Έγγραφο απαιτήσεων λογισμικού (SRS)

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

[SmartEMobility]



# Εισαγωγή

## 1.1 Εισαγωγή: σκοπός του λογισμικού

Σκοπός του λογισμικού SmartEMobility είναι να αποτελέσει εύχρηστο εργαλείο για τη διαχείριση ενός δικτύου σταθμών φόρτισης ηλεκτρικών οχημάτων καθώς και η βελτιστοποίηση και αυτοματοποίηση των διαδικασιών για την φόρτιση ηλεκτρικών οχημάτων.

## 1.2 Διεπαφές (interfaces)

### 1.2.1 Διεπαφές με εξωτερικά συστήματα

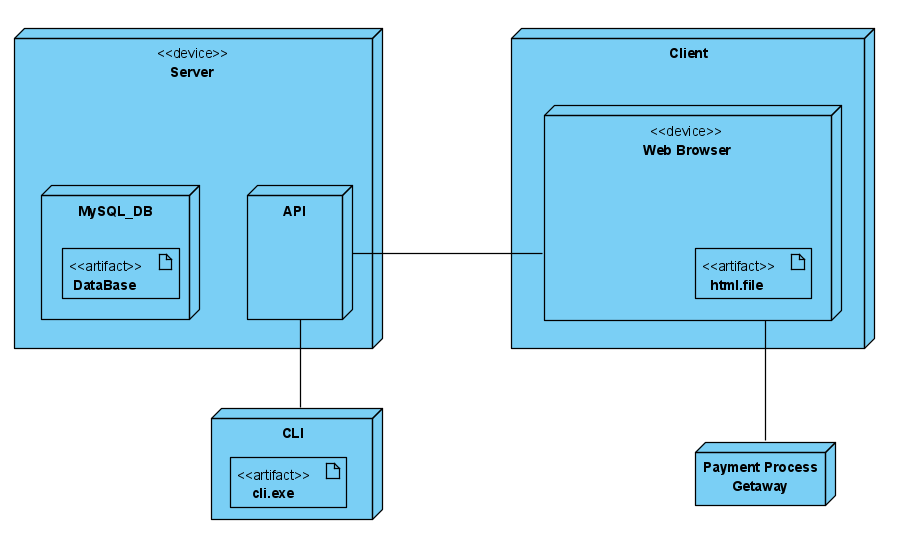
Front-end software: Web application

Back-end software: Command Line Interface

Database software: MySQL

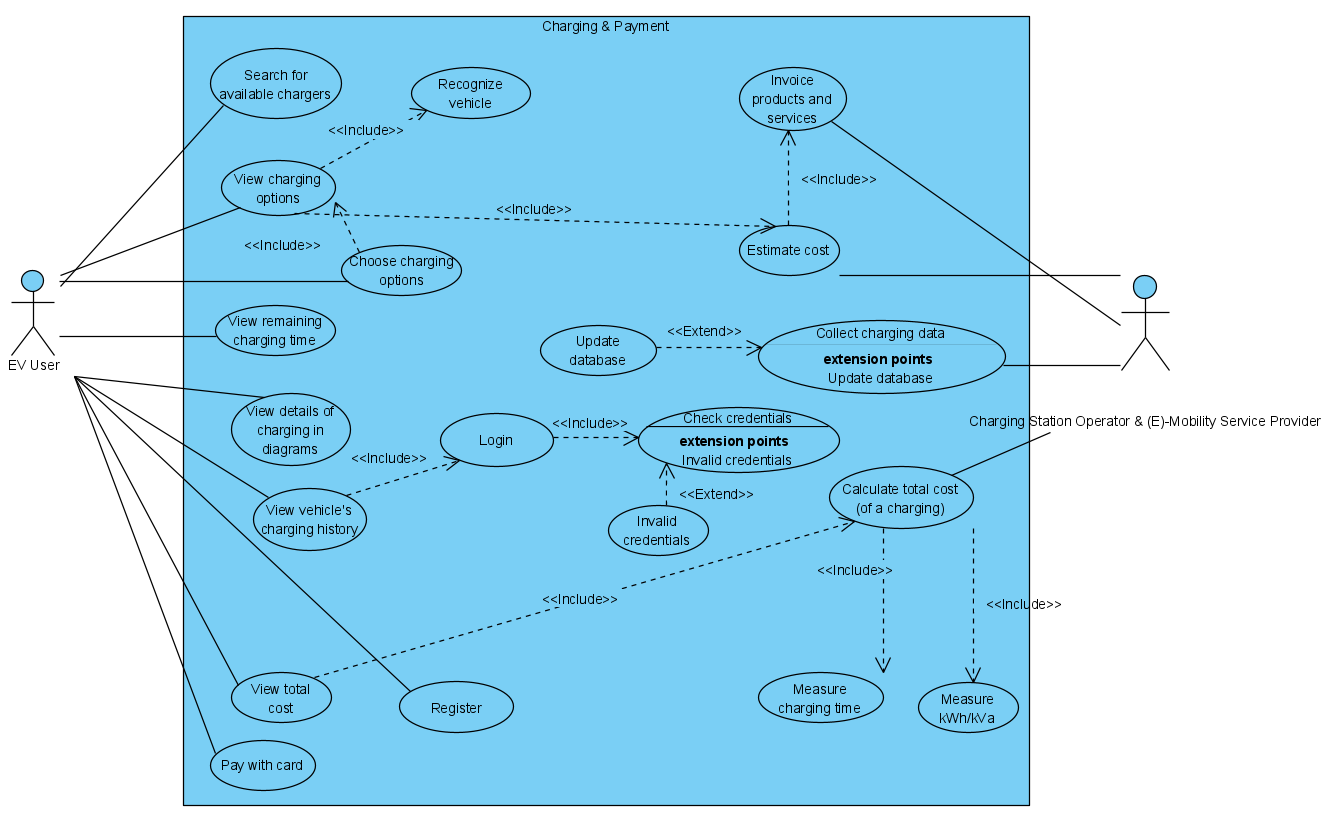
API

***Deployment Diagram***

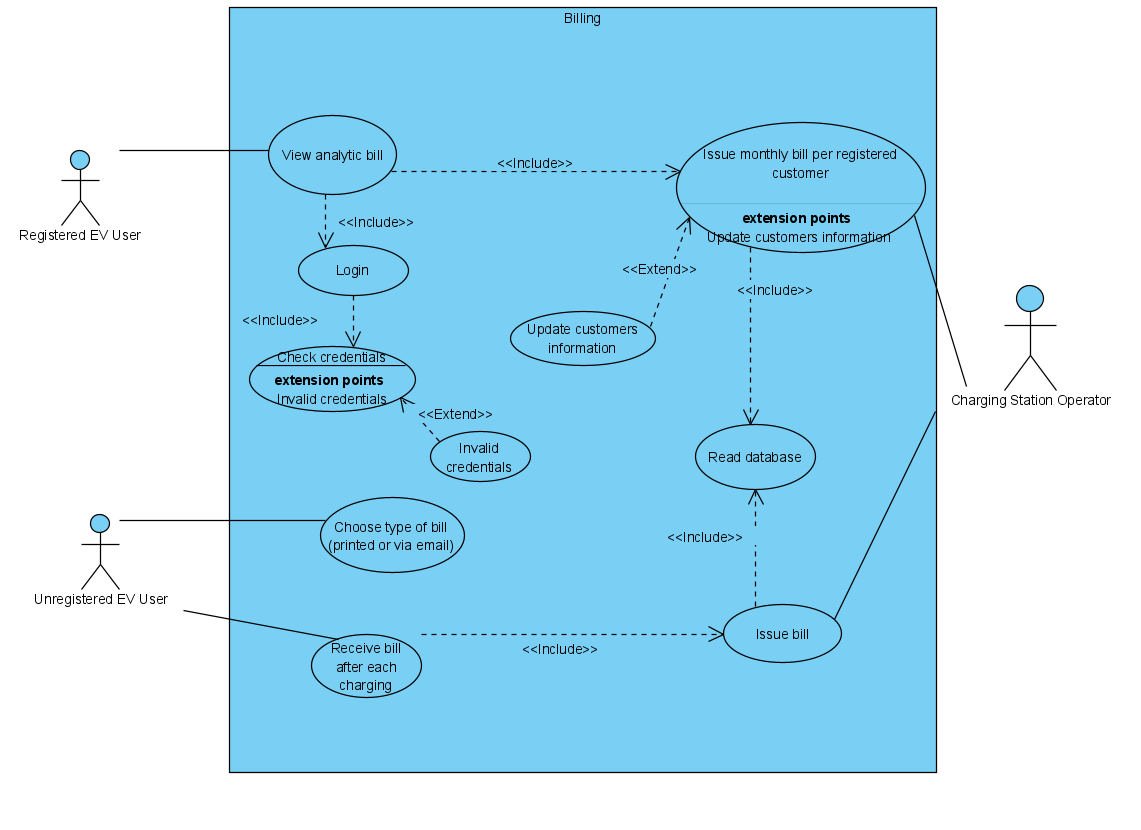


### 1.2.2 Διεπαφές με το χρήστη

**Use Case 1: Φόρτιση και πληρωμή (charging infrastructure)**



**Use Case 2: Billing**



# Αναφορές - πηγές πληροφοριών

* <https://www.virta.global/smart-charging>
* E-Mobility Systems Architecture: a model-based framework for managing complexity and interoperability [BenediktKirpes, PhilippDanner, RobertBasmadjian, Hermann de Meer & ChristianBecker]

# Προδιαγραφές απαιτήσεων λογισμικού

## 3.1 Περιπτώσεις χρήσης

### 3.1.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: Φόρτιση και Πληρωμή

#### 3.1.1.1 Χρήστες (ρόλοι) που εμπλέκονται

Οι χρήστες του συστήματος είναι ο διαχειριστής του δικτύου σταθμών φόρτισης ηλεκτρικών οχημάτων και οι ιδιοκτήτες των ηλεκτρικών οχημάτων εγγεγραμμένοι και μη.

