγής των δεδομένων στην βάση δεδομένων InsertIntoDB.

```php
function InsertIntoDB(&$formvars)
{
    $insert_query = 'insert into '.$this->tablename.'(
            email,
            username,
            password,
            confirmcode,
            BTC_Adrdress,
            Public_Key
            )
            values
            (
            "' . $this->SanitizeForSQL($formvars['email']) . '",
            "' . $this->SanitizeForSQL($formvars['username']) . '",
            "' . md5($formvars['password']) . '",
            "' . $confirmcode . '",
            "' . $this->SanitizeForSQL($formvars['BTC_Adrdress']) . '",
            "' . $this->SanitizeForSQL($formvars['Public_Key']) . '"
            )';
    if(!mysql_query( $insert_query ,$this->connection))
    {
        $this->HandleDBError("Error inserting data to the table\nquery:$insert_query");
        return false;
    }
    return true;
}
```

Εικόνα 65 Συνάρτηση Insert to DB

Ενδεικτικός κώδικας που εμπεριέχεται στην σελίδα login.php (**Εικόνα 40**)

```php
<?PHP
require_once("./include/membersite_config.php");

if(isset($_POST['submitted']))
{
   if($fgmembersite->Login())
   {
        $fgmembersite->RedirectToURL("login-home.php");
   }
}
```

Εικόνα 66 Κώδικας εισόδου χρήστη

Συνάρτηση Login για την διαδικασία εισόδου του χρήστη

```
function Login()
{
    if(empty($_POST['username']))
    {
        $this->HandleError("UserName is empty!");
        return false;
    }

    if(empty($_POST['password']))
    {
        $this->HandleError("Password is empty!");
        return false;
    }

    $username = trim($_POST['username']);
    $password = trim($_POST['password']);

    if(!isset($_SESSION)){ session_start(); }
    if(!$this->CheckLoginInDB($username,$password))
    {
        return false;
    }

    $_SESSION[$this->GetLoginSessionVar()] = $username;

    return true;
}
```

Εικόνα 67 Συνάρτηση Login

Έλεγχος στη βάση δεδομένων με την συνάρτηση CheckLoginDB

```
function CheckLoginInDB($username,$password)
{
    if(!$this->DBLogin())
    {
        $this->HandleError("Database login failed!");
        return false;
    }
    $username = $this->SanitizeForSQL($username);
    $pwdmd5 = md5($password);
    $qry = "Select username, email from $this->tablename where username='$username' and
        password='$pwdmd5' and confirmcode='y'";

    $result = mysql_query($qry,$this->connection);

    if(!$result || mysql_num_rows($result) <= 0)
    {
        $this->HandleError("Error logging in. The username or password does not match");
        return false;
    }

    $row = mysql_fetch_assoc($result);


    $_SESSION['name_of_user']  = $row['username'];
    $_SESSION['email_of_user'] = $row['email'];
    return true;
}
```

Εικόνα 68 Συνάρτηση Check Login In DB

Στην περίπτωση που ο χρήστης δεν έχει στην διάθεση του τον κωδικό πρόσβασης πατώντας το κατάλληλο σύνδεσμο (**Εικόνα 40**) Forgot Password θα μεταφερθεί στη σελίδα για επαναφορά του κωδικού του με αποστολή στο email του χρήστη. (**Εικόνα 42**).

Ο κώδικας που τρέχει ενδεικτικά είναι ο παρακάτω.

```
if(isset($_POST['submitted']))
{
   if($fgmembersite->EmailResetPasswordLink())
   {
        $fgmembersite->RedirectToURL("reset-pwd-link-sent.html");
        exit;
   }
}
```

Εικόνα 69 Κώδικας σελίδας επαναφοράς κωδικού

Γίνονται έλεγχοι μέσω συναρτήσεων ώστε να αποσταλεί ο σύνδεσμος με τον κωδικό πρόσβασης.

```
function EmailResetPasswordLink()
{
    if(empty($_POST['email']))
    {
        $this->HandleError("Email is empty!");
        return false;
    }
    $user_rec = array();
    if(false === $this->GetUserFromEmail($_POST['email'], $user_rec))
    {
        return false;
    }
    if(false === $this->SendResetPasswordLink($user_rec))
    {
        return false;
    }
    return true;
}
```

Εικόνα 70 Συνάρτηση Email Reset Password Link

Αφού γίνει η σύνδεση με την βάση δεδομένων ελέγχεται αν υπάρχει στην βάση και αποθηκεύει το αποτέλεσμα.

```
function GetUserFromEmail($email,&$user_rec)
{
    if(!$this->DBLogin())
    {
        $this->HandleError("Database login failed!");
        return false;
    }
    $email = $this->SanitizeForSQL($email);

    $result = mysql_query("Select * from $this->tablename where email='$
        email'",$this->connection);

    if(!$result || mysql_num_rows($result) <= 0)
    {
        $this->HandleError("There is no user with email: $email");
        return false;
    }
    $user_rec = mysql_fetch_assoc($result);


    return true;
}
```

Εικόνα 71 Συνάρτηση Get User From Email

Με την χρήση της βιβλιοθήκης PHPMailer γίνεται η αποστολή του email με το link για την αλλαγή του κωδικού πρόσβασης.

```
function SendResetPasswordLink($user_rec)
{
    require 'PHPMailer/PHPMailerAutoload.php';
    $mail = new PHPMailer();
    $mail->isSMTP();
    $mail->Host = 'smtp.gmail.com';
    $mail->CharSet = 'utf-8';
    $mail->SMTPAuth = true;
    $mail->Username =          // SMTP username
    $mail->Password = | // SMTP password
    $mail->SMTPSecure = 'tls';
    $mail->Port = 587;
    $mail->isHTML(true);
    $email = $user_rec['email'];
    $name=$user_rec['username'];
    $mail->CharSet = 'utf-8';
    $mail->AddAddress($email,$name);
    $mail->Subject = "Your reset password request at ".$this->sitename;
    $mail->From = $this->GetFromAddress();
    $link = $this->GetAbsoluteURLFolder().
            '/resetpwd.php?email='.
            urlencode($email).'&code='.
            urlencode($this->GetResetPasswordCode($email));
    $mail->Body ="Hello ".$name."\r\n\r\n".
    "There was a request to reset your password at ".$this->sitename."\r\n".
    "Please click the link below to complete the request: \r\n".$link."\r\n"
        .
    "Regards,\r\n".
    "Webmaster\r\n".
    $this->sitename;

    if(!$mail->Send())
    {
        return false;
    }
    return true;
}
```

Εικόνα 72 Συνάρτηση Send Reset Password Link

Δίνεται ένα τυχαίο αλφαριθμητικό ως προσωρινός κωδικός .

```
function GetResetPasswordCode($email)
{
   return substr(md5($email.$this->sitename.$this->rand_key),0,10);
}
```

Εικόνα 73 Συνάρτηση Get Reset Password

Αφού σταλεί το email βάζοντας το στον browser (**Εικόνα 46**) θα τρέξει ο παρακάτω κώδικας

```
$success = false;
if($fgmembersite->ResetPassword())
{
    $success=true;
}
```

Εικόνα 74 Περιεχόμενο σελίδας resetpwd.php

Με βάση το μήνυμα που ελήφθη και το περιεχόμενο του email=aloukis%40hotmail.com&code=b96d869971 γίνεται έλεγχος του περιεχομένου και στέλνετε ο καινούριος κωδικός.

```php
function ResetPassword()
{
    if(empty($_GET['email']))
    {
        $this->HandleError("Email is empty!");
        return false;
    }
    if(empty($_GET['code']))
    {
        $this->HandleError("reset code is empty!");
        return false;
    }
    $email = trim($_GET['email']);
    $code = trim($_GET['code']);

    if($this->GetResetPasswordCode($email) != $code)
    {
        $this->HandleError("Bad reset code!");
        return false;
    }

    $user_rec = array();
    if(!$this->GetUserFromEmail($email,$user_rec))
    {
        return false;
    }

    $new_password = $this->ResetUserPasswordInDB($user_rec);
    if(false === $new_password || empty($new_password))
    {
        $this->HandleError("Error updating new password");
        return false;
    }

    if(false == $this->SendNewPassword($user_rec,$new_password))
    {
        $this->HandleError("Error sending new password");
        return false;
    }
    return true;
}
```

Εικόνα 75 Συνάρτηση Reset Password

Με την συνάρτηση ResetUserPasswordInDB δημιουργείται ένας νέος κωδικός που είναι αυτός που θα σταλεί με μήνυμα στον χρήστη.

```php
function ResetUserPasswordInDB($user_rec)
{
    $new_password = substr(md5(uniqid()),0,10);

    if(false == $this->ChangePasswordInDB($user_rec,$new_password))
    {
        return false;
    }
    return $new_password;
}

function ChangePasswordInDB($user_rec, $newpwd)
{
    $newpwd = $this->SanitizeForSQL($newpwd);

    $qry = "Update $this->tablename Set password='".md5($newpwd)."' Where  id_user=".$user_rec['id_user']."";

    if(!mysql_query( $qry ,$this->connection))
    {
        $this->HandleDBError("Error updating the password \nquery:$qry");
        return false;
    }
    return true;
}
```

Εικόνα 76 Συναρτήσεις Reset ,Change User Password In DB

Η συνάρτηση SendNewPassword είναι παρόμοια με την SendResetPasswordLink με την διαφορά ότι λαμβάνει ένα ακόμα όρισμα τον καινούριο κωδικό ο οποίος και αποστέλλεται στον χρήστη.

