

 SRS_v1.md

Software Requirements Specification document (SRS)

Έγγραφο απαιτήσεων λογισμικού

ΣΗΜΜΥ ΕΜΠ

Τεχνολογία Λογισμικού - Ροή Α

Ομάδα: "Mika Shady"

Μέλη:

- Μιχαήλ Μελέτιος 031 16 677
- Σταυρακάκης Κωνσταντίνος 031 16 155
- Γιάγκος Δημήτριος 031 16 302

1. Εισαγωγή

- 1.1 Σκοπός Λογισμικού
- 1.2 Διεπαφές
 - 1.2.1 Διεπαφές με εξωτερικά χαρακτηριστικά
 - 1.2.2 Διεπαφές με το χρήστη

2. Αναφορές και πηγές πληροφοριών

3. Προδιαγραφές Απαιτήσεων Λογισμικού

- 3.1 Περιπτώσεις Χρήσης
 - 3.1.1 Περίπτωση Χρήσης 1η
 - 3.1.1.1 Χρήστες που εμπλέκονται
 - 3.1.1.2 Προυποθέσεις εκτέλεσης
 - 3.1.1.3 Περιβάλλον εκτέλεσης
 - 3.1.1.4 Δεδομένα Εισόδου
 - 3.1.1.5 Παράμετροι
 - 3.1.1.6 Αλληλουχία Ενεργειών-Προσδοκόμενη συμπεριφορά
 - 3.1.2 Περίπτωση Χρήσης 2η
 - 3.1.2.1 Χρήστες που εμπλέκονται
 - 3.1.2.2 Προυποθέσεις εκτέλεσης
 - 3.1.2.3 Περιβάλλον εκτέλεσης
 - 3.1.2.4 Δεδομένα Εισόδου
 - 3.1.2.5 Παράμετροι
 - 3.1.2.6 Αλληλουχία Ενεργειών-Προσδοκόμενη συμπεριφορά
- 3.2 Απαιτήσεις Οργάνωσης Δεδομένων
 - 3.2.1 Περιγραφή Δεδομένων
 - 3.2.2 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα
 - 3.2.2.1 Στοιχεία Σταθμών

- 3.2.2.2 Δεδομένα Χρηστών
- 3.2.2.3 Δεδομένα Φόρτισης
- 3.2.2.4 Δεδομένα Οχήματος
- 3.2.3 Προδιαγραφές διατήρησης δεδομένων
- 3.3 Περιορισμοί σχεδίασης
- 3.4 Λοιπές Απαιτήσεις
 - 3.4.3 Απαιτήσεις Διαθεσιμότητας
 - 3.4.2 Απαιτήσεις Συντήρησης
 - 3.4.3 Απαιτήσεις Ασφάλειας
- 3.5 Απαιτήσεις Επιδόσεων

1. Εισαγωγή

1.1 Σκοπός λογισμικού

Το project μας έχει ως στόχο την εξυπηρέτηση και διευκόλυνση των ιδιοκτητών ηλεκτρικών οχημάτων αλλά και των ιδιοκτητών σταθμών φόρτισης ηλεκτρικών οχημάτων. Στην εφαρμογή που θα αναπτυχθεί θα έχει οι stakeholders θα έχουν πρόσβαση μέσω οποιουδήποτε browser.

Βασικός σκοπός της πλατφόρμας είναι να βοηθά του οδηγούς σ' όλα τα στάδια της διαδικασίας φόρτισης του οχήματος τους. Εύκολα και γρήγορα θα μπορούν να βρίσκουν σταθμό φόρτισης, ο οποίος θα έχει προκύψει από παραμέτρους που οι ίδιοι έχουν θέσει κατά την αναζήτηση. Επίσης, οι χρήστες της πλατφόρμας θα δέχονται υπηρεσίες διευκόλυνσης και κατά την διάρκεια της φόρτισης αλλά και κατά την πληρώμη, όπου θα έχουν την δυνατότητα να ολοκληρώσουν με διάφορους τρόπους. Τέλος, οι χρήστες θα έχουν την δυνατότητα να αξιολογούν την παραπάνω παροχή υπηρεσιών με στόχο την συλλογή κριτικών ώστε η εμπειρία των ίδιων να γίνεται καλύτερη και να διευκολύνουν τους ιδιοκτήτες των σταθμών στην εύρεση "αρνητικών" στοιχείων.

1.2 Διεπαφές

1.2.1 Διεπαφές με εξωτερικά χαρακτηριστικά

Προκειμένου να επιτευχθεί η όσο το δυνατόν καλύτερη εξυπηρέτηση των χρηστών της εφαρμογής, κρίνεται απαραίτητη η χρήση κάποιων εξωτερικών διεπαφών, πιο συγκεκριμένα κάποιων APIs τα οποία παρέχουν χρήσιμες πληροφορίες κυρίως για τη θέση των χρηστών στο χάρτη. Συγκεκριμένα τα APIs είναι τα εξής:

- **OpenStreetMaps (OSM) Maps API by Nominatim:** Πρόκειται για ένα API που προσφέρει την προβολή χαρτών στο χρήστη. Με αυτόν τον τρόπο ο χρήστης μπορεί να δει ποιοι σταθμοί φόρτισης βρίσκονται τριγύρω του και που ακριβώς βρίσκονται στο χάρτη
- **OpenStreetMaps (OSM) Geocoding API by Nominatim:** Πρόκειται για API που επιτρέπει την ακριβή αντιστοίχιση τοπωνυμίων με συντεταγμένες (και το αντίστροφο). Χρησιμοποιείται κυρίως για μέτρηση αποστάσεων ώστε να γίνουν οι καλύτερες δυνατές προτάσεις στους χρήστες όσον αφορά τους κοντινότερους σταθμούς φόρτισης.