#### 3.1.1.2 Προϋποθέσεις εκτέλεσης

Ο ιδιοκτήτης ηλεκτρικού οχήματος για να χρησιμοποιήσει την εφαρμογή μπορεί να είναι εγγεγραμμένος ή όχι. Αν δεν είναι εγγεγραμμένος χρήστης τότε πρέπει να προηγηθεί η πληρωμή της έναρξης της φόρτιση ενώ για τους εγγεγραμμένους χρήστες τηρείται μηνιαίος λογαριασμός και οι πληρωμές πραγματοποιούνται από σύστημα πληρωμών μία φορά το μήνα.

Πρόσβαση στα δεδομένα των χρηστών έχει μόνο ο διαχειριστής των σταθμών φόρτισης ο οποίος ταυτοποιείται από την εφαρμογή για την είσοδό του στο σύστημα. Επίσης κάθε εγγεγραμμένος χρήστης αφού ταυτοποιηθεί από το σύστημα έχει πρόσβαση στα προσωπικά του δεδομένα.

#### 3.1.1.3 Περιβάλλον εκτέλεσης

Τα use cases στο παραπάνω Use Case UML διάγραμμα για την πρώτη περίπτωση χρήσης εκτελούνται σε κάθε σύστημα που υποστηρίζει Command Line Interface και διαθέτει περιηγητή διαδικτύου για την περιήγηση στη διαδικτυακή εφαρμογή.

#### 3.1.1.4 Δεδομένα εισόδου

Δεδομένα εισόδου για τους χρήστες της εφαρμογής αποτελούν τα:

* Μη εγγεγραμμένοι χρήστες: αριθμός άδειας κυκλοφορίας οχήματος για ταυτοποίηση, επιλογή προγράμματος φόρτισης από τα εμφανιζόμενα, επιλογή τρόπου πληρωμής από τους εμφανιζόμενους, φίλτρα στη φόρμα αναζήτησης του πλησιέστερου διαθέσιμου σταθμού
* Εγγεγραμμένοι χρήστες: username και password για είσοδο στο προφίλ τους, επιλογή προγράμματος φόρτισης από τα εμφανιζόμενα, επιλογή τρόπου πληρωμής από τους εμφανιζόμενους, φίλτρα στη φόρμα αναζήτησης του πλησιέστερου διαθέσιμου σταθμού
* Διαχειριστής σταθμών φόρτισης: username και password, φίλτρα κατά την αναζήτηση στη βάση δεδομένων
* Στοιχεία του οχήματος που διαβάζονται από το σύστημα κατά το plug-in

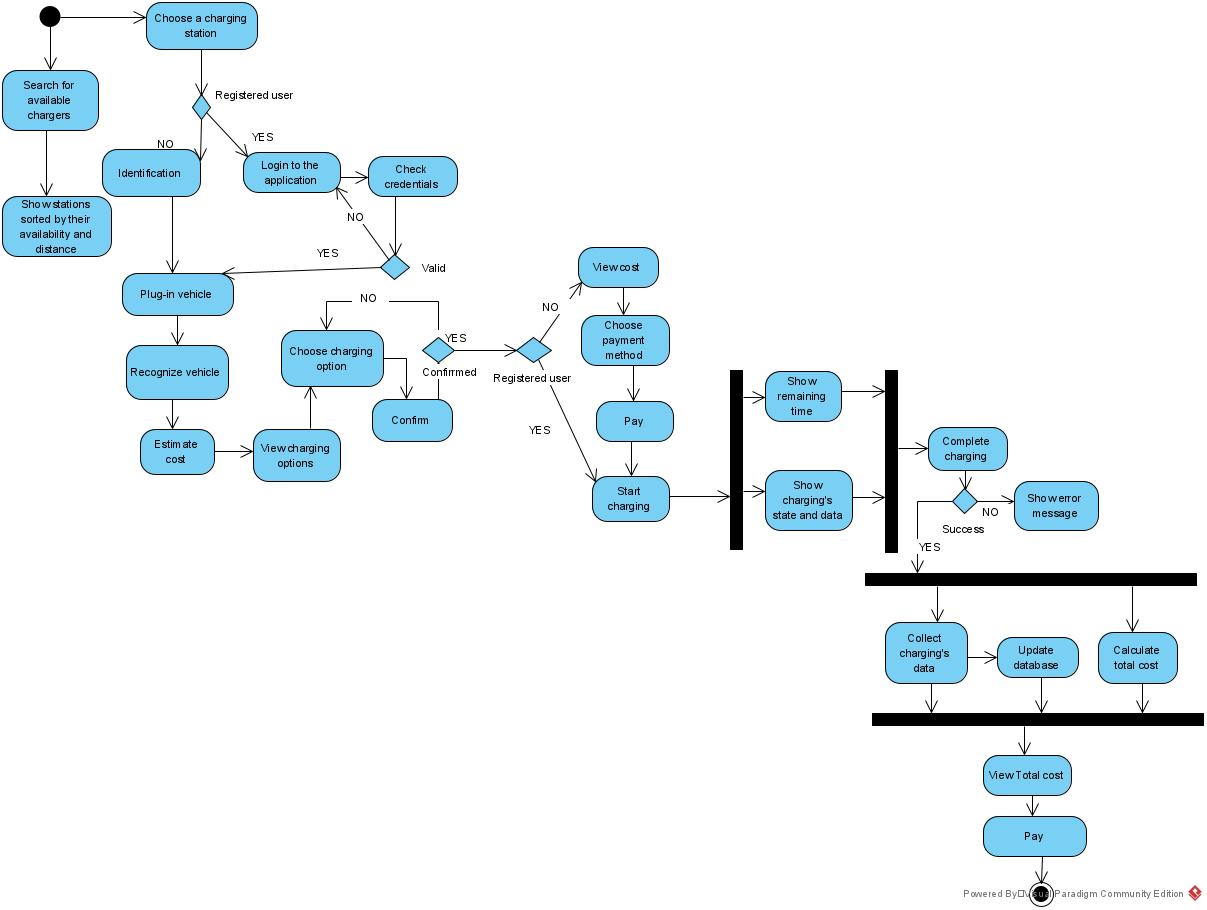
#### 3.1.1.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Oι χρήστες EV (EVU) μπορεί είτε να είναι συμβεβλημένοι με κάποιον CSO (Περίπτωση 1) είτε όχι (Περίπτωση 2).

