Έγγραφο απαιτήσεων λογισμικού(SRS)

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

[EVolution]

# Εισαγωγή

## 1.1 Εισαγωγή: σκοπός του λογισμικού

Οριοθέτηση του σκοπού του συστήματος

Το σύστημα μας αποτελείται από υπηρεσίες διαχείρισης της φόρτισης ηλεκτρικών οχημάτων, η οποία πραγματοποιείται σε δημόσιους αλλά και ιδιωτικούς χώρους. Στόχος του είναι η παροχή πληροφοριών στους ιδιοκτήτες των οχημάτων αυτών αναφορικά με τους σταθμούς φόρτισης που βρίσκονται κοντά τους, η δυνατότητα σύγκρισης τιμών και προγραμματισμού μεγάλων διαδρομών μέσω ενός διαδραστικού χάρτη καθώς και η παρουσίαση στατιστικών για το όχημά τους. Ένας επιπλέον στόχος είναι η ανώνυμη συλλογή στατιστικών από κάθε όχημα και σταθμό φόρτισης, η αξιόπιστη αποθήκευση τους και η παρουσίαση αναλυτικών στατιστικών ή/και προβλέψεων με προσιτό τρόπο σε εταιρείες παραγωγής ενέργειας, κατασκευής οχημάτων αλλά και στους σταθμούς φόρτισης. Τέτοια στατιστικά είναι απαραίτητα στις εταιρείες κατασκευής για τον έλεγχο και την βελτίωση των οχημάτων τους, στις εταιρείες παραγωγής ενέργειας για τη ρύθμιση των τιμών ρεύματος με βάση τον ανταγωνισμό αλλά και στους σταθμούς προκειμένου να βελτιώσουν τη συνολική ποιότητα υπηρεσιών προς τους καταναλωτές.

## 1.2 Διεπαφές (interfaces)

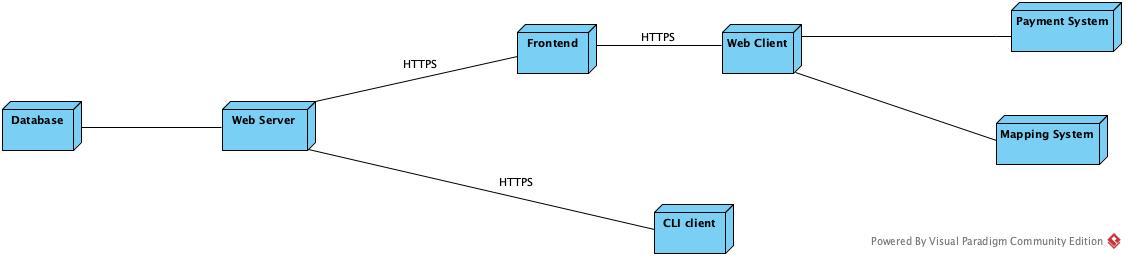
### 1.2.1 Διεπαφές με εξωτερικά συστήματα

Προδιαγραφή διεπαφών με εξωτερικά συστήματα και λογισμικό, με αναφορά σε πρότυπα ανταλλαγής δεδομένων και κλήσης υπηρεσιών. Χρήση διαγραμμάτων deployment UML.

Τα εξωτερικά συστήματα με τα οποία θα έρχεται σε επαφή η εφαρμογή περιλαμβάνουν:

* Τα API των διαφόρων τραπεζών, μέσω των οποίων θα πραγματοποιείται η πληρωμή για τη φόρτιση όταν αυτή πραγματοποιείται σε δημόσιους σταθμούς
* To API της υπηρεσίας που μας παρέχει τον χάρτη
* Αμφίδρομη επικοινωνία του Server που διατηρεί το UI και του Server που διατηρεί το API (υπάρχει περίπτωση να είναι ο ίδιος) με τον browser του χρήστη, χρησιμοποιώντας RESTful web services και https requests μέσω self signed certificate.
* Αμφίδρομη επικοινωνία του Server που περιέχει το API με το user interface, με χρήση RESTful web services. Μεταξύ των δύο άκρων αποστέλλονται πακέτα JSON με πρωτόκολλο HTTPS
* Επικοινωνία του API με άλλου είδους προγράμματα συμβατά με RESTful web services και HTTPS requests.
* Συνολικά η επικοινωνία μεταξύ των διαφορετικών ειδών API(του χρήστη, των σταθμών, των εταιρειών κλπ) επιτυγχάνεται μέσω RESTful web services.