Εκτός από τις εξωτερικές διεπαφές το σύστημα μας λειτουργεί με την υποστήριξη των εξής εσωτερικών διεπαφών:

1. MongoDB Community server : Πρόκειται για σύστημα με το οποίο διαχειριζόμαστε τη βάση δεδομένων του συστήματος μας. Επιλέχθηκε η MongoDB μιας και καλύπτει με άνεση τις όχι ιδιαίτερα μεγάλες απαιτήσεις του συστήματος από άποψη βάσης δεδομένων
2. Apache Tomcat Server : Για να επικοινωνήσει η εφαρμογή με τον έξω κόσμο χρειαζόμαστε έναν webserver. Η βασική μας απαίτηση είναι να μπορεί να εξυπηρετήσει requests που έρχονται από τον παγκόσμιο ιστό.
3. React Server : Για το development χρησιμοποιήθηκε ο server του React ωστόσο για το deployment της εφαρμογής δεν είναι απαραίτητη η χρήση του

Το λογισμικό παρέχεται στους χρήστες με 2 τρόπους-διεπαφές:

- **Front-end Εφαρμογή** : Μέσα απο τον ιστότοπο greencurrent.gr ο χρήστης θα έχει τη δυνατότητα να διαμορφώσει το προφίλ του, να αναζητήσει τους κοντινότερους σταθμούς φόρτισης, να πραγματοποιήσει πληρωμές φόρτισης και να δει στατιστικά των φορτίσεων του.
- **RESTful API**: Το σύστημά μας παρέχει ένα endpoint για διαπιστευμένη περιορισμένη πρόσβαση στη βάση δεδομένων μας, μέσω ενός REST API. Το REST API χρησιμοποιεί HTTP requests για την επικοινωνία των δύο προγραμμάτων. Το API εξυπηρετεί τον μορφότυπο .json ή .xml . Ενώ για να πάρει authorization ο χρήστης πρέπει αρχικά να κάνει ένα POST requests με τα στοιχεία του ώστε να του χορηγηθεί ένα token το οποίο θα "περνάει" μετά σε κάθε νέο του API request. Μόλις κάνει logout ο χρήστης το token θα λήγει.

