Έγγραφο απαιτήσεων λογισμικού (SRS)

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

[Plug and Drive]

# Εισαγωγή

## 1.1 Εισαγωγή: σκοπός του λογισμικού

Το λογισμικό αυτό έχει ως αντικείμενο την υποστήριξη ενός γενικευμένου πληροφοριακού συστήματος με στόχο την οργάνωση της φόρτισης των ηλεκτρικών οχημάτων. Οι εγγεγραμμένοι χρήστες της εφαρμογής μπορούν να απολαύσουν πληθώρα προνομοίων, ανάλογα με την ιδιότητά τους και τον σκοπό χρήσης της εφαρμογής.

Οι **ιδιοκτήτες ηλεκτρικών οχημάτων**, αφού επιλέξουν τον χώρο στάθμευσης και φόρτισης της επιλογής τους, θα μπορουν πλέον να αυτοματοποιήσουν τις διαδικασίες επιλογής προγράμματος φόρτισης και άμεσης πληρωμής μέσω της εφαρμογής. Παράλληλα, θα έχουν πρόσβαση στο ιστορικό και τα στατιστικά δεδομένα των φορτίσεών τους. Απο την άλλη, οι **κατασκευαστές ηλεκτρικών οχημάτων** θα έχουν στην διάθεση τους ενδιαφέροντα στατιστικά, χρήσιμα δεδομένα φόρτισης, καθώς και την ενεργειακή αποτίμηση των οχημάτων που έχουν διαθέσει στους καταναλωτές. Τέλος οι **διαχειριστές των σταθμών φόρτισης** θα μπορούν να επιβλέπουν και να ενημερώνουν την εφαρμογή για πιθανές αλλαγές στους χώρους parking καθώς και στα “πακέτα προσφορών και επιβραβεύσεων” που διαθέτουν, ενώ ταυτόχρονα θα έχουν στην διάθεσή τους χρήσιμα στατιστικά στοιχεία και δεδομένα για τις φορτίσεις του κάθε οχήματος στους σταθμούς τους.

Πέρα από αυτά, προβλέπεται και η ύπαρξη ενός **διαχειριστή συστήματος** ο οποίος θα μπορεί μέσω μιας εφαρμογής CLI (Command Line Interface) να δημιουργεί και να διαγράφει χρήστες, καθώς και να έχει πλήρη πρόσβαση στην ανάγνωση και εγγραφή δεδομένων.

## 1.2 Διεπαφές (interfaces)

### 1.2.1 Διεπαφές με εξωτερικά συστήματα

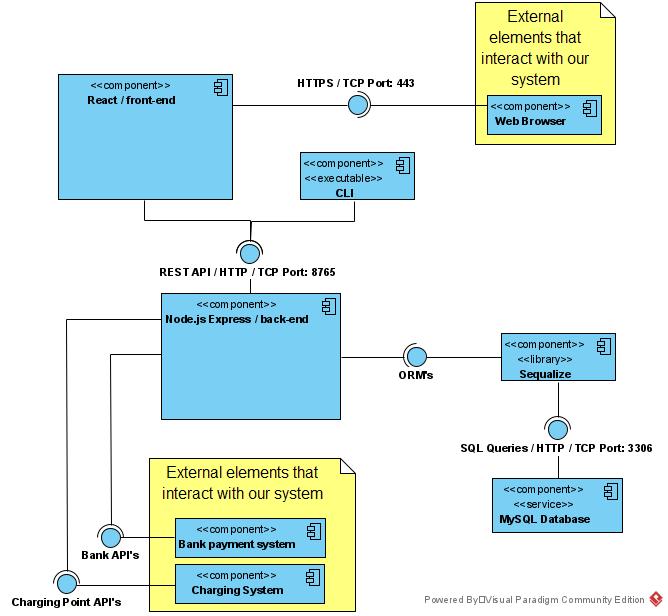
Το σύστημα στην πλήρη λειτουργία του θα χρησιμοποιεί τα API’s των εξής εξωτερικών συστημάτων:

* Τραπεζικό σύστημα (για τη διενέργεια ηλεκτρονικών συναλλαγών)
* Σημείο φόρτισης (για τη διενέργεια φορτίσεων οχημάτων)

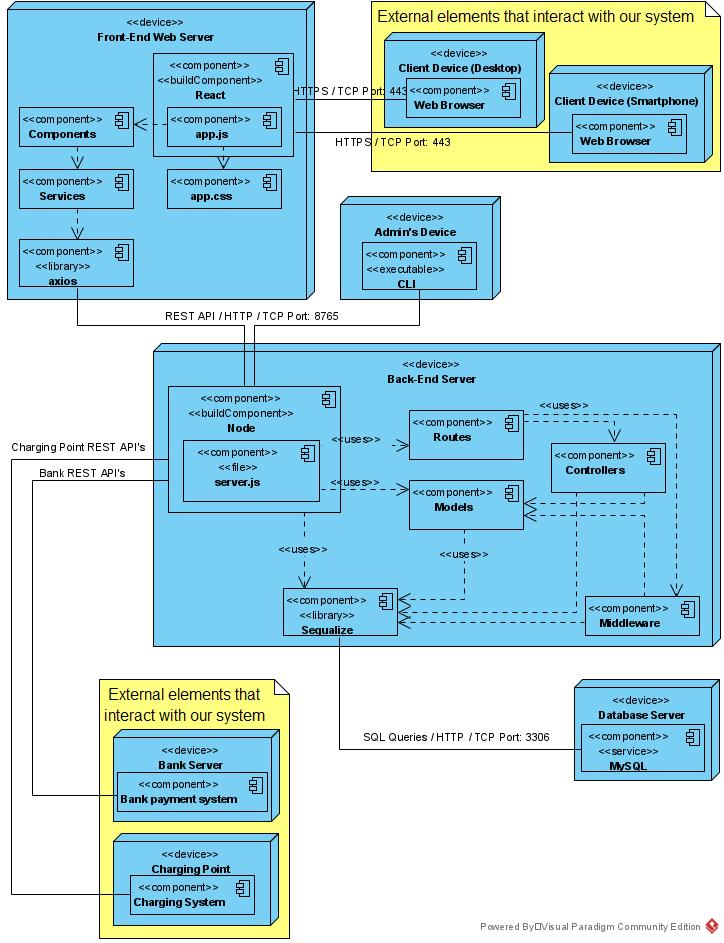
Εσωτερικά, έχουμε τις εξής διεπαφές:

* React web server (για την επικοινωνία client’s *browser* – *front-end*)
* Axios & REST API’s (για την επικοινωνία *front-end* – *back-end* καθώς και *CLI* – *back-end*)
* Sequalize ORM’s & MySQL queries (για την επικοινωνία *back-end* – *database*)

Παραθέτουμε το UML **Component diagram**:

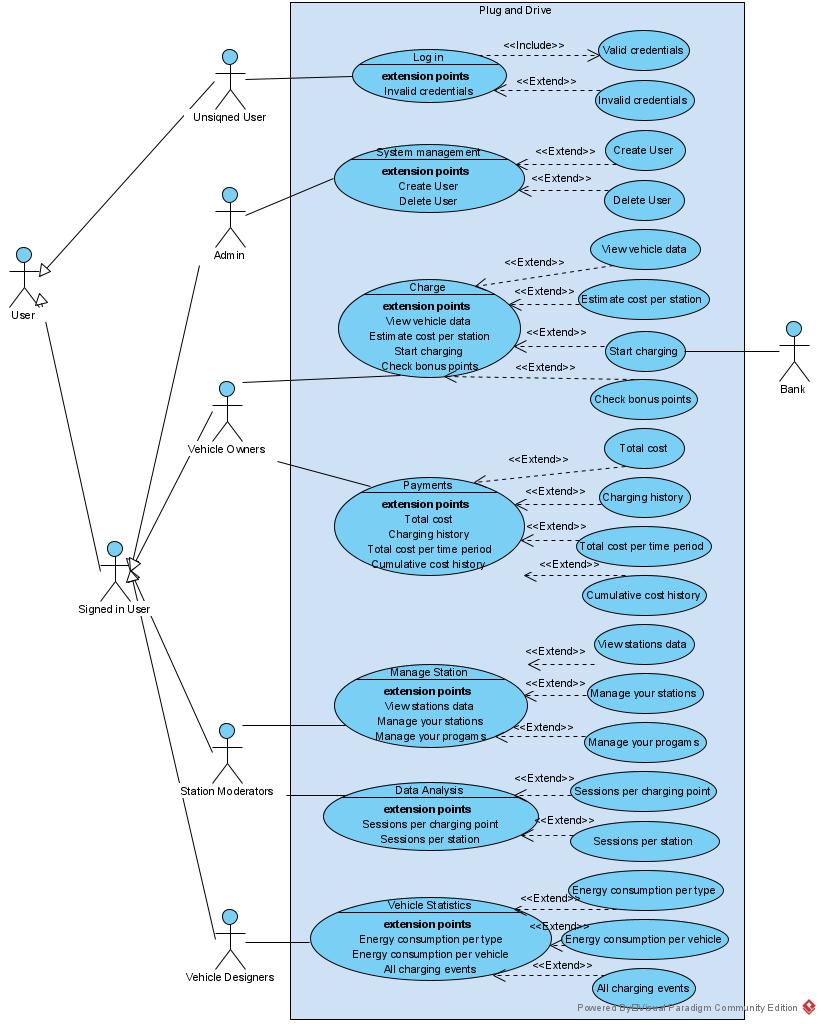


Παραθέτουμε το UML **Deployment Diagram**:



### 1.2.2 Διεπαφές με το χρήστη

Παραθέτουμε το UML **Use Case Diagram**:



# Αναφορές - πηγές πληροφοριών

N/A

# Προδιαγραφές απαιτήσεων λογισμικού

## 3.1 Περιπτώσεις χρήσης

### 3.1.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: Φόρτιση Οχήματος (Charge)

#### 3.1.1.1 Χρήστες (ρόλοι) που εμπλέκονται

* Ιδιοκτήτες οχημάτων (Vehicle Owners)

#### 3.1.1.2 Προϋποθέσεις εκτέλεσης

* Επιτυχές Login του χρήστη
* Ο χρήστης να έχει ρόλο ‘Vehicle Owner’

#### 3.1.1.3 Περιβάλλον εκτέλεσης

* Διεπαφή χρήστη του web based front-end
* Back-end server
* Database server
* Κατά την πλήρη λειτουργία του συστήματος, οι κλήσεις που δημιουργεί θα μπορούν να οδηγήσουν και στην ανάμιξη του συστήματος του σημείου φόρτισης και του εξυπηρετητή της Τράπεζας.

#### 3.1.1.4 Δεδομένα εισόδου

Το σύστημα απαιτεί από τον χρήστη τις παρακάτω εισόδους ανά λειτουργικότητα.

Συνθήκες εγκυρότητας θεωρούνται παντού οι τύποι των δεδομένων εισόδου που αναγράφονται.

**Στοιχεία αυτοκινήτου (View vehicle data)**

* Καμία *(Η διεπαφή επιστρέφει τα στοιχεία αυτοκινήτου με βάση το id του χρήστη)*

**Τιμολόγηση σταθμών (Estimate cost per station)**

* Καμία *(Η διεπαφή επιστρέφει την πλήρη λίστα σταθμών και προγραμμάτων φόρτισης)*

**Έναρξη φόρτισης (Start charging)**

* Επιθυμητό πρόγραμμα int
* Πόντοι επιβράβευσης double
* Πρωτόκολλο φόρτισης string
* Τρόπος πληρωμής enumeration(‘CASH’, ‘CREDIT\_CARD’)
* Σταθμός φόρτισης int

**Πόντοι επιβράβευσης (Check bonus points)**

* Καμία *(Η διεπαφή επιστρέφει ταους συνολικούς πόντους με βάση το id του χρήστη)*

#### 3.1.2.5 Παράμετροι

Το σύστημα παράγει και στέλνει στο back-end τις παρακάτω παραμέτρους ανά λειτουργικότητα.

Συνθήκες εγκυρότητας θεωρούνται παντού οι τύποι των δεδομένων εισόδου που αναγράφονται.

**Στοιχεία αυτοκινήτου (View vehicle data)**

* Owner id int

**Τιμολόγηση σταθμών (Estimate cost per station)**

* Owner id int

**Έναρξη φόρτισης (Start charging)**

* Owner id int
* Program id int
* Points double
* Protocol string
* Payment Method enumeration(‘CASH’, ‘CREDIT\_CARD’)
* Station id int

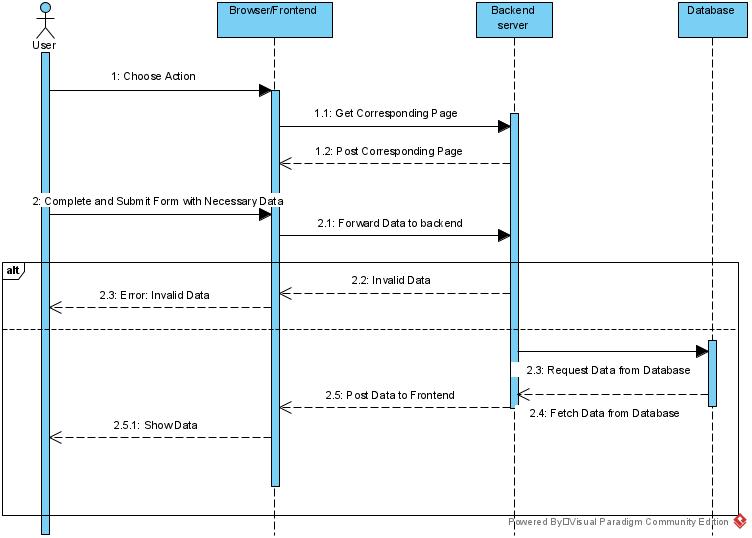
**Πόντοι επιβράβευσης (Check bonus points)**

* Owner id int

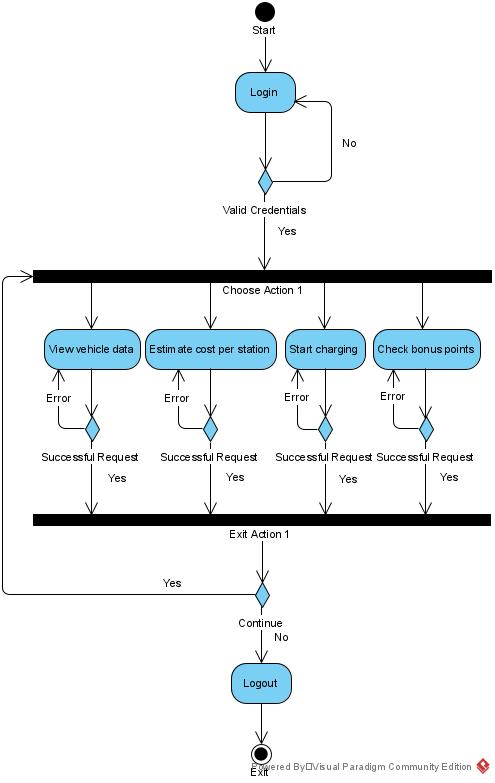
#### 3.1.1.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

1. Είσοδος στο σύστημα με τα διαπιστευτήρια (username + password)
2. Επιλογή επιθυμητής λειτουργίας
3. Είσοδος των ζητούμενων δεδομένων (δεν απαιτείται πάντα)
4. Έξοδος του συστήματος

Ακολουθεί το UML **Sequence Diagram**:



Ακολουθεί το UML **Activity Diagram**:



#### 3.1.1.7 Δεδομένα εξόδου

Το σύστημα επιστρέφει τις παρακάτω εξόδους ανά λειτουργικότητα.

