Έγγραφο απαιτήσεων λογισμικού (SRS)

ΠΡΟΣΑΡΜΟΓΗ ΤΟΥ ΑΝΤΙΣΤΟΙΧΟΥ ΕΓΓΡΑΦΟΥ ΤΟΥ ΠΡΟΤΥΠΟΥ ISO/IEC/IEEE 29148:2011

**Charge point helper**

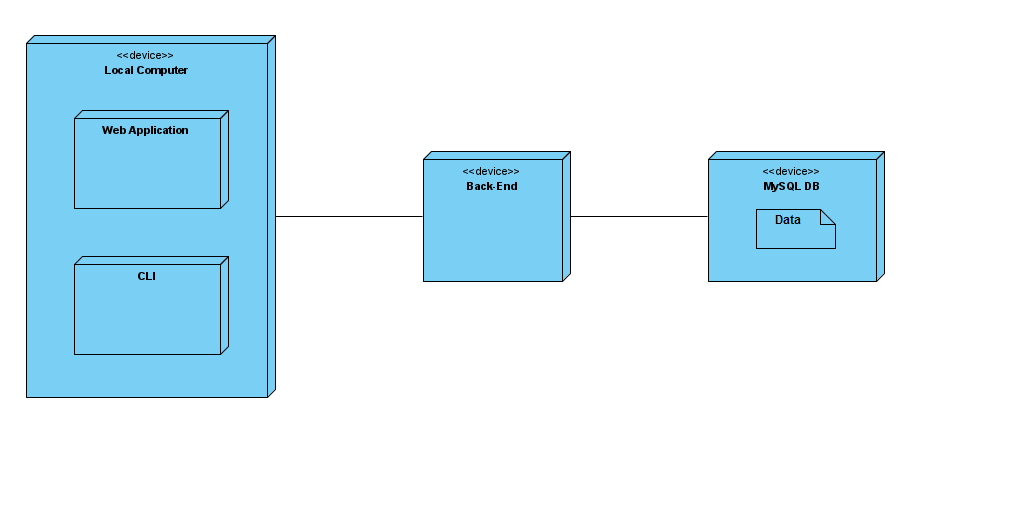
1.1 Εισαγωγή: σκοπός του λογισμικού

Σκοπός της δημιουργίας του project Charge point helper είναι η υποστήριξη των σταθμών φόρτισης ηλεκτρικών οχημάτων σε όλη την Ελλάδα με εύχρηστο προς τον χρήστη λογισμικό που θα επιτρέπει την σωστή λειτουργεία τους. Πιο συγκεκριμένα το Charge point helper θα υπάρχει σε κάθε φορτιστή με την μορφή web application και θα δίνει την ικανότητα στον κάτοχο του οχήματος να επιλέγει πόση φόρτιση θα ήθελε να βάλει στο όχημα του καθώς και το πρόγραμμα χρέωσης που θα θέλει να επιλέξει, επιπλέον στο τέλος της φόρτισης θα παρέχει την δυνατότητα πληρωμής με κάρτα. Επιπροσθέτως το Charge point helper με την άδεια του χρήστη θα έχει την δυνατότητα να συλλέγει πληροφορίες για τα στοιχεία του οχήματος και του κάθε συμβάντος φόρτισης.

1.2 Διεπαφές (interfaces)

1.2.1 Διεπαφές με εξωτερικά συστήματα

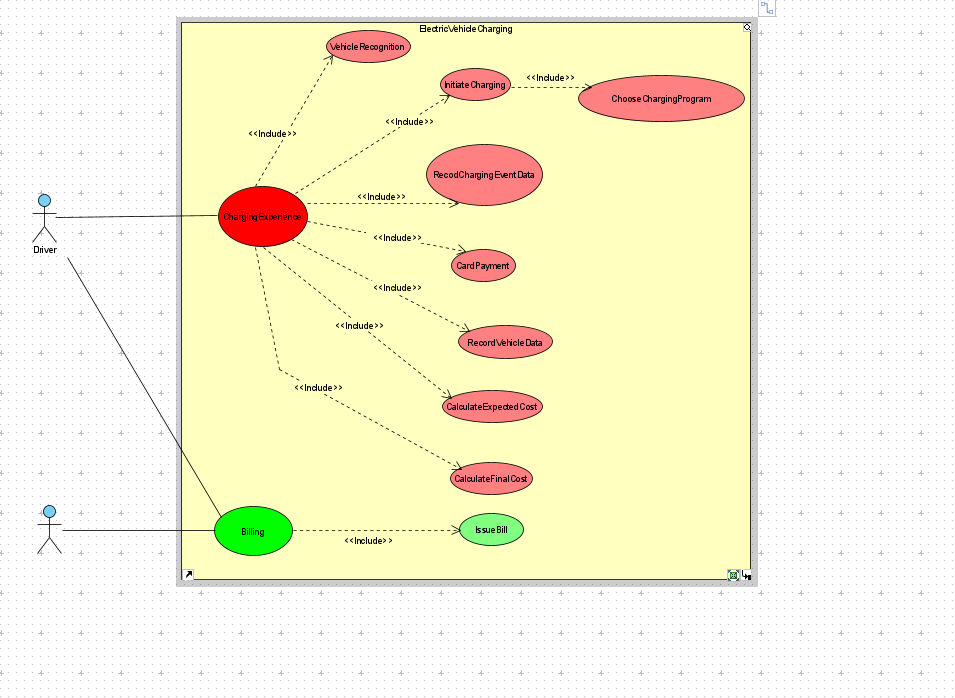
Το σύστημα μας προκείμενου να προσφέρει ασφάλεια στους χρήστης που θα κάνουν πληρωμές με κάρτα χρησιμοποίει Web banking API (adyen api)