Όσα περιγράφηκαν παραπάνω φαίνονται στο παρακάτω UML component διάγραμμα

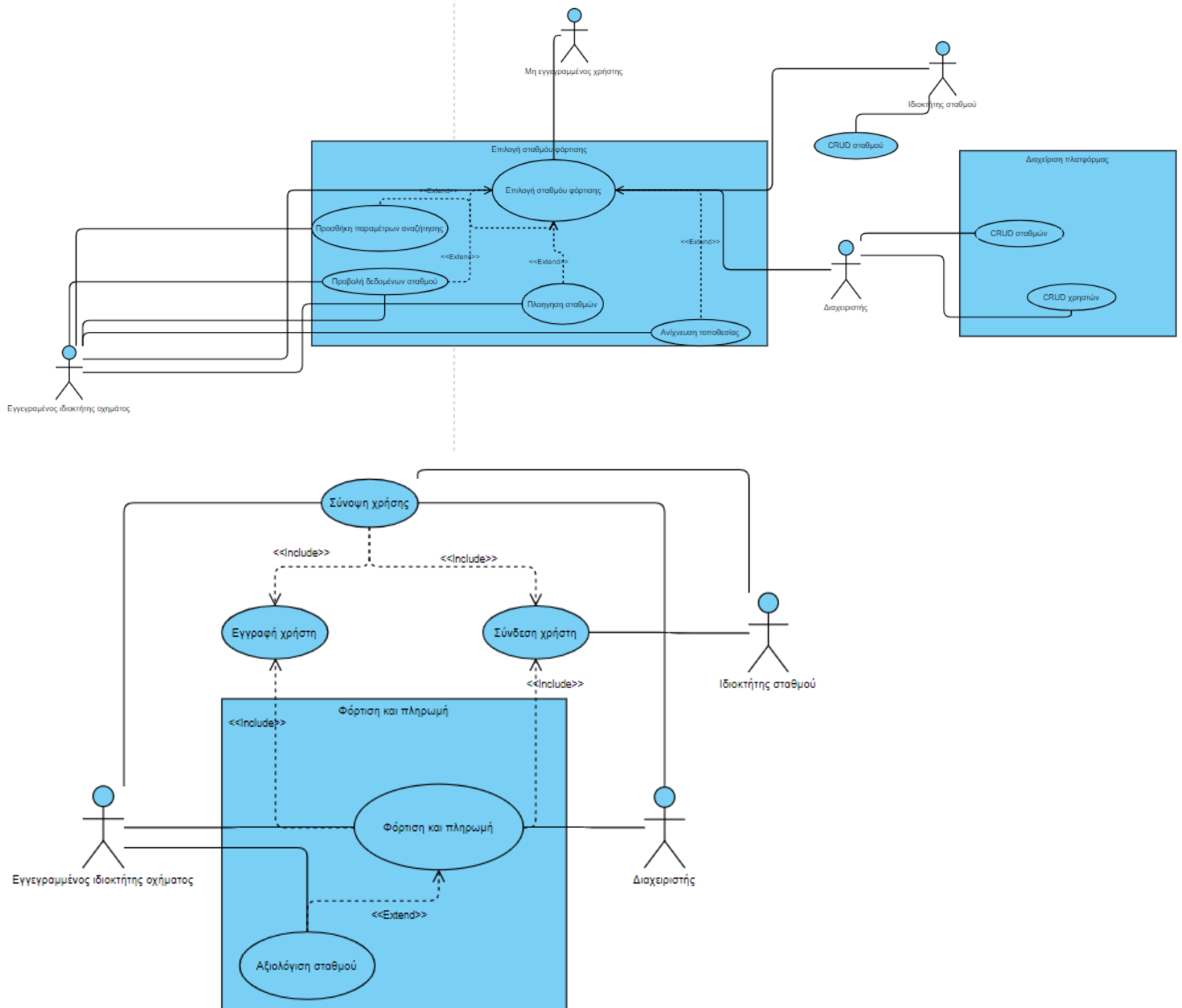
1.2.2 Διεπαφές με τον χρήστη

Στην εφαρμογή μας, ο διαχωρισμός των χρηστών θα έχει ως εξής: **1. Εγγεγραμμένοι χρήστες** Είναι οι χρήστες που έχουν ήδη πρόσβαση στο front-end και στο RESTful API επειδή έχουν δημιουργήσει λογαριασμό στο παρελθόν. Μπορούν να χρησιμοποιήσουν την εφαρμογή για αναζήτηση και πλοήγηση των σταθμών αλλά ταυτόχρονα έχουν πρόσβαση σε δικά τους στατιστικά στοιχεία που έχουν δημιουργήσει από προηγούμενες χρήσεις της εφαρμογής. Επιπλέον, μπορούν να αξιολογούν ένα σταθμό. Σημειώνεται ότι εγγεγραμμένος χρήστης μπορεί να είναι είτε ένας ιδιοκτήτης οχήματος είτε ένας ιδιοκτήτης σταθμού όπου ο καθένας θα έχει πρόσβαση σε διαφορετικές δυνατότητες της εφαρμογής μας.

2. Μη εγγεγραμμένοι χρήστες Πρόκειται για χρήστες της εφαρμογής που δεν διαθέτουν λογαριασμό χρήστη και μπορούν να χρησιμοποιούν το front-end της εφαρμογής μόνο για αναζήτηση και πλοήγηση στους σταθμούς φόρτισης.

3. Διαχειριστές Πρόκειται για τους χρήστες του back-end μας που είναι υπεύθυνοι για τη διαχείριση των εγγεγραμμένων χρηστών. Αυτοί έχουν τη δυνατότητα να διαγράφουν χρήστες και να τροποποιούν τις καταχωρήσεις του συστήματος όταν αυτό κρίνεται αναγκαίο. Επίσης, οι διαχειριστές έχουν και δικαιώματα χρήστη προφανώς.

Από κάτω παρουσιάζουμε και 2 UML Use Cases διαγράμματα.



2. Αναφορές και πηγές πληροφοριών

3. Προδιαγραφές απαιτήσεων λογισμικού

3.1 Περιπτώσεις χρήσης

3.1.1 1η περίπτωση χρήσης: Επιλογή σταθμού φόρτισης

3.1.1.1 Χρήστες που εμπλέκονται

Στην συγκεκριμένη λειτουργία έχουν πρόσβαση τόσο οι εγγεγραμμένοι όσο και οι μη εγγεγραμμένοι χρήστες, αλλά και οι ιδιοκτήτες σταθμών.

3.1.1.2 Προυποθέσεις εκτέλεσης

Η μόνη απαίτηση είναι η σύνδεση του χρήστη στο διαδίκτυο. Για προτάσεις σταθμών, απαιτείται ο χρήστης να εισάγει την τοποθεσία του είτε χειροκίνητα είτε δίνοντας πρόσβαση σε υπηρεσία γεω-εντοπισμού.