Συνθήκες εγκυρότητας θεωρούνται παντού:

1. η εγκυρότητα της εισόδου (αλλιώς επιστροφή error 400)
2. η άδεια πρόσβασης του χρήστη στα συγκεκριμένα δεδομένα (αλλιώς επιστροφή error 401)
3. η ύπαρξη των συγκεκριμένων δεδομένων στη βάση (αλλιώς επιστροφή error 402)

**Στοιχεία αυτοκινήτου (View vehicle data)**

* Brand string
* Type string
* Model string
* Release year string
* Usable battery size (kWh) double
* Average consumption (kWh) double
* Current battery charge (kWh) double
* Owner id int

**Τιμολόγηση σταθμών (Estimate cost per station)**

* Station id int
* Charging Program id int
* Price per kWh (€) double
* Bonus points per kWh double
* Total Cost double
* Bonus Points double

**Έναρξη φόρτισης (Start charging)**

* Total cost double

**Πόντοι επιβράβευσης (Check bonus points)**

* Total bonus points int

#### 3.1.1.8 Παρατηρήσεις

N/A.

### 3.1.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: Δαπάνες (Payments)

#### 3.1.2.1 Χρήστες (ρόλοι) που εμπλέκονται

* Ιδιοκτήτες οχημάτων (Vehicle Owners)

#### 3.1.2.2 Προϋποθέσεις εκτέλεσης

* Επιτυχές Login του χρήστη
* Ο χρήστης να έχει ρόλο ‘Vehicle Owner’

#### 3.1.2.3 Περιβάλλον εκτέλεσης

* Διεπαφή χρήστη του web based front-end
* Back-end server
* Database server

#### 3.1.2.4 Δεδομένα εισόδου

Το σύστημα απαιτεί από τον χρήστη τις παρακάτω εισόδους ανά λειτουργικότητα.

Συνθήκες εγκυρότητας θεωρούνται παντού οι τύποι των δεδομένων εισόδου που αναγράφονται.

**Συνολικές δαπάνες (Total cost)**

* Καμία *(Η διεπαφή επιστρέφει τις συνολικές δαπάνες μέσω κάρτας και μετρητών του τρέχοντος ιδιοκτήτη οχήματος με βάση το id του χρήστη)*

**Ιστορικό φορτίσεων (Charging history)**

* Καμία *(Η διεπαφή επιστρέφει το ιστορικό των φορτίσεων του τρέχοντος ιδιοκτήτη οχήματος με βάση το id του)*

**Συνολικές δαπάνες σε χρονική περίοδο (Total cost per time period)**

* Ημερμηνία Έναρξης date
* Ημερομηνία Λήξης date

**Έκδοση περιοδικού λογαριασμού (Cumulative cost history)**

* Καμία *(Η διεπαφή επιστρέφει το ιστορικό των φορτίσεων του χρήστη με βάση το id του φροντίζοντας να υπολογίσει συσσωρευτικά το αθροιστικό κόστος σε βάθος χρόνου)*

#### 3.1.2.5 Παράμετροι

Το σύστημα παράγει και στέλνει στο back-end τις παρακάτω παραμέτρους ανά λειτουργικότητα.

Συνθήκες εγκυρότητας θεωρούνται παντού οι τύποι των δεδομένων εισόδου που αναγράφονται.

**Συνολικές δαπάνες (Total cost)**

* Owner id int

**Ιστορικό φορτίσεων (Charging history)**

* Owner id int

**Συνολικές δαπάνες σε χρονική περίοδο (Total cost per time period)**

* Owner id int
* Start Date date
* End Date date

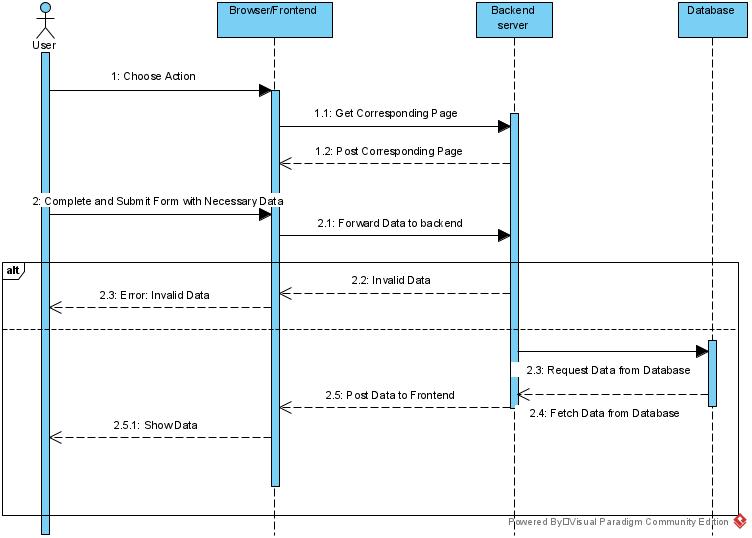
**Έκδοση περιοδικού λογαριασμού (Cumulative cost history)**

* Owner id int

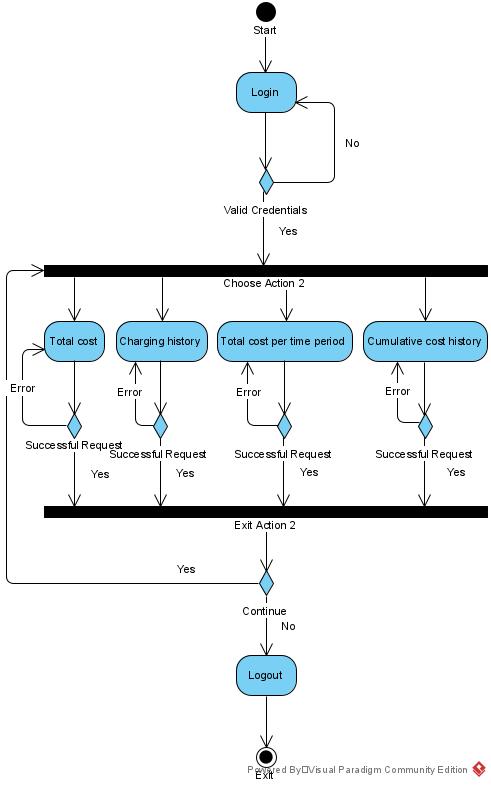
#### 3.1.2.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

1. Είσοδος στο σύστημα με τα διαπιστευτήρια (username + password)
2. Επιλογή επιθυμητής λειτουργίας
3. Είσοδος των ζητούμενων δεδομένων (δεν απαιτείται πάντα)
4. Έξοδος του συστήματος

Ακολουθεί το UML **Sequence Diagram**:



Ακολουθεί το UML **Activity Diagram**:



#### 3.1.2.7 Δεδομένα εξόδου

Το σύστημα επιστρέφει τις παρακάτω εξόδους ανά λειτουργικότητα.

