Έγγραφο απαιτήσεων λογισμικού (SRS)



EVolution ~ Supporting Electric Vehicles

**Πίνακας Περιεχομένων**

[Εισαγωγή](#_yyvtto28tlf7)

[1.1 Σκοπός του λογισμικού](#_ajlixp2gu6h2)

[1.2 Διεπαφές (interfaces)](#_3hv5xrepsf2m)

[1.2.1 Διεπαφές με εξωτερικά συστήματα](#_bzkommp6vtlx)

[1.2.2 Διεπαφές με το χρήστη](#_bv2092c2tfe5)

[Προδιαγραφές απαιτήσεων λογισμικού](#_zcfmae5opvm7)

[2.1 Περιπτώσεις χρήσης](#_85yzjb1qyx4j)

[2.1.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: Διαδικασία της φόρτισης](#_q3dtkx6lomuq)

[2.1.1.1 Χρήστες (ρόλοι) που εμπλέκονται](#_g1brsbelgj6g)

[2.1.1.2 Προϋποθέσεις εκτέλεσης](#_jx9btq1j1j28)

[2.1.1.3 Περιβάλλον εκτέλεσης](#_nm6mjpdk7mjy)

[2.1.1.4 Δεδομένα εισόδου](#_cpv8yjibltow)

[2.1.2.5 Παράμετροι](#_hrppjhblgtuu)

[2.1.1.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά](#_6cfn9540ghy)

[2.1.1.7 Δεδομένα εξόδου](#_i2a5gk1egvjb)

[2.1.1.8 Παρατηρήσεις](#_x9r758imsuvg)

[2.1.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: Έκδοση συγκεντρωτικού λογαριασμού και ιστορικού φορτίσεων](#_u8nhemak0493)

[2.1.2.1 Χρήστες (ρόλοι) που εμπλέκονται](#_kkasneezj2is)

[2.1.2.2 Προϋποθέσεις εκτέλεσης](#_yeu0ejg2bidz)

[2.1.2.3 Περιβάλλον εκτέλεσης](#_c8gggog0oprc)

[2.1.2.4 Δεδομένα εισόδου](#_7aeb9wwmwj31)

[2.1.2.5 Παράμετροι](#_uqps3to2mh7u)

[2.1.2.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά](#_ahgm28xsvqck)

[2.1.2.7 Δεδομένα εξόδου](#_jpf4cdsrhk74)

[2.1.2.8 Παρατηρήσεις](#_h7dhy24vq5ry)

[2.1.3 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 3: Εμφάνιση στατιστικών που αφορούν τις θέσεις φόρτισης/στάθμευσης και τους διάφορους τύπους φορτιστών](#_jg3c5i1bzse0)

[2.1.3.1 Χρήστες (ρόλοι) που εμπλέκονται](#_52i0bu14zod3)

[2.1.3.2 Προϋποθέσεις εκτέλεσης](#_3gmw8v6s2as2)

[2.1.3.3 Περιβάλλον εκτέλεσης](#_b8zxf61bdaor)

[2.1.3.4 Δεδομένα εισόδου](#_1484bfdylqae)

[2.1.3.5 Παράμετροι](#_h5t40sd7fhq)

[2.1.3.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά](#_ly5ylo6difet)

[2.1.3.7 Δεδομένα εξόδου](#_k9dhygyxn12k)

[2.1.3.8 Παρατηρήσεις](#_fm7lp525ltwd)

[2.2 Απαιτήσεις επιδόσεων](#_az9zzr2e7y9z)

[2.3 Απαιτήσεις οργάνωσης δεδομένων](#_ex2doejl6wn7)

[2.3.1 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα](#_mwytvsqa2e1s)

[2.3.2 Μοντέλο δεδομένων](#_mqrqxya5fwj9)

[2.4 Περιορισμοί σχεδίασης](#_6uawteex6hg2)

[2.5 Λοιπές απαιτήσεις](#_ewn73unbcl)

[2.5.1 Απαιτήσεις διαθεσιμότητας λογισμικού](#_bd78vlg5immr)

[2.5.2 Απαιτήσεις ασφάλειας](#_p8v40rsm4chl)

[2.5.3 Απαιτήσεις συντήρησης](#_qgd4cyijfxnk)

# 

# Εισαγωγή

## 1.1 Σκοπός του λογισμικού

Το σύστημα που αναπτύσσουμε στοχεύει στην εξυπηρέτηση των ιδιοκτητών ηλεκτρονικών οχημάτων, διευκολύνοντας τη διαδικασία των φορτίσεων των οχημάτων τους, καθώς και της πληρωμής. Με απεικονίσεις δεδομένων που συλλέγονται κατά τη φόρτιση, υπολογισμό ενός προσεγγιστικού συνολικού κόστους για τις φορτίσεις, αλλά και παροχή δυνατότητας πληρωμής μέσω της εφαρμογής με κάρτα, το σύστημα μας ελπίζουμε πως θα δώσει στους ιδιοκτήτες των ηλεκτρικών οχημάτων δυνατότητες κλειδιά για κάθε φόρτιση των οχημάτων τους, και θα βοηθήσει στην πλήρη αυτοματοποίηση τους.

Ωστόσο, το σύστημα μας στοχεύει παράλληλα και στην εξυπηρέτηση των ιδιοκτητών των σταθμών στάθμευσης/φόρτισης, συγκρατώντας και προβάλλοντας σημαντικά στοιχεία σχετικά με τις θέσεις που οι ίδιοι κατέχουν. Συγκεκριμένα, καταγράφονται και προβάλλονται, με διαγράμματα, στατιστικά τόσο για τους φορτιστές της κατοχής τους, όσο και για τις θέσεις. Τα δεδομένα αυτά, δύνανται να χρησιμοποιηθούν για μεγαλύτερη βελτίωση της παροχής των υπηρεσιών από τους σταθμούς, αλλά και για κατανόηση της αποδοτικότητας και απήχησης τους.

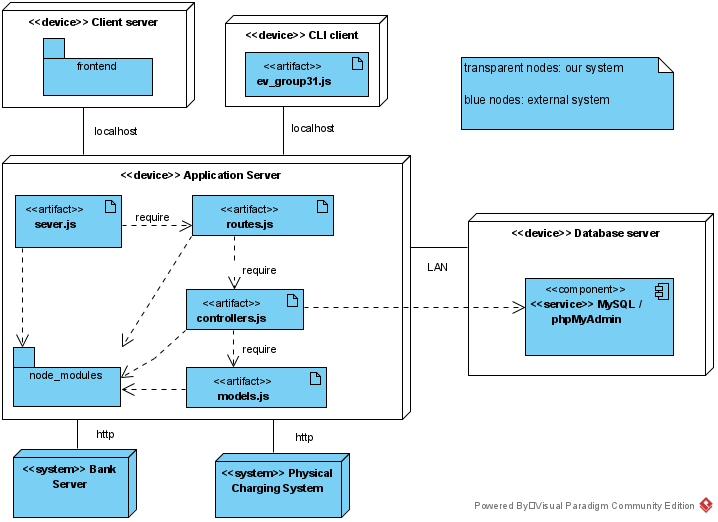
Στην ολοένα και αυξανόμενη βιομηχανία της ηλεκτροκίνησης, προσπαθούμε να σταθούμε συνοδοιπόροι και να συμβάλουμε στην ευκολότερη καθιέρωση της στη χώρα μας. Άλλωστε στόχος μας είναι η περαιτέρω ανάπτυξη του λογισμικού μας στο μέλλον, με ακόμα περισσότερες παροχές.

## 1.2 Διεπαφές (interfaces)

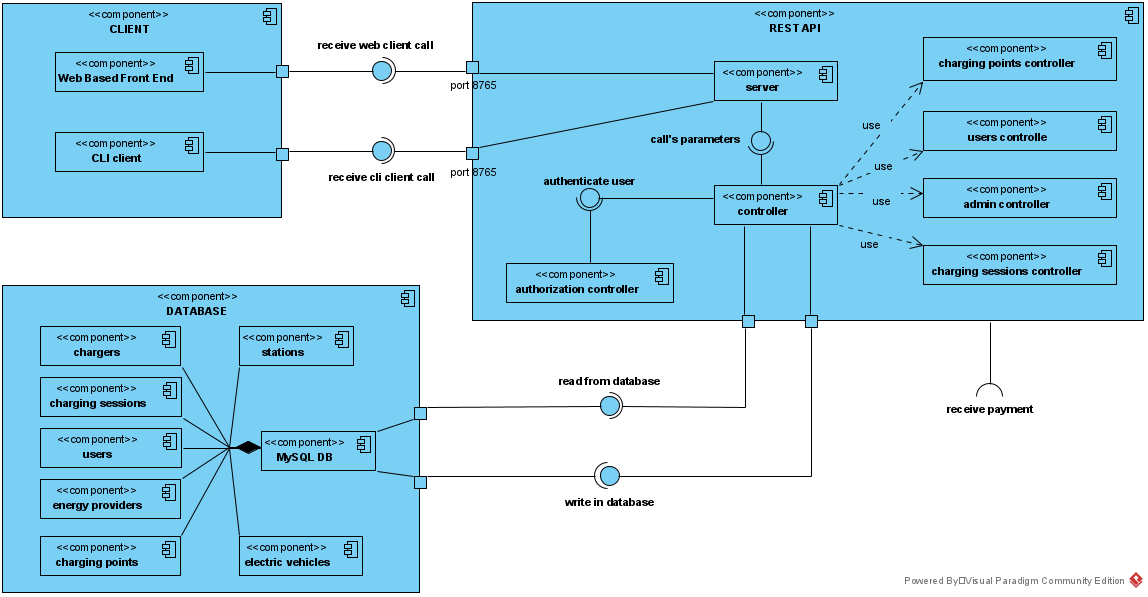
### 1.2.1 Διεπαφές με εξωτερικά συστήματα

Το σύστημα μας αλληλεπιδρά με αρκετά εξωτερικά συστήματα, προκειμένου να μπορεί να καλύπτει τις δυνατότητες που επιθυμούμε να προσφέρονται στους χρήστες:

1. Σύστημα φόρτισης ηλεκτρικών αυτοκινήτων των σταθμών: Το σύστημα μας δεν υλοποιεί το ίδιο την φόρτιση ενός αυτοκινήτου, αλληλεπιδρά απλώς με μια διεπαφή του συστήματος φόρτισης που υπάρχει στους συνεργαζόμενους σταθμούς προκειμένου να μπορεί να ελέγξει την αρχή, παύση και ολοκλήρωση της φόρτισης και να πάρει από αυτό δεδομένα σχετικά με την εκάστοτε φόρτιση.
2. Σύστημα διαδικτυακών πληρωμών: Το σύστημα μας δεν δύναται να πραγματοποιεί το ίδιο πληρωμές με κάρτα, απλώς συνεργάζεται άμεσα και αλληλεπιδρά με σύστημα ασφαλών διαδικτυακών πληρωμών, στο οποίο το λογισμικό μας δίνει τα στοιχεία της κάρτας του εκάστοτε χρήστη των θέσεων φόρτισης για να φροντίσει να ολοκληρώσει την πληρωμή.
3. PHP MyAdmin: Πρόκειται για το σύστημα που επιλέξαμε για την υποστήριξη της βάσης δεδομένων του λογισμικού μας. Το ίδιο υποστηρίζει τόσο MySQL όσο και MariaDB, και δίνει ένα αρκετά φιλικό UI μέσω του διαδικτύου για έλεγχο της βάσης, επιτρέποντας παράλληλα και την χρήση της βάσης υποβάλλοντας queries μέσω κώδικα στο backend.



*Εικόνα 1. UML Deployment Diagram*

**

*Εικόνα 2. UML Component Diagram*



### 1.2.2 Διεπαφές με το χρήστη

Αναφέρουμε πως η πλατφόρμα μας αναγνωρίζει μόνο τους ακόλουθους τρεις (3) τύπους χρηστών:

1. Ιδιοκτήτης Ηλεκτρικού/ων Οχήματος/ων: Ο χρήστης αυτός είναι άμεσα συνδεδεμένος με ένα ή περισσότερα ηλεκτρικά οχήματα, τα δεδομένα των οποίων κρατούνται στη βάση δεδομένων μας. Έχει δυνατότητες έναρξης και ελέγχου των φορτίσεων των οχημάτων του, καθώς και διατηρείται γι’ αυτόν ιστορικό φορτίσεων και πληρωμών.
2. Ιδιοκτήτης χώρου φόρτισης/στάθμευσης: Ο χρήστης αυτός είναι άμεσα συνδεδεμένος με συγκεκριμένες θέσεις φόρτισης (και μηχανήματα που παρέχουν ηλεκτρική ενέργεια στο οχήματα που σταθμεύουν σε αυτές). Μπορεί να διατηρεί ιστορικό για τα δεδομένα που αφορούν τις φορτίσεις σε θέσεις που του ανήκουν.
3. Διαχειριστής συστήματος: Ο ίδιος είναι υπεύθυνος για την δημιουργία και καταχώρηση νέων χρηστών στο σύστημα και φέρει, ακόμα, δικαιώματα που του επιτρέπουν να ανεβάζει ετεροχρονισμένα δεδομένα φορτίσεων, αρκεί τα ίδια να έχουν κατάλληλη csv μορφή. Ακόμα, μπορεί να ελέγχει τους διάφορους χρήστες.