3.1.1.3 Περιβάλλον εκτέλεσης

- Web application
- REST Api

3.1.1.4 Δεδομένα εισόδου

Φίλτρα:

Column	Data type
Γεωγραφικές συντεταγμένες	Decimal Field
Ακτίνα επιλογής	Decimal Field
Τύπος φορτιστή	Varchar
Τύπος σταθμού	Varchar
Προτεινόμενη αλυσίδα	Varchar
Μέγιστη τιμή ανά KWh	Decimal

Επιλογές ταξινόμησης σταθμών, με βάση:

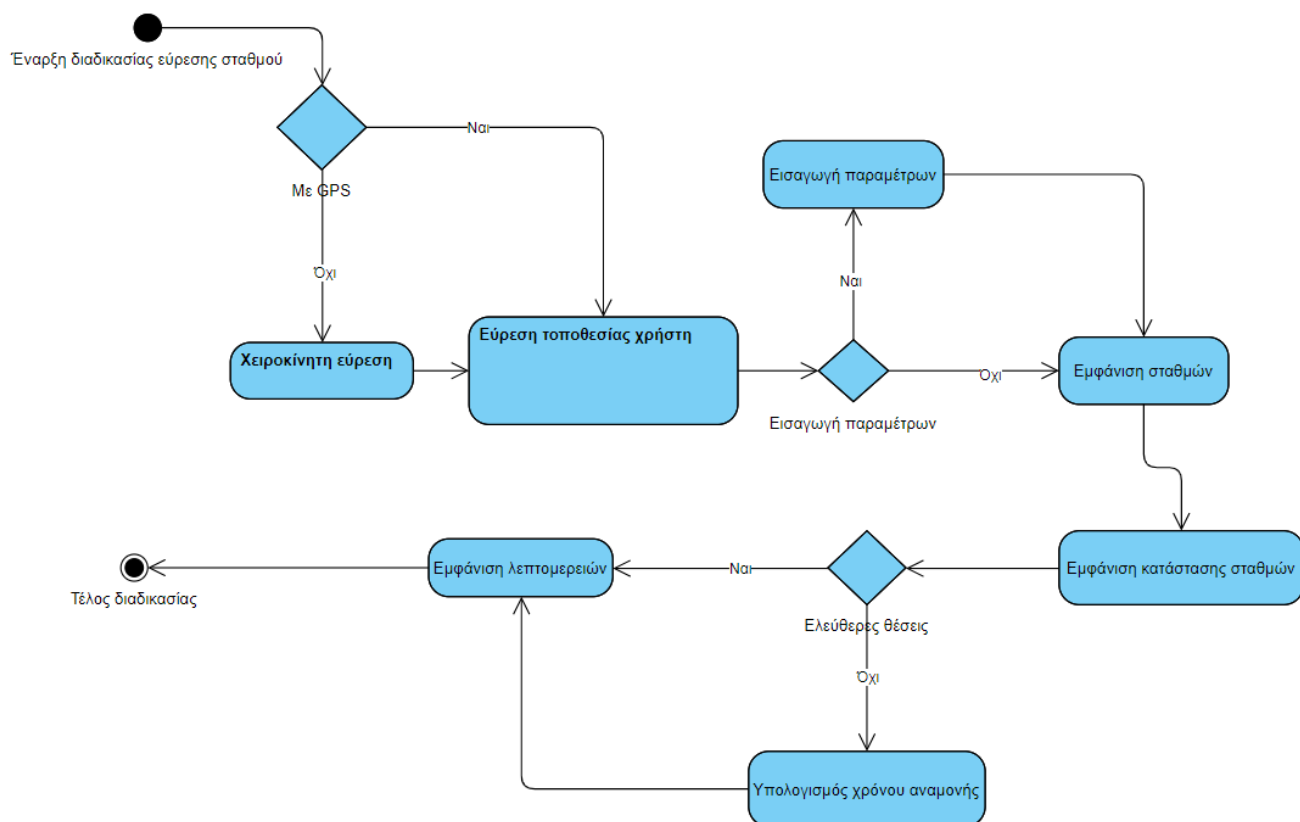
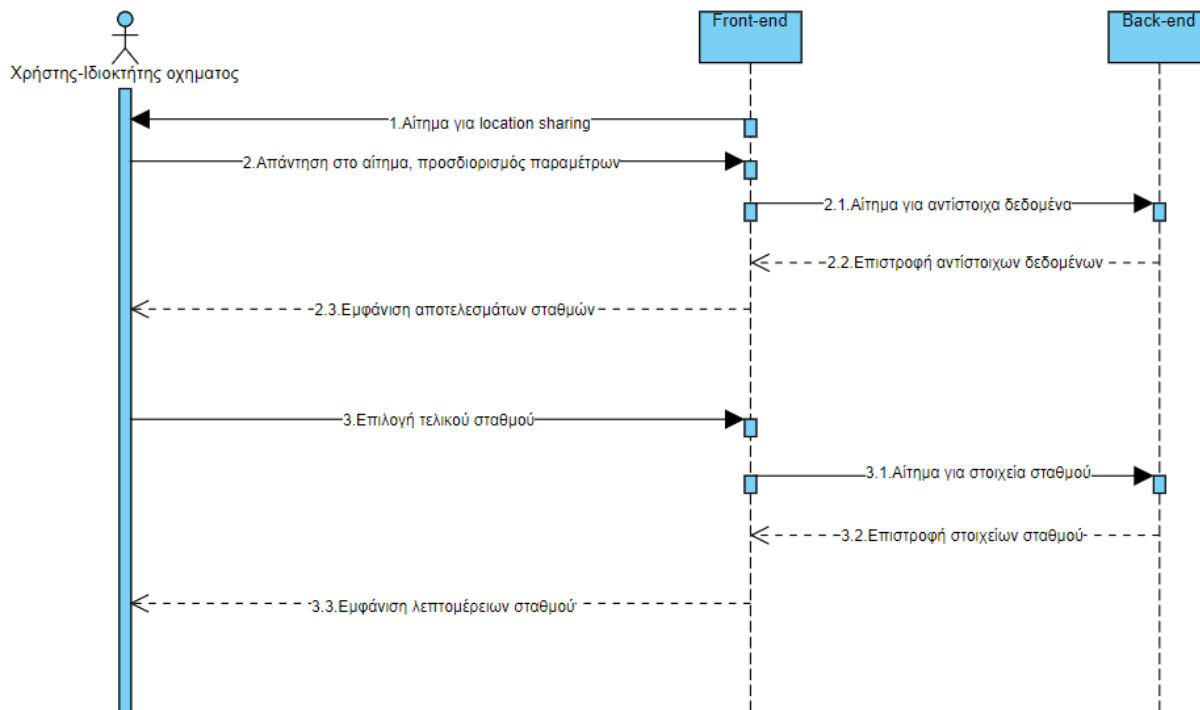
- Απόσταση
- Αξιολόγηση