Συνθήκες εγκυρότητας θεωρούνται παντού:

1. η εγκυρότητα της εισόδου (αλλιώς επιστροφή error 400)
2. η άδεια πρόσβασης του χρήστη στα συγκεκριμένα δεδομένα (αλλιώς επιστροφή error 401)
3. η ύπαρξη των συγκεκριμένων δεδομένων στη βάση (αλλιώς επιστροφή error 402)

**Συνολικές δαπάνες (Total cost)**

* Card int
* Cash int

**Ιστορικό φορτίσεων (Charging history)**

* Session id int
* Started on datetime
* Finished on datetime
* Energy (kwh) double
* Point id int
* Protocol string
* Payment enumeration (‘CASH’, ‘CREDIT\_CARD’)
* Bonus points energy (kwh) int
* Total cost (€) double
* Vehicle id int
* Station id int

**Συνολικές δαπάνες σε χρονική περίοδο (Total cost per time period)**

* Total cost int

**Έκδοση περιοδικού λογαριασμού (Cumulative cost history)**

* Session id int
* Finished on datetime
* Total Cost int

#### 3.1.2.8 Παρατηρήσεις

Ν/Α

### ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 3: Διαχείριση σταθμών φόρτισης (Manage Station)

#### 3.1.3.1 Χρήστες (ρόλοι) που εμπλέκονται

* Διαχειριστές σταθμών (Stations administrators)

#### 3.1.3.2 Προϋποθέσεις εκτέλεσης

* Επιτυχές Login του χρήστη
* Ο χρήστης να έχει ρόλο ‘Station Administrator’

#### 3.1.3.3 Περιβάλλον εκτέλεσης

* Διεπαφή χρήστη του web based front-end
* Back-end server
* Database server

#### 3.1.3.4 Δεδομένα εισόδου

Το σύστημα απαιτεί από τον χρήστη τις παρακάτω εισόδους ανά λειτουργικότητα.

Συνθήκες εγκυρότητας θεωρούνται παντού οι τύποι των δεδομένων εισόδου που αναγράφονται.

**Στοιχεία σταθμών (View stations data)**

* Καμία *(Η διεπαφή επιστρέφει αναλυτικά όλους τους σταθμούς και τα αντίστοιχα σημεία φόρτισης, προγράμματα και προμηθευτές που χειρίζεται ο χρήστης με βάση το id του)*

**Διαχείριση σταθμών φόρτισης (Manage your stations)**

* Σταθμός φόρτισης int
* Τοποθεσία string
* Όνομα Εταιρείας string
* Αριθμός Τηλεφώνου int
* Διαχειριστής σταθμού int
* Πάροχος int
* Σημείο φόρτισης int

**Διαχείριση προγραμμάτων φόρτισης (Manage your programs)**

* Πρόγραμμα φόρτισης int
* Όνομα προγράμματος string
* Κόστος ανά kWh double
* Πόντοι ανά kWh double
* Σταθμός φόρτισης int

#### 3.1.3.5 Παράμετροι

Το σύστημα παράγει και στέλνει στο back-end τις παρακάτω παραμέτρους ανά λειτουργικότητα.

Συνθήκες εγκυρότητας θεωρούνται παντού οι τύποι των δεδομένων εισόδου που αναγράφονται.

**Στοιχεία σταθμών (View stations data)**

* Station administrator id int

**Διαχείριση σταθμών φόρτισης (Manage your stations)**

* Station administrator id int
* Station id int
* Location string
* Company Name string
* Phone Number int
* Station Moderator id int
* Provider id int
* Point id int

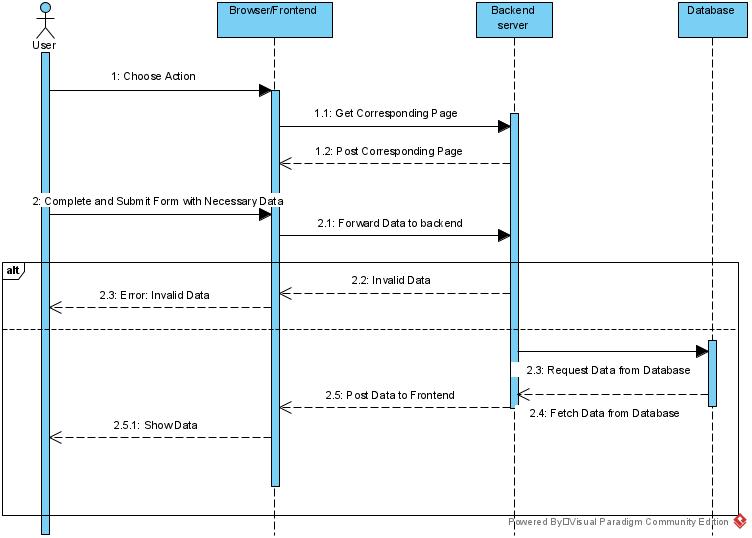
**Διαχείριση προγραμμάτων φόρτισης (Manage your programs)**

* Station administrator id int
* Program id int
* Program name string
* Price per kWh double
* Bonus point per kWh double
* Station id int

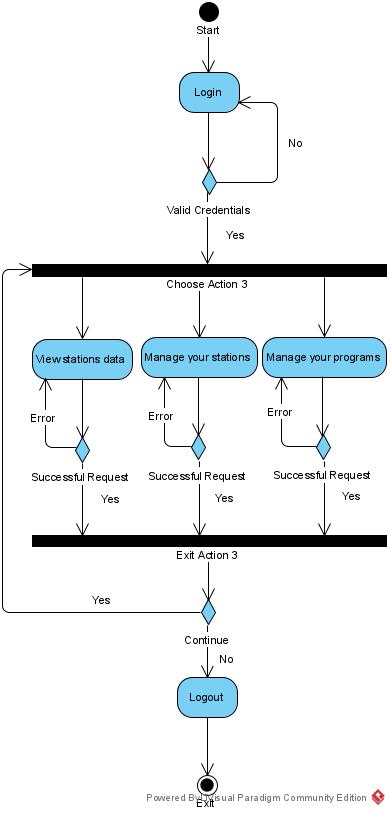
#### 3.1.3.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

1. Είσοδος στο σύστημα με τα διαπιστευτήρια (username + password)
2. Επιλογή επιθυμητής λειτουργίας
3. Είσοδος των ζητούμενων δεδομένων (δεν απαιτείται πάντα)
4. Έξοδος του συστήματος

Ακολουθεί το UML **Sequence Diagram**:



Ακολουθεί το UML **Activity Diagram**:



#### 3.1.3.7 Δεδομένα εξόδου

Το σύστημα επιστρέφει τις παρακάτω εξόδους ανά λειτουργικότητα.