Μετά το Login ο χρήστης θα μεταφερθεί στην αρχική σελίδα και θα μπορεί να δημιουργήσει αιτήματα για συλλογή δεδομένων.

Για την σελίδα δημιουργίας αιτήματος (**Εικόνα 47**) Create data ο κώδικας ενδεικτικά είναι ο εξής

```
if(!$fgmembersite->CheckLogin())
{
    $fgmembersite->RedirectToURL("login.php");
    exit;
}
$fgmembersite->ConfirmDate();
if(isset($_POST['submitted']))
{
   if($fgmembersite->send_data())
   {

        $fgmembersite->RedirectToURL("access-controlled.php");
        exit;
   }
}
```

Εικόνα 77 Περιεχόμενο σελίδας για την δημιουργία αιτήματος

Γίνεται ο έλεγχος του χρήστη μέσω της συνάρτησης CheckLogin. Στην περίπτωση που δεν έχει γίνει είσοδος στο σύστημα μεταφέρεται ο χρήστης στην σελίδα για να κάνει login.

```
function CheckLogin()
{
     if(!isset($_SESSION)){ session_start(); }

     $sessionvar = $this->GetLoginSessionVar();

     if(empty($_SESSION[$sessionvar]))
     {
        return false;
     }
     return true;
}
```

Εικόνα 78 Συνάρτηση Check Login

Η συνάρτηση ConfirmDate χρησιμοποιείτε για να αλλάζει η κατάσταση του request σε σχέση με την ημερομηνία που βρίσκετε. Οι καταστάσεις που μπορεί να βρεθεί ένα request είναι να είναι ανενεργό, ενεργό, ολοκληρωμένο χρονικά και πληρωμένο/διευθετημένο.

```
function ConfirmDate()
{
     $user_rec = array();

     if(!$this->Updatestatus($user_rec))
     {
          return false;
     }
     if(!$this->Updatestatus2($user_rec))
     {
          return false;
     }

     return true;

}
```

Εικόνα 79 Συνάρτηση Confirm Date

```php
function Updatestatus(&$user_rec)
{
    if(!$this->DBLogin())
    {
        $this->HandleError("Database login failed!");
        return false;
    }
    date_default_timezone_set('Europe/Athens');
    $today = date("Y-m-d H:i:s");
    $result = mysql_query("Select id_request from request where Time_e > '$today' AND Time_s < '$today'",$this->connection);
    if(!$result || mysql_num_rows($result) <= 0)
    {
        $this->HandleError("Wrong confirm code.");
        return false;
    }
    $row = mysql_fetch_assoc($result);
    $user_rec['id_request'] = $row['id_request'];

        $qry = "Update request Set status='1' Where Time_e > '$today'AND Time_s < '$today'";//active

    if(!mysql_query( $qry ,$this->connection))
    {
        $this->HandleDBError("Error inserting data to the table\nquery:$qry");
        return false;
    }
    return true;
}
```

Εικόνα 81 Συνάρτηση Update Status

```php
function Updatestatus2(&$user_rec)
{
    if(!$this->DBLogin())
    {
        $this->HandleError("Database login failed!");
        return false;
    }
    date_default_timezone_set('Europe/Athens');
    $today = date("Y-m-d H:i:s");
    $result = mysql_query("Select id_request from request where Time_e < '$today'",$this->connection);
    if(!$result || mysql_num_rows($result) <= 0)
    {
        $this->HandleError("Wrong confirm code.");
        return false;
    }
    $row = mysql_fetch_assoc($result);
    $user_rec['id_request'] = $row['id_request'];

        $qry = "Update request Set status='2' Where Time_e < '$today'";//disactive

    if(!mysql_query( $qry ,$this->connection))
    {
        $this->HandleDBError("Error inserting data to the table\nquery:$qry");
        return false;
    }
    return true;
}
```

Εικόνα 80 Συνάρτηση Update Status 2

Η συνάρτηση send_data έχει πολλά κοινά με την συνάρτηση RegisterUser καθώς χρησιμοποιείτε μια συνάρτηση για την συλλογή των δεδομένων και μία για την αποθήκευση στον κατάλληλο πίνακα.

```php
function send_data()
{
    if(!isset($_POST['submitted']))
    {
        return false;
    }

    $formvars = array();

    $this->CollectsendDataSubmission($formvars);

    if(!$this->SaveToDatabaserequest($formvars))
    {
        return false;
    }

    return true;
}
```

Εικόνα 82 Συνάρτηση Send Data

Στην συνάρτηση SaveTodatabaserequest όπως και στην SaveToDatabase ελέγχεται αν υπάρχει ο πίνακας (request) και αν δεν υπάρχει τότε τον δημιουργεί. Στην συνέχεια γίνεται η αποθήκευση των δεδομένων.

Ενδεικτικός κώδικας της συνάρτησης SaveTodatabaserequest για την δημιουργία του πίνακα.

```
function CreateTablerequest()
{
    $qry = "Create Table Request (".
            "id_request INT NOT NULL AUTO_INCREMENT PRIMARY KEY,".
            "Time_s DATETIME   NOT NULL ,".
            "Time_e DATETIME   NOT NULL ,".
            "latitude DECIMAL( 10,6 ) NOT NULL ,".
            "longitude DECIMAL( 10,6 ) NOT NULL ,".
            "range INT (3)   NOT NULL ,".
            "max_price DECIMAL( 8,4 ) NOT NULL ,".
            "user_id INT( 5) NOT NULL, ".
            "status INT (4) NOT NULL, ".
            "num_clients INT (4) NOT NULL, ".
            "category VARCHAR (20) NOT NULL, ".
            "FOREIGN KEY (user_id) REFERENCES users(id_user)
            ON UPDATE CASCADE
            ON DELETE CASCADE".
            ")";

    if(!mysql_query($qry,$this->connection))
    {
        $this->HandleDBError("Error creating the table \nquery was\n $qry");
        return false;
    }

    return true;
}
```

Εικόνα 83 Συνάρτηση Create Table Request

Για τις σελίδες HISTORY, HISTORY FROM ALL USERS (**Εικόνα 48 - 49**) εμφανίζονται τα αιτήματα του χρήστη αλλά και συνολικά όλων των χρηστών.

Ενδεικτικός κώδικας των σελίδων είναι παρόμοιος καθώς η μόνη διαφορά τους είναι ότι στην μία περίπτωση εμφανίζονται τα αιτήματα ενός χρήστη ενώ στην άλλη περίπτωση όλων των χρηστών.

```
if(!$fgmembersite->CheckLogin())
  {
      $fgmembersite->RedirectToURL("login.php");
      exit;
  }
$fgmembersite->DBLogin();
$user= $fgmembersite->iduser();
$search = "SELECT id_request,Time_s,Time_e,latitude,longitude,location_r,category,max_price,status
      FROM request
      WHERE user_id = '$user' ";

    $result=mysql_query($search);
     if ($result==false)
         {
            die(mysql_error());
         }
    $count=mysql_num_rows($result);
    echo "<table id='request' class='table table-striped table-bordered' cellspacing='0' width='100%'>
        <thead>
            <tr>
                <th>Number of request</th>
                <th>Time start</th>
                <th>Time end</th>
                <th>Location - Range</th>
                <th>Type of data</th>
                <th>Price </th>
                <th>Status</th>
            </tr>
        </thead>
        <tfoot>
            <tr>
            <th>Number of request</th>
                <th>Time start</th>
                <th>Time end</th>
                <th>Location - Range</th>
                <th>Type of data</th>
                <th>Price </th>
                <th>Status</th>
                </tr>
        </tfoot>
        <tbody>";
```

```php
for ($i=0;$i<$count;++$i)
  {
    $row=mysql_fetch_array($result);
    $id_request=$row["id_request"];
    $Time_s=$row["Time_s"];
    $Time_e=$row["Time_e"];
    $latitude=$row["latitude"];
    $longitude=$row["longitude"];
    $location_r=$row["location_r"];
    $datatype=$row["category"];
    $max_price=$row["max_price"];
    $status=$row["status"];
    $idnum=$i+1;
    echo "<tr><td>$id_request</td>
      <td>$Time_s</td>
      <td>$Time_e</td>
      <td>$latitude $longitude -- $location_r</td>
      <td>$datatype</td>
      <td>$max_price</td>
      <td>$status</td>

      </tr>
      </tbody>";

  }

  echo "</table>";
```

Εικόνα 84 Περιεχόμενο σελίδας HISTORY

Στο μέρος του Server Ο διαχειριστής της πλατφόρμας είναι αυτός που αναλαμβάνει την ολοκλήρωση των πληρωμών (**Εικόνα 53**).