1.2.2 Διεπαφές με το χρήστη

Η εφαρμογή μας αναγνωρίζει 4 ρόλους

* Επώνυμος χρήστης: Πρόκειται για τον χρήστη που έχει επαφή με το front end του REST API και έχει την δυνατότητα να βάλει ενέργεια στο όχημα το, να κάνει πληρωμή και να δει τις συναλλαγές που είχε με τον φορτιστή και το όχημα του το χρονικό διάστημα που έχει επιλέξει.
* Ανώνυμος χρήστης: Πρόκειται για τον χρήστη που έχει επαφή με το front end του REST API και έχει την δυνατότητα μόνο βάλει ενέργεια στο όχημα του να κάνει πληρωμή.
* Ιδιοκτήτης parking: Πρόκειται για τον χρήστη που έχει επαφή με το front end του REST API και μπορεί να δει τις συναλλαγές που έχουν γίνει με τον φορτιστή από οποιοδήποτε όχημα στο χρονικό διάστημα που έχει επιλέξει.
* Διαχειριστής : Πρόκειται για τον χρήστη που έχει επαφή με το back end και είναι υπεύθυνος για την ομαλή λειτουργεία της εφαρμογής και μπορεί να τροποποίηση και να προσθέσει δεδομένα στην βάση δεδομένων.



3. Προδιαγραφές απαιτήσεων λογισμικού

3.1 Περιπτώσεις χρήσης

3.1.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: Έκδοση περιοδικού λογαριασμού ανά όχημα

3.1.1.1 Χρήστες (ρόλοι) που εμπλέκονται

Οι ρόλοι που εμπλέκονται στην λειτουργεία έκδοσης περιοδικού λογαριασμού είναι ο ιδιοκτήτης parking και ο επώνυμος χρήστης. Ποιο συγκεκριμένα θεωρούμε ότι ο ιδιοκτήτης parking είναι και αυτός ένας χρήστης ο οποίος όμως διαθέτει την δυνατότητα να έχει πρόσβαση σε όλα τα δεδομένα συναλλαγών όλων των χρηστών ενώ ο επώνυμος χρήστης θα έχει την διανοητά να δει τις συναλλαγές που έχει πραγματοποίηση με το όχημα του το χρονικό διάστημα στο οποίο θα επιλέξει.

3.1.1.2 Προϋποθέσεις εκτέλεσης

Θα πρέπει ο χρήστης να έχει κάνει τουλάχιστον μια συναλλαγή με τον φορτιστή έτσι ώστε να έχουν δημιουργηθεί δεδομένα (για τον ιδιοκτήτη parking απαιτείτε να έχει πραγματοποιηθεί τουλάχιστον μια συναλλαγή από κάποιον χρήστη )

Θα πρέπει ο χρήστης να έχει βάλει το αναγνωριστικό του οχήματος του

3.1.1.3 Περιβάλλον εκτέλεσης

To περιβάλλον εκτέλεσης είναι το web application μέσο του οπίου ο επώνυμος χρήστης ή ο ιδιοκτήτης του parking θα ζήτηση να δει τις συναλλαγές που επιθυμεί και θα σταλούν τα ανάλογα get μέσο queries και του back end μας στην βάση δεδομένων και ανάλογα με τις πληροφορίες που θα ζήτηση ο χρήστης η βάση δεδομένων μας θα απάντηση είτε με κάποια πληροφορία είτε με κάποιο σφάλμα.