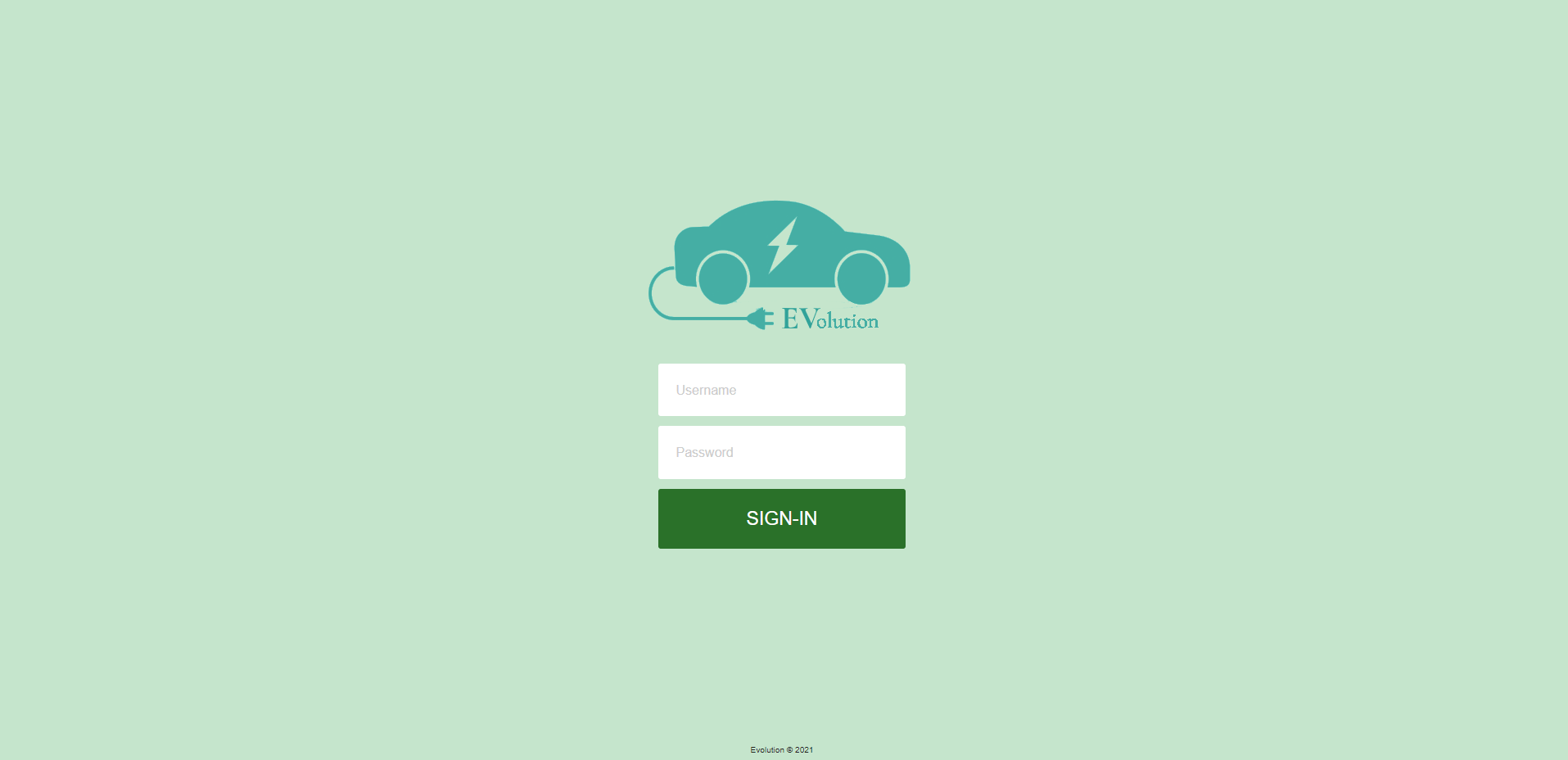
Η αλληλεπίδραση των χρηστών με το λογισμικό μας γίνεται με τους ακόλουθους δύο (2)τρόπους:

1. Frontend εφαρμογή: Πρόκειται για ένα UI φιλικό προς το χρήστη και αρκετά εύκολο στη χρήση. Προσφέρει δυνατότητα login και logout, ενώ παράλληλα αναγνωρίζει το ρόλο του εκάστοτε χρήστη και του επιτρέπει να έχει πρόσβαση στις παροχές του λογισμικού προς αυτόν με τρόπο κατανοητό και όμορφα οπτικοποιημένο. Το ίδιο τρέχει σε Web Browser και ουσιαστικά αποτελεί τον κορμό χειρισμού των δυνατοτήτων του λογισμικού.
2. Cli εφαρμογή: Πρόκειται για ένα λειτουργικό command line interface που παρέχει δυνατότητες ως προς τους Admin, τις οποίες δεν μπορεί να έχει μέσω της Frontend εφαρμογής. Συγκεκριμένα, η καταχώρηση νέων χρηστών στο σύστημα, η ενημέρωση των ήδη καταγεγραμμένων, αλλά και αρκετές ακόμα λειτουργίες προσφέρονται αυστηρά και μόνο μέσω Cli και μόνο εφόσον διαπιστωθεί πως ο χρήστης έχει δικαιώματα Admin.



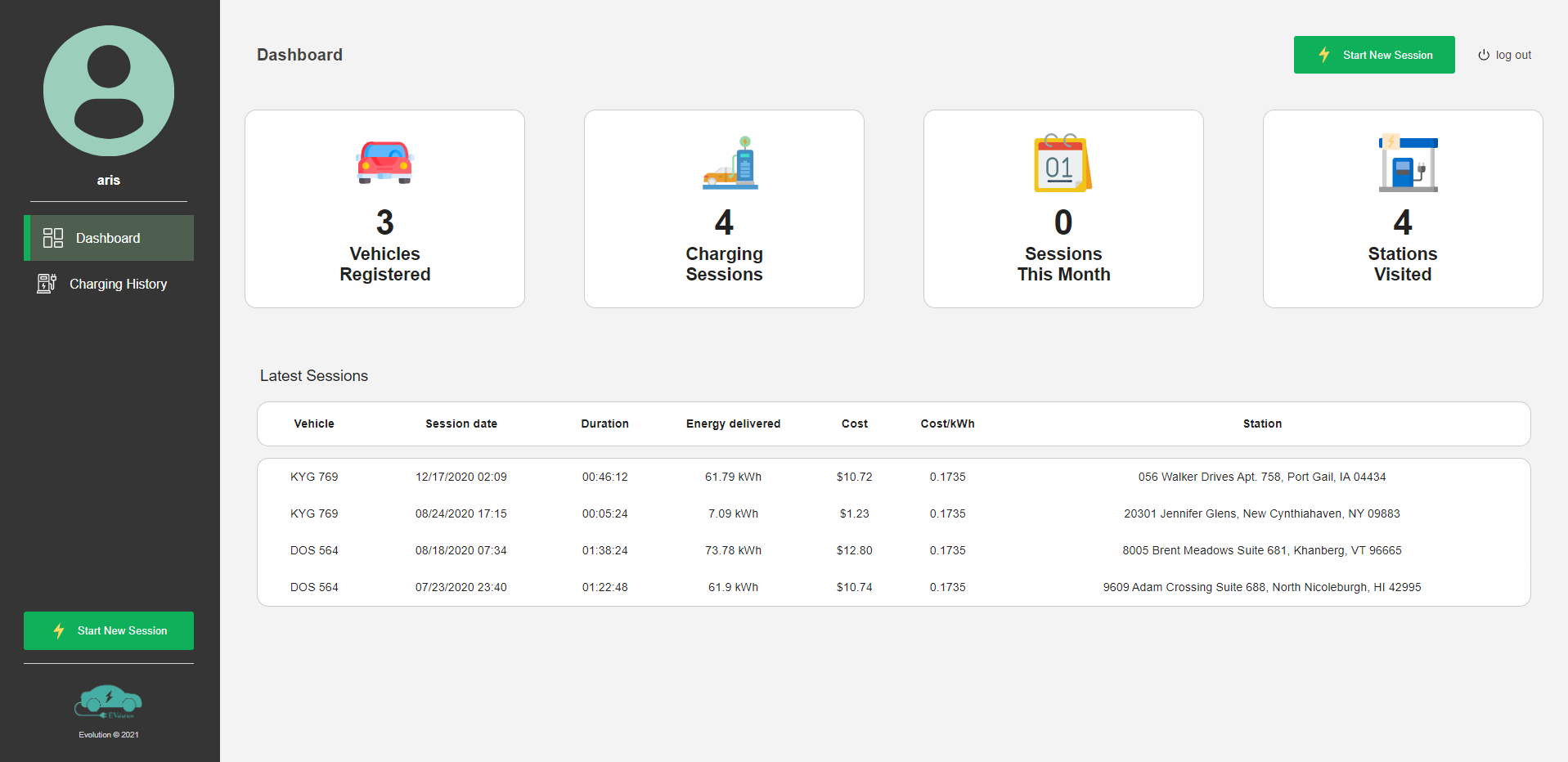
|  |  |  |  |
| --- | --- | --- | --- |
|  | Ιδιοκτήτης Ηλεκτρικού/ων Οχήματος/ων | Ιδιοκτήτης χώρου στάθμευσης | Διαχειριστής συστήματος |
| Cli εφαρμογή | ✓ | ✓ | ✓ |
| Frontend εφαρμογή | ✓ | ✓ | ✓ |

Παρέχονται μερικά χαρακτηριστικά στιγμιότυπα οθόνης από τη Frontend εφαρμογή, για καλύτερη κατανόηση της λειτουργίας της, ενώ κάποια πιο ειδικά στιγμιότυπα τοποθετούνται στις περιπτώσεις χρήσης, για την καλύτερη κατανόηση τους. Ακόμη, παρέχεται ένα στιγμιότυπο οθόνης από τη cli εφαρμογή μας, για καλύτερη αποτύπωση της λειτουργίας της:

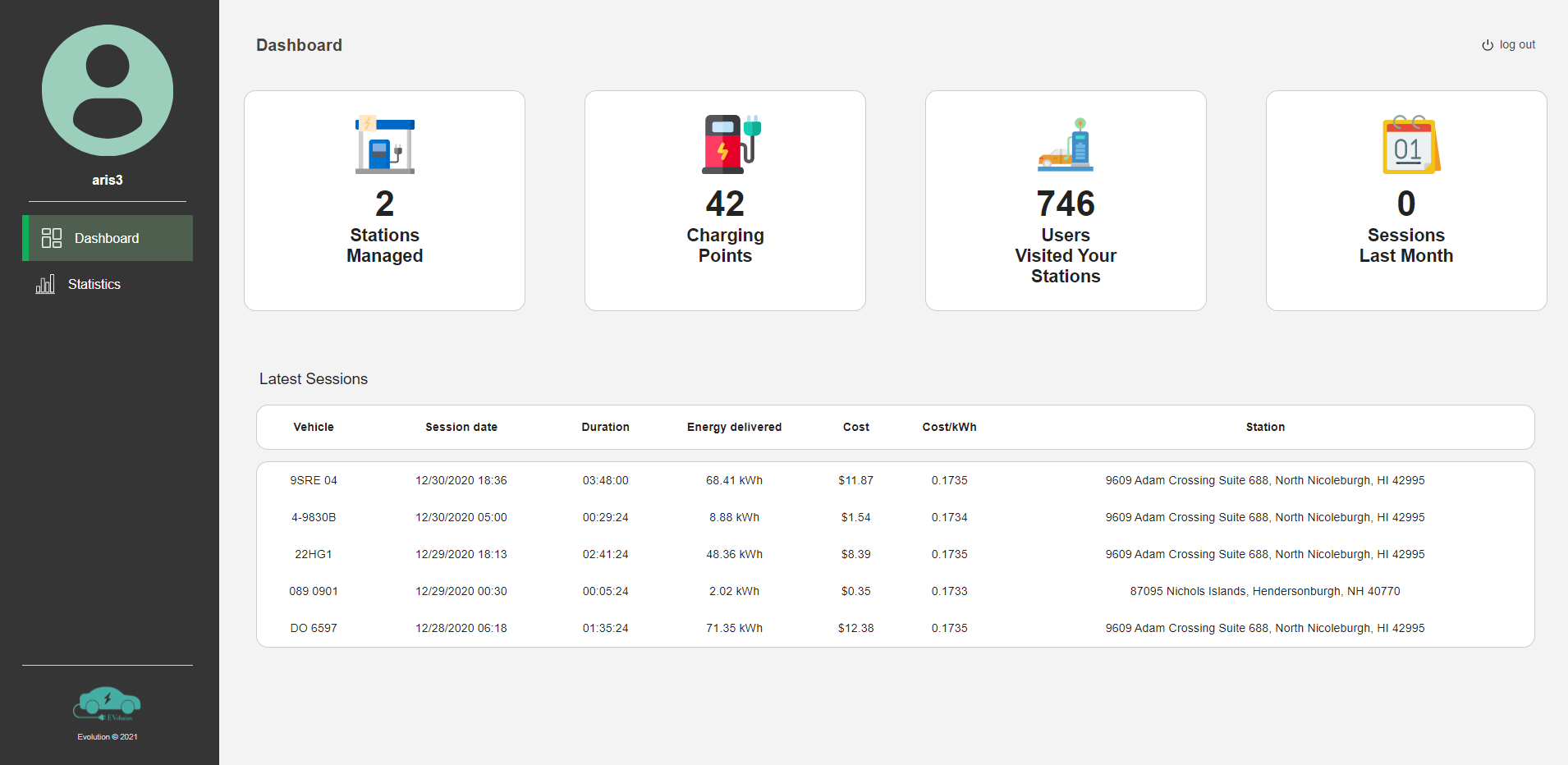


*Εικόνα 3. Σελίδα login στην frontend εφαρμογή*



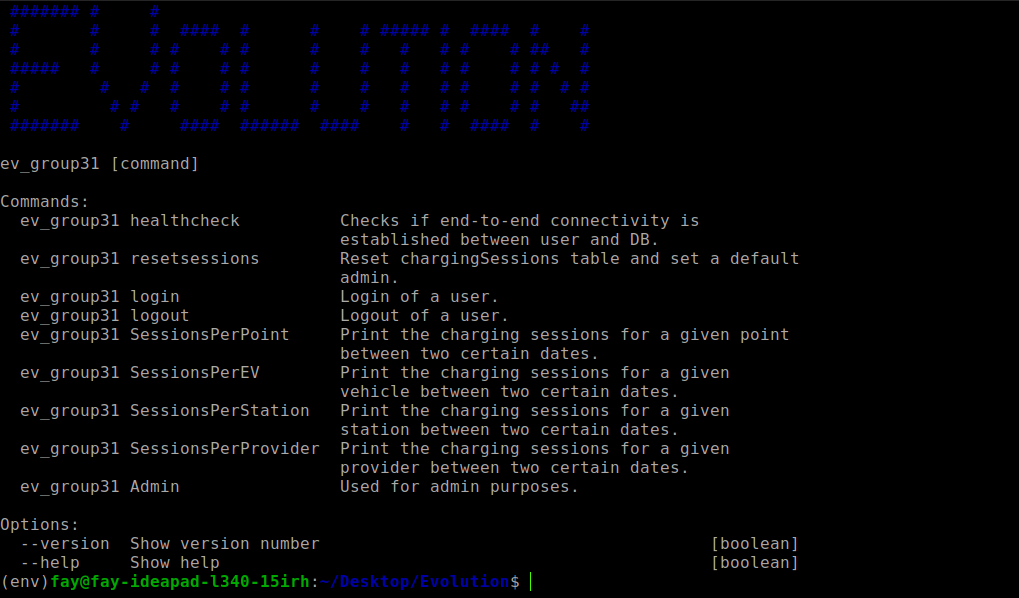
**

*Εικόνα 4. Αρχική σελίδα για τους χρήστες που αναγνωρίζονται ως “Ιδιοκτήτης Ηλεκτρικού/ων Οχήματος/ων”*

**

*Εικόνα 5: Αρχική σελίδα για τους χρήστες που αναγνωρίζονται ως “Ιδιοκτήτης χώρου στάθμευσης/φόρτισης”*



**

*Εικόνα 6: Εικόνα από τη cli εφαρμογή για την εντολή ev\_group31 --help*

# Προδιαγραφές απαιτήσεων λογισμικού

## 2.1 Περιπτώσεις χρήσης

### 2.1.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: Διαδικασία της φόρτισης

#### *2.1.1.1 Χρήστες (ρόλοι) που εμπλέκονται*

Αυτή η χρήση του λογισμικού αφορά τον εκάστοτε χρήστη που αναγνωρίζεται ως “Ιδιοκτήτης Ηλεκτρικού/ων Οχήματος/ων”, ο οποίος συνδέεται στην εφαρμογή προκειμένου να ξεκινήσει μια φόρτιση.

#### *2.1.1.2 Προϋποθέσεις εκτέλεσης*

Για να μπορέσει να εκτελεστεί αυτή η περίπτωση χρήσης, θα πρέπει αρχικά ο ιδιοκτήτης να έχει σταθμεύσει το όχημα του σε μια θέση που παρέχει φόρτιση. Κατόπιν, θα πρέπει να έχει συνδέσει το όχημα του με τον κατάλληλο εξοπλισμό. Επίσης, είναι απαραίτητο ο χρήστης να έχει ενεργό λογαριασμό στο σύστημα και να έχει καταχωρήσει τα δεδομένα του οχήματος του σε αυτό. Ακόμα, χρειάζεται να συνδεθεί στην εφαρμογή.



#### *2.1.1.3 Περιβάλλον εκτέλεσης*

Η περίπτωση χρήσης αυτής εκτελείται στη διεπαφή του χρήστη. Ωστόσο, χρειάζεται να “χτυπηθεί” το DBMS με χρήση του Rest API.

#### *2.1.1.4 Δεδομένα εισόδου*

Ως δεδομένα εισόδου το σύστημα λαμβάνει από το χρήστη:

* Τον επιθυμητό χρόνο φόρτισης
* Την επιθυμία για μερική/ολική φόρτιση
* Τα στοιχεία της κάρτας του χρήστη (σε περίπτωση επιλογής πληρωμής με κάρτα)

Ενώ, παράλληλα, είσοδο έχει και από το μηχάνημα της φόρτισης:

* Τη θέση φόρτισης (που συνδέεται άμεσα με το σταθμό και το είδος του φορτιστή)
* Το όχημα
* Το ποσοστό φόρτισης του οχήματος

#### *2.1.1.5 Παράμετροι*

|  |  |  |
| --- | --- | --- |
| **Πεδίο** | **Τύπος Δεδομένων** | **Περιγραφή** |
| electricVehicleId | Integer | Το id του οχήματος προς φόρτιση |
| chargingPointId | Integer | Το id της θέσης στην οποία πρόκειται να γίνει η φόρτιση |
| totalCost | Float | Το συνολικό κόστος της φόρτισης |
| paymentType | String | Η επιθυμητή μέθοδος πληρωμής για την φόρτιση |
| energyDelivered | Float | Το ποσό της ενέργειας που μεταφέρθηκε στο όχημα κατά τη φόρτιση |
| pointsAwarded | Float | Οι πόντοι επιβράβευσης που κέρδισε ο χρήστης από τη φόρτιση |
| startTime | Date | Η χρονική στιγμή έναρης της φόρτισης |
| endTime | Date | Η χρονική στιγμή λήξης της φόρτισης |



#### *2.1.1.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά*

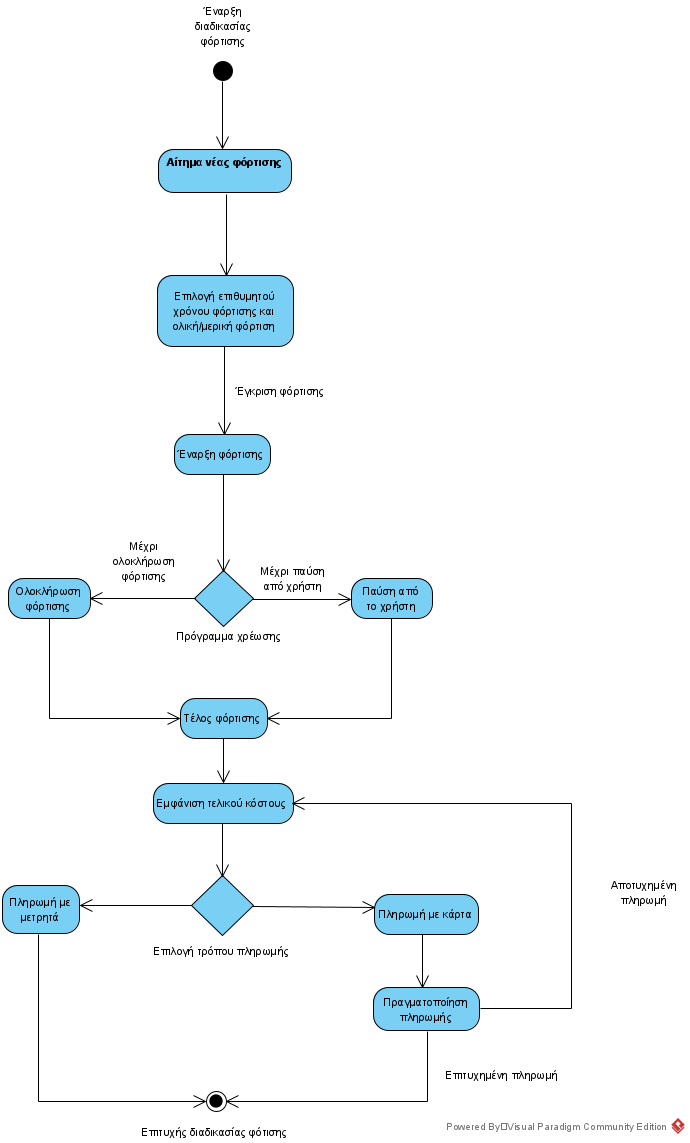
Θα περιγράψουμε με τη μορφή βημάτων την αλληλουχία ενεργειών τόσο όσων αφορά τη βασική, όσο και τις εναλλακτικές ροές. Στη συνέχεια, θα παρατεθούν τα διαγράμματα UML Activity και UML Sequence που αφορούν αυτή την περίπτωση χρήσης.

Βασική ροή:

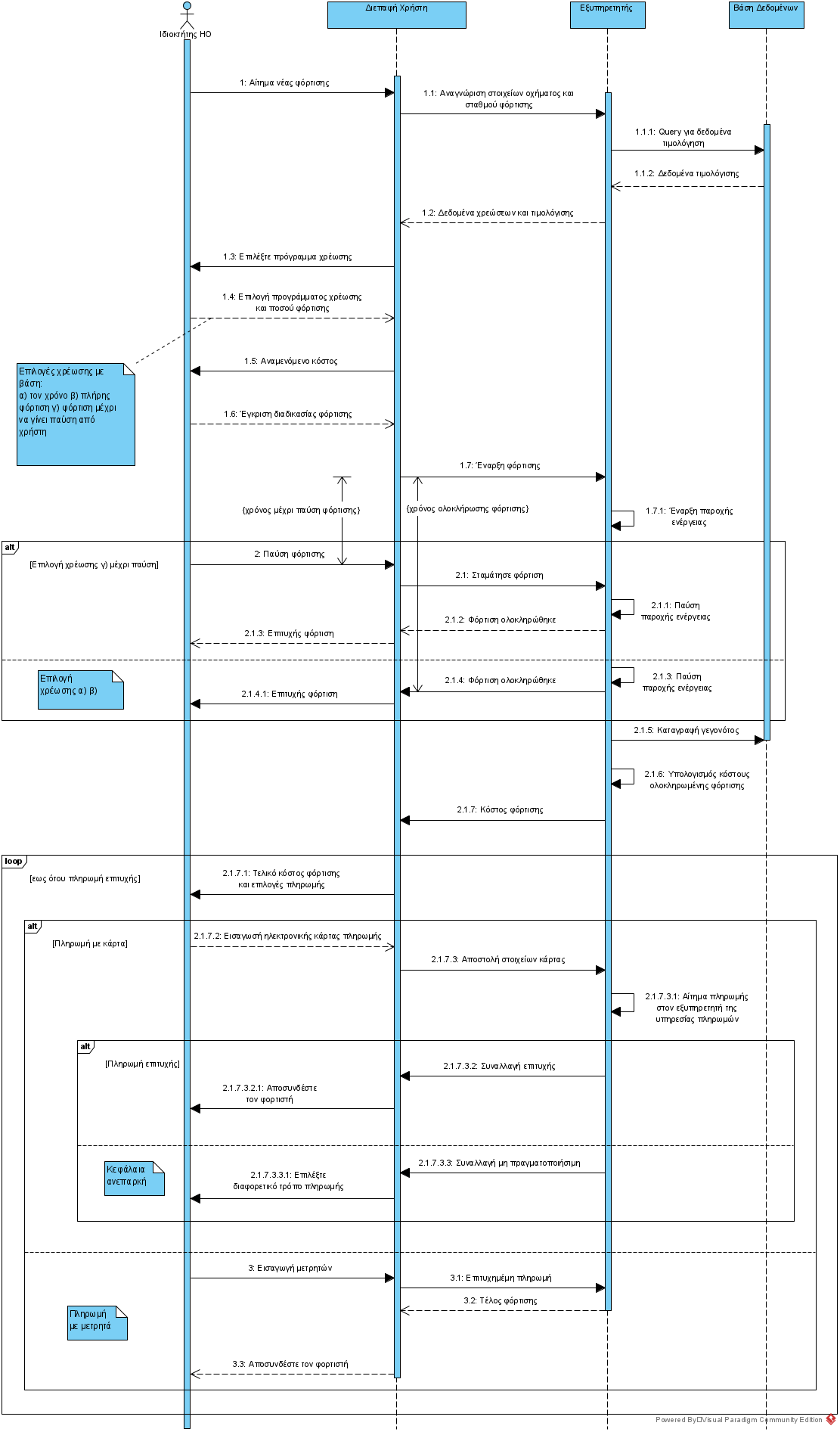
* Βήμα 1: Ο χρήστης αιτείται νέα φόρτιση.
* Βήμα 2: Ο χρήστης επιλέγει τον επιθυμητό χρόνο φόρτισης καθώς και αν θέλει ολική ή μερική φόρτιση.
* Βήμα 3: Η φόρτιση εκκινείται.
* Βήμα 4: Γίνεται η χρέωση του χρήστη εως ότου να ολοκληρωθεί η φόρτιση.
* Βήμα 5: Η φόρτιση ολοκληρώνεται.
* Βήμα 6: Εμφανίζεται το τελικό κόστος της φόρτισης.
* Βήμα 7: Ο χρήστης επιλέγει να πληρώσει με κάρτα.
* Βήμα 8: Η πληρωμή ολοκληρώνεται επιτυχώς.
* Βήμα 9: Η διαδικασία της φόρτισης ολοκληρώνεται επιτυχώς.