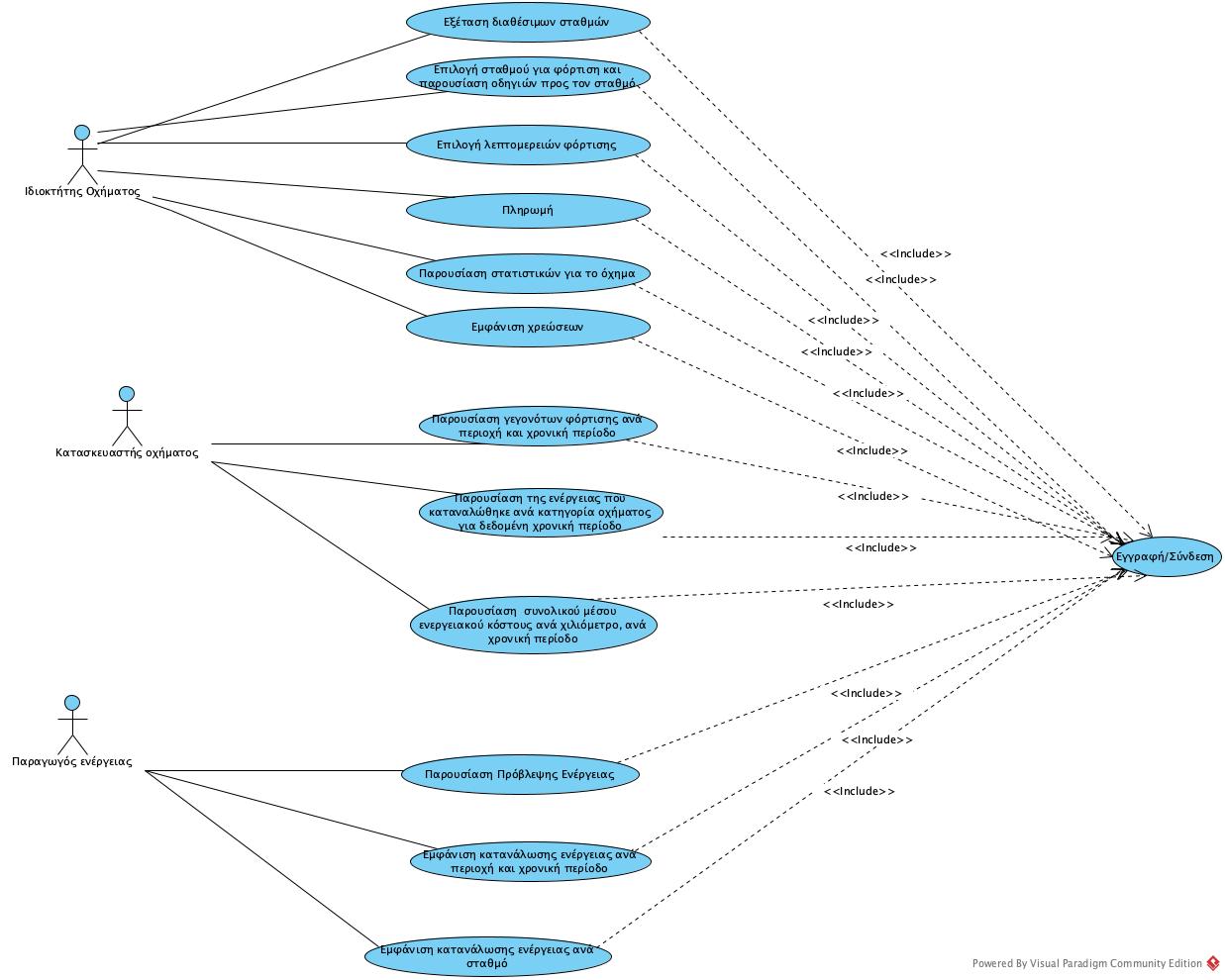
**Deployment diagram:**

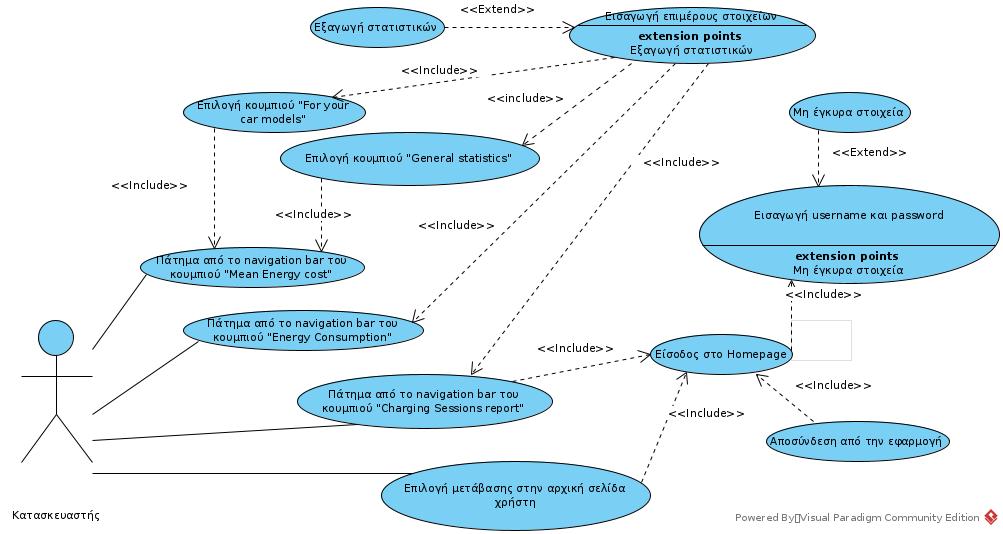
****

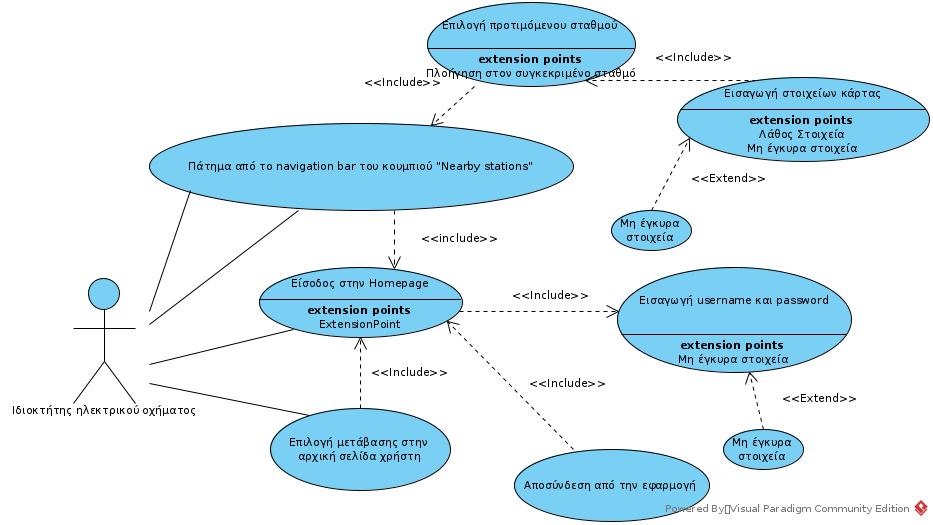
### 1.2.2 Διεπαφές με το χρήστη

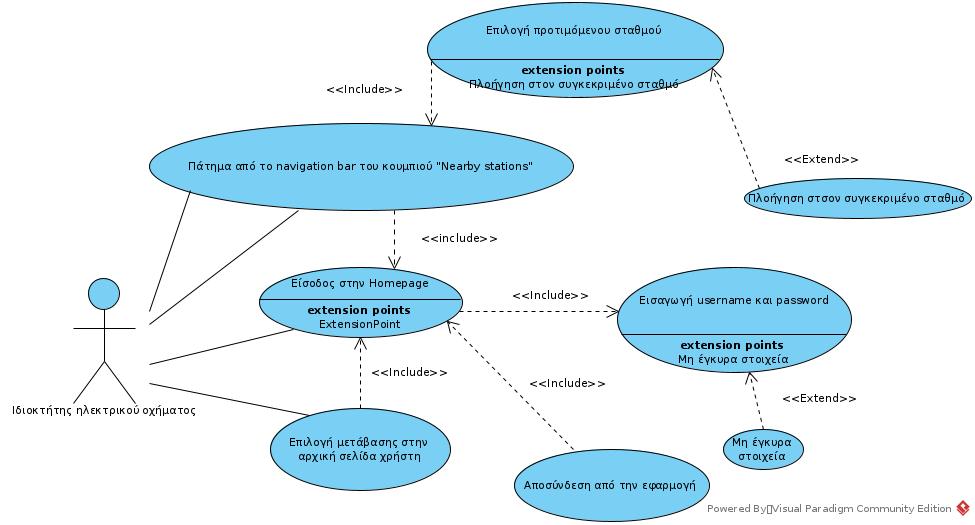
Προδιαγραφή διεπαφών με το χρήστη. Μοντέλο UseCase (UML).

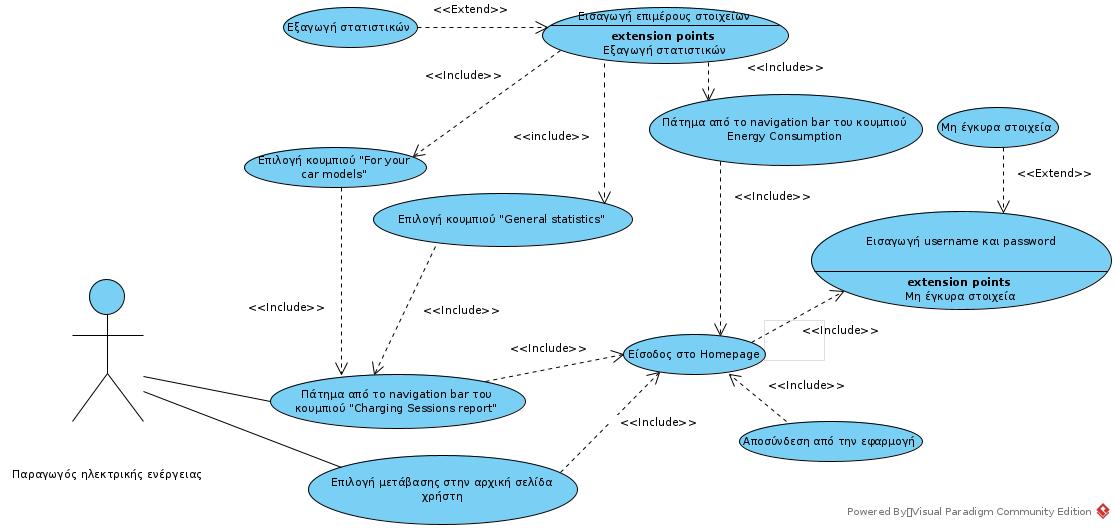
* User Interaction



* Statistics for Vehicle Manufactures Use Case Diagram
* Payment Use Case Diagram



* Willing to charge Diagram
* Statistics for Energy Providers Diagram



# Αναφορές - πηγές πληροφοριών

Αναφορά σε πηγές πληροφοριών στο μέτρο της αναγκαιότητας για την κατανόηση του συστήματος

ΝΑ ΜΗΝ ΣΥΜΠΛΗΡΩΘΕΙ ΑΝ ΔΕΝ ΑΠΑΙΤΕΙΤΑΙ

-

# Προδιαγραφές απαιτήσεων λογισμικού

## 3.1 Περιπτώσεις χρήσης

 Λεπτομερής προδιαγραφή των λειτουργιών του λογισμικού σε επίπεδο περιπτώσεων χρήσης. Ο αριθμός των περιπτώσεων χρήσης ανάλογα με τον αριθμό των μελών της ομάδας σύμφωνα με την εκφώνηση.