3.1.1.5 Παράμετροι

Column	Data type	Constraints
Γεωγραφικές συντεταγμένες	Decimal Field	Έγκυρες γεωγρ. συντ
Ακτίνα επιλογής	Decimal Field	> = 0
Τύπος φορτιστή	Varchar	Επιλογή από υπάρχοντες
Τύπος σταθμού	Varchar	Επιλογή από υπάρχοντες
Προτεινόμενη αλυσίδα	Varchar	Επιλογή από υπάρχουσες
Μέγιστη τιμή ανά KWh	Decimal	> =0

3.1.1.6 Αλληλουχία ενεργειών - προσδοκόμενη συμπεριφορά

1. Πρώτα απ'όλα ο χρήστης θα ρωτάται από την εφαρμογή αν θέλει να μοιράσει την τοποθεσία του. Αν ο ίδιος δεν επιθυμεί location sharing τότε θα υπάρχει η δυνατότητα να τοποθετήσει χειροκίνητα την τοποθεσία του την δεδομένη στιγμή.
2. Ο χρήστης τώρα μπορεί να προσθέσει διάφορες παραμέτρους στην αναζήτηση του, οι οποίες αναφέρθηκαν παραπάνω και μπορεί να υποβάλλει το αίτημα του.
3. Αφού υποβληθεί το αίτημα του χρήστη, ανάλογα με το αν υπάρχουν ελεύθερες θέσεις φόρτισης εκείνη την στιγμή, θα έχουμε είτε εμφάνιση των πλησιέστερων σταθμών που πληρούν τις προϋποθέσεις που έχει ζητήσει ο χρήστης είτε θα εμφανίζεται ο ελάχιστος εκτιμώμενος χρόνος αναμονής έως ότου αδειάσει κάποια θέση στον εκάστοτε σταθμό.
4. Με βάση τις παραπάνω επιλογές, τελικά ο χρήστης θα επιλέγει τον οριστικό σταθμό που προτιμά. Ακολουθούν Activity και sequence Διαγράμματα



3.1.2 2η περίπτωση χρήσης: Φόρτιση και Πληρωμή

3.1.1.1 Χρήστες που εμπλέκονται

Μόνο οι εγεγραμμένοι χρήστες μπορούν να χρησιμοποιήσουν αυτή τη λειτουργία τόσο οι χρήστες όσο και οι ιδιοκτήτες των σταθμών

3.1.1.2 Προυποθέσεις εκτέλεσης

Για να εκτελεστεί αυτή η λειτουργία πρέπει οι ιδιοκτήτες αυτοκινήτων, αλλά και οι ιδιοκτήτες των σταθμών φόρτισης να κάνουν Login στο λογαριασμό τους

3.1.1.3 Περιβάλλον εκτέλεσης

- Web Application
- RESTful API

3.1.1.4 Δεδομένα Εισόδου

Column	Data Type
VehicleID	Integer Field
UserID	Integer Field
ChargingStationID	Integer Field
ConnectionStartTime	Date Field
ConnectionEndTime	Date Field
ChargingProgram	Char Field
Cost	Double Field
PaymentType	Char Field
Rating	Integer Field