Εναλλακτικές ροές:

* Στο βήμα 4 θα μπορούσε η φόρτιση να μην ολοκληρωθεί λόγω πληρότητας, αλλά λόγω της χειροκίνητης παύσης της από το χρήστη. Στην περίπτωση αυτή τα υπόλοιπα βήματα συνεχίζουν κανονικά, απλά το ποσό είναι αυτό που έχει καταγραφεί ως εκείνη τη στιγμή.
* Στο βήμα 7 ο χρήστης θα μπορούσε να επιλέξει πληρωμή με μετρητά και, έτσι, η διαδικασία να πάει απευθείας στο βήμα 9.
* Στο βήμα 8 θα μπορούσε να υπάρξει πρόβλημα κατά την προσπάθεια πληρωμής με κάρτα, ο χρήστης να ενημερωθεί για την αποτυχημένη προσπάθεια και η διαδικασία να ανακατευθυνθεί στο βήμα 6.



*Εικόνα 7: UML Activity Diagram για το συγκεκριμένο Use Case*



*Εικόνα 8: UML Sequence Diagram για το συγκεκριμένο Use Case*

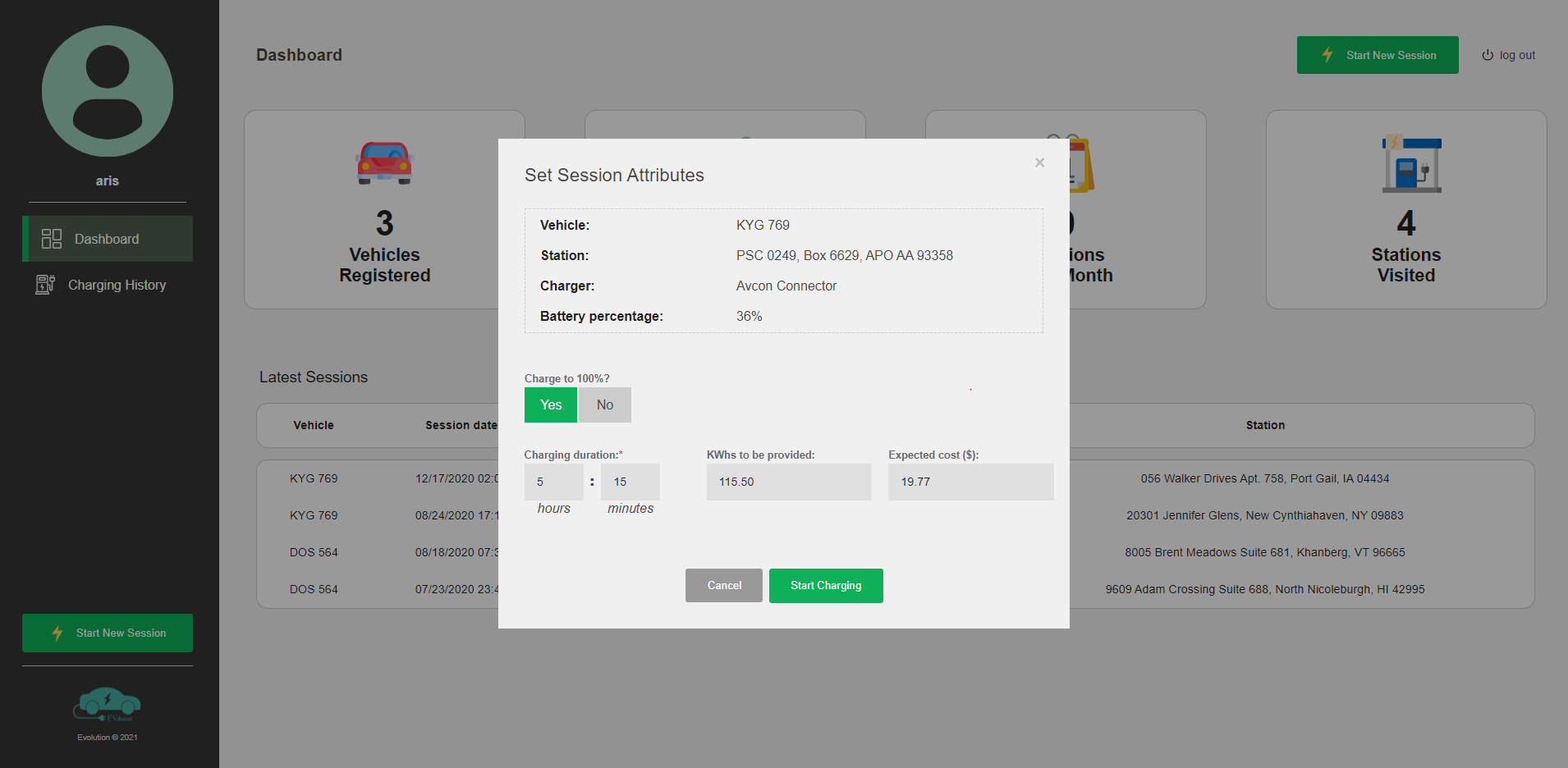
#### *2.1.1.7 Δεδομένα εξόδου*

Τα δεδομένα εξόδου της συγκεκριμένης περίπτωσης χρήσης είναι:

* Το συνολικό κόστος της φόρτισης
* Το ποσοστό φόρτισης του οχήματος την κάθε χρονική στιγμή
* Το ποσό ενέργειας που μεταφέρθηκε στο όχημα

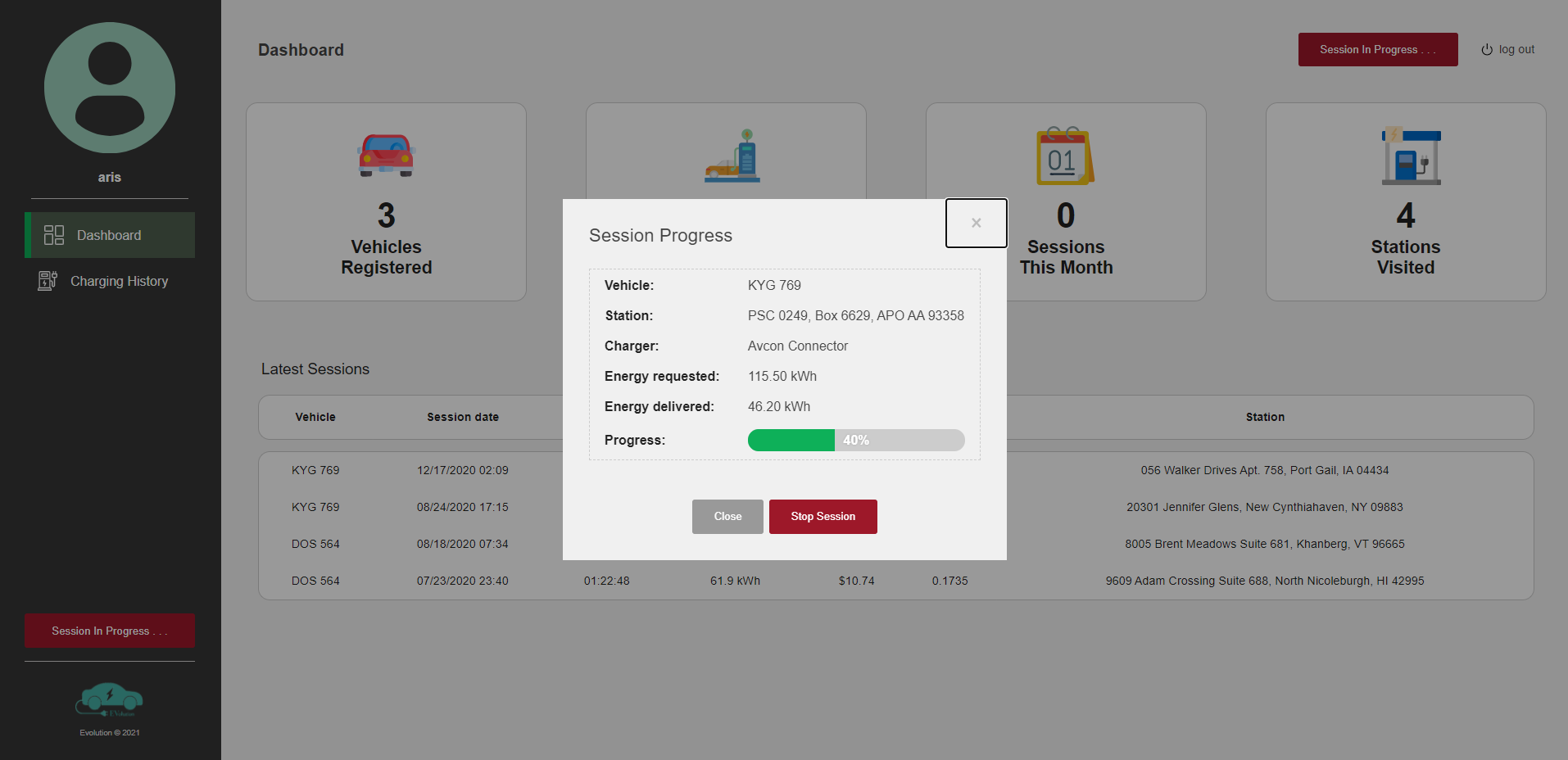
#### *2.1.1.8 Παρατηρήσεις*

Στη συνέχεια φαίνονται κάποια στιγμιότυπα οθόνης από τη frontend εφαρμογή που αφορούν το use case αυτό:

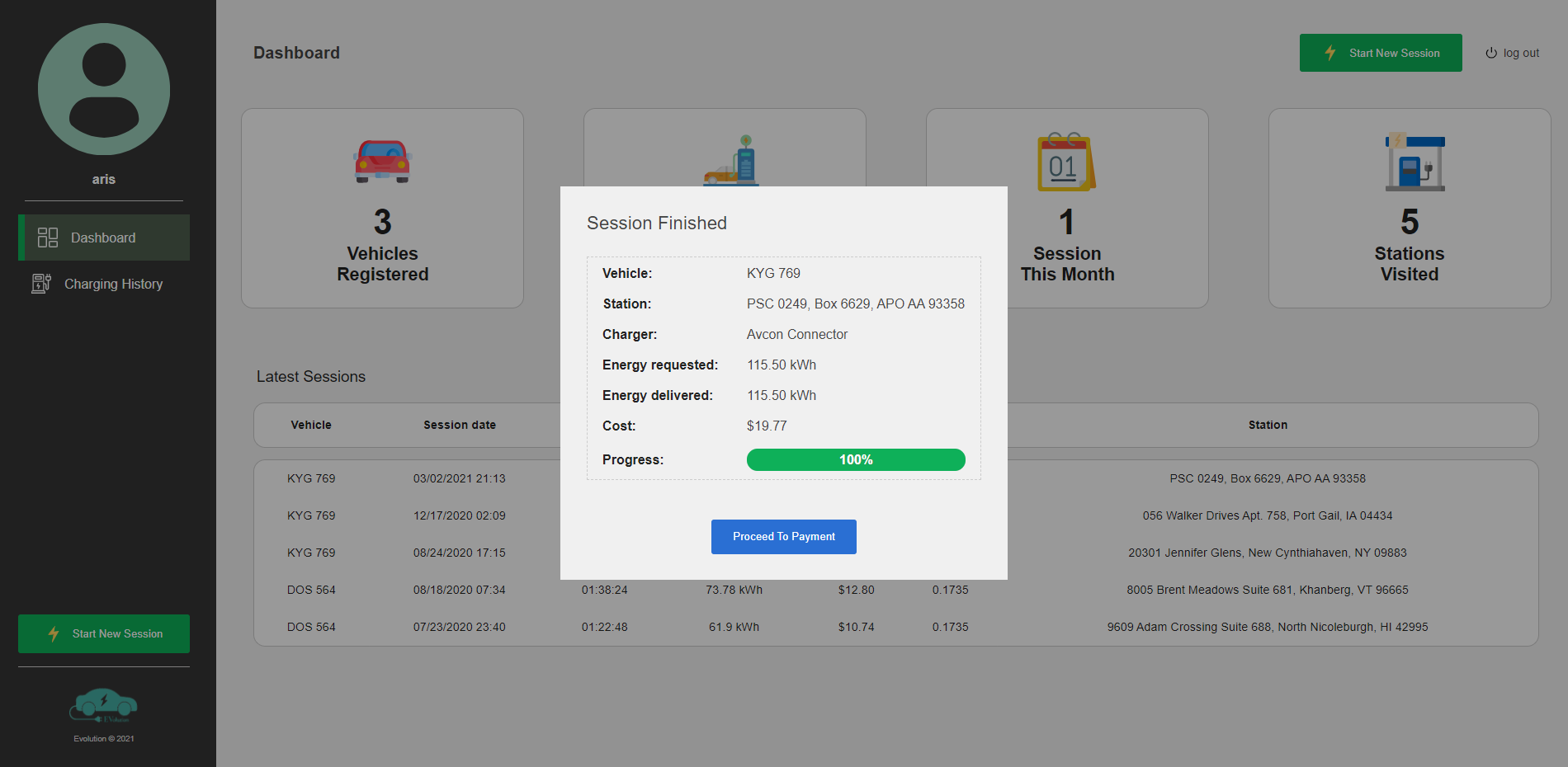


*Εικόνα 9: Εικόνα από τη φόρμα που αποστέλλεται από το χρήστη για την εκκίνηση μιας νέας φόρτισης*



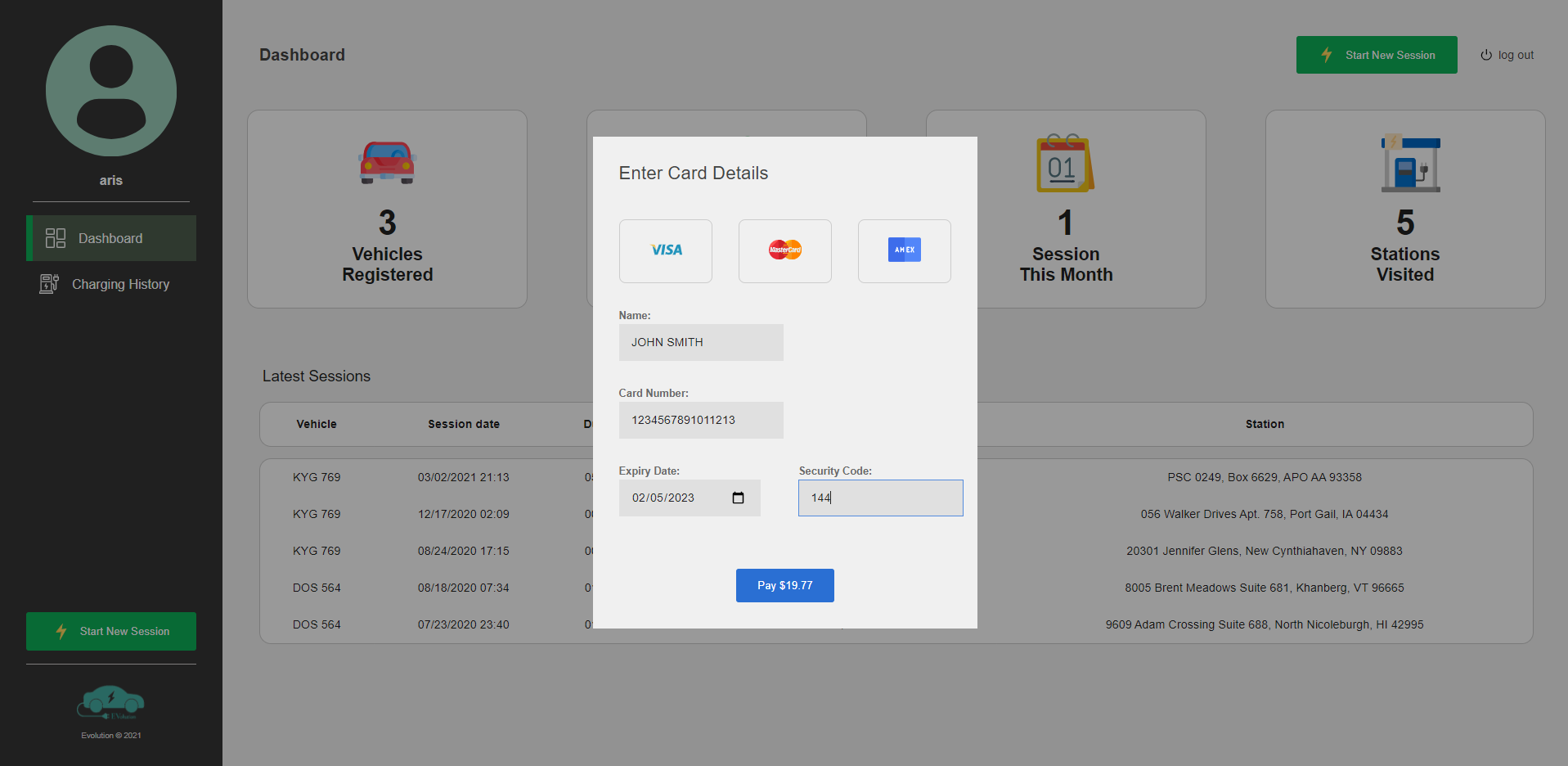
**

*Εικόνα 10: Εικόνα από τη φόρμα που εμφανίζεται στο χρήστη κατά τη διάρκεια της φόρτισης (διακρίνεται η επιλογή του χρήστη να διακόψει τη φόρτιση ανά πάσα στιγμή αν το επιθυμεί)*

**

*Εικόνα 11: Εικόνα από τη φόρμα που εμφανίζεται στο χρήστη κατά την ολοκλήρωση της φόρτισης*



**

*Εικόνα 12 Εικόνα από τη φόρμα που εμφανίζεται στο χρήστη για τη συμπλήρωση των στοιχείων της κάρτας του (στην περίπτωση επιλογής πληρωμής με κάρτα)*

### 2.1.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: Έκδοση συγκεντρωτικού λογαριασμού και ιστορικού φορτίσεων

#### *2.1.2.1 Χρήστες (ρόλοι) που εμπλέκονται*

Αυτή η χρήση του λογισμικού αφορά τον εκάστοτε χρήστη που αναγνωρίζεται ως “Ιδιοκτήτης Ηλεκτρικού/ων Οχήματος/ων”, ο οποίος συνδέεται στην εφαρμογή προκειμένου να δει τον συγκεντρωτικό του λογαριασμό και το ιστορικό φορτίσεων.

#### *2.1.2.2 Προϋποθέσεις εκτέλεσης*

Ο ιδιοκτήτης ηλεκτρικού οχήματος θα πρέπει να έχει έναν πιστοποιημένο λογαριασμό. Επίσης, θα πρέπει όλα τα δεδομένα που αφορούν τις φορτίσεις του/ων οχήματος/ων να έχουν καταγραφεί στη βάση μας.

#### *2.1.2.3 Περιβάλλον εκτέλεσης*

Η περίπτωση χρήσης αυτής εκτελείται στη διεπαφή του χρήστη. Ωστόσο, χρειάζεται να “χτυπηθεί” το DBMS με χρήση του Rest API.

#### *2.1.2.4 Δεδομένα εισόδου*

Ο χρήστης χρειάζεται να παρέχει ως είσοδο στο σύστημα τα ακόλουθα:

* Ημερομηνία έναρξης για την προβολή του ιστορικού
* Τελική ημερομηνία για την προβολή του ιστορικού

#### *2.1.2.5 Παράμετροι*



|  |  |  |
| --- | --- | --- |
| **Πεδίο** | **Τύπος Δεδομένων** | **Περιγραφή** |
| userId | Integer | Το id του χρήστη |
| station | json | Η καταχώρηση του εκάστοτε σταθμού |
| charger | json | Η καταχώρηση του εκάστοτε φορτιστή |
| sessionId | Integer | Το id της εκάστοτε φόρτισης |
| electricVehicle | json | Η καταχώρηση του εκάστοτε οχήματος |
| startTime | Date | Η ημερομηνία έναρξης της εκάστοτε φόρτισης |
| endTime | Date | Η ημερομηνία λήξης της εκάστοτε φόρτισης |
| energyDelivered | Float | Η ενέργεια που μεταφέρθηκε στην εκάστοτε φόρτιση |
| costPerKwh | Float | Το κόστος της κιλοβατώρας |
| totalCost | Float | Το συνολικό κόστος της εκάστοτε φόρτισης |
| points | Float | Το πλήθος των πόντων ανταμοιβής για την εκάστοτε φόρτιση |

#### *2.1.2.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά*

Θα περιγράψουμε με τη μορφή βημάτων την αλληλουχία ενεργειών τόσο όσων αφορά τη βασική, όσο και τις εναλλακτικές ροές. Στη συνέχεια, θα παρατεθούν τα διαγράμματα UML Activity και UML Sequence που αφορούν αυτή την περίπτωση χρήσης.

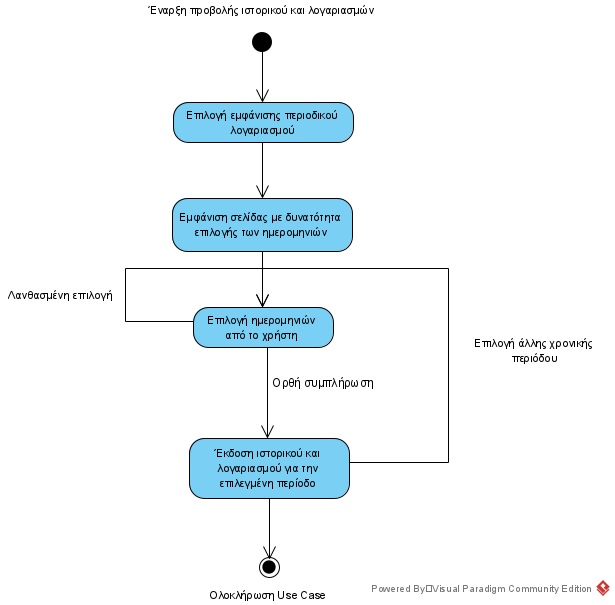
Βασική ροή:

* Βήμα 1: Ο χρήστης επιλέγει να έχει πρόσβαση στον περιοδικό λογαριασμό και το ιστορικό φορτίσεων του.
* Βήμα 2: Μέσω δύο εύχρηστων κουμπιών επιλέγει τις ημερομηνίες που θέλει να οριοθετούν το ιστορικό φορτίσεων που θα προβληθούν σε αυτόν.
* Βήμα 3: Οι ημερομηνίες που συμπλήρωσε είναι έγκυρες κι έτσι του εμφανίζονται τα δεδομένα των φορτίσεων που οριοθετούνται από αυτές σε συγκεντρωτική και αναλυτική μορφή.
* Βήμα 4: Ο χρήστης επιλέγει είτε να δει το ιστορικό και εξέρχεται από τη σελίδα παρουσίασης του ιστορικού.