### 3.1.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: **Φόρτιση**

#### 3.1.1.1 Χρήστες (ρόλοι) που εμπλέκονται

Αναφορά στους ρόλους που αφορά η περίπτωση χρήσης

Εμπλέκονται ο ιδιοκτήτης ηλεκτρικού οχήματος, οι χώροι στάθμευσης(parking), οι δήμοι και λοιποί παροχείς σταθμών φόρτισης εκτός οργανωμένων χώρων στάθμευσης και έμμεσα οι λειτουργοί συστήματος μεταφοράς και διανομής ηλεκτρικής ενέργειας (ΑΔΜΗΕ, ΔΕΔΔΗΕ)*.*

3.1.1.2 Προϋποθέσεις εκτέλεσης

Καταγραφή των συνθηκών που πρέπει να ισχύουν ώστε να μπορεί να εκτελεστεί η περίπτωση χρήσης

Θα πρέπει αρχικά να υπάρχει ένα ηλεκτρικό όχημα το οποίο να χρειάζεται φόρτιση, ο ιδιοκτήτης του οποίου να έχει προσωπικό λογαριασμό στην εφαρμογή μας, και να υπάρχει διαθέσιμη θέση σε κάποιον ενεργό σταθμός φόρτισης (οργανωμένο ή μη).

#### 3.1.1.3 Περιβάλλον εκτέλεσης

Αναφορά στο περιβάλλον στο οποίο εκτελείται η περίπτωση χρήσης. Πχ "διαδικτυακή διεπαφή χρήστη", "DBMS" κλπ

Το περιβάλλον στο οποίο εκτελείται η περίπτωση χρήσης είναι η διαδικτυακή διεπαφή χρήστη καθώς και η επικοινωνία του API με την βάση για καταχώρηση των δεδομένων που λαμβάνονται κατά την φόρτιση.

#### 3.1.1.4 Δεδομένα εισόδου

Καταγραφή δεδομένων εισόδου και εξόδου και συνθηκών εγκυρότητας αυτών.

Στα δεδομένα εισόδου συγκαταλέγονται:

* Τα προσωπικά στοιχεία του χρήστη (username, password) για την εισαγωγή στον προσωπικό λογαριασμό του χρήστη (sign in),
* Τα στοιχεία του σταθμού φόρτισης
* Τα χρήματα και οι πόντοι που θέλει ο χρήστης να δαπανήσει στην συγκεκριμένη φόρτιση

#### 3.1.1.5Παράμετροι

Καταγραφή παραμέτρων και συνθηκών εγκυρότητας αυτών, εφόσον υπάρχουν παράμετροι.

Θα πρέπει ο χρήστης να έχει δημιουργήσει ένα έγκυρο προφίλ (το οποίο θα κατοχυρώνει πως είναι κάτοχος ηλεκτρικού οχήματος).

#### 3.1.1.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Περιγραφή με κείμενο (Βήμα 1, Βήμα 2 κλπ) και διαγράμματαUML αλληλουχίας (Sequence) και δραστηριοτήτων (Activity). Περιλαμβάνεται η συμπεριφορά σε απρόβλεπτες καταστάσεις και σφάλματα (εναλλακτικές ροές).

**Βήμα 1:**Είσοδος στην αρχική σελίδα ως επισκέπτης

**Βήμα 2:**Επιλογή κουμπιού Sign in 

**Βήμα 3:**Εισαγωγή username, password

**Βήμα 4:** (Αρχική σελίδα χρήστη)  από το navigation bar, στην καρτέλα services, του κουμπιού “Charge now”

**Βήμα 5:** Εισαγωγή στοιχείων οχήματος, επιλογή προγράμματος χρέωσης από τα διατιθέμενα και εισαγωγή στοιχείων σταθμού φόρτισης (και προαιρετικά εισαγωγή ενδεχόμενου επιθυμητού ποσού που ο ιδιοκτήτης επιθυμεί να δαπανήσει για την εν λόγω φόρτιση )

**Βήμα 6:**Επιλογή κουμπιού “Start charging”

**Βήμα 7:**Εμφάνιση εκτιμώμενου χρόνου φόρτισης

**Βήμα 8:** Εμφάνιση κατάλληλου μηνύματος ολοκλήρωσης φόρτισης

**Βήμα 9:**  Υπολογισμός κόστους φόρτισης και εμφάνιση του και παράλληλα καταγραφή και εμφάνιση γεγονότων και δεδομένων φόρτισης

**Βήμα 10:**Αν ο χρήστης έχει επιλέξει να πληρώσει στην εφαρμογή με κάρτα, πάτημα κουμπιού continue to payment και συνέχεια στο βήμα 12. Επίσης υπάρχει η επιλογή για προβολή δεδομένων / γεγονότων φόρτισης σε διάγραμμα 

**Βήμα 11:** Επιλογή απεικόνισης στατιστικών που αφορούν τα χιλιόμετρα μεταξύ φορτίσεων ή στατιστικών που αφορούν τα χρήματα που ξοδεύει ο χρήστης ανά μήνα σε φορτίσεις. Συνέχεια στο βήμα 17.

**Βήμα 12:**(Σελίδα με το χρωστούμενο υπόλοιπο όπου υπάρχει μπάρα για πληκτρολόγηση του ποσού που θα πληρωθεί και των πόντων που θα χρησιμοποιήσει). Συμπληρώνεται το ποσό και οι πόντοι που ο χρήστης επιθυμεί να αποπληρώσει.