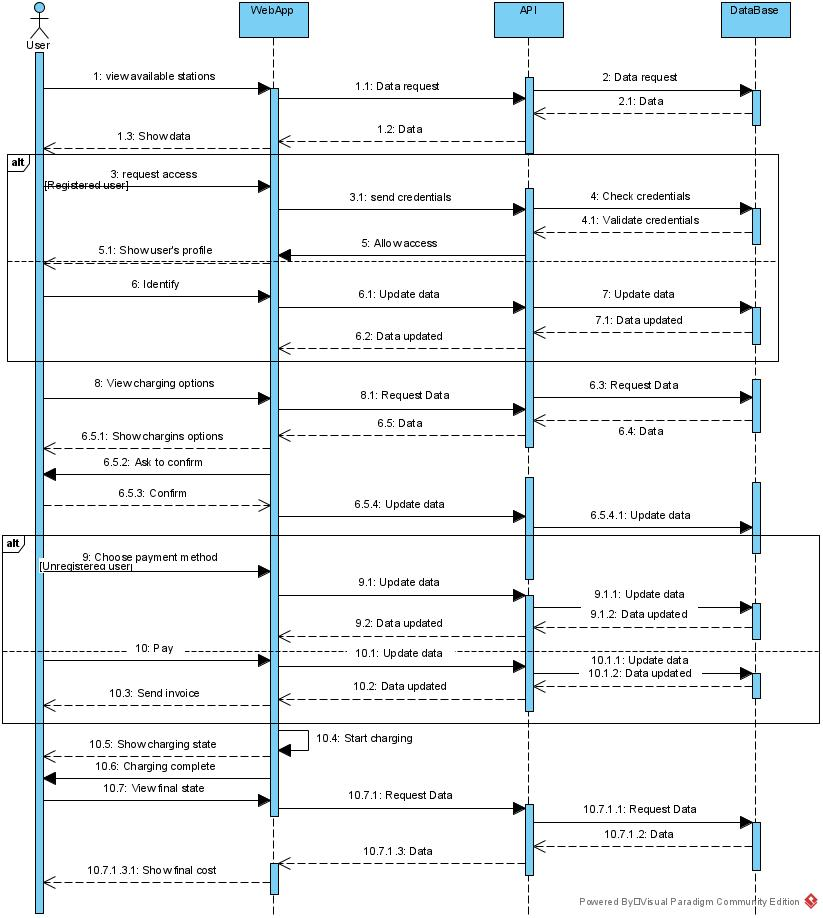
Βήματα για τη φόρτιση ενός EV με χρήση του συστήματος:

1. Ο EVU μέσω της διαδικτυακή μας εφαρμογή μπορεί να δει χάρτη με τους διαθέσιμους σταθμούς φόρτισης εφαρμόζοντας φίλτρα (CSO, τύπος οχήματος, απόσταση από ορισμένη τοποθεσία) και του δίνεται η δυνατότητα να κάνει κράτηση ενός συγκεκριμένου φορτιστή για διάστημα έως μιάμισης ώρας εισάγοντας τον αριθμό άδειας κυκλοφορίας (ΑΑΚ) του οχήματός του. Το βήμα αυτό δεν απαιτεί ο EVU να είναι συμβεβλημένος με κάποιον CSO. Επίσης για κάθε σταθμό φόρτισης η εφαρμογή εμφανίζει το κόστος φόρτισης. (Οι εγγεγραμμένοι χρήστες όταν χρησιμοποιούν την εφαρμογή συνδεδεμένοι στο λογαριασμό τους έχουν πρόσβαση σε ειδικές προσφορές).
2. Μετάβαση στο σταθμό φόρτισης.
3. Για τους μη εγγεγραμμένους χρήστες απαιτείται ταυτοποίηση με χρήση του ΑΑΚ του οχήματος.
4. Plugin του αυτοκινήτου και αναγνώριση των στοιχείων του οχήματος δηλ χαρακτηριστικά μπαταρίας και προδιαγραφές φόρτισης (μοντέλο, χωρητικότητα μπαταρίας, τύπος φόρτισης (AC ή DC)).
5. Εμφάνιση των επιλογών φόρτισης και του κόστους κάθε επιλογής.
6. Ο EVU επιλέγει ένα πρόγραμμα φόρτισης, το σύστημα του ζητάει να επιβεβαιώσει την επιλογή του (είτε από το φορτιστή είτε μέσω της εφαρμογής).
7. Σε αυτό το σημείο οι μη εγγεγραμμένοι χρήστες πρέπει να επιλέξουν τρόπο πληρωμής και να πληρώσουν (με κάρτα στο μηχάνημα).
8. Έναρξη φόρτισης. Κατά τη διάρκεια της φόρτισης στοιχεία όπως η ισχύς (kVA, kWh) και ο υπολειπόμενος χρόνος φόρτισης εμφανίζονται στην οθόνη του φορτιστή και στην εφαρμογή. Μέσω της εφαρμογής ο εγγεγραμμένος χρήστης μπορεί να δει το ιστορικό φορτίσεων για κάθε όχημά του (φορτίσεις που έγιναν σε σταθμούς του συγκεκριμένου CSO).
9. Ολοκλήρωση της φόρτισης. Ενημέρωση του ιστορικού φορτίσεων του οχήματος εάν το όχημα είναι καταχωρημένο από εγγεγραμμένο χρήστη. Παράλληλα ενημερώνεται η βάση δεδομένων του CSO όπου καταγράφονται στοιχεία για όλες τις φορτίσεις.
10. O λογαριασμός του εγγεγραμμένου χρήστη συνδέεται με σύστημα πληρωμών και μέσω της εφαρμογής μπορεί να επιλέξει τον τρόπο πληρωμής.

***Activity Diagram***

****

***Sequence Diagram***

****

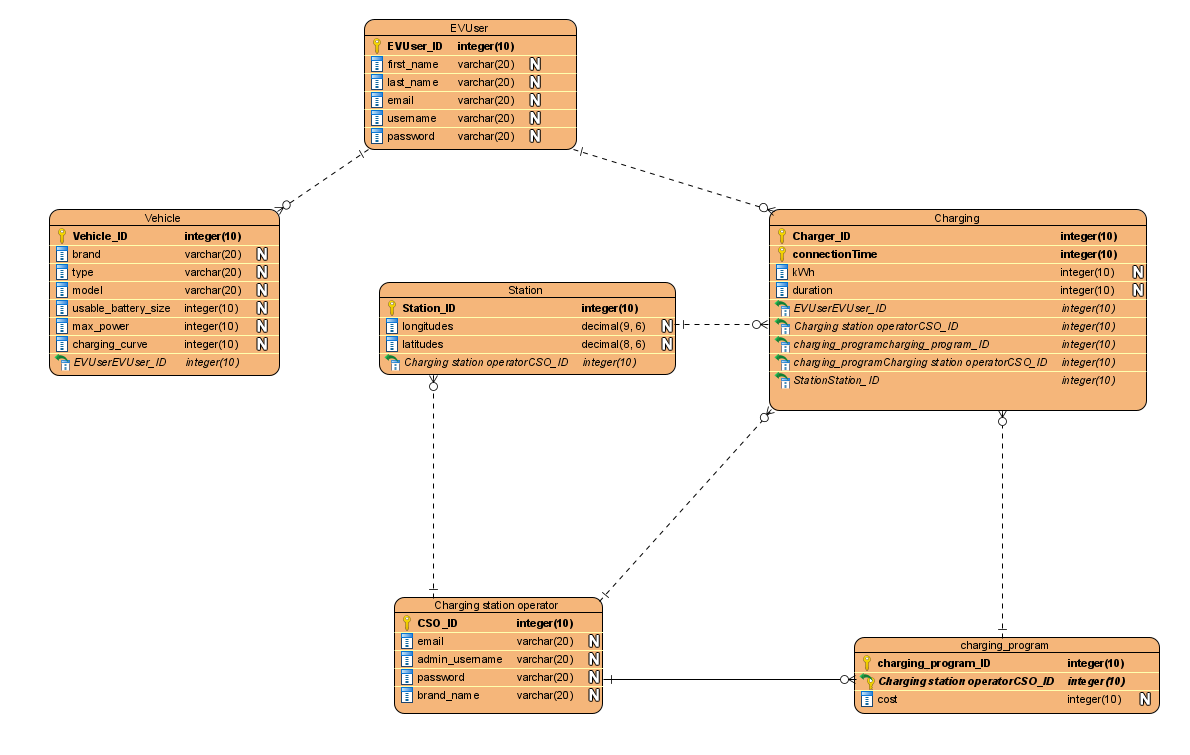
#### 3.1.1.7 Δεδομένα εξόδου

Την έξοδο του συστήματος αποτελούν δεδομένα της βάσης δεδομένων που παρουσιάζονται σε φιλική προς το χρήστη μορφή όπως για παράδειγμα λίστες και διαγράμματα.