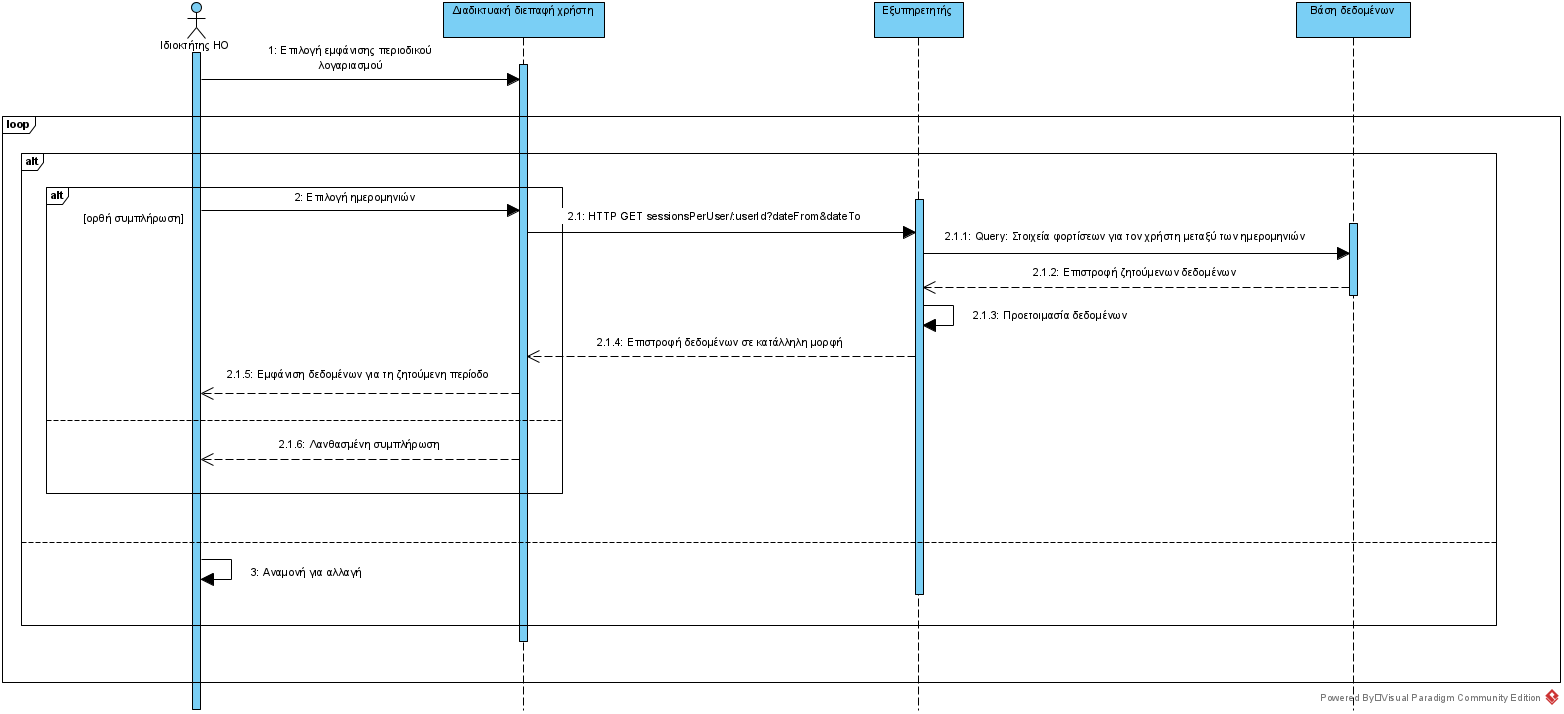
Πιθανές εναλλακτικές ροές:

* Στο Βήμα 3 ο χρήστης θα μπορούσε να είχε εισάγει άκυρες ημερομηνίες κι έτσι να ανακατευθυνόταν στο Βήμα 2.
* Στο Βήμα 4 ο χρήστης ενδεχομένως να επέλεγε να δει το ιστορικό και για κάποιες άλλες ημερομηνίες, τότε θα επέστρεφε στο Βήμα 3 για να αλλάξει την επιλογή του.



*Εικόνα 13: UML Activity Diagram για το συγκεκριμένο Use Case*





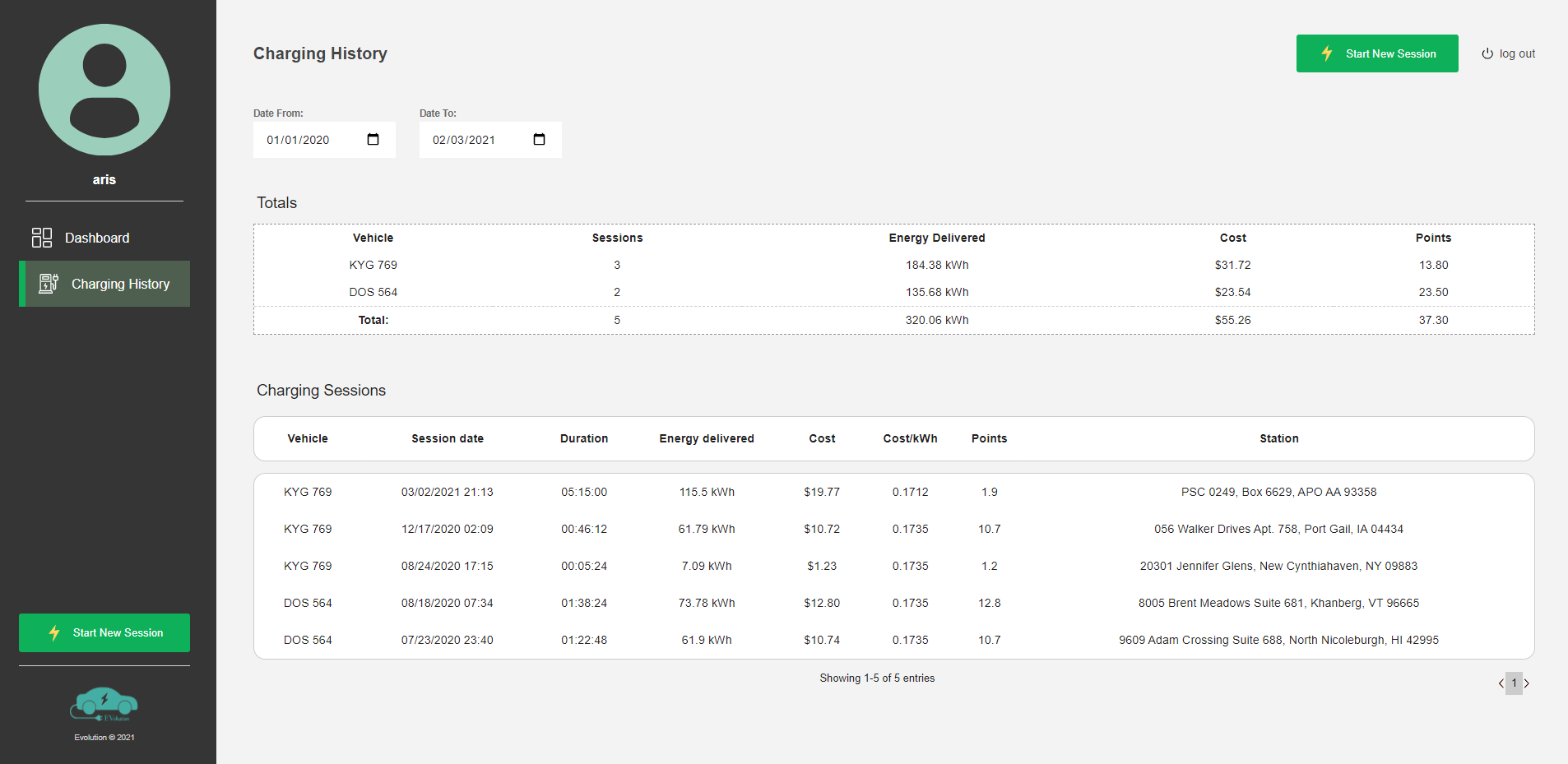
*Εικόνα 14: UML Sequence Diagram για το συγκεκριμένο Use Case*

#### *2.1.2.7 Δεδομένα εξόδου*

Ως έξοδο η συγκεκριμένη περίπτωση χρήσης εμφανίζει συγκεντρωτικά το λογαριασμό για την επιλεχθείσα περίοδο, μαζί με τους πόντους επιβράβευσης γι’ αυτή, αλλά και το σύνολο των φορτίσεων και της ενέργειας που μεταφέρθηκε. Ακόμη, εμφανίζει αναλυτικά τα δεδομένα των φορτίσεων αυτών.

#### *2.1.2.8 Παρατηρήσεις*

Στη συνέχεια παρατίθεται ένα στιγμιότυπο οθόνης από τη frontend εφαρμογή του συστήματος μας που δείχνει το πως μοιάζει η έξοδος του συγκεκριμένου Use Case για μια επιλεγμένη χρονική περίοδο.



*Εικόνα 15: Εικόνα από τη frontend εφαρμογή που δείχνει τη μορφή της εξόδου του συγκεκριμένου Use Case*



### 2.1.3 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 3: Εμφάνιση στατιστικών που αφορούν τις θέσεις φόρτισης/στάθμευσης και τους διάφορους τύπους φορτιστών

#### *2.1.3.1 Χρήστες (ρόλοι) που εμπλέκονται*

Αυτή η χρήση του λογισμικού αφορά τον εκάστοτε χρήστη που αναγνωρίζεται ως “Ιδιοκτήτης χώρου στάθμευσης/φόρτισης”, ο οποίος συνδέεται στην εφαρμογή προκειμένου να δει κάποια στατιστικά και συγκεντρωτικά στοιχεία που αφορούν τον/τους σταθμούς που ο ίδιος κατέχει.

#### *2.1.3.2 Προϋποθέσεις εκτέλεσης*

Για να μπορέσει να εκτελεστεί η περίπτωση χρήσης αυτή, θα πρέπει ο χρήστης να έναν πιστοποιημένο λογαριασμό που έχει συνδεθεί με τις θέσεις στάθμευσης/φόρτισης που είναι στην κατοχή του (καθώς και με τα μηχανήματα που βρίσκονται σε αυτές). Επίσης, ο χρήστης θα πρέπει να συνδεθεί στο σύστημα. Τέλος, απαραίτητη προϋπόθεση είναι όλα τα δεδομένα των φορτίσεων να έχουν καταγραφεί.

#### *2.1.3.3 Περιβάλλον εκτέλεσης*

Η περίπτωση χρήσης αυτής εκτελείται στη διεπαφή του χρήστη. Ωστόσο, χρειάζεται να “χτυπηθεί” το DBMS με χρήση του Rest API.

#### *2.1.3.4 Δεδομένα εισόδου*

Η περίπτωση χρήσης αυτή λαμβάνει από το χρήστη τα ακόλουθα δεδομένα εισόδου από το χρήστη:

* Χρονική περίοδος (χρόνος ή μήνας) για την παρουσίαση διαγράμματος με το πλήθος φορτίσεων
* Επιθυμητός τύπος διαγράμματος για τα στατιστικά που αφορούν στα διαφορετικά είδη φορτιστών

#### *2.1.3.5 Παράμετροι*



|  |  |  |
| --- | --- | --- |
| **Πεδίο** | **Τύπος Δεδομένων** | **Περιγραφή** |
| chargingPoint | json | Η καταχώρηση της εκάστοτε θέσης φόρτισης |
| costPerKwh | Float | Το κόστος της κιλοβατώρας |
| electricVehicle | json | Η καταχώρηση του εκάστοτε οχήματος |
| endTime | Date | Η χρονική στιγμή λήξης της εκάστοτε φόρτισης |
| energyDelivered | Float | Η ενέργεια που μεταφέρθηκε στην εκάστοτε φόρτιση |
| paymentType | String | Ο τύπος πληρωμής που επιλέχθηκε στην εκάστοτε φόρτιση |
| pointsAwarded | Float | Οι πόντοι επιβράβευσης για την εκάστοτε φόρτιση |
| startTime | Date | Η χρονική στιγμή έναρξης της εκάστοτε φόρτισης |
| station | json | Η καταχώρηση του εκάστοτε σταθμού |
| totalCost | Float | Το συνολικό κόστος της εκάστοτε φόρτισης |

#### *2.1.3.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά*

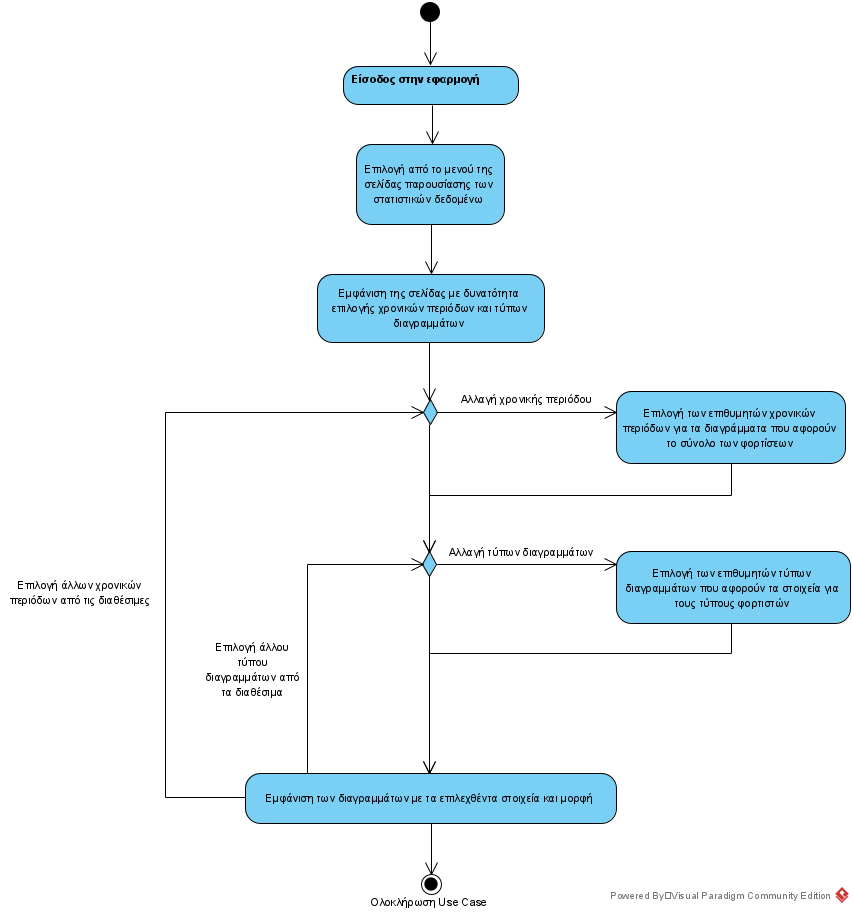
Θα περιγράψουμε με τη μορφή βημάτων την αλληλουχία ενεργειών τόσο όσων αφορά τη βασική, όσο και τις εναλλακτικές ροές. Στη συνέχεια, θα παρατεθούν τα διαγράμματα UML Activity και UML Sequence που αφορούν αυτή την περίπτωση χρήσης.