**Βήμα 13:**Πάτημα κουμπιού “Next ”

**Βήμα 14:**(Ασφαλής σελίδα τράπεζας για την εισαγωγή της κάρτας) Εισαγωγή ζητούμενων στοιχείων κάρτας

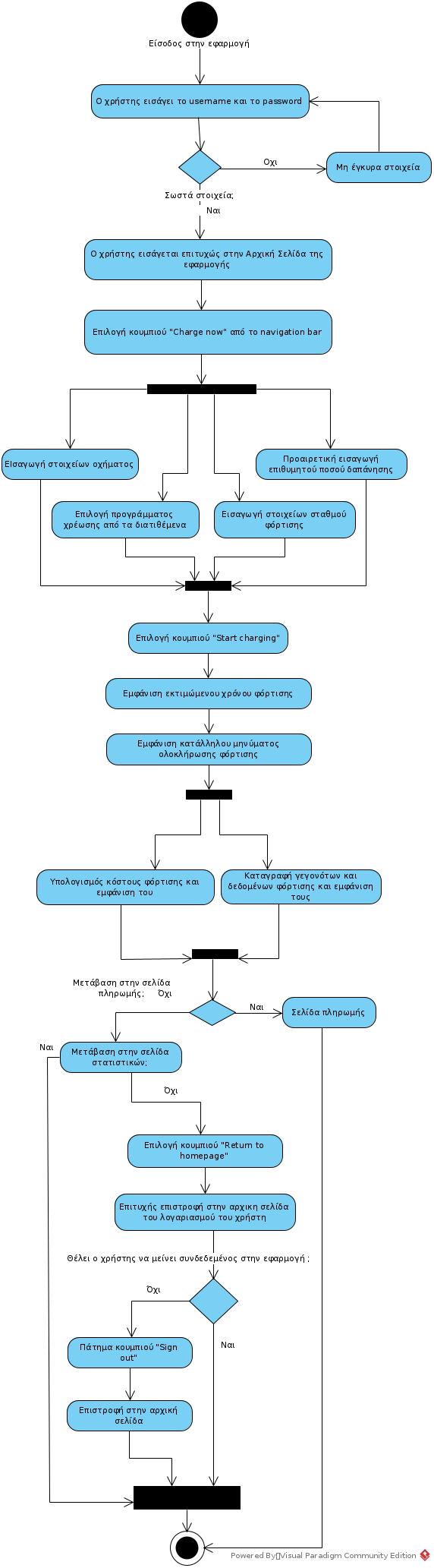
**Βήμα 15:**Πάτημα κουμπιού “Start payment”

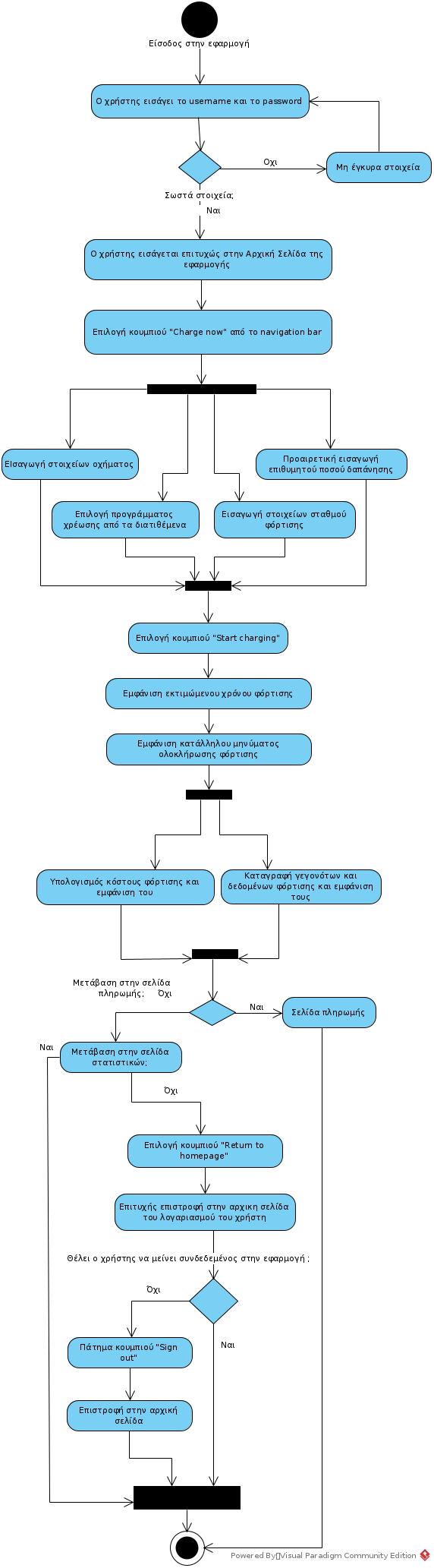
**Βήμα 16:**Εμφάνιση κατάλληλου μηνύματος ανάλογα με την εκτέλεση της πληρωμής (αν η πληρωμή πραγματοποιήθηκε εμφάνιση μηνύματος “Successful payment”, πόντων που έχουν κερδηθεί, συνολικών πόντων και οφειλόμενου ποσού μετά την πληρωμή , σε περίπτωση λάθους, εμφάνιση “Unsuccessful payment” και ανακατεύθυνση στην αρχική σελίδα)

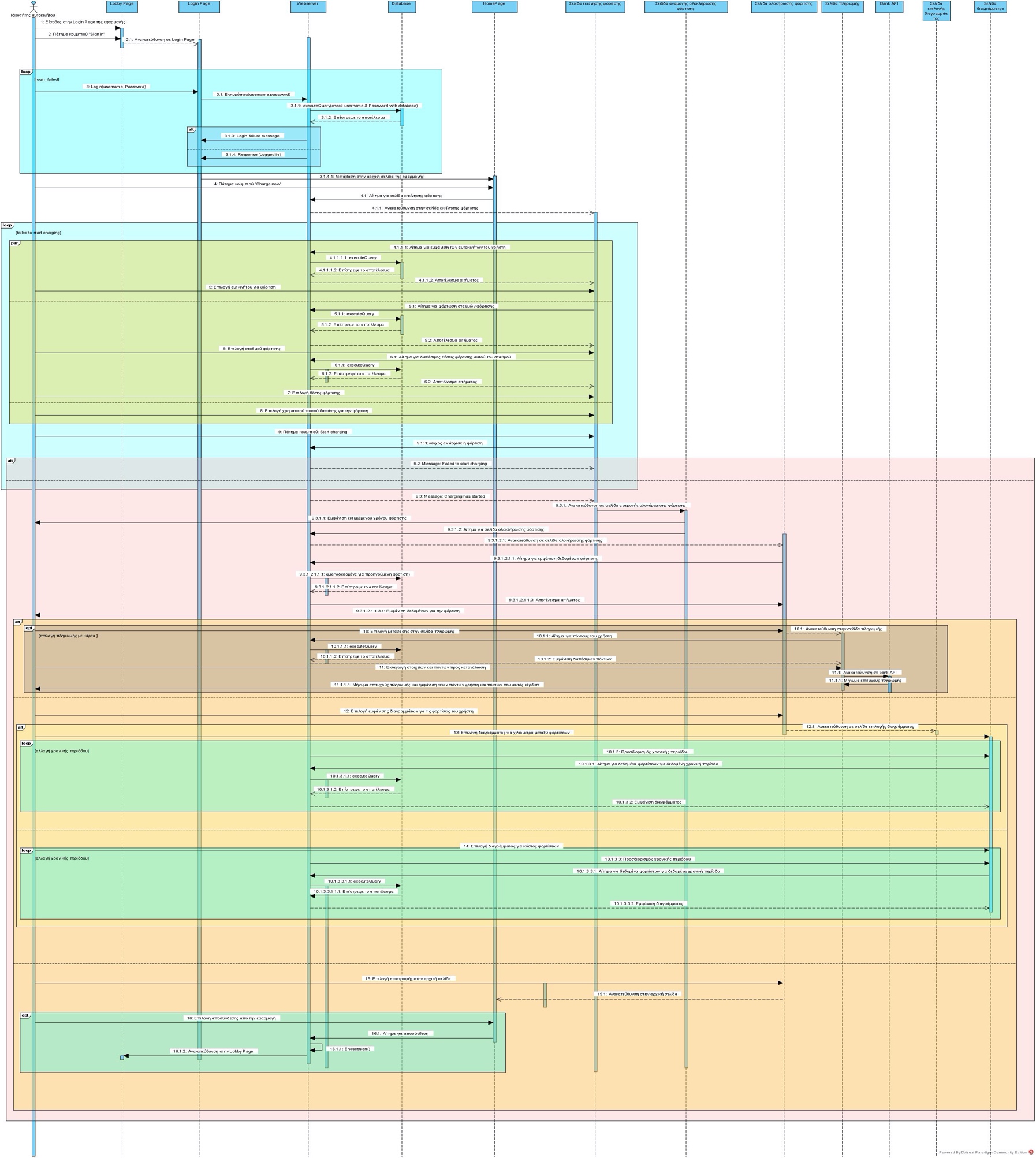
**Βήμα 17:**Αποσύνδεση από την εφαρμογή, αν ο χρήστης το επιθυμεί