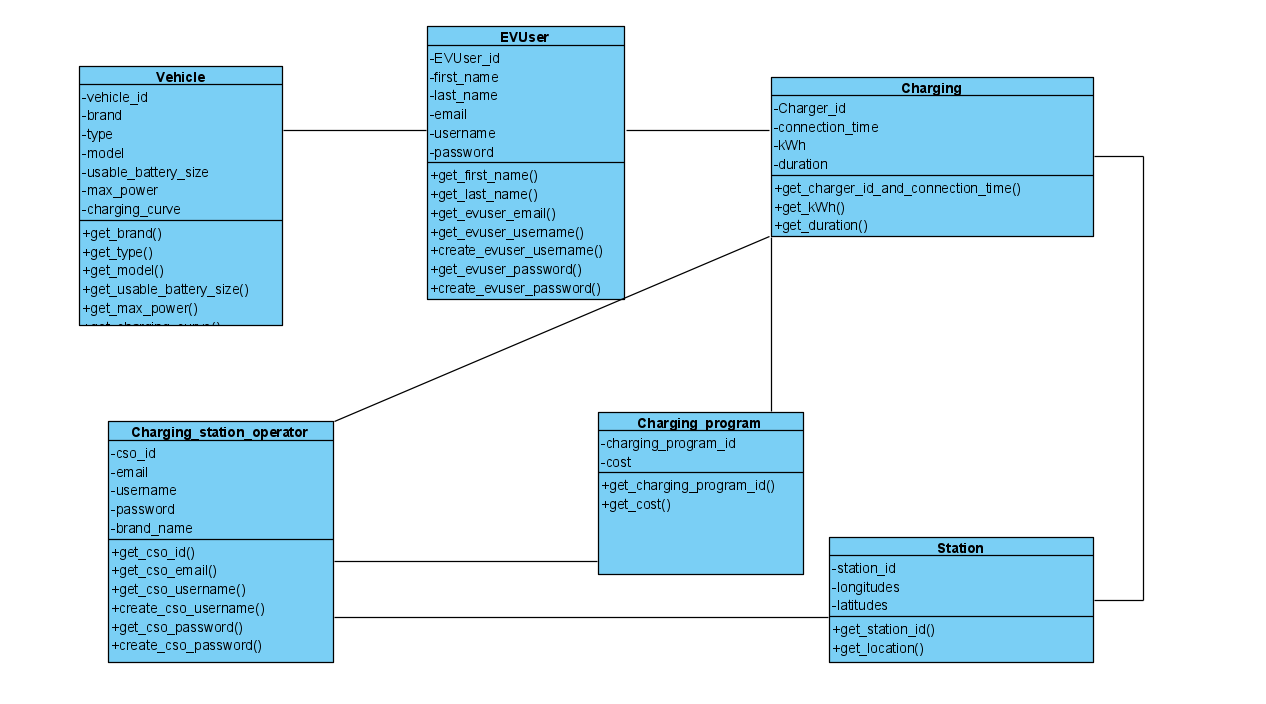
#### 3.1.1.8 Παρατηρήσεις

Παραθέτονται τα Entity Relationship και Class Diagrams που αφορούν το σχεδιασμό του λογισμικού.

***ER Diagram***



***Class Diagram***



### 3.1.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: Billing

#### 3.1.2.1 Χρήστες (ρόλοι) που εμπλέκονται

Οι εμπλεκόμενοι χρήστες είναι ο διαχειριστής του σταθμού φόρτισης και οι χρήστες των ηλεκτρικών οχημάτων που χρησιμοποιούν το σταθμό.

#### 3.1.2.2 Προϋποθέσεις εκτέλεσης

Ο ιδιοκτήτης ηλεκτρικού οχήματος για να χρησιμοποιήσει την εφαρμογή μπορεί να είναι εγγεγραμμένος χρήστης ή όχι. Αν δεν είναι εγγεγραμμένος χρήστης τότε πρέπει να προηγηθεί η πληρωμή της έναρξης της φόρτιση ενώ για τους εγγεγραμμένους χρήστες τηρείται μηνιαίος λογαριασμός και οι πληρωμές πραγματοποιούνται από σύστημα πληρωμών μία φορά το μήνα.

#### 3.1.2.3 Περιβάλλον εκτέλεσης

Τα use cases στο παραπάνω Use Case UML διάγραμμα για την πρώτη περίπτωση χρήσης εκτελούνται σε κάθε σύστημα που υποστηρίζει Command Line Interface και διαθέτει περιηγητή διαδικτύου για την περιήγηση στη διαδικτυακή εφαρμογή.

Εξωτερική διεπαφή τράπεζας (Payment Process Getaway).