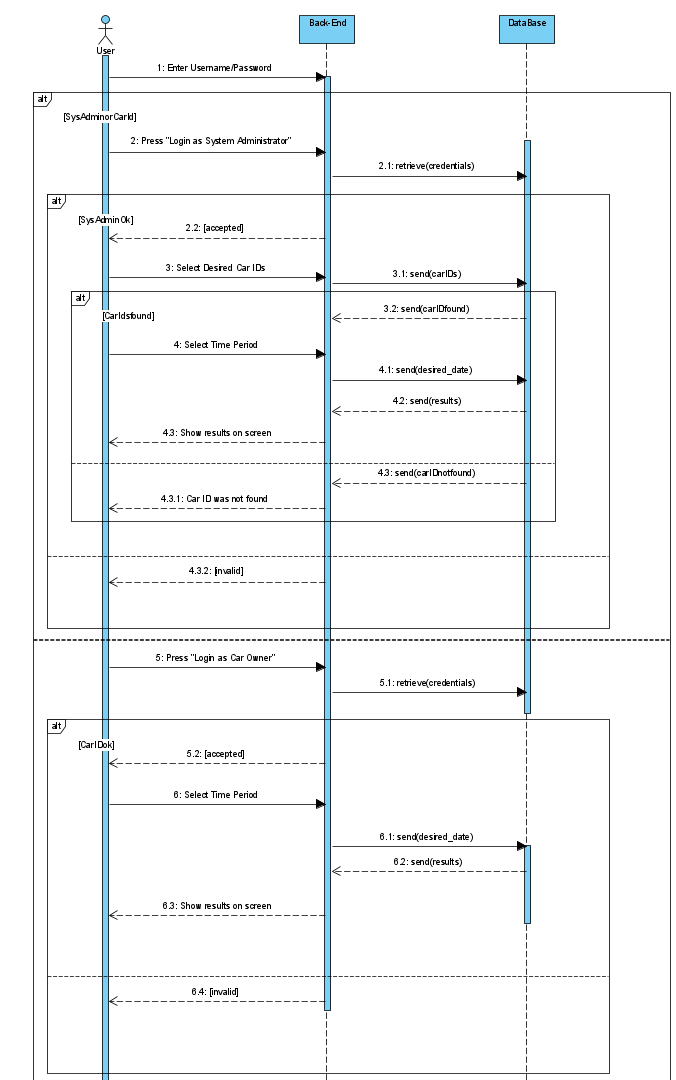
3.1.1.4 Δεδομένα εισόδου

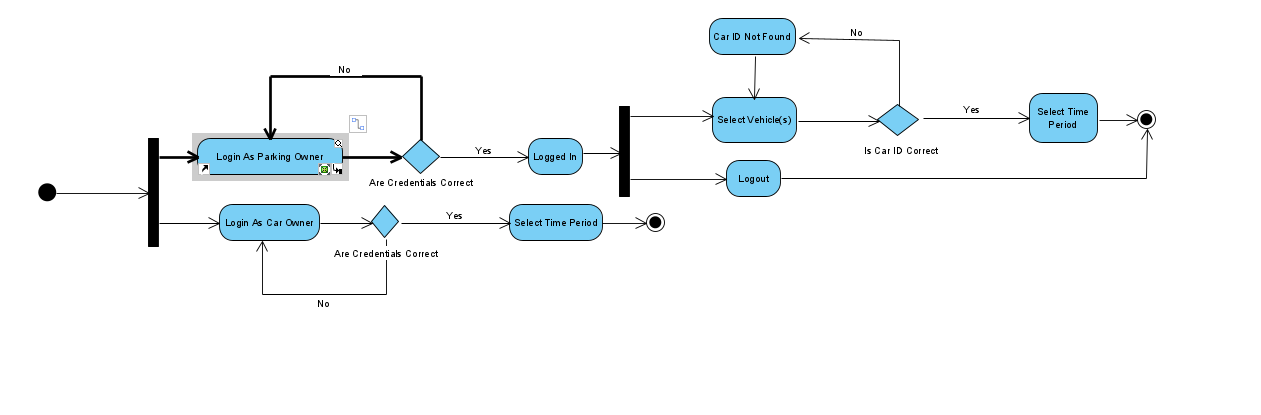
|  |  |
| --- | --- |
| Πεδίο | Τύπος δεδομένων |
| Owner login | Char field |
| Vehicle id | Char field |
| Start date | Int field |
| End date | Int field |