**Διαγράμματα UML:**

* Charging activity diagram:

**

**

* Charging sequence diagram:

#### 3.1.1.7 Δεδομένα εξόδου

Διαγράμματα UMLαλληλουχίας για την παραγωγή δεδομένων εξόδου. Ως δεδομένα εξόδου νοούνται όλα τα δεδομένα του συστήματος τα οποία δημιουργούνται ή μεταβάλλονται κατά την εκτέλεση (αν υπάρχουν τέτοια)

Στα δεδομένα εξόδου συγκαταλέγονται τα εξής:

* Εμφάνιση μηνύματος ολοκλήρωσης φόρτισης.
* Καταγραφή γεγονότων/δεδομένων φόρτισης
* Απεικόνιση τους σε διάγραμμα
* Συλλογή δεδομένων οχήματος
* Υπολογισμός κόστους φόρτισης όταν αυτή ολοκληρώνεται
* Σε περίπτωση επιλογής πληρωμής στην εφαρμογή, υπολογισμός πόντων που κερδήθηκαν από την συγκεκριμένη φόρτιση και τελικών συνολικών πόντων.

#### 3.1.1.8 Παρατηρήσεις

Ο,τι δεν εντάσσεται στα προηγούμενα, εφόσον υπάρχει

-

### 3.1.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: **Χρήση πληρωμής**

#### 3.1.2.1 Χρήστες (ρόλοι) που εμπλέκονται

Αναφορά στους ρόλους που αφορά η περίπτωση χρήσης

Στην πληρωμή εμπλέκονται ο ιδιοκτήτης του ηλεκτρικού οχήματος που έχει την οφειλή και η αντίστοιχη τράπεζα που επιλέγει να χρησιμοποιήσει ,μέσω της οποίας θα εκτελέσει την πληρωμή.

#### 3.1.2.2 Προϋποθέσεις εκτέλεσης

Καταγραφή των συνθηκών που πρέπει να ισχύουν ώστε να μπορεί να εκτελεστεί η περίπτωση χρήσης

Αρχικά ο ιδιοκτήτης του αυτοκινήτου θα πρέπει να έχει συνδεθεί στον προσωπικό του λογαριασμό και να έχει κάνει μία ή περισσότερες φορτίσεις στο ηλεκτρικό του όχημα ώστε να έχει κάποιο ανεξόφλητο υπόλοιπο. Στην συνέχεια για να γίνει η ηλεκτρονική πληρωμή θα πρέπει ο χρήστης να διαθέτει λογαριασμό σε κάποια τράπεζα που να εμπεριέχει το ποσό που θέλει να πληρώσει καθώς και αντίστοιχη κάρτα συμβατή με τα είδη καρτών που δέχεται η εφαρμογή (MasterCard, American Express, Visa….).

#### 3.1.2.3 Περιβάλλον εκτέλεσης

Αναφορά στο περιβάλλον στο οποίο εκτελείται η περίπτωση χρήσης. Πχ "διαδικτυακή διεπαφή χρήστη", "DBMS" κλπ

Το περιβάλλον το οποίο χρησιμοποιείται είναι αρχικά η διαδικτυακή διεπαφή χρήστη (UI), όπου ο χρήστης θα βλέπει το ποσό που χρωστάει και θα επιλέγει το ποσό που θέλει να πληρώσει. Επίσης θα χρησιμοποιείται η βάση δεδομένων μέσω του API για την ανάκτηση προσωπικών δεδομένων του χρήστη. Τέλος, θα χρησιμοποιηθεί και η βάση δεδομένων της εκάστοτε τράπεζας (μέσω του αντίστοιχου API), για ταυτοποίηση των στοιχείων της κάρτας του και μεταφορά των χρημάτων, στην ασφαλή σελίδα της τράπεζας όπου θα έχει μεταφερθεί.

#### 3.1.2.4 Δεδομένα εισόδου

Καταγραφή δεδομένων εισόδου και εξόδου και συνθηκών εγκυρότητας αυτών.

Στα δεδομένα εισόδου συγκαταλέγονται:

* Τα προσωπικά στοιχεία του χρήστη (username, password) για την εισαγωγή στον προσωπικό λογαριασμό του χρήστη (sign in)
* Το ποσό που επιθυμεί να αποπληρώσει ο χρήστης καθώς και η προαιρετική επιλογή να εισάγει εάν θέλει να ξοδέψει συγκεκριμένους από τους πόντους του.

#### 3.1.2.5 Παράμετροι

Καταγραφή παραμέτρων και συνθηκών εγκυρότητας αυτών, εφόσον υπάρχουν παράμετροι.

Πρέπει η κάρτα που εισάγει ο χρήστης να είναι υπαρκτή, να εμπεριέχει το ποσό που αυτός επιθυμεί να αποπληρώσει και το είδος της να υποστηρίζεται από την εφαρμογή. Επίσης για να πραγματοποιηθεί η συναλλαγή θα πρέπει ο χρήστης να έχει συνδεθεί στον προσωπικό του λογαριασμό και να υπάρχει σύνδεση στο διαδίκτυο. Επιπλέον επιβάλλεται το πόσο που εισάγεται για να αποπληρωθεί να είναι μικρότερο ή ίσο από το ποσό οφειλής καθώς επίσης και το συνολικό αντίτιμο που αντιστοιχεί στους πόντους που επιθυμεί να χρησιμοποιήσει, να μην ξεπερνά το ποσό οφειλής. Τέλος, η συναλλαγή θεωρείται έγκυρη μόνο όταν εμφανιστεί το μήνυμα “Successful payment”. Αν για οποιονδήποτε λόγο προκύψει κάποιο σφάλμα (παραδείγματος χάρη ξαφνική απενεργοποίηση κινητού, διακοπή σύνδεσης στο διαδίκτυο κτλ) πριν την εμφάνιση του εν λόγω μηνύματος η εντολή πληρωμής θα ακυρώνεται και τα χρήματα δεν θα δεσμεύονται.

#### 3.1.2.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Περιγραφή με κείμενο (Βήμα 1, Βήμα 2 κλπ) και διαγράμματαUML αλληλουχίας (Sequence) και δραστηριοτήτων (Activity). Περιλαμβάνεται η συμπεριφορά σε απρόβλεπτες καταστάσεις και σφάλματα (εναλλακτικές ροές).

**Βήμα 1:** Είσοδος στην αρχική σελίδα ως επισκέπτης

**Βήμα 2:** Επιλογή κουμπιού Sign in

**Βήμα 3:**Εισαγωγή username, password

**Βήμα 4:**(Αρχική σελίδα χρήστη) Επιλογή από το navigation bar, στην καρτέλα services, του κουμπιού “Payment ”

**Βήμα 5:**(Σελίδα με το χρωστούμενο υπόλοιπο όπου υπάρχει μπάρα για πληκτρολόγηση του ποσού που θα πληρωθεί και των πόντων που θα χρησιμοποιήσει). Συμπληρώνεται το ποσό και οι πόντοι που ο χρήστης επιθυμεί να αποπληρώσει.