Συνθήκες εγκυρότητας θεωρούνται παντού:

1. η εγκυρότητα της εισόδου (αλλιώς επιστροφή error 400)
2. η άδεια πρόσβασης του χρήστη στα συγκεκριμένα δεδομένα (αλλιώς επιστροφή error 401)
3. η ύπαρξη των συγκεκριμένων δεδομένων στη βάση (αλλιώς επιστροφή error 402)

**Στοιχεία σταθμών (View stations data)**

* Σταθμοί διαχείρισης
  + Station id int
  + Location string
  + Company Name string
  + Phone Number int
  + Station Moderator id int
  + Provider id int
* Σημεία φόρτισης των σταθμών διαχείρισης
  + Station id int
  + Point id int
* Προγράμματα φόρτισης των σταθμών διαχείρισης
  + Station id int
  + Program id int
  + Program name string
  + Price per kWh double
  + Bonus point per kWh double
* Προμηθευτές των σταθμών διαχείρισης
  + Station id int
  + Provider id int
  + Provider Name string

**Διαχείριση σταθμών φόρτισης (Manage your stations)**

* Στην περίπτωση ύπαρξης σταθμού με το id της εισόδου:
  + Μήνυμα: *Updated Station and Added Point*
* Στην περίπτωση μη ύπαρξης σταθμού με το id της εισόδου:
  + Μήνυμα: *Added Station*

**Διαχείριση προγραμμάτων φόρτισης (Manage your programs)**

* Στην περίπτωση ύπαρξης προγράμματος φόρτισης με το id της εισόδου:
  + Μήνυμα: *Updated Program with given ID*
* Στην περίπτωση μη ύπαρξης προγράμματος φόρτισης με το id της εισόδου:
  + Μήνυμα: *Added Program*

#### 3.1.3.8 Παρατηρήσεις

Ν/Α

### 3.1.4 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 4: Ανάλυση δεδομένων σταθμών (Data Analysis)

#### 3.1.4.1 Χρήστες (ρόλοι) που εμπλέκονται

* Διαχειριστές σταθμών (Stations Administrators)

#### 3.1.4.2 Προϋποθέσεις εκτέλεσης

* Επιτυχές Login του χρήστη
* Ο χρήστης να έχει ρόλο ‘Station Administrator’

#### 3.1.4.3 Περιβάλλον εκτέλεσης

* Διεπαφή χρήστη του web based front-end
* Back-end server
* Database server

#### 3.1.4.4 Δεδομένα εισόδου

Το σύστημα απαιτεί από τον χρήστη τις παρακάτω εισόδους ανά λειτουργικότητα.

Συνθήκες εγκυρότητας θεωρούνται παντού οι τύποι των δεδομένων εισόδου που αναγράφονται.

**Φορτίσεις ανά σημείο φόρτισης (Sessions per charging point)**

* Σημείο φόρτισης int
* Ημερομηνίας έναρξης date
* Ημερομηνία λήξης date

**Φορτίσεις ανά σταθμό φόρτισης (Sessions per station)**

* Σημείο φόρτισης int
* Ημερομηνίας έναρξης date
* Ημερομηνία λήξης date

#### 3.1.4.5 Παράμετροι

Το σύστημα παράγει και στέλνει στο back-end τις παρακάτω παραμέτρους ανά λειτουργικότητα.

Συνθήκες εγκυρότητας θεωρούνται παντού οι τύποι των δεδομένων εισόδου που αναγράφονται.

**Φορτίσεις ανά σημείο φόρτισης (Sessions per charging point)**

* Station administrator id int
* Point id int
* Start Date date
* End Date date

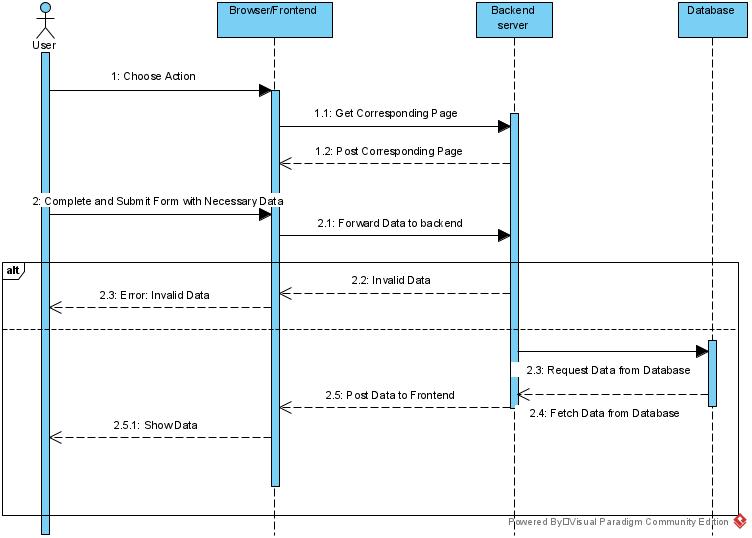
**Φορτίσεις ανά σταθμό φόρτισης (Sessions per station)**

* Station administrator id int
* Station id int
* Start Date date
* End Date date

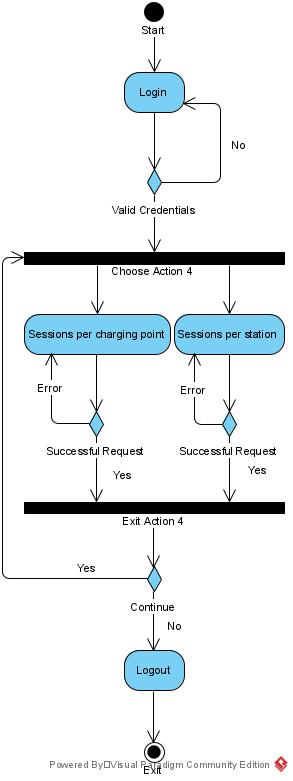
#### 3.1.4.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

1. Είσοδος στο σύστημα με τα διαπιστευτήρια (username + password)
2. Επιλογή επιθυμητής λειτουργίας
3. Είσοδος των ζητούμενων δεδομένων (δεν απαιτείται πάντα)
4. Έξοδος του συστήματος

Ακολουθεί το UML **Sequence Diagram**:



Ακολουθεί το UML **Activity Diagram**:



#### 3.1.4.7 Δεδομένα εξόδου

Το σύστημα επιστρέφει τις παρακάτω εξόδους ανά λειτουργικότητα.

Συνθήκες εγκυρότητας θεωρούνται παντού:

1. η εγκυρότητα της εισόδου (αλλιώς επιστροφή error 400)
2. η άδεια πρόσβασης του χρήστη στα συγκεκριμένα δεδομένα (αλλιώς επιστροφή error 401)
3. η ύπαρξη των συγκεκριμένων δεδομένων στη βάση (αλλιώς επιστροφή error 402)

**Φορτίσεις ανά σημείο φόρτισης (Sessions per charging point)**

* Point id int
* Point Operator string
* Request Timestamp datetime
* Period From date
* Period To date
* Number Of Charging Sessions int
* Για κάθε φόρτιση:
  + Session index int
  + Session id int
  + Started On datetime
  + Finished On datetime
  + Protocol string
  + Energy Delivered int
  + Payment string
  + Vehicle Type string

**Φορτίσεις ανά σταθμό φόρτισης (Sessions per station)**

* Station id int
* Operator string
* Request Timestamp datetime
* Period From date
* Period To date
* Total Energy Delivered int
* Number Of Charging Sessions int
* Number Of Active Points int
* Για κάθε σημείο φόρτισης:
  + Point id int
  + Point Sessions int
  + Energy Delivered int

#### 3.1.4.8 Παρατηρήσεις

N/A

### 3.1.5 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 5: Στατιστικά οχημάτων (Vehicle Statistics)

#### 3.1.5.1 Χρήστες (ρόλοι) που εμπλέκονται

* Κατασκευαστές οχημάτων (Vehicle Designers)

#### 3.1.5.2 Προϋποθέσεις εκτέλεσης

* Επιτυχές Login του χρήστη
* Ο χρήστης να έχει ρόλο ‘Vehicle Designer’

#### 3.1.5.3 Περιβάλλον εκτέλεσης

* Διεπαφή χρήστη του web based front-end
* Back-end server
* Database server

#### 3.1.5.4 Δεδομένα εισόδου

Το σύστημα απαιτεί από τον χρήστη τις παρακάτω εισόδους ανά λειτουργικότητα.