Βασική ροή:

* Βήμα 1: Ο χρήστης συνδέεται στην εφαρμογή.
* Βήμα 2: Επιλέγει την επιθυμητή σελίδα για να έχει πρόσβαση στα στατιστικά δεδομένα.
* Βήμα 3: Μέσω dropdown μπάρας επιλέγει για τα στοιχεία που αφορούν τα στατιστικά στοιχεία για τους διάφορους τύπους φορτιστών τη μορφή των διαγραμμάτων που επιθυμεί να βλέπει. Παράλληλα, για το σύνολο των φορτίσεων σε χρονικές περιόδους επιλέγει για ποια χρονική περίοδο επιθυμεί να βλέπει τα αντίστοιχα διαγράμματα.
* Βήμα 4: Τα δεδομένα εμφανίζονται σε αυτόν και ο ίδιος εξέρχεται από τη σελίδα προβολής των στατιστικών δεδομένων.

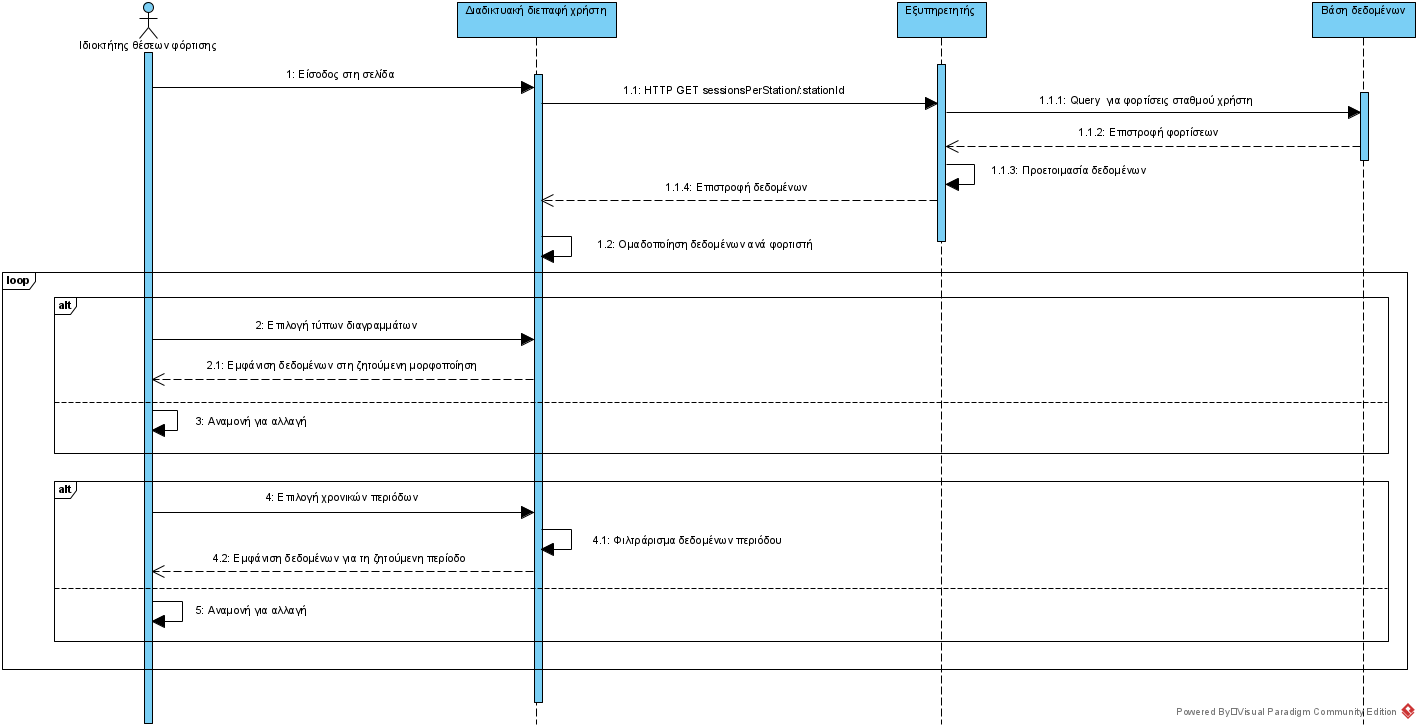
 Πιθανές εναλλακτικές ροές:

* Στο Βήμα 4 θα μπορούσε να επιλέξει κάποια άλλη μορφή για τα διαγράμματα που παρουσιάζονται σε αυτόν. Τότε θα επέστρεφε στο Βήμα 3 για να αλλάξει την επιλογή του.
* Στο Βήμα 4 ο χρήστης θα μπορούσε να αποφασίσει πως θέλει να αλλάξει τις χρονικές περιόδους που του παρουσιάζονται. Στην περίπτωση αυτή θα γύριζε στο Βήμα 3.

**

*Εικόνα 16: UML Activity Diagram για το συγκεκριμένο Use Case*



**

*Εικόνα 17: UML Sequence Diagram για το συγκεκριμένο Use Case*

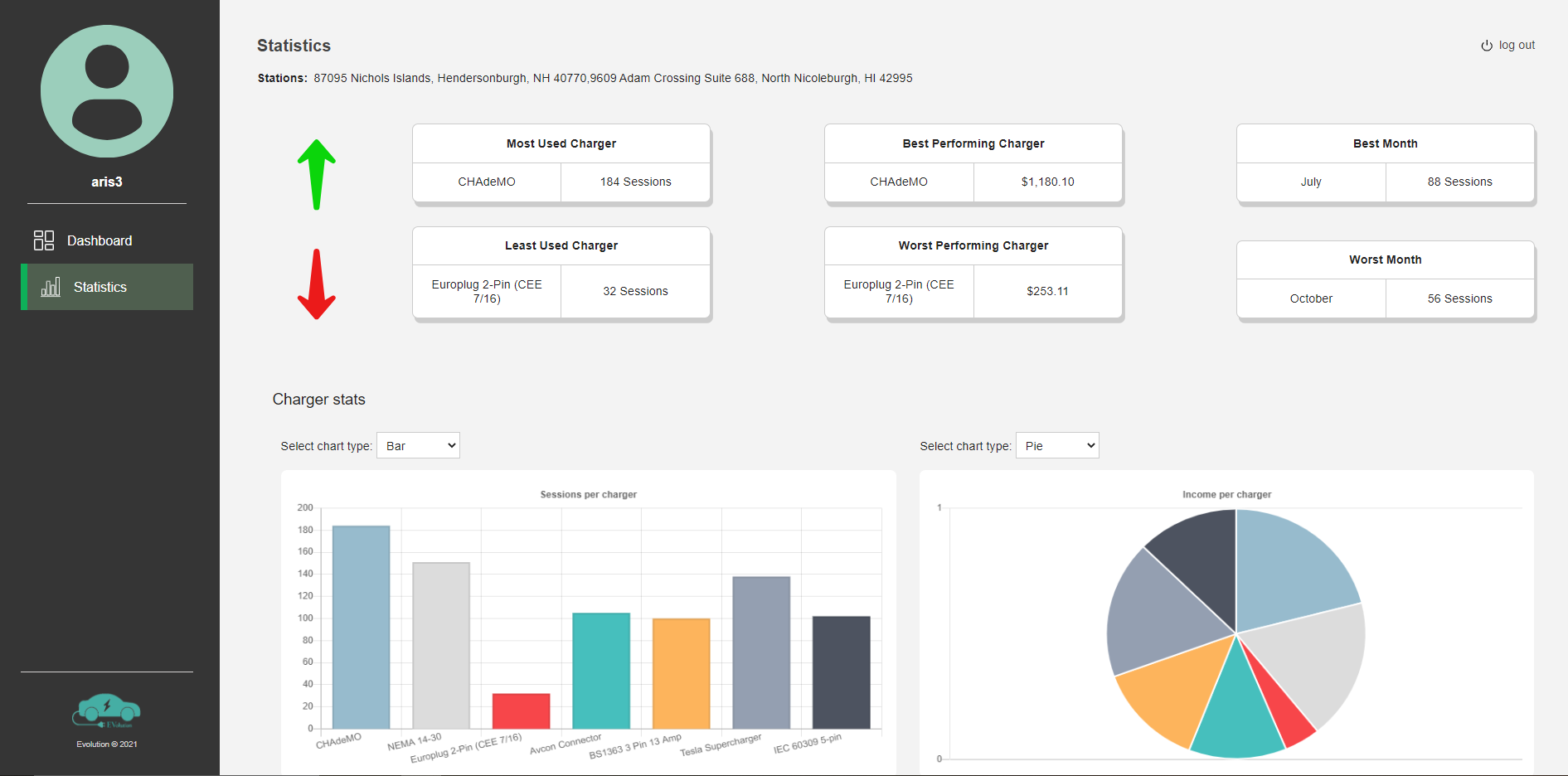
#### *2.1.3.7 Δεδομένα εξόδου*

Ως έξοδος θα δίνονται στο χρήστη τα δεδομένα που επιλέγει να δει. καθώς επίσης και τα διάφορα διαγράμματα της επιθυμητής μορφής και για τις επιθυμητές χρονικές περιόδους.

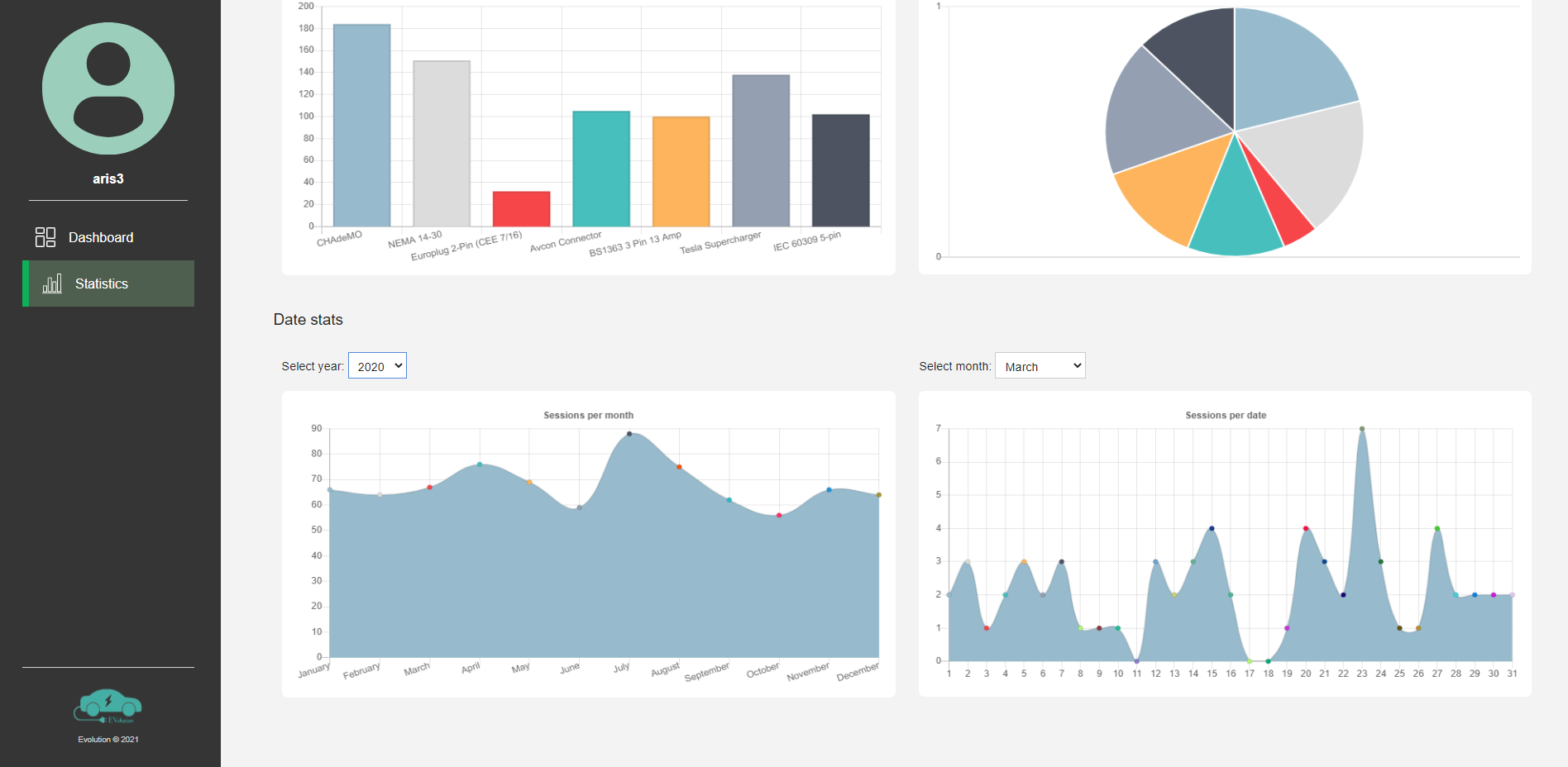
#### *2.1.3.8 Παρατηρήσεις*

Παραθέτουμε κάποια στιγμιότυπα οθόνης από την frontend εφαρμογή όπου φαίνεται η μορφή που έχουν τα διάφορα δεδομένα εξόδου που προβάλλονται στον χρήστη, καθώς και τα διαφορετικά διαγράμματα.