3.1.1.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Όταν χρησιμοποιεί το λογισμικό ο ιδιοκτήτη parking τότε αρχικά πρέπει να βάλει τα στοιχεία του στο πεδίο Admin login στην συνέχεια να επιλέξει για ποια οχήματα θέλει να δει τις συναλλαγές φόρτισής βάζοντας για κάθε ξεχωριστό όχημα το αναγνωριστικό του στο πέδιλο Vehicle id και στην συνέχεια να επιλέξει το διάστημα από το οποίο επιθυμεί να δει τις συναλλαγές φόρτισης με την βοήθεια τον πεδίων Start date, End date. Αφού ακολουθήσει τα παραπάνω βήματα τότε θα του εμφανιστή ένας πίνακας που θα περιέχει τις συναλλαγές φόρτισης για κάθε όχημα μέσα στο χρονικό διάστημα το οποίο επέλεξε.

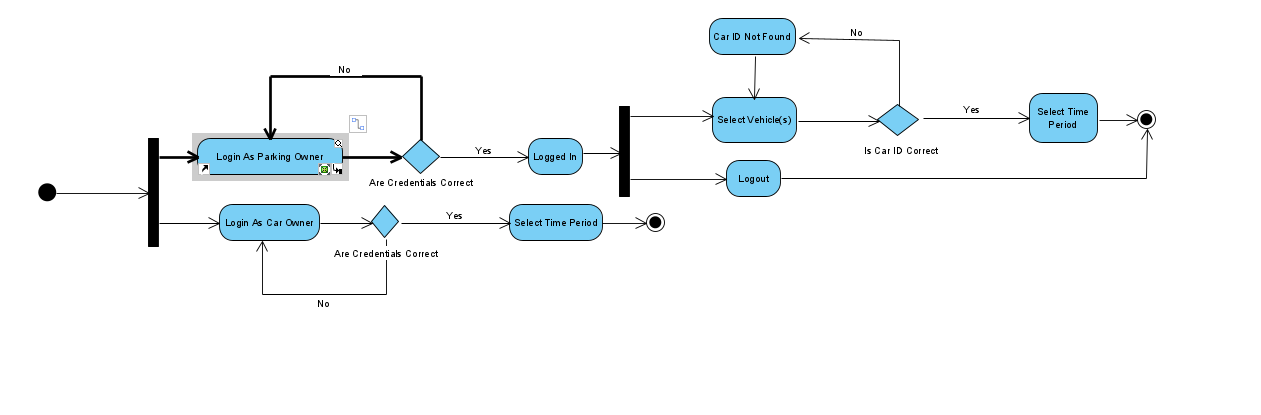
Όταν χρησιμοποιεί το λογισμικό ένας επώνυμος χρήστης τότε αρχικά πρέπει να βάλει το αναγνωριστικό του οχήματος του στο πεδίο Vehicle id και στην συνέχεια να επιλέξει το διάστημα από το οποίο επιθυμεί να δει τις συναλλαγές φορτίσεις με την βοήθεια τον πεδίων Start date, End date. Αφού ακολουθήσει τα παραπάνω βήματα τότε θα του εμφανιστή ένας πίνακας που θα περιέχει τις συναλλαγές φόρτισης για το όχημα του μέσα στο χρονικό διάστημα το οποίο επέλεξε.





3.1.1.7 Δεδομένα εξόδου

|  |  |
| --- | --- |
| Πεδίο | Τύπος δεδομένων |
| Transactions | char array |



3.1.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: Πληρωμή και συλλογή δεδομένων

3.1.2.1 Χρήστες (ρόλοι) που εμπλέκονται

Οι ρόλοι που εμπλέκονται στην λειτουργεία πληρωμής και συλλογής δεδομένων είναι ο επώνυμος και ο ανώνυμος χρήστης. Ποιο συγκεκριμένα ο επώνυμος χρήστης βάζει το αναγνωριστικό του οχήματος και στη συνέχεια το λειτουργικό μας σύστημα του δίνει την δυνατότητα να επιλέξει πρόγραμμα φόρτισης και ποσότητα ενέργειάς που θέλει να αγοράσει τέλος κάνει την πληρωμή, συλλέγονται τα στοιχεία για αυτή τι φόρτιση και τελειώνει η διαδικασία ενώ ο ανώνυμος χρήστης ακολουθεί την ίδια διαδικασία με την διαφορά ότι δεν βάζει το αναγνωριστικό του οχήματος του και όταν συλλέγονται τα στοιχεία για την φόρτιση το πεδίο με το αναγνωριστικό του οχήματος μένει κενό.