**Βήμα 6:**Πάτημα κουμπιού “Next ”

**Βήμα 7:**(Ασφαλής σελίδα τράπεζας για την εισαγωγή της κάρτας) Εισαγωγή ζητούμενων στοιχείων κάρτας

**Βήμα 8:**Πάτημα κουμπιού “Start payment”

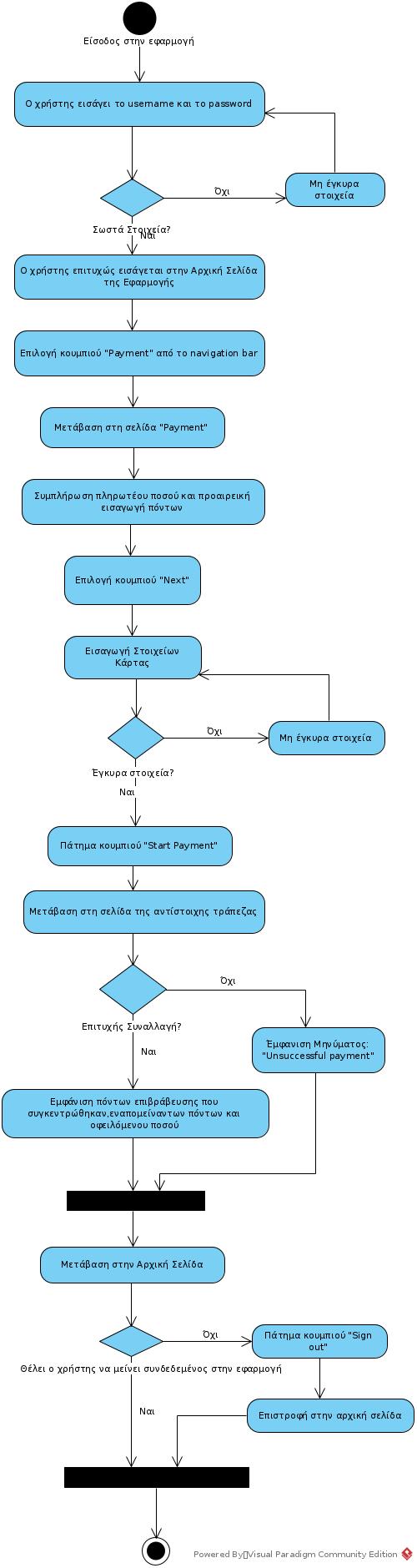
**Βήμα 9:**Εμφάνιση κατάλληλου μηνύματος ανάλογα με την εκτέλεση της πληρωμής (αν η πληρωμή πραγματοποιήθηκε εμφάνιση μηνύματος “Successful payment”, πόντων που έχουν κερδηθεί, συνολικών πόντων και οφειλόμενου ποσού μετά την πληρωμή , σε περίπτωση λάθους, εμφάνιση “Unsuccessful payment” και ανακατεύθυνση στην αρχική σελίδα)

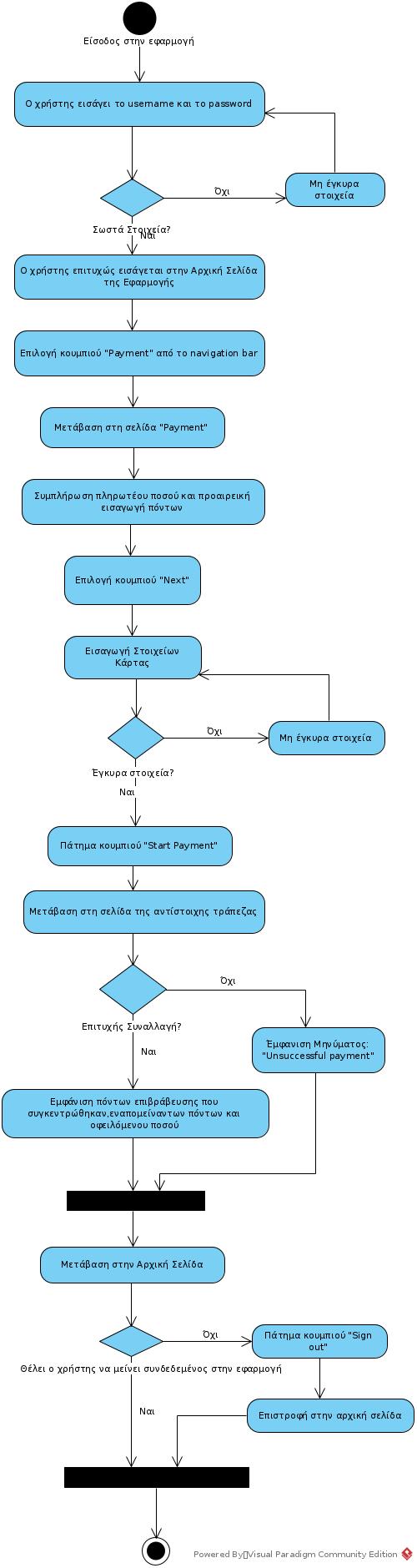
**Βήμα 10:**Επιστροφή στην αρχική σελίδα χρήστη

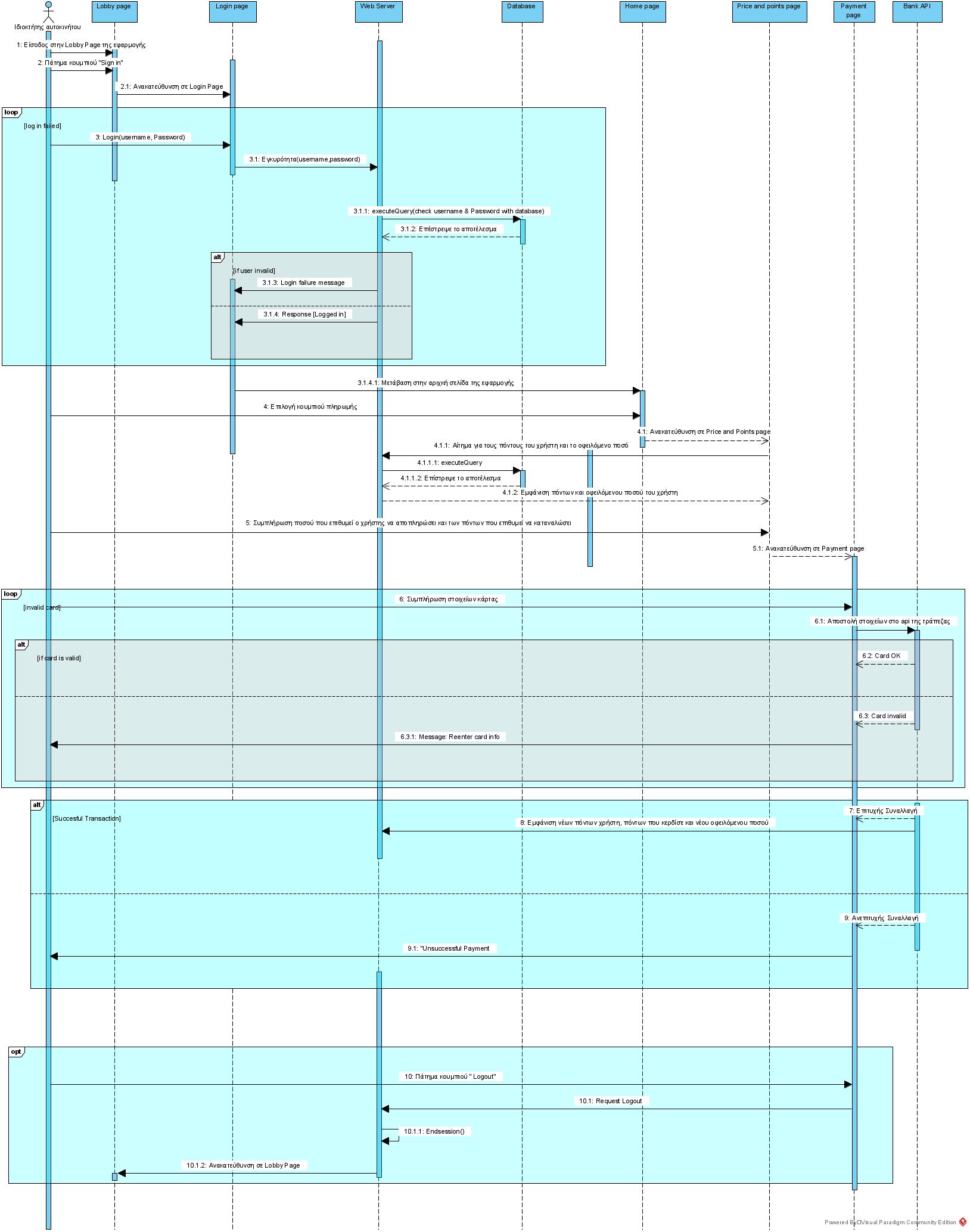
**Βήμα 11:** Αποσύνδεση από την εφαρμογή, αν ο χρήστης το επιθυμεί