Συνθήκες εγκυρότητας θεωρούνται παντού οι τύποι των δεδομένων εισόδου που αναγράφονται.

**Κατανάλωση ενέργειας ανά τύπο οχήματος (Energy consumption per type)**

* Ημερομηνίας έναρξης date
* Ημερομηνία λήξης date

**Κατανάλωση ενέργειας ανά όχημα (Energy consumption per vehicle)**

* Ημερομηνίας έναρξης date
* Ημερομηνία λήξης date

**Ιστορικό φορτίσεων (All charging events)**

* Ημερομηνίας έναρξης date
* Ημερομηνία λήξης date

#### 3.1.5.5 Παράμετροι

Το σύστημα παράγει και στέλνει στο back-end τις παρακάτω παραμέτρους ανά λειτουργικότητα.

Συνθήκες εγκυρότητας θεωρούνται παντού οι τύποι των δεδομένων εισόδου που αναγράφονται.

**Κατανάλωση ενέργειας ανά τύπο οχήματος (Energy consumption per type)**

* Designer id int
* Start Date date
* End Date date

**Κατανάλωση ενέργειας ανά όχημα (Energy consumption per vehicle)**

* Designer id int
* Start Date date
* End Date date

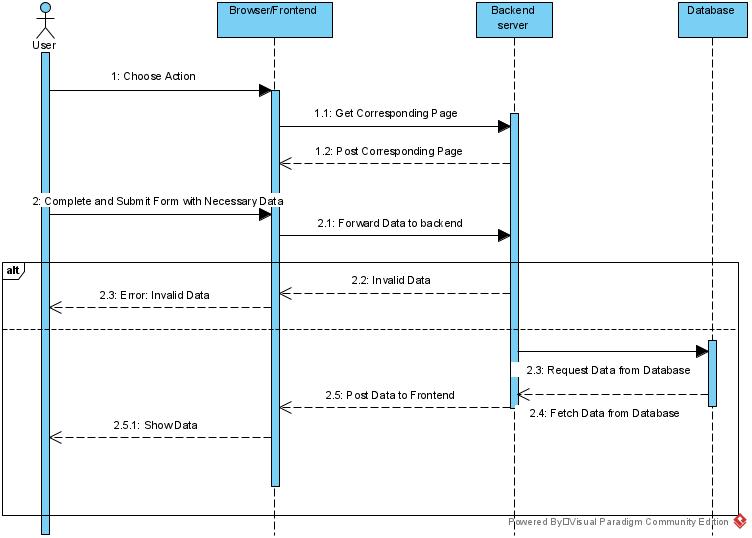
**Ιστορικό φορτίσεων (All charging events)**

* Designer id int
* Start Date date
* End Date date

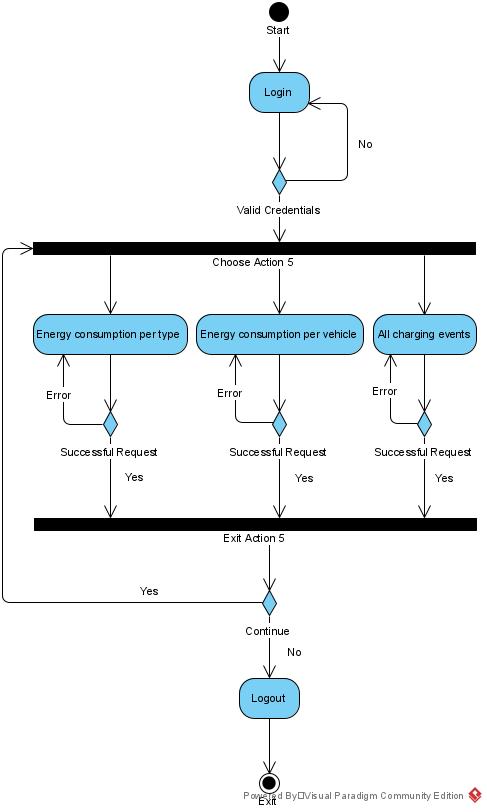
#### 3.1.5.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

1. Είσοδος στο σύστημα με τα διαπιστευτήρια (username + password)
2. Επιλογή επιθυμητής λειτουργίας
3. Είσοδος των ζητούμενων δεδομένων (δεν απαιτείται πάντα)
4. Έξοδος του συστήματος

Ακολουθεί το UML **Sequence Diagram**:



Ακολουθεί το UML **Activity Diagram**:



#### 3.1.5.7 Δεδομένα εξόδου

Το σύστημα επιστρέφει τις παρακάτω εξόδους ανά λειτουργικότητα.

Συνθήκες εγκυρότητας θεωρούνται παντού:

1. η εγκυρότητα της εισόδου (αλλιώς επιστροφή error 400)
2. η άδεια πρόσβασης του χρήστη στα συγκεκριμένα δεδομένα (αλλιώς επιστροφή error 401)
3. η ύπαρξη των συγκεκριμένων δεδομένων στη βάση (αλλιώς επιστροφή error 402)

**Κατανάλωση ενέργειας ανά τύπο οχήματος (Energy consumption per type)**

* Vehicle Type string
* Total Energy Delivered (kWh) int

**Κατανάλωση ενέργειας ανά όχημα (Energy consumption per vehicle)**

* Vehicle id int
* Total Energy Delivered (kWh) int

**Ιστορικό φορτίσεων (All charging events)**

* Session id int
* Started on datetime
* Finished on datetime
* Energy Delivered (kwh) double
* Protocol string
* Payment Method enumeration (‘CASH’, ‘CREDIT\_CARD’)
* Bonus points energy int
* Total cost (€) double
* Vehicle id int
* Station id int
* Point id int
* Program id int