Ενδεικτικός κώδικας εμφάνισης της λίστας για να γίνουν οι πληρωμές.

```php
public function getcompletedRequests(){
    $stmt = $this->con->prepare("SELECT id_request, Time_s, Time_e, latitude, longitude,
        location_r, max_price, user_id, status, num_clients, category FROM request WHERE
        status ="."3");
    $stmt->execute();
    $stmt->bind_result($id, $timestart, $timeend, $latitude, $longitude, $location, $maxprice
        , $userid, $status, $numclients, $category);

    $requests = array();

    while($stmt->fetch()){
        $temp = array();
        $temp['id']=$id;
        $temp['timestart'] = $timestart;
        $temp['timeend'] = $timeend;
        $temp['latitude'] = $latitude;
        $temp['longitude'] = $longitude;
        $temp['location'] = $location;
        $temp['maxprice'] = $maxprice;
        $temp['userid'] = $userid;
        $temp['status'] = $status;
        $temp['numclients'] = $numclients;
        $temp['category'] = $category;
        array_push($requests, $temp);
    }

    $stmt->close();

    return $requests;
}
```

Εικόνα 85 Συνάρτηση getcompletedRequests

Η επόμενη ενέργεια είναι η πληρωμή των Data producers.

```
    $db = new DbOperation();

    $winner = $db->getWinner($requestid);

    if($winner){

        echo "<h2>Winner is </h2>";
        echo "<h3>". $winner['btcaddress']."</h3>";

        try{
            $transaction = $card->createTransaction($winner['btcaddress'], $winner['amount'], 'BTC');
            $transaction->commit();
        }catch(Exception $e){
            echo "May be the bitcoin address is incorrect";
            echo $e->getMessage();
        }
    }else{

        echo "No one won";

    }

}else{
    header('Location: index.php');
}
```

Εικόνα 86 Κώδικας πληρωμής

Μέσω της getWinner τρέχουν τρεις συναρτήσεις η getShortlistedUsers η getRequestCoordinates και η haversineGreatCircleDistance.

```
public function getWinner($requestid){
    $shortlistedUsers = $this->getShortlistedUsers($requestid);

    $coordinates = $this->getRequestCoordinates($requestid);

$winners = array();

//echo count($shortlistedUsers);
    foreach($shortlistedUsers as $user){

        //echo $this->haversineGreatCircleDistance($user['latitude'], $user['longitude'],
        $coordinates['latitude'], $coordinates['longitude']) . '<br />';

        if($this->haversineGreatCircleDistance($user['latitude'], $user['longitude'], $
            coordinates['latitude'], $coordinates['longitude'])<=$user['location_r']){

            array_push($winners, $user);
        }
    }
    return $winners;
}
```

Εικόνα 87 Συνάρτηση getWinner

Η getShortlistedUsers επιλέγει τους χρήστες με βάση κάποια χαρακτηριστικά που έχει δώσει ο Requester. Στην συνάρτηση επιλέγονται και αποθηκεύονται όσοι έχουν κάνει αίτημα συμμετοχής σε συγκεκριμένο request που εξετάζεται κάθε φορά και πληροί κάποιες προδιαγραφές όπως το ποσό διάθεσης να είναι μικρότερο ή ίσο από το ποσό της προσφοράς. Στην περίπτωση που υπάρχουν περισσότερες από μια ίδιες τιμές λαμβάνετε κατά εκτίμηση της SQL. Τέλος επιλέγονται όχι περισσότεροι από αυτούς που έχει ορίσει ο Requester.

```php
private function getShortlistedUsers($requestid){
    $stmt = $this->con->prepare("select id_auction, amount, data, id_appclients, id_request,
        latitude, longitude, (SELECT location_r FROM request where id_request = ?) as location_r,
        (SELECT num_clients FROM request where id_request = ?) as numclients, (SELECT bitcoinadr
        FROM appclients WHERE appclients.id_appclient = auction.id_appclients) as btcaddress
        FROM auction WHERE id_request = ? AND amount < (SELECT max_price FROM request WHERE
        id_request = ?) ORDER BY amount ASC LIMIT numclients;");
    $stmt->bind_param("iiii", $requestid, $requestid, $requestid);
    $stmt->execute();
    $stmt->bind_result($id, $amount, $data, $idappclient, $idrequest, $latitude, $longitude,$
        locationr, $btcaddress,$numclients);

    $selectedUsers = array();

    while($stmt->fetch()){
        $temp = array();
        $temp['id'] = $id;
        $temp['amount'] = $amount;
        $temp['data'] = $data;
        $temp['idappclient'] = $idappclient;
        $temp['idrequest'] = $idrequest;
        $temp['location_r']=$locationr;
        $temp['latitude']  = $latitude;
        $temp['longitude'] = $longitude;
        $temp['btcaddress'] = $btcaddress;
        $temp['numclients'] = $numclients;
        array_push($selectedUsers, $temp);
    }

    return $selectedUsers;
}
```

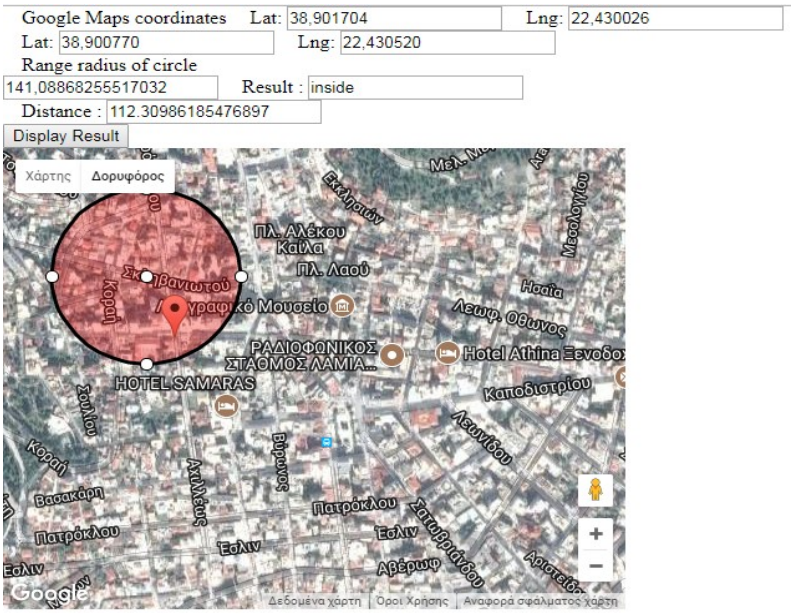
Εικόνα 88 Συνάρτηση getGetShortlistedUsers

Στην συνέχεια οι συναρτήσεις getRequestCoordinates και haversineGreatCircleDistance αποθηκεύουν το γεωγραφικό μήκος και πλάτος του κάθε Request ώστε να γίνει η σύγκριση με τις αντίστοιχες συντεταγμένες των App Clients Users αλλά και με βάση την ακτίνα στην οποία βρίσκονται.

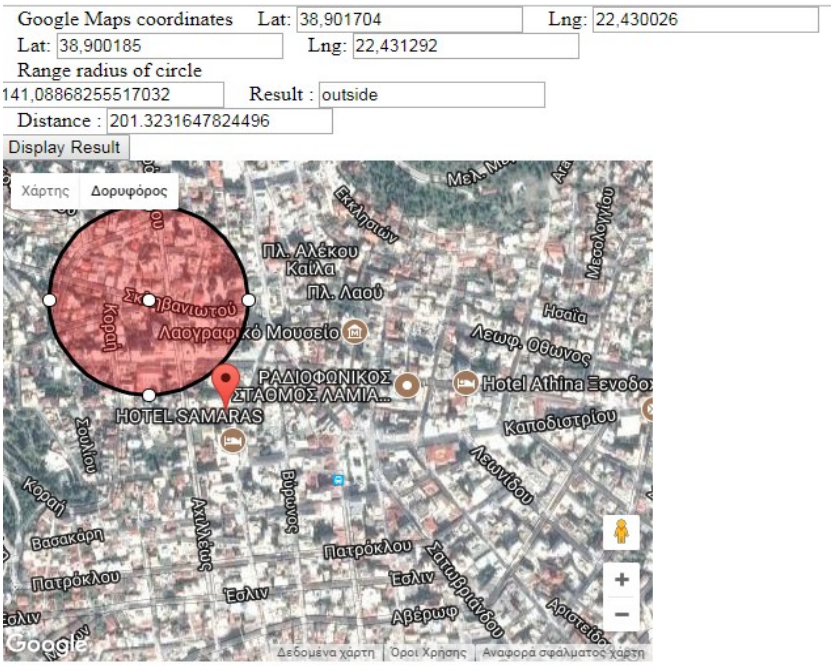
```php
private function getRequestCoordinates($requestid){
    $stmt = $this->con->prepare("SELECT latitude, longitude FROM request WHERE id_request = ?");
    $stmt->bind_param("i", $requestid);
    $stmt->execute();
    $stmt->bind_result($latitude, $longitude);
    $stmt->fetch();
    $coordinates = array();
    $coordinates['latitude'] = $latitude;
    $coordinates['longitude'] = $longitude;
    return $coordinates;
}

public function haversineGreatCircleDistance($latitudeFrom, $longitudeFrom, $latitudeTo, $
    longitudeTo, $earthRadius = 6371000)//metre 6371~in km
{
    // convert from degrees to radians
    $latFrom = deg2rad($latitudeFrom);
    $lonFrom = deg2rad($longitudeFrom);
    $latTo = deg2rad($latitudeTo);
    $lonTo = deg2rad($longitudeTo);

    $latDelta = $latTo - $latFrom;
    $lonDelta = $lonTo - $lonFrom;

    $angle = 2 * asin(sqrt(pow(sin($latDelta / 2), 2) +
    cos($latFrom) * cos($latTo) * pow(sin($lonDelta / 2), 2)));
    return $angle * $earthRadius;
}
```

Εικόνα 89 Συναρτήσεις getRequestCoordinates και haversineGreatCircleDistance

Για την κατανόηση και την λύση αυτού του προβλήματος δημιουργήθηκε μια σελίδα. Δουλεύει πατώντας τα πλήκτρα του ποντικιού(αριστερό και δεξί click) και το κουμπί στην σελίδα Display Result. Πατώντας το αριστερό click δημιουργείτε ένας κόκκινος κύκλος και με το δεξί click δημιουργείτε μία πινέζα. Όταν η πινέζα βρίσκεται εκτός του κύκλου τότε πατώντας το κουμπί Display Result έχω ως αποτέλεσμα το μήνυμα outside διαφορετικά έχω inside. Η ακτίνα του κύκλου αλλά και η απόσταση της πινέζας από το κέντρο του κύκλου μετρώνται σε μέτρα.