3.1.2.2 Προϋποθέσεις εκτέλεσης

Απαιτείτε για τον επώνυμο χρήστη να διαθέτει το αναγνωριστικό του οχήματος και να έχει αρκετά χρήματά για να πληρώσει για την ενέργεια που επιθυμεί να αγοράσει ενώ για τον ανώνυμο χρήστη αρκεί να έχει τα απαιτούμενα χρήματα για την ενέργεια που επιθυμεί να αγοράσει.

3.1.2.3 Περιβάλλον εκτέλεσης

To περιβάλλον εκτέλεσης είναι το web application μέσο του οπίου ο επώνυμος χρήστης ή ο ανώνυμος χρήστης θα προσπαθήσει να κάνει κάποια συναλλαγή. Μέσω queries θα επικοινωνήσει το back end μας με την βάση δεδομένων και ανάλογα με τις πληροφορίες που θα ζήτηση ο χρήστης η βάση δεδομένων μας θα απάντηση είτε με κάποια πληροφορία είτε με κάποιο σφάλμα.

3.1.2.4 Δεδομένα εισόδου

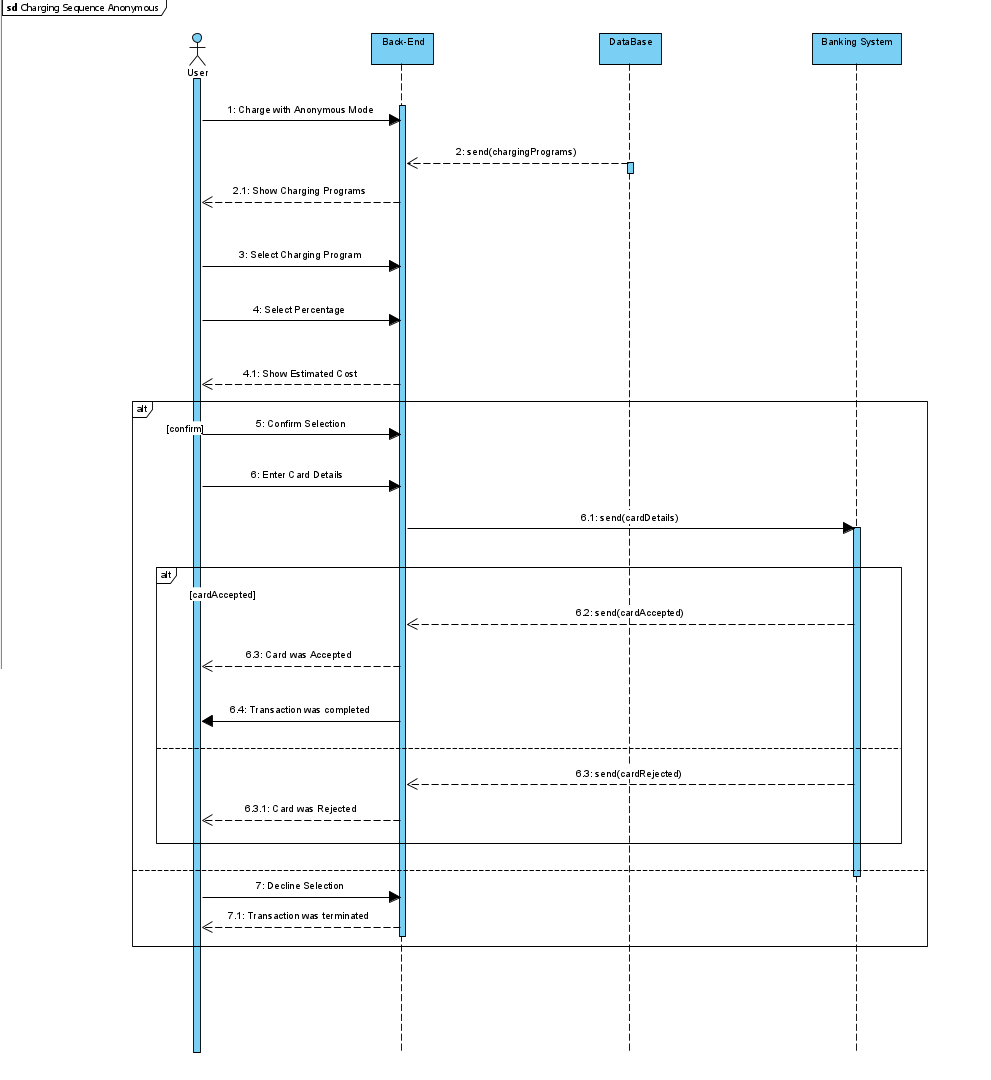
Καταγραφή δεδομένων εισόδου και εξόδου και συνθηκών εγκυρότητας αυτών.