3.1.1.5 Παράμετροι

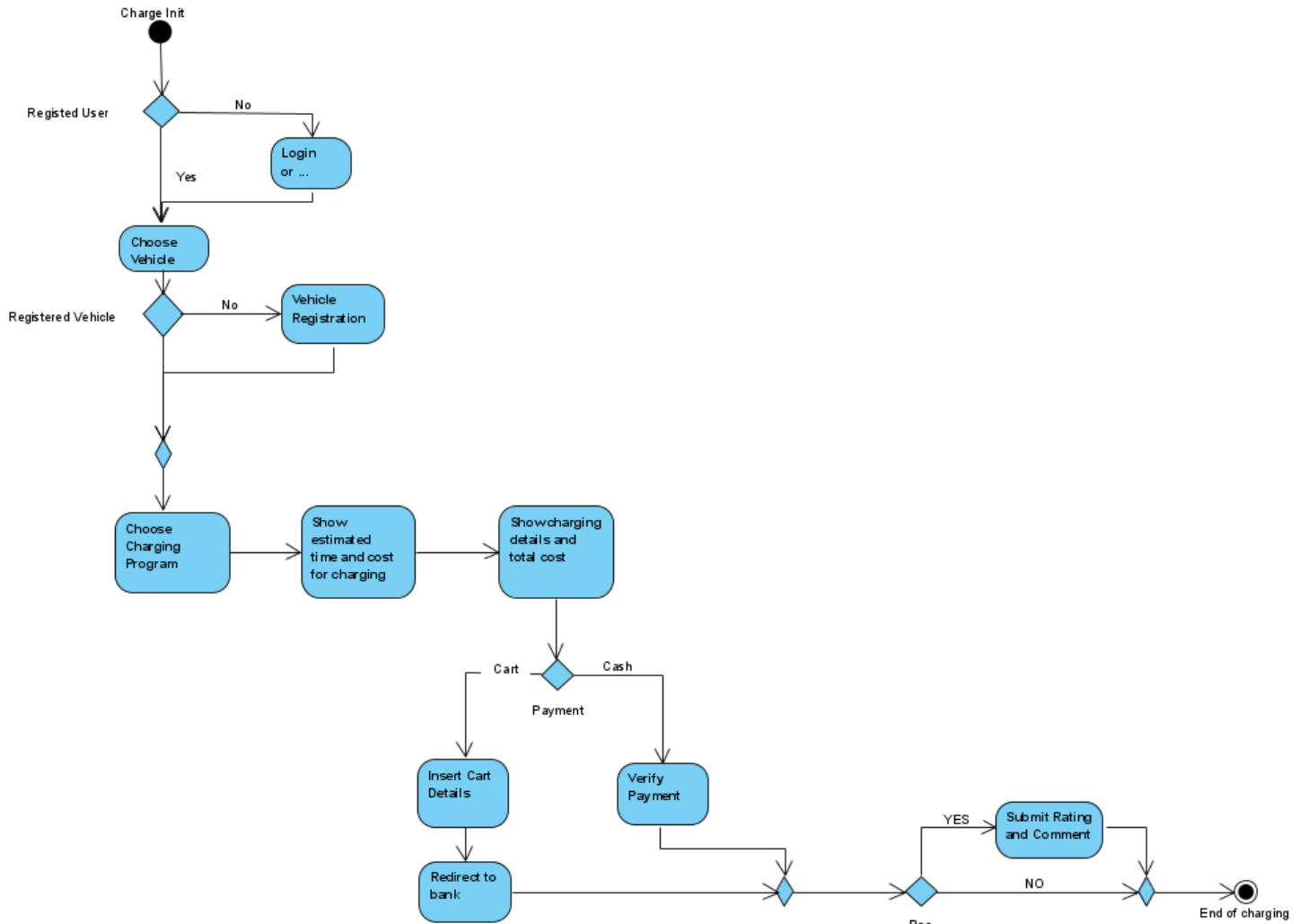
Πεδίο	Τύπος Δεδομένων	Περιορισμοί
VehicleID	Integer Field	Valid Vehicle
UserID	Integer Field	Valid User
ChargingStationID	Integer Field	Valid Station
ConnectionStartTime	Date Field	Valid Date
ConnectionEndTime	Date Field	Valid Date
ChargingProgram	Char Field	Choose from the provided choices
Cost	Double Field	>0
PaymentType	Char Field	Cart/Cash/Bill
Rating	Integer Field	0 to 5 Stars

3.1.1.6 Αλληλουχία ενεργειών-επιθυμητή συμπεριφορά

1. Εισαγωγή ID αυτοκινήτου
2. Επιλογή προγράμματος χρέωσης
3. Εκτίμηση Χρόνου φόρτισης και κόστους
4. Ολοκλήρωση φόρτισης εμφάνιση τελικού κόστους
5. Επιλογή τρόπου πληρωμής

6. Αξιολόγηση

Ακολουθεί το Activity UML διάγραμμα



3.2 Απαιτήσεις Οργάνωσης Δεδομένων

3.2.1 Περιγραφή Δεδομένων

Τα δεδομένα που διαχειρίζεται το λογισμικό αποθηκεύονται μέσω κατάλληλων μοντέλων στη βάση δεδομένων. Η βάση δεδομένων που επιλέχθηκε να χρησιμοποιηθεί είναι η MongoDB, όπου με queries πάνω στα collections θα αντλούμε την επιθυμητή πληροφορία ανάλογα με τα operations που θέλουμε να κάνουμε κάθε φορά. Ακόμα τα μοντέλα θα περιέχουν γεωγραφικά δεδομένα, ενώ επίσης στα media files της εφαρμογής θα υπάρχουν στατικές φωτογραφίες για την μορφοποίηση των web pages. Η καταχώρηση των δεδομένων μπορεί να γίνει μέσω φόρμών από τη web εφαρμογή ή μέσω POST request στο REST API. Αποθήκευση των δεδομένων γίνεται στο σκληρό δίσκο της εφαρμογής.