Εικόνα 90 Βοηθητική σελίδα Α



Εικόνα 91 Βοηθητική εφαρμογή Β

7.3.2.2. Android εφαρμογή

Οι χρήστες οι οποίοι θα διαθέσουν τα δεδομένα τους θα γίνει μέσω της Android εφαρμογής αποτελούμενη από 3 κύριες οθόνες (**Εικόνα 50-52**).

Για την υλοποίηση της Android εφαρμογής έχει γίνει χρήση κάποιων βοηθητικών κλάσεων. Αυτές είναι:

- Η κλάση Constants όπου δηλώνονται κάποιες σταθερές.

```
private static final String ROOT_URL = "http://bitcoinproject.website/alex/v1/";

public static final String URL_LOGIN = ROOT_URL + "login";
public static final String URL_REQUESTS = ROOT_URL + "requests";

public static final String URL_SAVE_AUCTION = ROOT_URL + "saveauction";
```

Εικόνα 92 Κλάση Constants

- Η κλάση RSA η οποία κάνει την κρυπτογράφηση.

```
public static byte[] encryptData(byte[] data, String key)
        throws Exception {

    byte[] keyBytes = decryptBASE64(key);

    X509EncodedKeySpec x509KeySpec = new X509EncodedKeySpec(keyBytes);
    KeyFactory keyFactory = KeyFactory.getInstance(KEY_ALGORITHM);
    Key publicKey = keyFactory.generatePublic(x509KeySpec);

    Cipher cipher = Cipher.getInstance(keyFactory.getAlgorithm());
    cipher.init(Cipher.ENCRYPT_MODE, publicKey);

    return cipher.doFinal(data);
}
```

Εικόνα 93 Κλάση RSA

- Η κλάση SharedPrefManager για την διαχείριση εισόδου, εξόδου του χρήστη από την εφαρμογή.

```
public void userLogin(User user) {
    SharedPreferences sharedPreferences = mCtx.getSharedPreferences(SHARED_PREF_NAME, Context.MODE_PRIVATE);
    SharedPreferences.Editor editor = sharedPreferences.edit();
    editor.putInt(KEY_USER_ID, user.getId());
    editor.putString(KEY_USER_BITCOINADDRESS, user.getBitcoinAddress());
    editor.putString(KEY_USER_PUBLIC_KEY, user.getPublicKey());
    editor.apply();
}

public boolean isLoggedIn() {
    SharedPreferences sharedPreferences = mCtx.getSharedPreferences(SHARED_PREF_NAME, Context.MODE_PRIVATE);
    if (sharedPreferences.getString(KEY_USER_BITCOINADDRESS, null) != null)
        return true;
    return false;
}

public User getUser() {
    SharedPreferences sharedPreferences = mCtx.getSharedPreferences(SHARED_PREF_NAME, Context.MODE_PRIVATE);
    return new User(
            sharedPreferences.getInt(KEY_USER_ID, -1),
            sharedPreferences.getString(KEY_USER_BITCOINADDRESS, null),
            sharedPreferences.getString(KEY_USER_PUBLIC_KEY, null)
    );
}

public void logout() {
    SharedPreferences sharedPreferences = mCtx.getSharedPreferences(SHARED_PREF_NAME, Context.MODE_PRIVATE);
    SharedPreferences.Editor editor = sharedPreferences.edit();
    editor.clear();
    editor.apply();
}
```

Εικόνα 94 Κλάση Shared Pref Manager

- Η κλάση User όπου δηλώνονται τα χαρακτηριστικά του κάθε χρήστη.

```
public class User {
    private int id;
    private String bitcoinAddress;
    private String publicKey;

    public User(int id, String bitcoinAddress, String publicKey) {
        this.id = id;
        this.bitcoinAddress = bitcoinAddress;
        this.publicKey = publicKey;
    }

    public int getId() { return id; }

    public String getBitcoinAddress() { return bitcoinAddress; }

    public String getPublicKey() { return publicKey; }
}
```

Εικόνα 95 Κλάση User

- Η κλάση UserRequest όπου δηλώνονται οι μεταβλητές για τον πίνακα Request.

```
public class UserRequest implements Serializable {
    private int id;
    private String timeS, timeE;
    private Double latitude, longitude;
    private int locationR;
    private String maxPrice;
    private int status, numclients;
    private String category;

    public UserRequest(){}

    public UserRequest(int id, String timeS, String timeE, Double latitude,
                       Double longitude, int locationR, String maxPrice,
                       int status, int numclients, String category) {
        this.id = id;
        this.timeS = timeS;
        this.timeE = timeE;
        this.latitude = latitude;
        this.longitude = longitude;
        this.locationR = locationR;
        this.maxPrice = maxPrice;
        this.status = status;
        this.numclients = numclients;
        this.category = category;
    }

    public int getId() { return id; }

    public String getTimeS() { return timeS; }

    public String getTimeE() { return timeE; }

    public Double getLatitude() { return latitude; }

    public Double getLongitude() { return longitude; }

    public int getLocationR() { return locationR; }

    public String getMaxPrice() { return maxPrice; }

    public int getStatus() { return status; }

    public int getNumclients() { return numclients; }

    public String getCategory() { return category; }
}
```

Εικόνα 97 Κλάση User Request

- Η κλάση VolleySingleton η οποία χρησιμοποιεί την Volley βιβλιοθήκη της HTTP για την επικοινωνία με το δίκτυο.

```
public class VolleySingleton {
    private static VolleySingleton mInstance;
    private RequestQueue mRequestQueue;

    private static Context mCtx;

    private VolleySingleton(Context context) {
        mCtx = context;
        mRequestQueue = getRequestQueue();
    }

    public static synchronized VolleySingleton getInstance(Context context) {
        if (mInstance == null) {
            mInstance = new VolleySingleton(context);
        }
        return mInstance;
    }

    public RequestQueue getRequestQueue() {
        if (mRequestQueue == null) {
            // getApplicationContext() is key, it keeps you from leaking the
            // Activity or BroadcastReceiver if someone passes one in.
            mRequestQueue = Volley.newRequestQueue(mCtx.getApplicationContext());
        }
        return mRequestQueue;
    }

    public <T> void addToRequestQueue(Request<T> req) { getRequestQueue().add(req); }
```

Εικόνα 98 Κλάση VolleySingleton

- MainActivity

Όταν ο χρήστης εκκινήσει για πρώτη φορά στην εφαρμογή (**Εικόνα 51**) θα του ζητηθεί να δηλώσει μια Bitcoin διεύθυνση για να πληρωθεί. Σε αυτό το σημείο θα τρέξει η MainActivity όπου θα γίνει έλεγχος αν έχει συμπληρωθεί η διεύθυνση και αν όχι τότε θα εμφανιστεί το κατάλληλο μήνυμα. Αν δεν υπάρχει χρήστης με αυτή την διεύθυνση τότε θα δημιουργηθεί νέος λογαριασμός και θα καταχωρηθεί στην βάση μαζί με ένα μοναδικό id αριθμό διαφορετικά θα γίνει είσοδος.

```
private void login() {
    final String bitCoinAddress = editTextBitcoinAddress.getText().toString().trim();

    if (TextUtils.isEmpty(bitCoinAddress)) {
        Toast.makeText(this, "Please enter bitcoin address", Toast.LENGTH_LONG).show();
        return;
    }

    progressDialog.setMessage("Login in...");
    progressDialog.show();

    StringRequest stringRequest = new StringRequest(Request.Method.POST, Constants.URL_LOGIN,
            new Response.Listener<String>() {
                @Override
                public void onResponse(String response) {
                    progressDialog.dismiss();
                    try {
                        JSONObject obj = new JSONObject(response);
                        if (!obj.getBoolean("error")) {
                            JSONObject jsonUser = obj.getJSONObject("user");
                            User user = new User(
                                    jsonUser.getInt("id"),
                                    jsonUser.getString("bitcoinaddress"),
                                    jsonUser.getString("publickey")
                            );
                            SharedPrefManager.getInstance(getApplicationContext()).userLogin(user);
                            startActivity(new Intent(getApplicationContext(), MapActivity.class));
                            finish();
                        } else {
                            Toast.makeText(MainActivity.this, obj.getString("message"), Toast.LENGTH_LONG).show();
                        }
                    } catch (JSONException e) {
                        e.printStackTrace();
                    }
                    //Toast.makeText(MainActivity.this, response, Toast.LENGTH_LONG).show();
                }
            }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            progressDialog.dismiss();
            Toast.makeText(MainActivity.this, error.getMessage(), Toast.LENGTH_LONG).show();
        }
    }) {
        @Override
        protected Map<String, String> getParams() throws AuthFailureError {
            Map<String, String> params = new HashMap<>();
            params.put("bitcoinaddress", bitCoinAddress);
            return params;
        }
    };
```