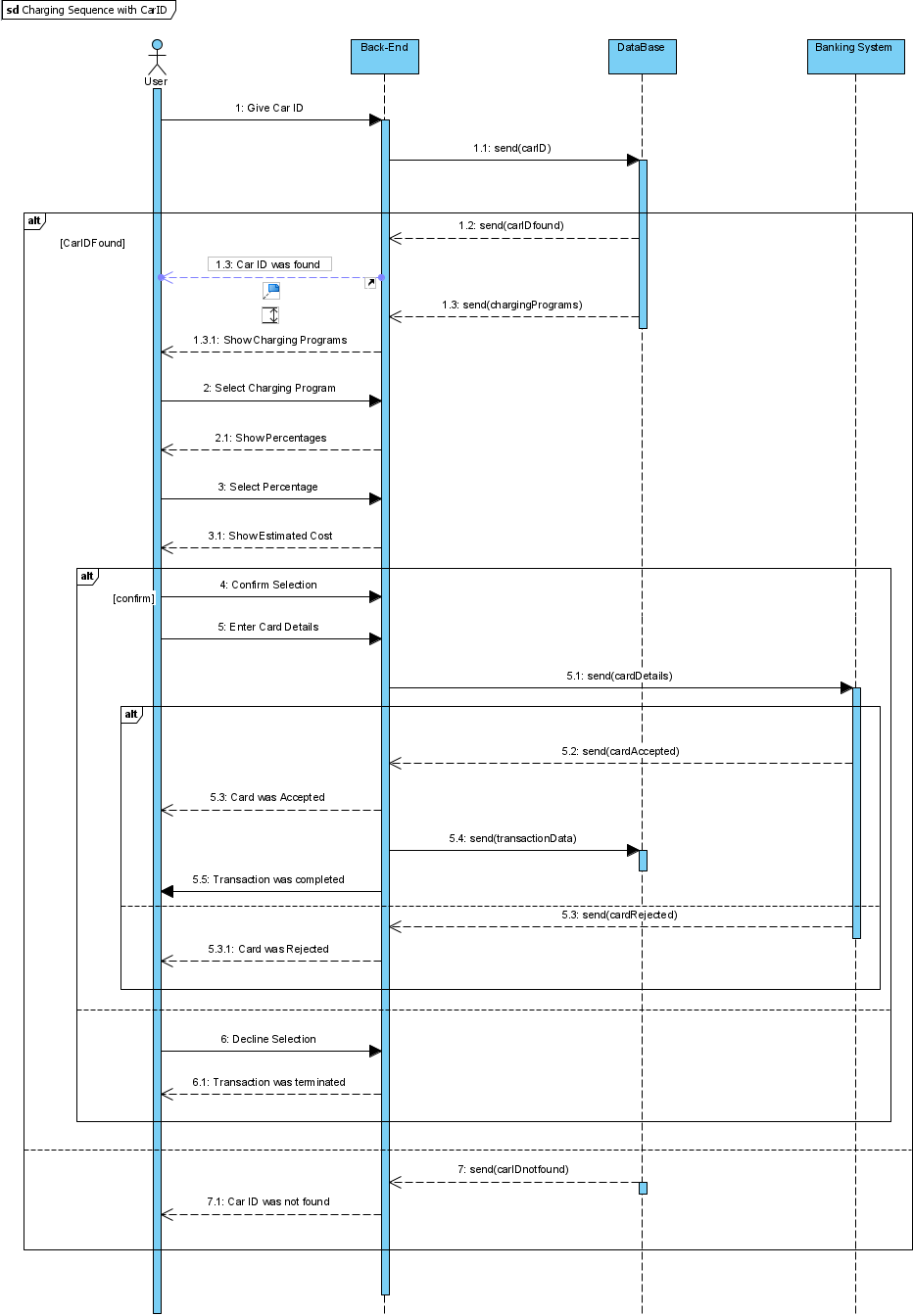
|  |  |
| --- | --- |
| Πεδίο | Τύπος δεδομένων |
| Vehicle id | Char field |
| Charging plan | Int field |
| Percentage of charge | Int field |
| Card information | Char field |