**Διαγράμματα UML:**

* Payment activity diagram:

**

**

* **Payment sequence diagram:

#### 3.1.2.7 Δεδομένα εξόδου

Διαγράμματα UMLαλληλουχίας για την παραγωγή δεδομένων εξόδου. Ως δεδομένα εξόδου νοούνται όλα τα δεδομένα του συστήματος τα οποία δημιουργούνται ή μεταβάλλονται κατά την εκτέλεση (αν υπάρχουν τέτοια)

Στα δεδομένα εξόδου συγκαταλέγονται τα εξής:

* Κατά την επιτυχή ολοκλήρωση της πληρωμής θα εμφανίζεται μήνυμα “Successful payment”.
* Κατά την μη επιτυχή ολοκλήρωση της πληρωμής θα εμφανίζεται μήνυμα “Unsuccessful payment”.
* Σε περίπτωση που μετά την συναλλαγή ο χρήστης έχει κερδίσει πόντους τότε αυτοί θα εμφανίζονται στην οθόνη.
* Οι συνολικοί πόντοι του χρήστη μετά την πληρωμή
* Το οφειλόμενο ποσό μετά την πληρωμή

#### 3.1.2.8 Παρατηρήσεις

Ο,τι δεν εντάσσεται στα προηγούμενα, εφόσον υπάρχει

-

### 3.1.3ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 3: **Πρόθεση φόρτισης**

#### 3.1.3.1Χρήστες (ρόλοι) που εμπλέκονται

Αναφορά στους ρόλους που αφορά η περίπτωση χρήσης

Εμπλέκεται ο οδηγός του αυτοκινήτου ο οποίος επιθυμεί να κάνει την φόρτιση και οι σταθμοί φόρτισης.

3.1.3.2Προϋποθέσεις εκτέλεσης

Καταγραφή των συνθηκών που πρέπει να ισχύουν ώστε να μπορεί να εκτελεστεί η περίπτωση χρήσης

Απαραίτητη προϋπόθεση είναι να υπάρχει ένα ηλεκτρικό όχημα το οποίο να χρειάζεται φόρτιση.

#### 3.1.3.3Περιβάλλον εκτέλεσης

Αναφορά στο περιβάλλον στο οποίο εκτελείται η περίπτωση χρήσης. Πχ "διαδικτυακή διεπαφή χρήστη", "DBMS" κλπ

Το περιβάλλον εκτέλεσης είναι:

* Η διαδικτυακή διεπαφή χρήστη όπου θα παρουσιάζονται οι τοποθεσίες του εκάστοτε σταθμού πάνω σε χάρτη καθώς και τα στοιχεία ( που αναφέρονται στο βήμα 6 του ερωτήματος 3.1.3.6) που αφορούν τον επιλεγόμενο σταθμό.
* Η βάση δεδομένων όπου είναι αποθηκευμένα και από την οποία αντλούνται τα δεδομένα που αφορούν το ποσοστό φόρτισης του αυτοκινήτου καθώς και διάφορα στοιχεία των σταθμών
* Το API μέσω του οποίου θα επιτυγχάνεται η επικοινωνία της διεπαφής χρήστη με την βάση δεδομένων.

#### 3.1.3.4Δεδομένα εισόδου

Καταγραφή δεδομένων εισόδου και εξόδου και συνθηκών εγκυρότητας αυτών.

Στα δεδομένα εισόδου συγκαταλέγονται:

* Τα προσωπικά στοιχεία του χρήστη (username, password) για την εισαγωγή στον προσωπικό λογαριασμό του χρήστη (sign in).
* Η επιλογή του σταθμού προς τον οποίον θα κατευθυνθεί ο οδηγός του αυτοκινήτου.

#### 3.1.3.5Παράμετροι

Καταγραφή παραμέτρων και συνθηκών εγκυρότητας αυτών, εφόσον υπάρχουν παράμετροι.

Θα πρέπει αρχικά το αυτοκίνητο να μην είναι πλήρως φορτισμένο (ώστε να έχει νόημα η περίπτωση χρήσης), να υπάρχει σύνδεση στο Internet (ώστε να είναι δυνατή η εμφάνιση των σταθμών στον χάρτη) καθώς επίσης και να είναι ενεργοποιημένο το GPS προκειμένου να είναι δυνατή η πλοήγηση στο επιλεγόμενο σταθμό.

#### 3.1.3.6Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Περιγραφή με κείμενο (Βήμα 1, Βήμα 2 κλπ) και διαγράμματα UML αλληλουχίας (Sequence) και δραστηριοτήτων (Activity). Περιλαμβάνεται η συμπεριφορά σε απρόβλεπτες καταστάσεις και σφάλματα (εναλλακτικές ροές).

**Βήμα 1:** Είσοδος στην αρχική σελίδα ως επισκέπτης

**Βήμα 2:** Επιλογή κουμπιού Sign in

**Βήμα 3:** Εισαγωγή username, password

**Βήμα 4:** Επιλογή από το navigation bar, στην καρτέλα services, του ~~Πάτημα~~ κουμπιού “Nearby stations ”

**Βήμα 5:** Εμφάνιση ενός χάρτη με όλους τους διαθέσιμους σταθμούς

**Βήμα 6 :** Επιλογή προτιμώμενου σταθμού

**Βήμα 7:** Απεικόνιση πληροφοριών που αφορούν τον συγκεκριμένο σταθμό (παραδείγματος χάρη αναφορά της κατάστασης των σταθμών κάθε στιγμή(μαύρος κεραυνός αν ο σταθμός είναι ενεργός και άσπρος ένα όχι), αν ο εκάστοτε σταθμός υποστηρίζει τιμολόγιο, τί τρόπους πληρωμής υποστηρίζει ο σταθμός, τιμοκατάλογο του σταθμού καθώς και αξιολογήσεις χρηστών για τους σταθμούς)