Εικόνα 99 Ενδεικτικός κώδικας της MainActivity

Από την μεριά του Server εκτελούνται τα παρακάτω μέσω της android εφαρμογής και του Framework Slim. Αφού λάβει τα δεδομένα ο Server θα δημιουργήσει ένα μοναδικό κλειδί και θα εισάγει την διεύθυνση του χρήστη αφού πρώτα γίνουν οι κατάλληλοι έλεγχοι όπως αν υπάρχει ήδη κάποιος χρήστης με την ίδια διεύθυνση.

```php
$app = new \Slim\Slim();

$app->post('/login', function () use ($app) {
    verifyRequiredParams(array('bitcoinaddress'));

    $response = array();

    $bitcoinaddress = $app->request->post('
        bitcoinaddress');

    $db = new DbOperation();

    if($db->userLogin($bitcoinaddress)){
        $response['error'] = false;
        $response['user'] = $db->getUserDetails($
            bitcoinaddress);
    }else{
        $response['error'] = true;
        $response['message'] = 'Invalid bitcoin
            address';
    }
    echoResponse(200, $response);
});
```

Εικόνα 101 Ενδεικτικός κώδικας από το Framework Slim

```php
public function userLogin($bitcoinaddress){
    if($this->isAddressExist($bitcoinaddress)){
        return true;
    }else{
        $public_key = $this->generateApiKey();
        $stmt = $this->con->prepare("INSERT INTO appclients (bitcoinadr,
            public_key) VALUES (?, ?)");
        $stmt->bind_param("ss",$bitcoinaddress, $public_key);
        if($stmt->execute())
            return true;
    }
    return false;
}

private function isAddressExist($bitcoinaddress){
    $stmt = $this->con->prepare("SELECT id_appclient FROM appclients WHERE
        bitcoinadr = ?");
    $stmt->bind_param("s",$bitcoinaddress);
    $stmt->execute();
    $stmt->store_result();
    $num_rows = $stmt->num_rows;
    $stmt->close();
    return $num_rows>0;
}
```

Εικόνα 100 Συναρτήσεις userLogin και isAddressExist

- MapActivity

Στην συνέχεια ο χρήστης θα μεταφερθεί στον χάρτη (**Εικόνα 52**) με όλες τις ενεργές τοποθεσίες όπου μπορεί να πάρει μέρος σε όποια επιθυμεί. Πατώντας την κάθε πινέζα εμφανίζεται η ημερομηνία που τελειώνει η προσφορά καθώς και το είδος αυτής.

```java
protected void onStart() {
    mGoogleApiClient.connect();
    mSensorManager.registerListener(this,
            mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER
                    | Sensor.TYPE_LIGHT),
            SensorManager.SENSOR_DELAY_NORMAL);
    super.onStart();
}
protected void onStop() {
    mGoogleApiClient.disconnect();
    mSensorManager.unregisterListener(this);
    super.onStop();
}
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_map);
    requestLocationPermission();
    userRequestList = new ArrayList<>();
    progressDialog = new ProgressDialog(this);
    mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
    mSensor = mSensorManager.getDefaultSensor(Sensor.TYPE_GRAVITY);
    mSensorLight = mSensorManager.getDefaultSensor(Sensor.TYPE_LIGHT);
    mGoogleApiClient = new GoogleApiClient.Builder(this)
            .addConnectionCallbacks(this)
            .addOnConnectionFailedListener(this)
            .addApi(LocationServices.API)
            .build();

    if (!SharedPrefManager.getInstance(this).isLoggedIn()) {
        finish();
        startActivity(new Intent(this, MainActivity.class));
    }
```

```
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    mMap.setOnInfoWindowClickListener(this);
    loadRequests();
}
private void loadRequests() {
    progressDialog.setMessage("Please wait...");
    progressDialog.show();

    StringRequest stringRequest = new StringRequest(Request.Method.GET, Constants.URL_REQUESTS,
            new Response.Listener<String>() {
                @Override
                public void onResponse(String response) {
                    progressDialog.dismiss();
                    try {
                        JSONObject obj = new JSONObject(response);
                        JSONArray jsonReq = obj.getJSONArray("requests");

                        for (int i = 0; i < jsonReq.length(); i++) {
                            JSONObject objReq = jsonReq.getJSONObject(i);
                            UserRequest userRequest = new UserRequest(
                                    objReq.getInt("id"),
                                    objReq.getString("time_s"),
                                    objReq.getString("time_e"),
                                    objReq.getDouble("latitude"),
                                    objReq.getDouble("longitude"),
                                    objReq.getInt("location_r"),
                                    objReq.getString("maxprice"),
                                    objReq.getInt("status"),
                                    objReq.getInt("numclients"),
                                    objReq.getString("category")
                            );
```

```
private void drawMarkers() {
    markers = new Marker[userRequestList.size()];

    for (int i = 0; i < userRequestList.size(); i++) {
        UserRequest userRequest = userRequestList.get(i);
        markers[i] = mMap.addMarker(
                new MarkerOptions()
                        .position(
                                new LatLng(userRequest.getLatitude(),
                                        userRequest.getLongitude())
                        ).title(userRequest.getTimeE())
                        .snippet(userRequest.getCategory()));
        markers[i].showInfoWindow();
    }

    mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(markers[0].getPosition(), 16));
}
```

```
public void onInfoWindowClick(Marker marker) {
    for (int i = 0; i < markers.length; i++) {
        if (marker.equals(markers[i])) {
            Location mLastLocation = LocationServices.FusedLocationApi.getLastLocation(mGoogleApiClient);
            UserRequest userRequest = userRequestList.get(i);
            Intent intent = new Intent(this, AuctionActivity.class);
            intent.putExtra("request", userRequest);

            intent.putExtra("latitude", mLastLocation.getLatitude());
            intent.putExtra("longitude", mLastLocation.getLongitude());

            if (userRequest.getCategory().equalsIgnoreCase("location")) {
                intent.putExtra("data", locationData);
                intent.putExtra("category", "Location");
            }
            if (userRequest.getCategory().equalsIgnoreCase("accelerometer")) {
                intent.putExtra("data", String.valueOf(acceleration));
                intent.putExtra("category", "Acceleration");
            }
            if (userRequest.getCategory().equalsIgnoreCase("gyroscope")) {
                intent.putExtra("data", String.valueOf(accuracy));
                intent.putExtra("category", "Accuraccy");
            }
            if (userRequest.getCategory().equalsIgnoreCase("light")) {
                intent.putExtra("data", String.valueOf(lightValue));
                intent.putExtra("category", "Light");
            }
            startActivity(intent);
            break;
        }
    }
}
```

Εικόνα 102 Ενδεικτικός κώδικας MapActivity

Από την μεριά του Server θα εκτελεστεί συνάρτηση ώστε να εμφανιστούν τα ενεργά αιτήματα

```php
$app->get('/requests', function() use ($app){
    $db = new DbOperation();

    $response = array();
    $response['error'] = false;
    $response['requests'] = $db->getUserRequests();

    echoResponse(200, $response);
});
```

Εικόνα 103 Ενδεικτικός κώδικας από το Framework Slim

```php
public function getUserRequests(){
    $stmt = $this->con->prepare("SELECT id_request, time_s, time_e, latitude, longitude,
        location_r, max_price, status, num_clients, category FROM request WHERE status = 1");
    $stmt->execute();
    $stmt->bind_result(
        $id, $times, $timee, $latitude, $longitude, $locationr, $maxprice, $status, $numclients,
            $category
    );
    $requests = array();

    while($stmt->fetch()){
        $temp = array();
        $temp['id'] = $id;
        $temp['time_s'] = $times;
        $temp['time_e'] = $timee;
        $temp['latitude'] = $latitude;
        $temp['longitude'] = $longitude;
        $temp['location_r'] = $locationr;
        $temp['maxprice'] = $maxprice;
        $temp['status'] = $status;
        $temp['numclients'] = $numclients;
        $temp['category'] = $category;

        array_push($requests, $temp);
    }

    return $requests;
}
```

Εικόνα 104 Συνάρτηση getUserRequests

- AuctionActivity

Η τελευταία οθόνη της εφαρμογής είναι αυτή της δήλωσης της προσφοράς (**Εικόνα 50**). Κάνοντας είσοδο για συμμετοχή σε ένα αίτημα εμφανίζεται εκτός από το είδος που φαίνεται και σε προηγούμενη φάση απεικονίζεται και η μέγιστη τιμή που πρόκειται να διατεθεί αλλά και σε τι εύρος μπορεί να βρίσκεται ο χρήστης της κινητής εφαρμογής. Στο κάτω μέρος και πριν το κουμπί αποθήκευσης βρίσκονται τα δεδομένα που θα διατεθούν και στο συγκεκριμένο είναι γεωγραφικό μήκος και πλάτος του χρήστη. Αυτή είναι η πληροφορία η οποία θα κρυπτογραφηθεί και θα αποθηκευτεί στην βάση.

```
private void saveAuction() throws IllegalBlockSizeException, InvalidKeyException,
        BadPaddingException, NoSuchAlgorithmException, NoSuchPaddingException {
    final String amount = editTextAmount.getText().toString().trim();
    final String encryptData = rsa.encryptData(data,publicKey);

    if (TextUtils.isEmpty(amount)) {
        Toast.makeText(this, "Please enter an amount first", Toast.LENGTH_LONG).show();
        return;
    }
    progressDialog.setMessage("Saving...");
    progressDialog.show();
    StringRequest stringRequest = new StringRequest(Request.Method.POST, Constants.URL_SAVE_AUCTION,
            new Response.Listener<String>() {
                @Override
                public void onResponse(String response) {
                    progressDialog.dismiss();
                    try {
                        JSONObject obj = new JSONObject(response);
                        Toast.makeText(getApplicationContext(), obj.getString("message"), Toast.LENGTH_LONG).show();
                    } catch (JSONException e) {
                        e.printStackTrace();
                    }
                }
            },
            new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {
                    progressDialog.dismiss();
                    Toast.makeText(getApplicationContext(), error.getMessage(), Toast.LENGTH_LONG).show();
                }
            }) {
        @Override
        protected Map<String, String> getParams() throws AuthFailureError {
            Map<String, String> params = new HashMap<>();
            params.put("amount", amount);
            params.put("data", encryptData);
            params.put("idappclient", String.valueOf(SharedPrefManager.
                    getInstance(getApplicationContext()).getUser().getId()));
            params.put("idrequest", String.valueOf(userRequest.getId()));
            params.put("latitude", String.valueOf(latitude));
            params.put("longitude", String.valueOf(longitude));
            return params;
        }
    };
```