3.2.2 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα

3.2.2.1 Στοιχεία Σταθμών

Για κάθε σταθμό θα διατίθενται ανοιχτά σε όλους τους χρήστες τα στοιχεία που αφορούν το σταθμό αυτό, όπως θέση του σταθμού στο χάρτη, αξιολογήσεις σταθμού, διαθέσιμες θέσεις φόρτισης σταθμού, χαρακτηριστικά φόρτισης των θέσεων του σταθμού (Watt, A, και άλλα τεχνικά στοιχεία). Ενώ ο ιδιοκτήτης κάθε σταθμού φόρτισης θα μπορεί να βλέπει και στατιστικά στοιχεία που αφορούν το σταθμό του, όπως σύνολο φορτίσεων που έγιναν σε κάποια περίοδο, κόστος φορτίσεων, μέσος χρόνος φόρτισης, "busy time intervals" μέσα στη μέρα κλπ.

3.2.2.2 Δεδομένα Χρηστών

Δεδομένα	Διαχειριστής	Χρήστης	3ος Χρήστης
Username	RW	RW	R
Password	W	W	-
email	RW	RW	-
First Name	RW	RW	-
Last Name	RW	RW	-
Address	RW	RW	-
Date of Birth	RW	RW	-
Ο κάθε χρήστης λοιπόν θα μπορεί να δει και να κάνει edit τα προσωπικά του στοιχεία με εξαίρεση το Password το οποίο θα είναι hashed μέσα στη βάση δεδομένων. Ο διαχειριστής έχει και αυτός πρόσβαση στα στοιχεία του χρήστη (ίσως περιοριστεί αυτό αργότερα), ενώ ο εξωτερικός χρήστης, δηλαδή κάποιος άλλος user της εφαρμογής είτε ανώνυμος είτε επώνυμος θα μπορεί να δει μονάχα το username ενός χρήστη.			

3.2.2.3 Δεδομένα Φόρτισης

Τα δεδομένα φόρτισης θα είναι ορατά σε αναλυτική μορφή στον ιδιοκτήτη του οχήματος που έγινε η φόρτιση καθώς και στον ιδιοκτήτη του σταθμού στην οποία έγινε η φόρτιση. Ακόμα τα δεδομένα αυτά δεν θα μπορούν να αλλάξουν μετά το πέρας της φόρτισης, ενώ επίσης ο χρήστης αν επιθυμεί θα μπορεί να επιτρέψει τη χρήση τους από την εφαρμογή για την δημιουργία στατιστικών που αφορούν την αγορά φόρτισης ηλεκτρικών αυτοκινήτων ολιστικά.

3.2.2.4 Δεδομένα Οχήματος

Κάθε όχημα φέρει μοναδικό κωδικό ID το οποίο συνδέεται με τον εκάστοτε ιδιοκτήτη. Μονάχα αυτός θα μπορεί να βλέπει τα στοιχεία του οχήματος του και όχι οι υπόλοιποι χρήστες της εφαρμογής. Ενώ ακόμα τα στοιχεία κάθε οχήματος θα μπορούν να βλέπουν οι κάτοχοι των σταθμών φόρτισης, μονάχα όμως δεδομένα που τους αφορούν, και όχι στοιχεία που σχετίζονται με τον ιδιοκτήτη.

3.2.3 Προδιαγραφές Διατήρησης Δεδομένων

Για τα δεδομένα που εισάγονται στο σύστημα μας λαμβάνουμε τις εξής αποφάσεις

1. Χρήστες που παραμένουν ανενεργοί για περισσότερο από 12 μήνες θα διαγράφονται με 2 month notice.
2. Expired token, σε περίπτωση υποκλοπής σε σύντομο χρονικό διάστημα να απαιτείται ανανέωση, περιορίζοντας το χρόνο που θα έχει κάποιος κακόβουλος χρήστης
3. Refresh token που απαιτεί πιστοποίηση εγκυρότητας του πελάτη για ασφάλεια.
4. Αποθήκευση στον Session Manager που παρέχει η εφαρμογή των Access και Refresh tokens στο αντίστοιχο session id. Χρήση του access token για πρόσβαση σε πόρους και αν έχει λήξει τότε χρησιμοποιούμε Refresh token.
5. Χρήστης παραμένει συνδεδεμένος μετά την πρώτη σύνδεση, ενώ αν δεν έχει υπάρξει αλληλεπίδραση του χρήστη με το σύστημα για 1 εβδομάδα ο χρήστης αποσυνδέεται αυτόματα και απαιτείται εκ νέου σύνδεση για χρήση της εφαρμογής.