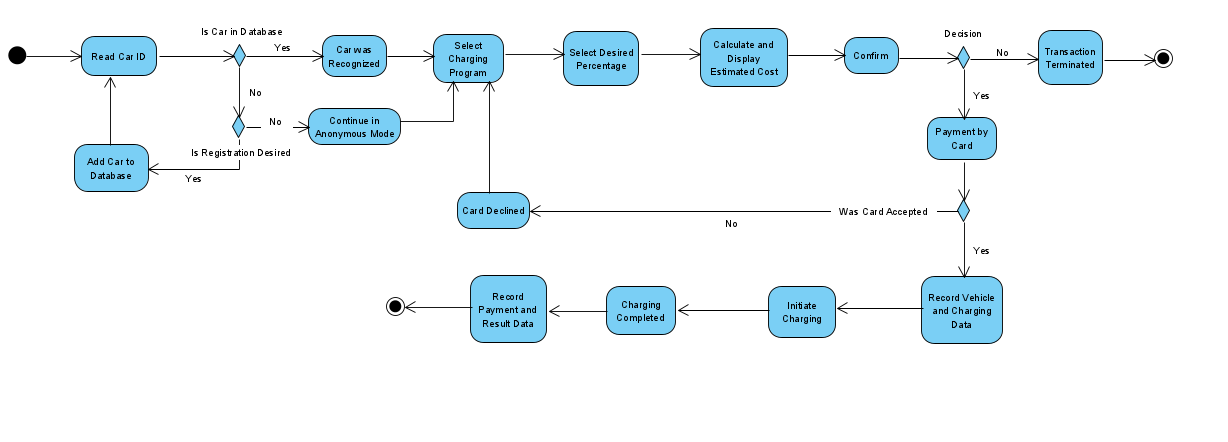
3.1.2.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Όταν χρησιμοποιεί το λογισμικό ο επώνυμος χρήστης αρχικά πρέπει να βάλει στο πεδίο Vehicle id το αναγνωριστικό του οχήματος του και αν είναι έγκυρο τότε θα εμφανιστούν στην οθόνη τα στοιχεία του οχήματος του. Στη συνέχεια χρειάζεται να επιλέξει το πρόγραμμα φόρτισης που επιθυμεί στο πεδίο Charging plan και να βάλει στο πεδίο Percentage of charge το ποσοστό ενέργειας που θέλει να βάλει στο όχημα του. Τέλος πρέπει να βάλει τα στοιχεία της κάρτας του στο πεδίο Card information. Αν τα παραπάνω στοιχεία είναι έγκυρα τότε θα γίνει καταγραφή αυτής τις συναλλαγής με το αναγνωριστικό του οχήματος του και θα πραγματοποιηθεί η φόρτιση.

Όταν χρησιμοποιεί το λογισμικό ο ανώνυμος χρήστης αρχικά πρέπει να επιλέξει το πρόγραμμα φορτίσεις που επιθυμεί στο πεδίο Charging plan και να βάλει στο πεδίο Percentage of charge το ποσοστό ενέργειας που θέλει να βάλει στο όχημα του. Τέλος πρέπει να βάλει τα στοιχεία της κάρτας του στο πεδίο Card information. Αν τα παραπάνω στοιχεία είναι έγκυρα τότε θα γίνει καταγραφή αυτής τις συναλλαγής χωρίς το αναγνωριστικό του οχήματος του και θα πραγματοποιηθεί η φόρτιση.