Εικόνα 105 Ενδεικτικός κώδικας AuctionActivity

Από την μεριά του Server θα εκτελεστεί η συνάρτηση αποθήκευσης των δεδομένων που λαμβάνει από την android συσκευή. Η αποθήκευση θα γίνει στον πίνακα auction.

```
$app->post('/saveauction', function() use ($app){
    verifyRequiredParams(array('amount','data', 'idappclient', 'idrequest'));
    $db = new DbOperation();

    $response = array();

    $amount = $app->request->post('amount');
    $data = $app->request->post('data');
    $idappclient = $app->request->post('idappclient');
    $idrequest = $app->request->post('idrequest');

    if($db->saveAuction($amount, $data, $idappclient, $idrequest)){
        $response['error'] = false;
        $response['message'] = 'Auction saved successfully';
    }else{
        $response['error'] = true;
        $response['message'] = 'Auction not saved';
    }
    echoResponse(201, $response);
});
```

Εικόνα 106 Ενδεικτικός κώδικας από το Framework Slim

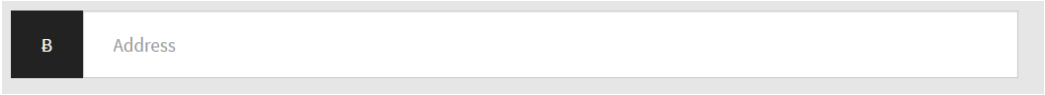
```
public function saveAuction($amount, $data, $idappclient, $
    idrequest){
    $stmt = $this->con->prepare('INSERT INTO auction (amount,
        data, id_appclients, id_request) VALUES (?, ?, ?, ?)');
    $stmt->bind_param("ssii", $amount, $data, $idappclient, $
        idrequest);
    if($stmt->execute())
        return true;
    return false;
}
```

Εικόνα 107 Συνάρτηση saveAuction

7.4. Ολοκληρωμένη χρήση των εφαρμογών

7.4.1. Δημιουργία λογαριασμού του Data consumer μέσω uphold sandbox

Η διαδικασία δημιουργίας ενός λογαριασμού γίνεται με την συμπλήρωση κάποιων βασικών πεδίων όπως username, email, date of birthday. Στην συνέχεια θα πρέπει να γίνει κατάθεση εικονικών χρημάτων στον λογαριασμό. Υπάρχουν ιστοσελίδες που παρέχουν αυτή την υπηρεσία όπως η http://faucet.haskoin.com/



Εικόνα 108 Πεδίο για την συμπλήρωση διεύθυνσης Bitcoin για την κατάθεση εικονικών χρημάτων

Σε αυτό το πεδίο εισάγετε η διεύθυνση Bitcoin που θα γίνει η κατάθεση. Η διεύθυνση του νέου λογαριασμού είναι η παρακάτω.

```
mwWmPbqZoSYg2CR32rTiQej1tVPFcQizVd
```

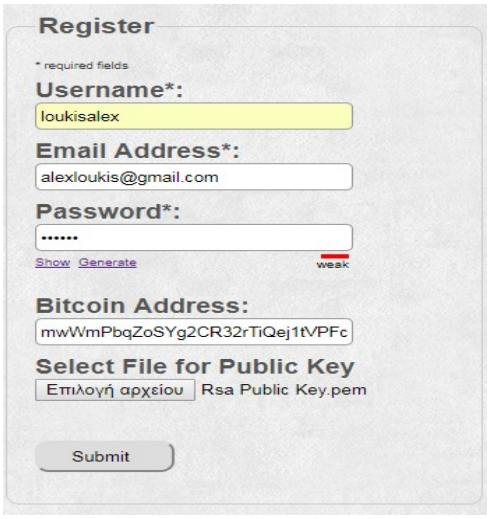
Εικόνα 109 Bitcoin διεύθυνση

Το νέο υπόλοιπο είναι

Εικόνα 110 Υπόλοιπο λογαριασμού της Uphold πλατφόρμας

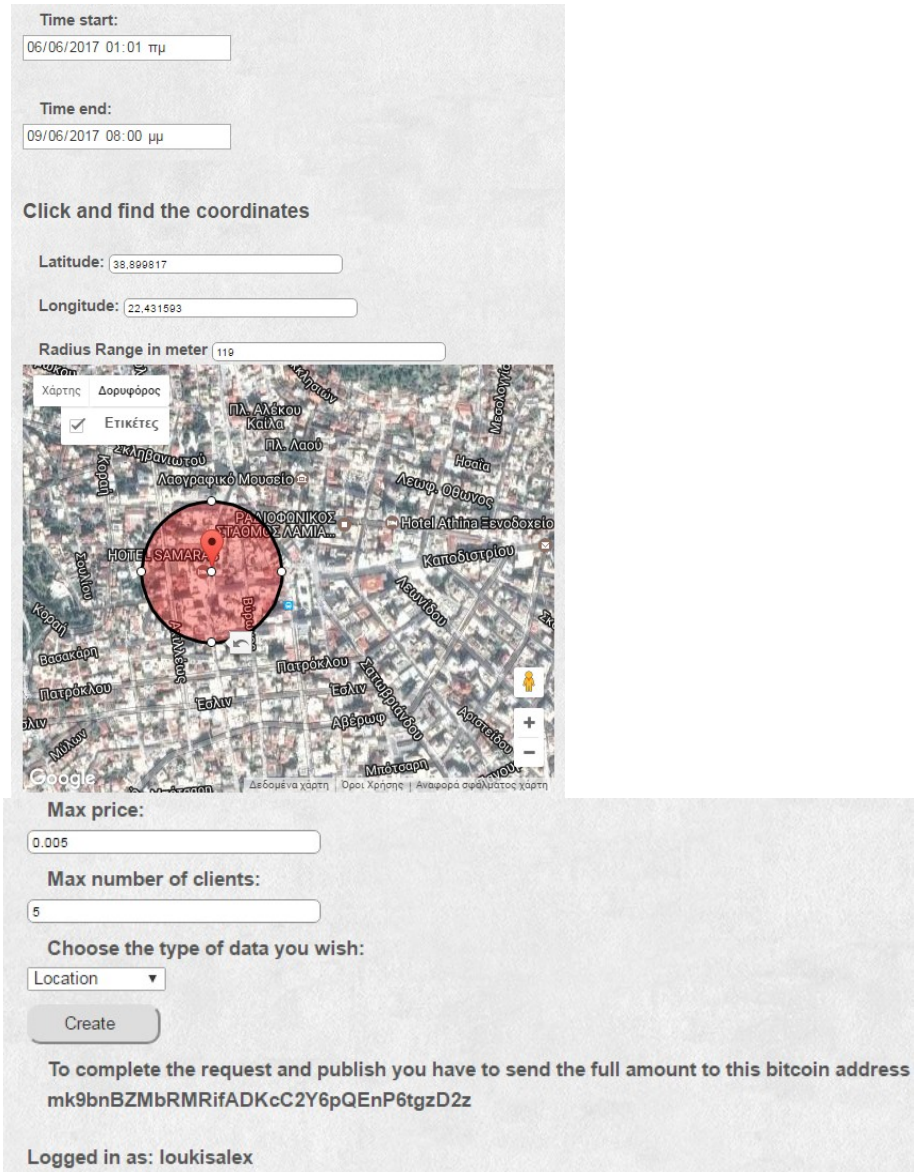
7.4.2. Δημιουργία λογαριασμού στην πλατφόρμα

Ο Data consumer μεταφέρεται στην σελίδα εγγραφής και συμπληρώνει τα απαραίτητα πεδία. Στην συνέχεια αφού λάβει τον κωδικό επιβεβαίωσης στο email του έχει την δυνατότητα να κάνει είσοδο και δημιουργία ερωτήματος.



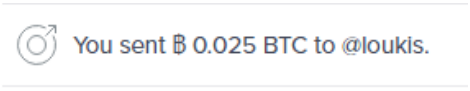
Εικόνα 111 Φόρμα εγγραφής νέου χρήστη

7.4.3. Create Data και πληρωμή του Διαχειριστή



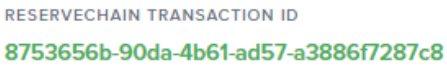
Εικόνα 112 Φόρμα δημιουργίας ερωτήματος

Συμπληρώνονται τα πεδία που απαιτούνται και γίνεται η πληρωμή στην διεύθυνση που αναγράφεται.