#### 3.1.2.4 Δεδομένα εισόδου

Ο εγγεγραμμένος χρήστης εισάγει το αναγνωριστικό του όνομα και τον μοναδικό κωδικό του, ενώ ο μη εγγεγραμμένος εισάγει τον αριθμό άδειας κυκλοφορίας του οχήματός του.

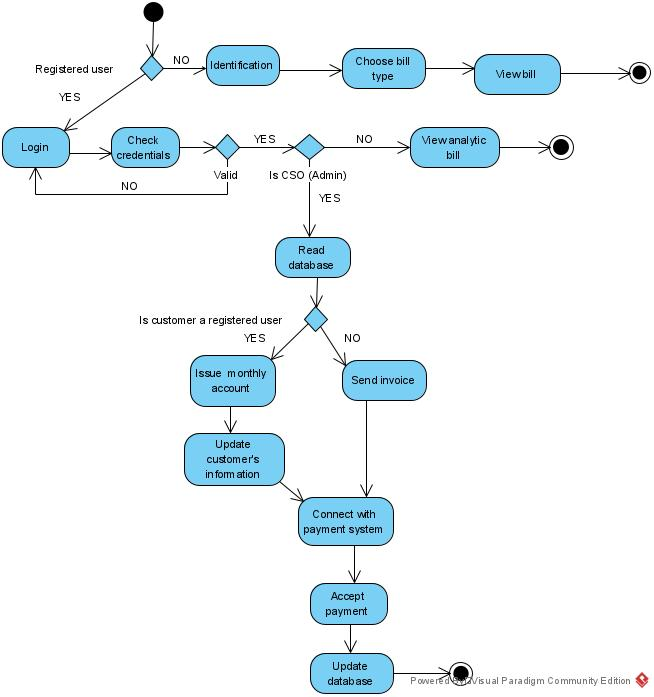
#### 3.1.2.5 Παράμετροι

Δεν υπάρχουν.

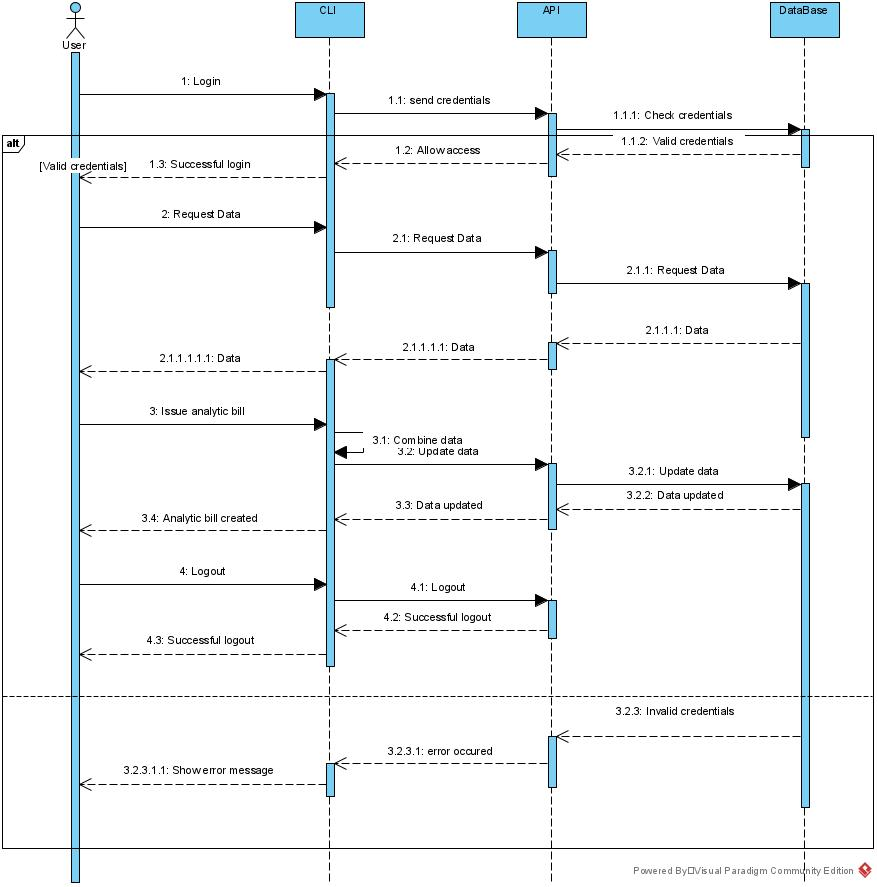
#### 3.1.2.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

1. Είσοδος του CSO στην εφαρμογή, έλεγχος εγκυρότητας username και password
2. Προσπέλαση δεδομένων, συνδυασμός και επεξεργασία τους
3. Για τους εγγεγραμμένους χρήστες: έκδοση μηνιαίου αναλυτικού λογαριασμού με βάση τα δεδομένα που καταχωρούνται για κάθε χρήστη ο οποίος αποστέλλεται στο email κάθε EV User, ανανέωση καρτέλας χρήστη με τον αναλυτικό λογαριασμό ώστε να τον βλέπει όταν εισέρχεται στην εφαρμογή.
4. Για τους μη εγγεγραμμένους χρήστες είτε εκτύπωση είτε αποστολή στο email τους αναλυτικού λογαριασμού. (η εφαρμογή παρέχει δυνατότητα επιλογής)