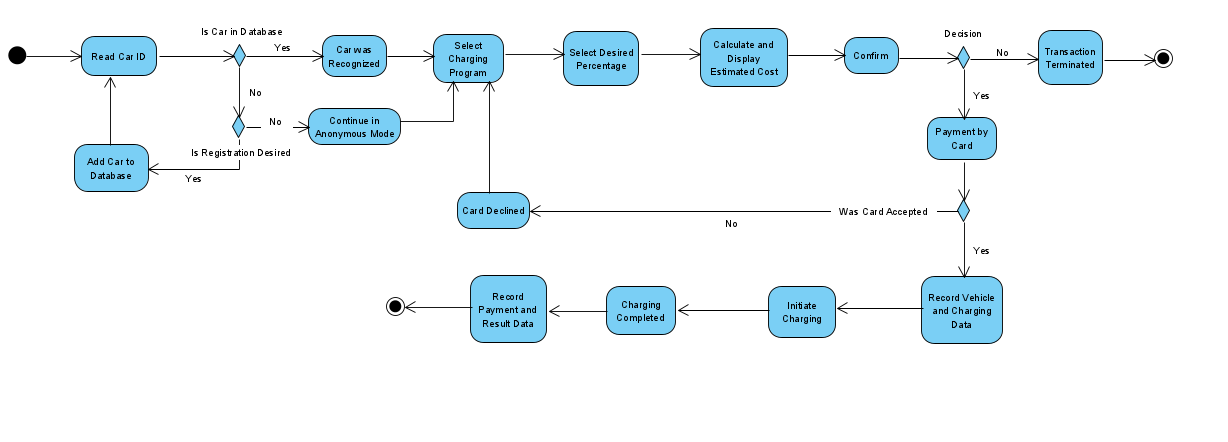






3.1.2.7 Δεδομένα εξόδου

|  |  |
| --- | --- |
| Πεδίο | Τύπος δεδομένων |
| Charge event | Char array |



3.2 Απαιτήσεις επιδόσεων

Το λογισμικό μας θα είναι διαθέσιμο κάθε στιγμή σε αυτούς που επιθυμούν να το χρησιμοποιήσουνε. Αναμένουμε σε ώρες αιχμής το μέγιστο φορτίο χρηστών που θα συνδεθούν ταυτόχρονα να μην ξεπερνάει τους 100 ταυτόχρονα. Επιπλέον το REST API μας είναι σχεδιασμένο έτσι ώστε να μπορεί να εξυπηρέτηση 100 request το δευτερόλεπτο.

3.3 Απαιτήσεις οργάνωσης δεδομένων

3.3.1 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα

Ο μόνος τρόπος με τον οποίο θα μπορεί να έχει κάποιος εκτός του διαχειριστή πρόσβαση στα δεδομένα που υπάρχουν στην βάση δεδομένων θα είναι αφού περάσουν από το API και να χρησιμοποιήσουν τα σωστά στοιχεία για την πρόσβαση στην βάση δεδομένων. Επομένως δεν υπάρχει κίνδυνος να υποκλαπούν δεδομένα.

Όσον αφορά τους περιορισμούς πρόσβαση σε δεδομένα.

* Ο ανώνυμος χρήσης δεν θα μπορεί να δει κάποια πληροφορία που υπάρχει στην βάση δεδομένων
* Ο επώνυμος χρήστης θα μπορεί να δει όλες συναλλαγές που έχει κάνει στον φορτιστή το χρονικό διάστημα που έχει επιλέξει
* Ο ιδιοκτήτης parking θα μπορεί να δει όλες τις συναλλαγές για οποιοδήποτε οχήματα επιλέξει και οποιαδήποτε στιγμή επιλέξει
* Ο διαχειριστής θα έχει πρόσβαση σε όλα τα δεδομένα τις βάσεις.

3.4 Περιορισμοί σχεδίασης

Back end

* Python
* Django
* Myshql

Front end

* Bootstrap
* Tachyons
* JQuery

3.5 Λοιπές απαιτήσεις

3.5.1 Απαιτήσεις διαθεσιμότητας λογισμικού

Λόγω τις φύσης του λογισμικού μας απαιτητέ να είναι διαθέσιμη προς του χρήστες κάθε στιγμή που υπάρχει σύνδεση με το διαδίκτυο η δυνατότητα για αγορά και πληρωμή ενέργειας καθώς αυτή είναι η κύρια λειτουργεία που επιτελούν τα μηχανήματα στα οποία λειτουργεί και η δυσλειτουργία του μπορεί να έχει σαν αποτέλεσμα είτε κάποιος χρήστης να μην μπορεί να συνέχιση την διαδρομή του λόγο έλλειψης ενέργειας είτε την απώλεια χρημάτων από τους ιδιοκτήτες τον φορτιστών. Επιπλέον είναι αναγκαίο να υπάρχει μεγάλη διαθεσιμότητά στις υπόλοιπες λειτουργίες του λογισμικού έτσι ώστε να διασφαλιστεί η ικανοποίηση τον χρηστών που χρησιμοποιούνε τους φορτιστές.

3.5.2 Απαιτήσεις ασφάλειας

Σημαντικό βάρος θα πρέπει να δοθεί στο κομμάτι τις προστασίας τον δεδομένων του χρήστη κατά την διάρκεια τις πληρωμής με πιστωτική κάρτα κάτι το οποίο θα επιτευχθεί με την χρήση ασφαλούς σύνδεσης (SSL).

3.5.3 Απαιτήσεις συντήρησης

Προκειμένου να πετύχουμε την ομαλή λειτουργεία του λογισμικού κατά την διάρκεια του κύκλου ζωής του χρειάζεται να υπάρχει τακτική επιτήρηση του λογισμικού για πιθανά σφάλματα που έχουν προκύψει είτε στο front end είτε στο back end και άμεση επιδιόρθωση τους. Επιπλέον χρειάζεται να γίνετε μια τακτική επίβλεψη τον δεδομένων έτσι ώστε να διασφαλίζεται ότι η ενέργεια που έχει πουληθεί αντίστοιχη στα χρήματα που έχουν εισπραχθεί από το σύστημα πληρωμών.