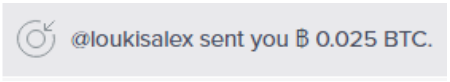
Εικόνα 113 Μήνυμα αποστολής Bitcoins

Μέσω του Transaction ID εξακριβώνεται η πληρωμή

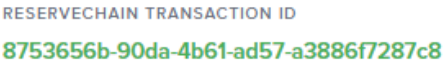


Εικόνα 114 Κωδικός συναλλαγής

Από την μεριά του διαχειριστή φαίνεται η συναλλαγή.



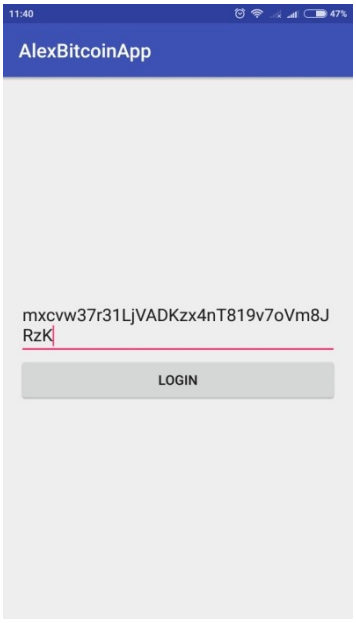
Εικόνα 115 Μήνυμα λήψης Bitcoins



Εικόνα 116 Κωδικός συναλλαγής

7.4.4. Δημιουργία λογαριασμού του Data producer μέσω uphold sandbox και είσοδο στην Android εφαρμογής

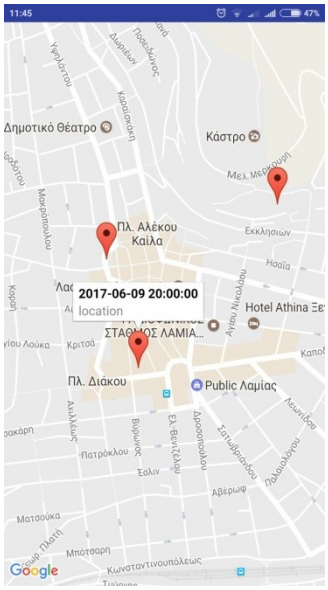
Δημιουργείται ένας λογαριασμός για τον Data producer μέσω του uphold όπως και στην περίπτωση του Data consumer με την διαφορά ότι δεν πραγματοποιείται κατάθεση εικονικών χρημάτων αλλά αποθήκευση της Bitcoin διεύθυνσης. Ανοίγοντας για πρώτη φορά την εφαρμογή συμπληρώνεται το πεδίο της διεύθυνσης.



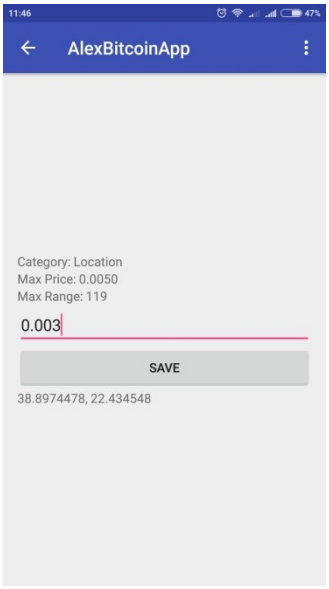
Εικόνα 117 Είσοδο χρήστη στην mobile εφαρμογή

7.4.5. Εμφάνιση όλων των ενεργών αιτημάτων και κατάθεση προσφοράς

Με την εμφάνιση του χάρτη αναζητάτε το αίτημα με την ημερομηνία τερματισμού που έχει δηλωθεί καθώς και το είδος των δεδομένων που θα συλλέξει η εφαρμογή. Στην συνέχεια η επόμενη οθόνη που εμφανίζεται είναι αυτή της προσφοράς όπου διακρίνεται η τοποθεσία, η μέγιστη τιμή διάθεσης, το εύρος του αιτήματος σε μέτρα και το πεδίο όπου ο χρήστης θα δώσει την προσφορά του και πρέπει να είναι μικρότερη από την μέγιστη τιμή. Τέλος φαίνεται και το γεωγραφικό μήκος και πλάτος του χρήστη που αποτελεί τα δεδομένα προς κρυπτογράφηση.



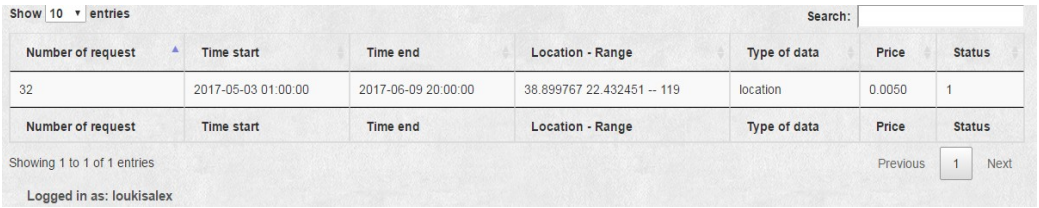
Εικόνα 119 Χάρτης με ενεργά αιτήματα



Εικόνα 118 Δήλωση προσφοράς σε αίτημα

7.4.6. Το μέρος του Data Consumer

Ο Data Consumer κατά την πλοήγηση του στην ιστοσελίδα πατώντας τον σύνδεσμο HISTORY θα δει όλα τα ενεργά ή μη ενεργά αιτήματα που έχει δημιουργήσει. Στην περίπτωση αυτή θα εμφανιστεί το παρακάτω.


Show 10 entries Search:

Number of request	Time start	Time end	Location - Range	Type of data	Price	Status
32	2017-05-03 01:00:00	2017-06-09 20:00:00	38.899767 22.432451 -- 119	location	0.0050	1
Number of request	Time start	Time end	Location - Range	Type of data	Price	Status

Showing 1 to 1 of 1 entries Previous 1 Next

Logged in as: loukisalex

Εικόνα 120 Σελίδα με το Ιστορικό του χρήστη

Όταν ολοκληρωθεί το αίτημα χρονικά η αλλαγή που θα γίνει θα είναι στην κατάσταση του.

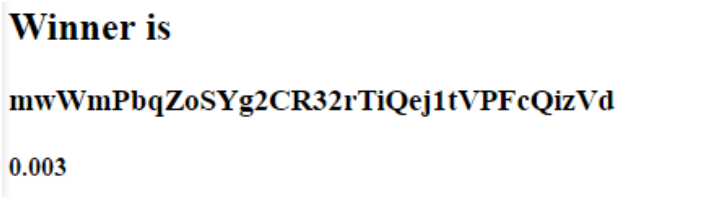
7.4.7. Το μέρος του Διαχειριστή

Το επόμενο βήμα είναι η πληρωμή των Data producers. Αυτό γίνεται από τον διαχειριστή για κάθε ένα αίτημα. Σε αυτό το σημείο εμφανίζονται μόνο τα απλήρωτα αλλά ολοκληρωμένα χρονικά αιτήματα.

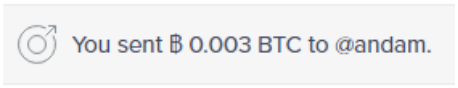
Id	Time Start	Time End	Latitude	Longitude	Location	Max Price	User Id	Status	Num Clients	Category	Winner
32	2017-05-03 01:00:00	2017-06-09 20:00:00	38.899767	22.432451	119	0.0050	19	3	5	location	Finalize Winner

Εικόνα 121 Σελίδα πληρωμής

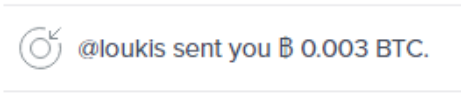
Όταν πατήσει το κουμπί Finalize Winner εμφανίζεται ο νικητής



Εικόνα 122 Εμφάνιση του νικητή ενός αιτήματος



Εικόνα 124 Μήνυμα αποστολής Bitcoins του Διαχειριστη

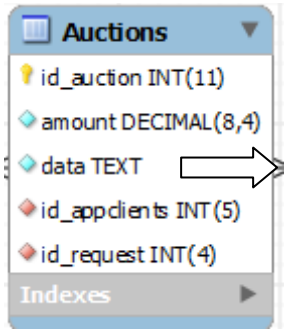


Εικόνα 123 Μήνυμα παραλαβής Bitcoins του Data producer

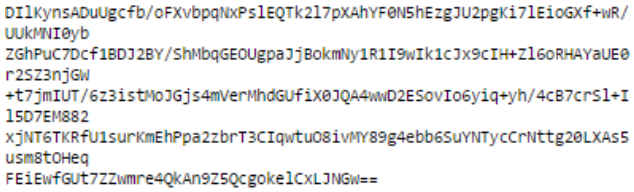
7.4.8. Αποκρυπτογράφηση

Για την αποκρυπτογράφηση έχει δημιουργηθεί μία σελίδα όπου εισάγεται η κρυπτογραφημένη πληροφορία που έχει δοθεί από στον Data consumer και μαζί με το ιδιωτικό κλειδί το οποίο διαθέτει μόνο αυτός πραγματοποιείται η αποκρυπτογράφηση. Η πληροφορία που κρυπτογραφείτε και αποκρυπτογραφείτε έχει την παρακάτω μορφή.

Κρυπτογραφημένα δεδομένα από τον πίνακα auctions



Εικόνα 125 Πίνακας Auctions



Εικόνα 126 Κρυπτογραφημένο κείμενο

Το συγκεκριμένο αίτημα αφορούσε γεωγραφικό μήκος και πλάτος οπότε στην αποκρυπτογράφηση θα εμφανιστεί η παρακάτω πληροφορία.