#### 3.1.5.8 Παρατηρήσεις

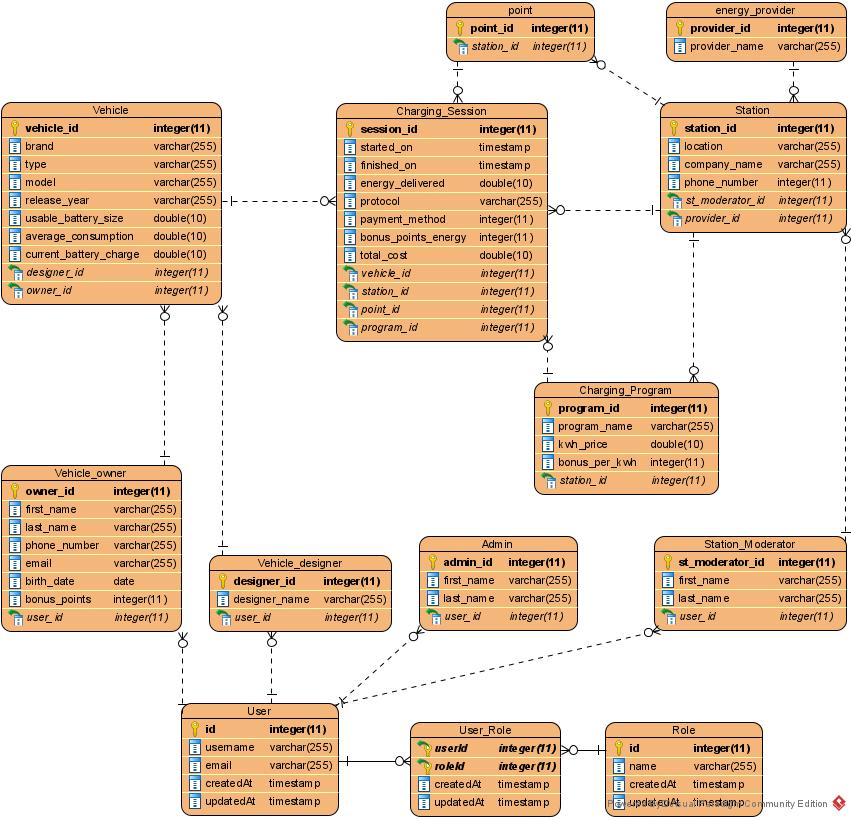
N/A

## 3.2 Απαιτήσεις επιδόσεων

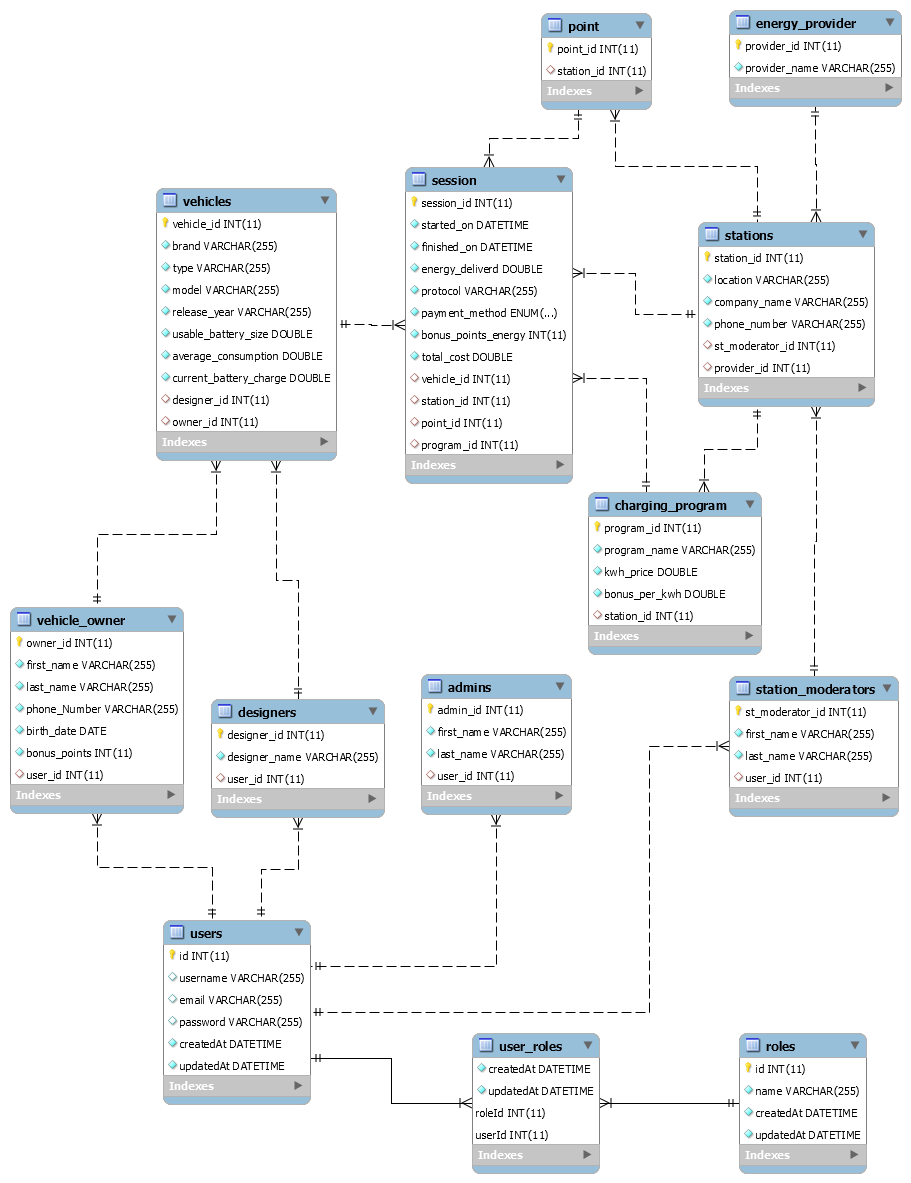
Το λογισμικό μας θα είναι συνεχώς διαθέσιμο σε αυτούς που επιθυμούν να το χρησιμοποιήσουν. Αναμένουμε σε ώρες αιχμής φορτίο 500 ταυτόχρονα συνδεδεμένων χρηστών τους οποίους και μπορούμε να διαχειριστούμε. Επιπλέον, το REST API μας μπορεί να καλύψει 50 requests το δευτερόλεπτο προς όλα τα resources. Όσον αφορά τους χρόνους απόκρισης, η σύνδεση χρήστη δε θα πρέπει να υπερβαίνει τα 2 δευτερόλεπτα, ενώ οι εμφανίσεις των δεδομένων στα αντίστοιχα use cases δε θα πρέπει να υπερβαίνουν τα 3 δευτερόλεπτα.

## 3.3 Απαιτήσεις οργάνωσης δεδομένων

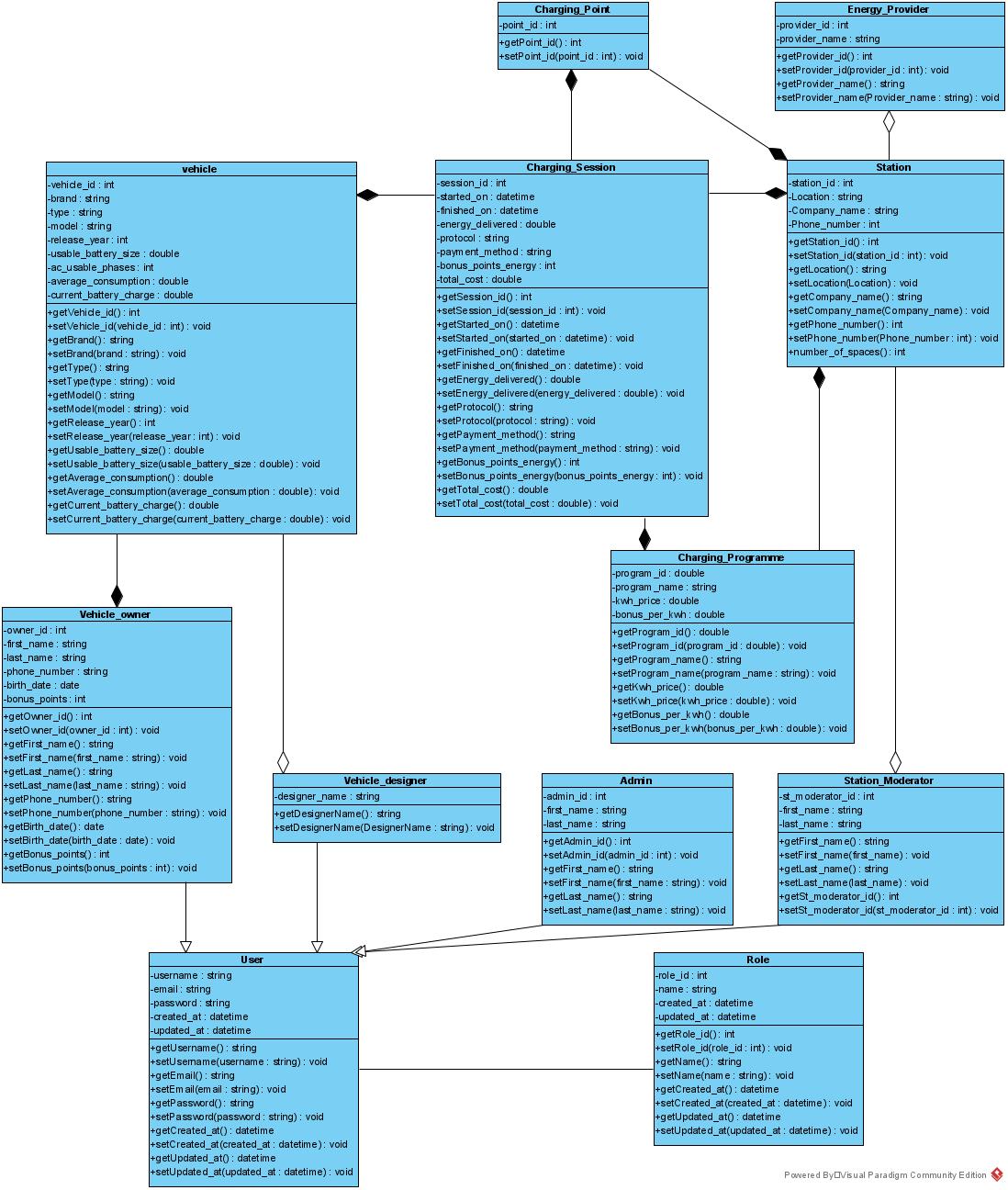
Ακολουθεί το UML **Entity Relation Diagram**:



Ακολουθεί το MySQL Workbench **Model View Diagram**:



Ακολουθεί το UML **Class Diagram:**



### 3.3.1 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα

Κάθε τύπος πρόσβασης (Parking - Data Analyst, Parking - Admin,...) έχει περιορισμένη πρόσβαση στα δεδομένα αναλόγως τα Privileges που του ανήκουν. Όσων αφορά τα προσωπικά δεδομένα όλων των χρηστών (username, password,...), είναι ορατά μόνο από τους διαχειριστές της πλατφόρμας και φυλάσσονται σε ξεχωριστή βάση δεδομένων από αυτήν που έχουν πρόσβαση οι χρήστες.

## 3.4 Περιορισμοί σχεδίασης

Front-End

* Για τη μορφοποίηση, τη διαδραστικότητα και τη γενική διαμόρφωση της ιστοσελίδας θα γίνει η χρήση **Javascript**, η οποία συνεπάγεται κάποιους περιορισμούς στην ταχύτητα, ανάλογα με τη χρήση της, καθώς και τη μεταβλητότητα που μπορεί να παρουσιάσει σε διαφορετικούς browsers.
* Στην συγγραφή και οργάνωση του κώδικα Javascript αξιοποιούμε τη βιβλιοθήκη **React**, η οποία μας παρέχει μεγαλύτερη ευκολία στην δημιουργία του front-end σε σύγκριση με απλό κώδικα Javascript. Η ευκολία αυτή προκύπτει από τη μορφοποίηση και τις διευκολύνσεις που προσφέρει το React επιτρέποντας μεγαλύτερη ταχύτητα συγγραφής του κώδικα καθώς και την ύπαρξη υποστήριξης για αναδυόμενα ερωτήματα λόγω της ευρείας χρήσης της. Επίσης, η χρήση του αντικειμενοστραφούς προγραμματισμού εντός του πλαισίου της React προσφέρει κάποια επιπλέον πλεονεκτήματα όπως η ξεκάθαρη οργάνωση και δομή του κώδικα, η ευελιξία μέσω του πολυμορφισμού κλπ.
* Μέσα στα αρχεία της Javascript ενσωματώνουμε κώδικες **HTML** για την κατασκευή του σκελετού της εφαρομογής και **CSS** για την μορφοποίησή της

Back-End

* Χρήση της γλώσσας **Javascript** για την υλοποίηση του back-end της εφαρμογής.
* Χρήση του **NodeJS** και του εργαλείου αυτόματου χτισίματος **npm** για αυτοματισμό της επίλυσης εξαρτήσεων λογισμικού εντός του project και αυτόματη εκτέλεση των τεστ λειτουργικότητας.
* Για τη συγγραφή δοκιμαστικού (test) κώδικα και τη διενέργεια ελέγχων στα endpoints του back-end χρησιμοποιήθηκαν τα **chai** και **mocha**

CLI

* Το Command Line Interface γράφτηκε σε Python με χρήση της βιβλιοθήκης **argparse** (parser for command-line options, arguments and sub-commands).
* Η διενέργεια ελέγχων έγινε με το εργαλείο **pytest** (full-featured Python testing tool)

Version Control

* Χρήση του Git (και πιο συγκεκριμένα της πλατφόρμας Github) για την διαδικασία version control του συστήματος.

## 3.5 Λοιπές απαιτήσεις

### 3.5.1 Απαιτήσεις διαθεσιμότητας λογισμικού

Η εφαρμογή είναι επιθυμητό να είναι διαθέσιμη οποιαδήποτε ώρα της ημέρας για όλα τα είδη χρηστών.

* Χρήση αντιγράφων της βάσης δεδομένων και εναλλακτικών servers οι οποίοι θα υποστηρίζουν την κίνηση της πλατφόρμας σε περίπτωση συντήρησης ενός server. Αυτό είναι επιθυμητό σε όλες τις φάσεις υιοθέτησης της πλατφόρμας καθώς στο αρχικό στάδιο οι χρήστες θα είναι λίγοι αλλά οι ανάγκες σε αλλαγές και βελτιώσεις του λογισμικού πολλές ενώ στη συνέχεια η βάση χρηστών θα είναι μεγαλύτερη.
* Συνέπεια στην δυνατότητα πρόσβασής της.
* Συμφωνία με πάροχο υπηρεσιών διαδικτύου, ο οποίος μπορεί να παρέχει πλήρη διαθεσιμότητα μέσω των δικτυακών εγκαταστάσεών του στο διαδίκτυο.

### 3.5.2 Απαιτήσεις ασφάλειας

* Ασφαλή κανάλια για τη σύνδεση των χρηστών.
* Χρήση HTTPS και SSL certificate.
* Αυτόματη έξοδος χρηστών ανενεργών για μεγάλο χρονικό διάστημα.
* Κωδικοί πρόσβασης και άλλα ευαίσθητα προσωπικά δεδομένα δεν θα πρέπει να εμφανίζονται ή να αποθηκεύονται σε υπολογιστές χρηστών μέσω cookies ή με οποιοδήποτε άλλον τρόπο.
* Οι κωδικοί πρόσβασης θα πρέπει να είναι κρυπτογραφημένοι στη βάση δεδομένων.

### 3.5.3 Απαιτήσεις συντήρησης

* Λόγω της στατικής φύσης των δεδομένων, αρκεί να μπορεί το σύστημα να τροφοδοτηθεί από κάποιο backup αποθηκευτικό χώρο.
* Συντήρηση του συστήματος θα εκτελείται κάθε 3 μήνες και σε έκτακτες περιστάσεις. Η συχνότητα αυτή υπόκειται σε αλλαγές ανάλογα με τον αριθμό χρηστών και την απόδοση του συστήματος.