***Activity Diagram***

****

***Sequence Diagram***

**

## 3.2 Απαιτήσεις επιδόσεων

Το λογισμικό πρέπει να ανταποκρίνεται στις αντίστοιχες κλήσεις εξωτερικών συστημάτων γρήγορα (ώστε να εξασφαλίζεται σε κάθε περίπτωση ελάχιστο δυνατό latency). Επιπρόσθετα, πρέπει το throughput δεδομένου των quotas για κάθε χρήστη να είναι μέγιστο, δηλαδή να τηρείται πάντα το μέγιστο επιτρεπόμενο δυνατό πλήθος αιτήσεων.

## 3.3 Απαιτήσεις οργάνωσης δεδομένων

### 3.3.1 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα

Εφαρμόζουμε κατάλληλους περιορισμούς (quotas), με σκοπό τον έλεγχο ορθολογικής κατανάλωσης από κάθε χρήστη των υπηρεσιών που προσφέρουμε. Για την διαχείριση της αξιοποίησης των πόρων, ελέγχουμε τον όγκο των δεδομένων χρησιμοποιούνται σε κάθε κλήση των χρηστών, και θέτουμε άνω όριο απόκτησης πόρων για κάθε μέρα σε όλους τους χρήστες. Επομένως, λόγω περιορισμένων πόρων, το λογισμικό δεν δύναται να ανταποκριθεί σε όλες τις κλήσεις των χρηστών.

## 3.4 Περιορισμοί σχεδίασης

Για το back-end χρησιμοποιούμε Java8, MySQL14.14, RESTful API. Για το CLI χρησιμοποιούμε Python3.6 και για το build automation tool χρησιμοποιούμε Gradle. Απαιτείται εναρμόνιση με το πρότυπο ISO/IEC/ IEEE 42010 και με το πρωτόκολλο OCPP σχετικά με την επικοινωνία μεταξύ του σταθμού φόρτισης και του συστήματος διαχείρισης.

## 3.5 Λοιπές απαιτήσεις

### 3.5.1 Απαιτήσεις διαθεσιμότητας λογισμικού

Κάθε αίτηση να λαμβάνει σωστή απάντηση, χωρίς μη προγραμματισμένα μηνύματα σφαλμάτων. Είναι σημαντική η ενημέρωση των χρηστών για τις ώρες και μέρες συντήρησης του συστήματος εκ των προτέρων, ώστε όποιοι έχουν ανάγκη να προσπελάσουν τα δεδομένα, να κάνουν τον ανάλογο προγραμματισμό. Για τους διαχειριστές των σταθμών θα ήταν πρόβλημα η αδυναμία πρόσβασης στα δεδομένα για μεγάλο χρονικό διάστημα, καθώς κάτι τέτοιο θα είχε οικονομικές συνέπειες και θα επηρέαζε αρνητικά τη θέση τους στην αγορά.

### 3.5.2 Απαιτήσεις ασφάλειας

* Επίπεδα προνομίων πρόσβασης με στόχο της διατήρηση της ακεραιότητας και αξιοπιστίας των δεδομένων. Οι διαχειριστές των σταθμών έχουν πρόσβαση σε όλα τα δεδομένα, οι εγγεγραμμένοι χρήστες έχουν πρόσβαση μόνο στα ορισμένα δεδομένα που συνδέονται με το λογαριασμό τους και οι μη εγγεγραμμένοι χρήστες έχουν πρόσβαση μόνο σε δεδομένα που αφορούν κάθε φόρτιση του οχήματός τους μόνο κατά τη διάρκεια της φόρτισης χωρίς τη δυνατότητα να δουν ιστορικό φορτίσεων.
* Οι κωδικοί (passwords) αποθηκεύονται σε μορφή που τους καθιστά μη ανακτήσιμους, χρησιμοποιώντας hashing.
* Ασφαλής αποθήκευση των κλειδιών που χρησιμοποιούνται για το REST ΑPΙ.
* Δυνατότητα πρόσβασης στη βάση δεδομένων έχει μόνο το REST API. H πρόσβαση αυτή αξιοποιεί την ασφάλεια που παρέχουν τα διάφορα πρωτόκολλα επικοινωνίας στα οποία βασίζεται το API.

### 3.5.3 Απαιτήσεις συντήρησης

Ο έγκαιρος εντοπισμός των αστοχιών και η συντήρηση του συστήματος κρίνονται απαραίτητα για τη διάθεση μιας πλήρους λειτουργικής και αποκρίσιμης εφαρμογής. Για την εκτίμηση των απαιτήσεων της συντήρησης του λογισμικού χρησιμοποιούμε τη μετρική *MTTR* (μέσος χρόνος αποκατάστασης σφαλμάτων).