3.3 Περιορισμοί σχεδίασης

Ακολουθούν τα τεχνολογικά πρότυπα που έχουν επιλεγεί για την ανάπτυξη της εφαρμογής

Backend

Tool	Operation
MongoDB	Database Construction

Tool	Operation
JPA-Hibernate(Java 1.8)	Database Connection/Operations
Spring Boot	Protocol for Deployment
OpenStreetMaps API	User's location Data

Frontend

Tool	Operation
React.js	Retrieve Data from API
Bootstrap v4.1.3	Web Page Functionality and Layout
Ink (in conjunction with react)	CLI construction

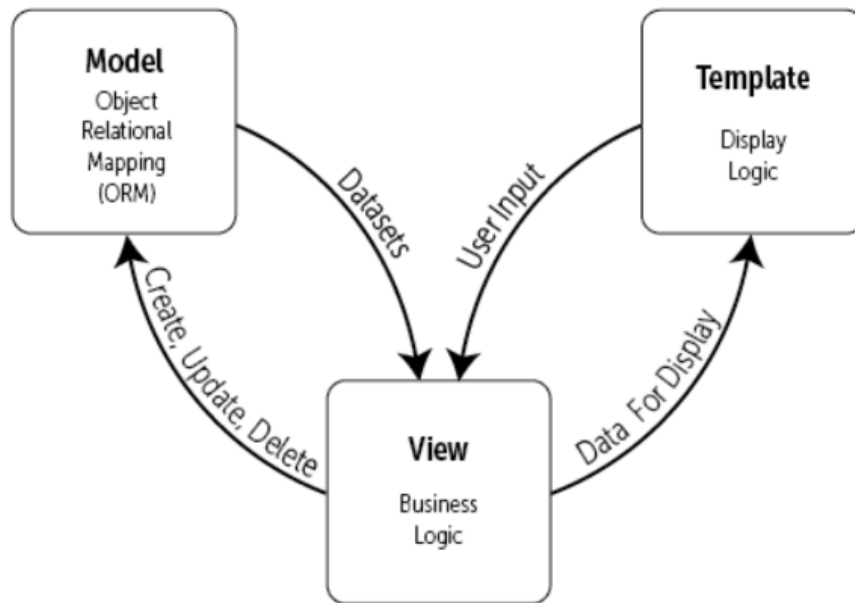
Deployment

Tool	Operation
Apache Tomcat	Deploy Backend
npm	React Deployment

Extras

Tool	Operation
Spock	Unit-Testing
GNU Make	Build Automation

Το πρότυπο σχεδίασης γύρω από το οποίο έγινε η ανάπτυξη της εφαρμογής είναι το MVT (Model-View-Template)



3.4 Λοιπές Απαιτήσεις

3.4.1 Απαιτήσεις διαθεσιμότητας λογισμικού

Σενάριο	Χρήστης	Αναγνώστης	Διαχειριστής
Χρήση Front-end ,εύρεση σταθμού	YES	YES	YES
Πραγματοποίηση Φόρτισης μέσω εφαρμογής	YES	NO	YES
Διαχείριση Προφίλ	YES	NO	YES
Διαχείριση Λογαριασμών	NO	NO	YES
Χρήση API	YES	NO	YES

3.4.2 Απαιτήσεις Συντήρησης

- Αναπροσαρμογές κώδικα για παραχώρηση ή περιορισμό δικαιωμάτων των χρηστών ,καθώς και για αντιμετώπιση πιθανών προβλημάτων
- Συχνή ανανέωση των SSL Certificates για να προστατευθεί η ανταλλαγή ευαίσθητων πληροφοριών μεταξύ χρήστη και συστήματος.
- Συχνός έλεγχος για δεδομένα που είναι απαρχαιωμένα στη βάση
- Ανταπόκριση στις κριτικές των χρηστών για αλλαγές ή προσθήκες στις περιπτώσεις χρήσης της εφαρμογής

3.4.3 Απαιτήσεις Ασφάλειας

Ιδιαίτερη προσοχή θα δοθεί στον τομέα της ασφάλειας των προσωπικών στοιχείων. Θα χρησιμοποιηθεί ειδική δομή δεδομένων της java που προσφέρει hashing για τους κωδικούς πρόσβασης, ενώ το σύστημα δεν θα ξαναζητήσει ποτέ απο το χρήστη κατά τη διάρκεια της χρήσης της εφαρμογής να επιβεβαιώσει εκ νέου τα στοιχεία του προκειμένου να αποφευχθούν πιθανές υποκλοπές δεδομένων. Παρόμοιοι περιορισμοί έχουν τεθεί και στις λειτουργίες του REST API οι οποίες απαιτούν user authentication για να εκτελεστούν.

3.5 Απαιτήσεις Επιδόσεων

Το λογισμικό μας θα είναι συνεχώς διαθέσιμο σε αυτούς που επιθυμούν να το χρησιμοποιήσουν. Σε ώρες αιχμής αναμένεται φορτίο 300 ταυτόχρονα συνδεδεμένων χρηστών. Ενώ το REST API θα μπορεί να καλύψει έως και 100 requests το δευτερόλεπτο προς τη βάση μας