*Εικόνα 18: Εικόνα από τη frontend εφαρμογή όπου φαίνεται η μορφή που έχουν τα δεδομένα εξόδου και τα διαγράμματα που αφορούν τους διάφορους τύπους φορτιστών*

**

*Εικόνα 19: Εικόνα από τη frontend εφαρμογή όπου φαίνονται τα διαγράμματα που αφορούν το σύνολο των φορτίσεων σε διαφορετικές περιόδους*

## 2.2 Απαιτήσεις επιδόσεων

Αναμένουμε εως και τουλάχιστον 100 ταυτόχρονα συνδεδεμένους χρήστες στην εφαρμογή μας, στην αρχή, ενώ για το σύντομο μέλλον το μέγεθος αυτό ανέρχεται στους 500 και έχουμε φροντίσει αυτό το φόρτο να είναι διαχειρίσιμο. Παράλληλα, επιθυμούμε να έχουμε χρόνο απόκρισης στα αιτήματα της τάξης των ms, καθώς πρόκειται για μια εφαρμογή με ορισμένα της use case να αφορούν το χειρισμό γεγονότων. Επιπρόσθετα, τα τρέχοντα δεδομένα μας υπολογίζονται σε GB, ενώ στο σύντομο μέλλον αναμένεται η μέτρηση τους σε TB, επομένως οφείλουμε να μπορούμε να υποστηρίξουμε με τη βάση μας αυτό τ φορτίο. Τέλος, αναμένουμε μερικές εκατοντάδες ταυτόχρονα χτυπήματα στο REST API, τα οποία το ίδιο είναι σε θέση να καλύψει.

## 2.3 Απαιτήσεις οργάνωσης δεδομένων

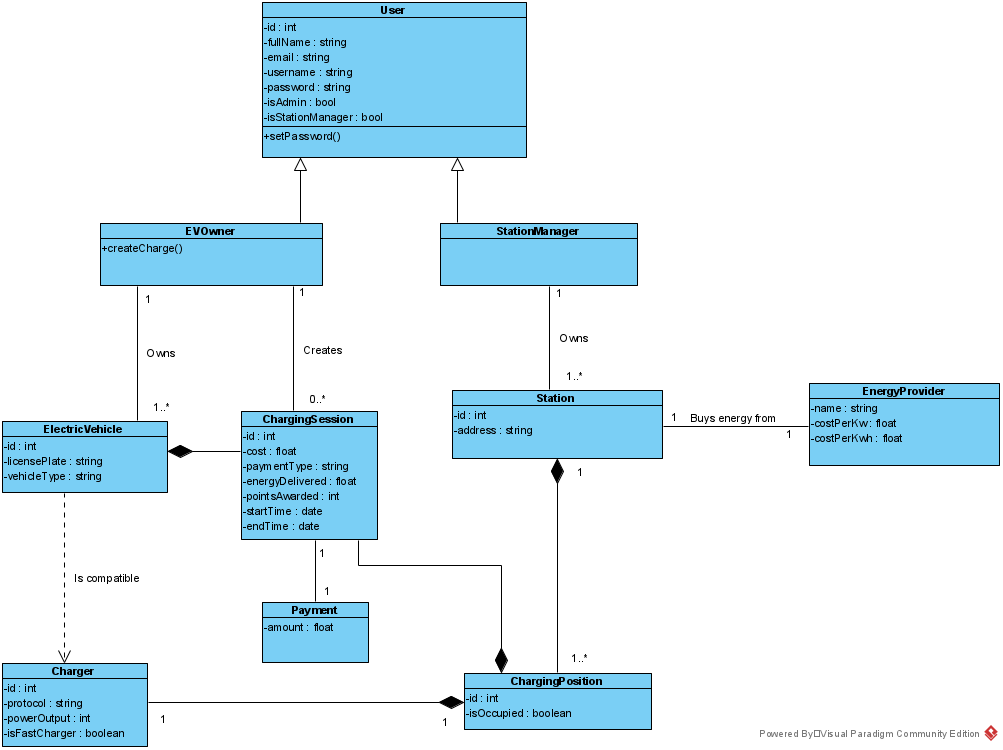
### 2.3.1 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα

Έχουμε επιλέξει το σύστημα μας να επιτρέπει πρόσβαση στα δεδομένα που διατηρεί μόνο σε πιστοποιημένους χρήστες και μόνο εφόσων τα δεδομένα αυτά τους αφορούν. Επομένως, κανένας εξωτερικός χρήστης ή λογισμικό δεν θα έχει πρόσβαση στα δεδομένα που διατηρούνται από το λογισμικό μας. Όλες οι ανταλλαγές δεδομένων θα περνούν από RESTful API. Επίσης, κάθε χρήστης θα πιστοποιείται με το email του και κάποιο password προκειμένου να εξασφαλίζεται πως μόνο ο ίδιος θα έχει πρόσβαση σε δεδομένα που είναι σχετικά με αυτόν (και με τον τρόπο αυτό τα αντιμετωπίζει και η βάση μας).



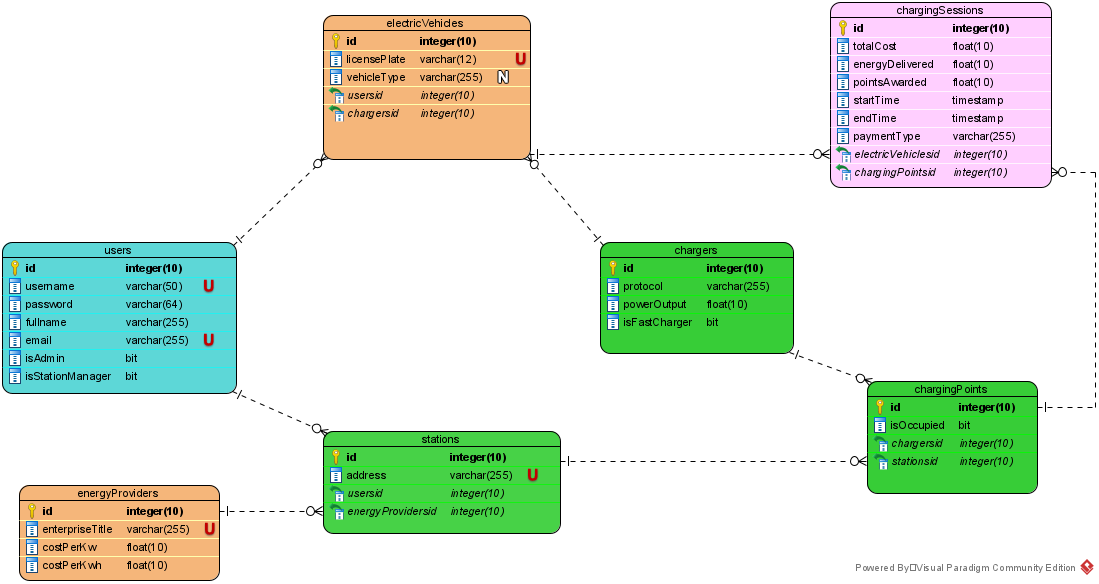
### 2.3.2 Μοντέλο δεδομένων

Στο σημείο αυτό παραθέτουμε τα διαγράμματα UML Class και ER προκειμένου να γίνει εμφανής τόσο η προγραμματιστική δομή του API μας, όσο και η το σχεσιακό σχήμα της βάσης δεδομένων μας.



*Εικόνα 20: UML Class Diagram*



**

*Εικόνα 21: ER Diagram*

## 2.4 Περιορισμοί σχεδίασης

Οι βασικότεροι περιορισμοί σχεδίασης οφείλονται στις διάφορες τεχνολογίες και σχεδιαστικές επιλογές που λάβαμε για την ανάπτυξη του λογισμικού μας. Συγκεκριμένα, η επιλογή για χρήση MariaDB για τη βάση δεδομένων μας, μας οδήγησε στους συνήθεις περιορισμούς που ακολουθούν μια σχεσιακή βάση δεδομένων. Επιπρόσθετα, η επιλογή το cli να επικοινωνεί με το REST API κι όχι απευθείας με τη βάση μας οδήγησε σε περιορισμούς ως το σχεδιασμό του backend, προσθέτοντας τα αντίστοιχα endpoints που αφορούν μόνο στο cli. Τέλος, η επιλογή χρήσης του runtime node.js μας οδήγησε στην ανάγκη δημιουργίας του package.json για τις διάφορες εξαρτήσεις.

## 2.5 Λοιπές απαιτήσεις

### 2.5.1 Απαιτήσεις διαθεσιμότητας λογισμικού

Η βασικότερη απαίτηση από το λογισμικό μας είναι η υψηλή του διαθεσιμότητα. Δεδομένου πως κάθε ιδιοκτήτης ηλεκτρικού οχήματος μπορεί να επιλέξει να φορτίσει το όχημα του οποιαδήποτε χρονική στιγμή, επιθυμούμε το σύστημα μας να είναι διαθέσιμο. Παράλληλα, επιθυμούμε συνέπεια των δεδομένων, τουλάχιστον όσων αφορά τα οικονομικά στοιχεία, καθώς θέλουμε τόσο ο ιδιοκτήτης του οχήματος, καθώς και ο εκάστοτε ιδιοκτήτης χώρου στάθμευσης/φόρτισης, να γνωρίζουν τα ίδια (και τα σωστά) οικονομικά δεδομένα.

### 2.5.2 Απαιτήσεις ασφάλειας

Στο σημείο αυτό παραθέτουμε τις απαιτήσεις ασφαλείας που καλύπτονται από το λογισμικό που αναπτύξαμε:

* Η ανταλλαγή δεδομένων μεταξύ χρήστη και εξυπηρετητή γίνεται μόνο με ασφαλή κρυπτογραφημένο τρόπο, σύμφωνα με το πρωτόκολλο HTTPS.
* Για κάθε λογαριασμό στη βάση μας αποθηκεύεται το password κρυπτογραφημένο. Για την ακριβεία, με χρήση ασφαλούς hash function εξάγεται το hashing value του εκάστοτε password, το οποίο και κρατείται για την ταυτοποίηση του χρήστη κατά την είσοδο του.
* Κενά password δεν θα γίνονται αποδεκτά.
* Κάθε λογαριασμός οφείλει να έχει διαφορετικό email που τον ταυτοποιεί, καθώς και μοναδικό username.
* Διατηρούνται backups των δεδομένων σε replica servers τα οποία ανανεώνονται το αργότερο κάθε βράδυ. Σκοπός μας είναι η όσο το δυνατόν μεγαλύτερη διατήρηση της συνέπειας των δεδομένων, όπως αναφέρθηκε ρητά προηγουμένως.

### 2.5.3 Απαιτήσεις συντήρησης

Όσων αφορά τις απαιτήσεις συντήρησης, αυτές συνοψίζονται στη συνέχεια:

* Περιοδική παροχή αναβαθμίσεων τις εφαρμογής, με βελτιώσεις πιθανών σφαλμάτων που γίνονται αντιληπτά από τους χρήστες ή/και προσθήκες στις λειτουργίες που επιτελούνται.
* Περιοδικός έλεγχος εγγεγραμμένων χρηστών και δεδομένων για έλεγχο εγκυρότητας αυτών, καθώς επίσης και για διαγραφή δεδομένων χρηστών που είναι ανενεργοί για ένα χρονικό διάστημα που ξεπερνάει κάποιο προκαθορισμένο όριο.
* Ανανέωση πιστοποιητικών.