Εικόνα 127 Αποτέλεσμα αποκρυπτογράφησης

8. Συμπέρασμα -Μελλοντικές εφαρμογές

Παρουσιάστηκε μια βασική ιδέα του Mobile Crowd Sensing (MSC), η υλοποίηση μιας πρωτότυπης εφαρμογής ανταλλαγής δεδομένων που αξιοποιεί την δύναμη των χρηστών και των αισθητήρων σε μεγάλη κλίμακα καθώς και την πληρωμής μέσω της τεχνολογίας του Bitcoin. Θέλοντας με αυτό τον τρόπο να δείξουμε την σημασία που έχουν οι πληροφορίες από τους αισθητήρες. Επίσης γίνεται κατανοητό πως αυτοί που παράγουν τα δεδομένα δεν είναι κατά ανάγκη οι μόνοι ωφελούμενοι. Κάποιες από τις μελλοντικές χρήσεις τέτοιων συστημάτων θα μπορούσε να είναι από κατασκευαστές αυτοκινήτων οι οποίοι ενδιαφέρονται για μία σταθερή ροή των δεδομένων στο δρόμο για να λύσουν κυκλοφοριακά προβλήματα, ή στην εύρεση χώρου στάθμευσης σε κάποιο παρκινγκ ή κάποια εταιρεία πρόγνωσης καιρού και πολλά άλλα.

Με την κρυπτογράφηση και την χρησιμοποίηση της πλατφόρμας πληρωμής του ψηφιακού νομίσματος Bitcoin διασφαλίζεται

- η ασφάλεια,
- η διατήρηση της ανωνυμίας
- η εγκυρότητα
- και η ακεραιότητα των δεδομένων

ωστόσο απαιτείται περαιτέρω έρευνα για την βελτίωση της τεχνολογίας αυτής.

Μία αναβάθμιση της προτεινόμενης λύσης θα μπορούσε να είναι η μεταφορά όλης της υλοποίησης σε κάποια αποκεντρωμένη πλατφόρμα blockchain. Με αυτό τον τρόπο θα μπορούσε να εξαλειφθεί η ανάγκη για την ύπαρξη μίας έμπιστης τρίτης οντότητας. Η λειτουργικότητα που η παρουσία της προσφέρει θα μπορούσε να αντικατασταθεί από smart contracts, των οποίων η ορθή λειτουργία επιβεβαιώνεται κρυπτογραφικά.

Βιβλιογραφία- Ιστοσελίδες-Άρθρα

1. Layla Pournajaf, Daniel A. Garcia-Ulloa, Li Xiong, Vaidy Sunderam, Participant Privacy in Mobile Crowd Sensing Task Management: A Survey of Methods and Challenges
2. Manoop Talasila, Reza Curtmola, and Cristian Borcea, Mobile Crowd Sensing
3. Stylianos Gisdakis, Thanassis Giannetsos, Panos Papadimitratos, Data Verification and Privacy- Respecting User Remunation in Mobile Crowd Sensing
4. Raghu K. Ganti, Fan Ye, and Hui Lei, Mobile Crowdsensing: Current State and Futere Challenges
5. Daojng He, Sammy Chan and Mohsen Guizani, User Privacy and Data Trustworthiness in Mobile Crowd Sensing
6. Xinglin Zhang, Zheng Yang, Wei Sun, Yunhao Liu, Incentives for Mobile Crowd Sensing: A Survey
7. *Ioannis Krontiris, Marc Langheinrich, and Katie Shilton,* Trust and Privacy in Mobile Experience Sharing: Future Challenges and Avenues for Research
8. Hong Lu, Jun Yang, Zhigang Liu, Nicholas D. Lane, The Jigsaw Continuous Sensing Engine for Mobile Phone Applications
9. Kay Noyen, Dirk Vollandy, Dominic Worner and Elgar Fleisch, When Money Learns to Fly: Towards Sensing as a Service Applications Using Bitcoin
10. P. Dutta *et al.*, "Demo Abstract: Common Sense: Participatory Urban Sensing Using A Network of Handheld Air Quality Monitors," *Proc. ACM SenSys*, 2009, pp. 349–50
11. Beginning HTML, XHTML, CSS and JavaScript - Jon Duckett
12. Java_All-in-One_For_Dummies__4th_edition
13. Jeff Friesen - Learn Java for Android Development – 2013
14. B. Hull *et al.*, "Cartel: A Distributed Mobile Sensor Computing System," *Proc. SenSys*, 2006, pp. 125–38.
15. P. Mohan, V. Padmanabhan, and R. Ramjee, "Nericell: RICH monitoring of Road and Traffic Conditions Using Mobile Smartphones," *Proc. ACM SenSys*, 2008, pp. 323 –36.
16. S. Mathur *et al.*, "Parknet: Drive-by Sensing of Road-Side Parking Statistics," *Proc. ACM MobiSys*, 2010, pp. 123–36
17. S. B. Eisenman *et al.*, "The Bikenet Mobile Sensing System for Cyclist Experience Mapping," *Proc. SenSys*,Nov. 2007
18. S. Reddy *et al.*, "Image Browsing, Processing, and Clustering for Participatory Sensing: Lessons from a Dietsense Prototype," *Proc. EmNets*, 2007, pp. 13–17.
19. J. Krumm, "A Survey of Computational Location Privacy," *Personal and Ubiquitous Computing*, vol. 13, no.6, 2009, pp. 391–99.

20. L. Sweeney, “k-Anonymity: A Model for Protecting Privacy,” *Int’l. J. Uncertainty, Fuzziness, and Knowledge- Based Systems*, vol. 10, no. 5, Oct. 2002, pp. 557–70.
21. A. Yao, “Protocols for Secure Computations,” *Proc. IEEE Symp. Foundations of Comp. Sci.*, 1982, pp. 160–64
22. R. Agrawal and R. Srikant, “Privacy Preserving Data Mining,” *Proc. ACM Conf. Mgmt. of Data*, May 2000, pp. 439–50.
23. R. K. Ganti *et al.*, “Poolview: Stream Privacy for Grassroots Participatory Sensing,” *Proc. SenSys ’08*, 2008, pp. 281–94. S. Saroiu and A. Wolman, “Enabling New Mobile Applications With Location Proofs,” *Proc. HotMobile*, 2009, pp. 1–6.
24. Learning PHP, MySQL, JavaScript, CSS & HTML5 - Robin Nixon
25. PHP Cookbook, Third Edition - David Sklar & Adam Trachtenberg
26. PHP.MySQL.JavaScript.and.HTML5.All-in-One.For.Dummies
27. Android Programming Guide - Android App Development Learn In A Day! by OS Swift{2nd edition}
28. Mastering Bitcoin [e-book] O’ Reilly Media. Available at http://cdn.oreillystatic.com/oreilly/booksamplers/9781449374044_sampler.pdf
29. The Bitcoin Revolution: The history, mystery and what it all means. Anderson, B. (2014)
30. How to buy, trade and profit with Bitcoin: A jump-start guide Edelson, R. (2014)
31. Bitcoin and the future of money. Pagliery, J. (2014)
32. Virtual currency: The Bitcoin guide Roy, L. (2012)
33. Ανάλυση της αρχιτεκτονικής του Android http://developer.android.com/guide/basics/what-is-android.html
34. https://en.wikipedia.org/wiki/Android_(operating_system)#Linux_kernel
35. Πληροφορίες σχετικά με τη δομή και τη χρήση του αρχείου AndroidManifest.xml http://developer.android.com/guide/topics/manifest/manifest-intro.html
36. Οι φάσεις του σχεδιασμού μιας εφαρμογής http://developer.android.com/guide/developing/index.html
37. https://en.wikipedia.org/wiki/Android_(operating_system)
38. https://en.wikipedia.org/wiki/Android_(operating_system)#Software_stack
39. http://www.w3schools.com/
40. https://en.wikipedia.org/wiki/Android_software_development

41. Η κοινότητα προγραμματιστών “Stack Overflow” παρέχει ένα μέρος για ερωτήσεις και απαντήσεις στα περισσότερα προβλήματα που μπορεί να συναντήσει ένας developer. www.stackoverflow.com
42. www.slimframework.com
43. www.rsa.com
44. https://en.wikipedia.org/wiki/PHP
45. https://en.wikipedia.org/wiki/HTML
46. https://en.wikipedia.org/wiki/XAMPP
47. https://en.wikipedia.org/wiki/MySQL
48. https://en.wikipedia.org/wiki/Java_(programming_language)
49. https://en.wikipedia.org/wiki/JavaScript
50. https://en.wikipedia.org/wiki/Internet_of_things
51. https://en.wikipedia.org/wiki/RSA_(cryptosystem)
52. https://en.wikipedia.org/wiki/Android_version_history
53. https://en.wikipedia.org/wiki/Cryptocurrency
54. https://en.wikipedia.org/wiki/Bitcoin
55. https://en.wikipedia.org/wiki/Crowdsensing
56. https://en.wikipedia.org/wiki/Cascading_Style_Sheets
57. https://en.wikipedia.org/wiki/JQuery
58. https://en.wikipedia.org/wiki/JSON
59. http://www.wolframalpha.com
60. http://www.eett.gr
61. http://www.cryptosys.net
62. www.rsasecurity.com
63. www.cryptogram.gr
64. www.securitymanager.gr
65. www.itl.nist.gov
66. www.ierf.org
67. https://bitcoinx.gr/
68. http://www.naftemporiki.gr/
69. https://www.cryptovest.co.uk/
70. http://json.org/
71. https://uphold.com/