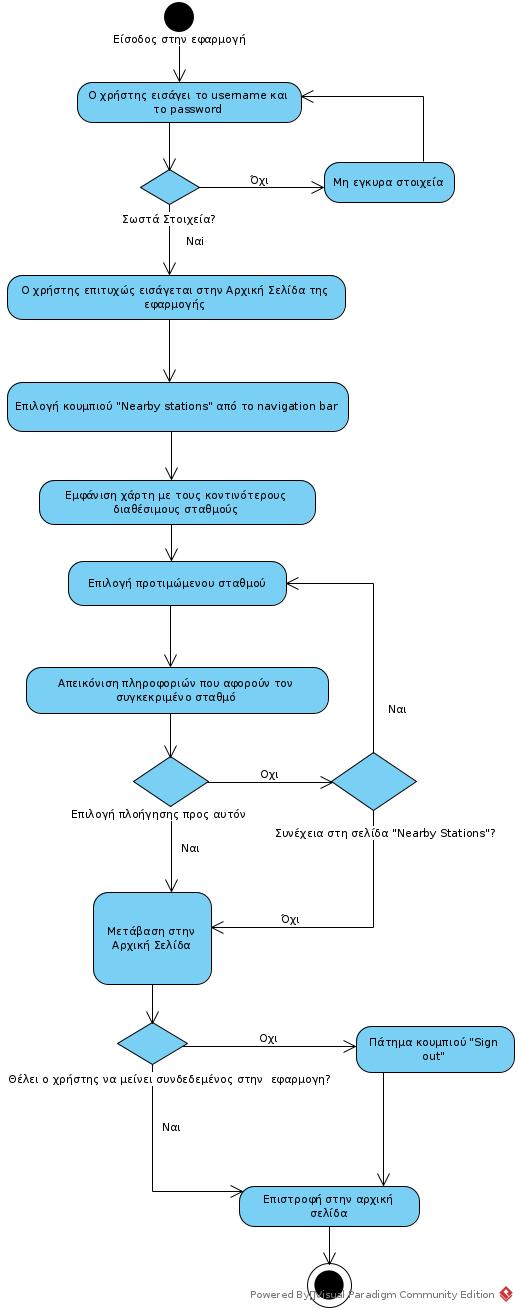
**Βήμα 8:** Δυνατότητα πλοήγησης προς αυτόν

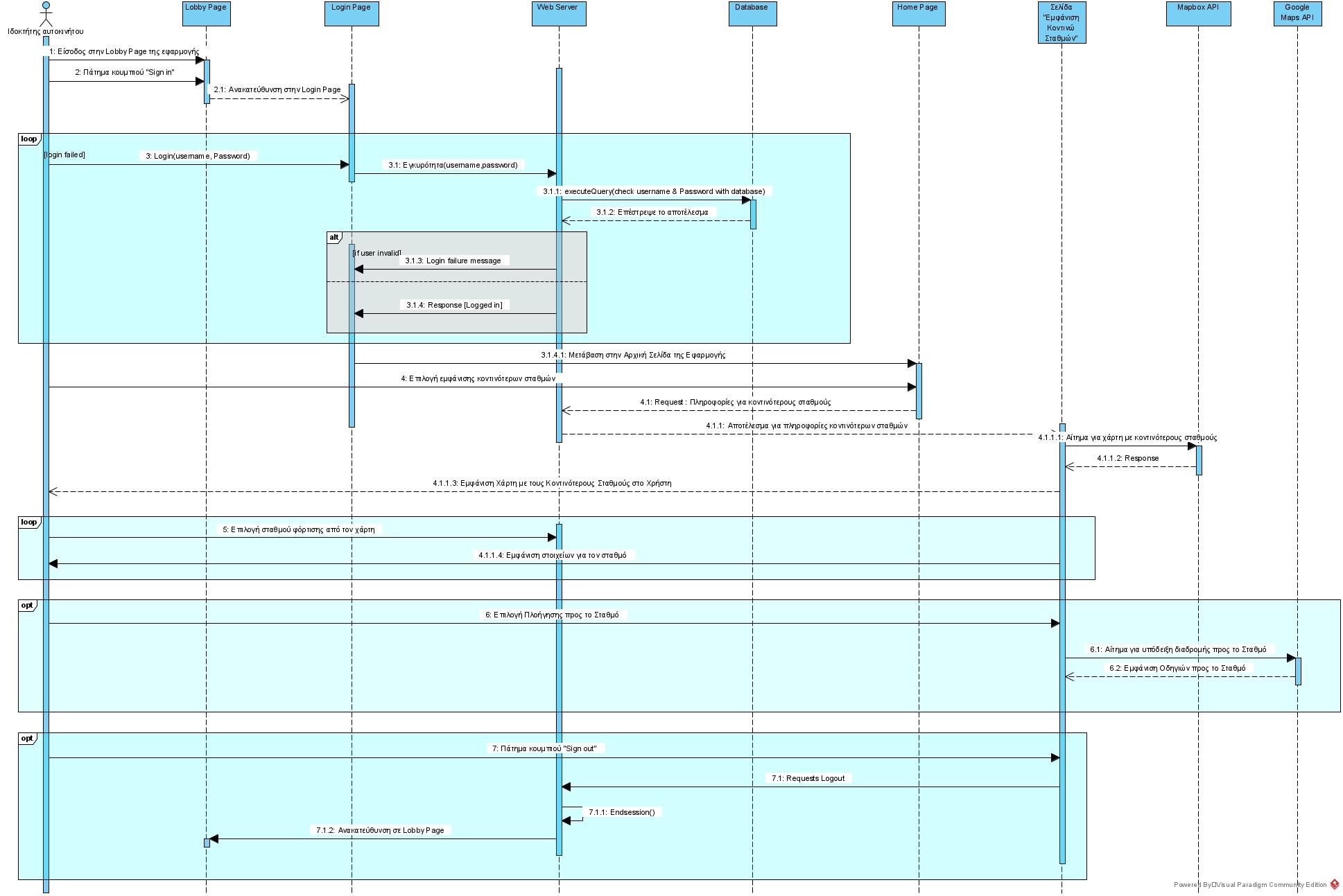
**Βήμα 9**: Επιστροφή στην αρχική σελίδα χρήστη

**Βήμα 10:** Αποσύνδεση από την εφαρμογή, αν ο χρήστης το επιθυμεί

**Διαγράμματα UML:**

* Willing to charge activity diagram:

**

* **Willing to charge sequence diagram:

#### 3.1.3.7Δεδομένα εξόδου

Διαγράμματα UML αλληλουχίας για την παραγωγή δεδομένων εξόδου. Ως δεδομένα εξόδου νοούνται όλα τα δεδομένα του συστήματος τα οποία δημιουργούνται ή μεταβάλλονται κατά την εκτέλεση (αν υπάρχουν τέτοια)

Στα δεδομένα εξόδου συγκαταλέγονται τα εξής:

* Δυναμικός χάρτης με τους σταθμούς
* Απεικόνιση στοιχείων που αφορούν τους σταθμούς σε μορφή JSON (π.χ. αναφορά της κατάστασης των σταθμών κάθε στιγμή(μαύρος κεραυνός αν ο σταθμός είναι ενεργός και άσπρος ένα όχι), αν ο εκάστοτε σταθμός υποστηρίζει τιμολόγιο, τί τρόπους πληρωμής υποστηρίζει ο σταθμός, τιμοκατάλογο του σταθμού καθώς και αξιολογήσεις χρηστών για τους σταθμούς)
* Οδηγίες πλοήγησης προς το σταθμό επιλογής

### 3.1.4ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 4: **Στατιστικά και άλλα στοιχεία για ανεφοδιασμό που αφορούν παραγωγούς Ενέργειας )**

#### 3.1.4.1Χρήστες (ρόλοι) που εμπλέκονται

Αναφορά στους ρόλους που αφορά η περίπτωση χρήσης

Οι χρήστες που εμπλέκονται είναι οι παραγωγοί ηλεκτρικής ενέργειας και οι διανομείς ηλεκτρικής ενέργειας που βλέπουν τα στοιχεία και τα στατιστικά που αφορούν την κατανάλωση ενέργειας. Σε ένα δεύτερο επίπεδο εμπλέκονται επιπλέον οι ιδιοκτήτες των οχημάτων και των σταθμών φόρτισης καθώς τα δεδομένα που παρέχονται σχετικά με την ηλεκτρική ενέργεια προέρχονται σε μεγάλο βαθμό από τις φορτίσεις που αυτοί πραγματοποιούν ή που γίνονται στον σταθμό τους αντίστοιχα.

#### 3.1.4.2Προϋποθέσεις εκτέλεσης

Καταγραφή των συνθηκών που πρέπει να ισχύουν ώστε να μπορεί να εκτελεστεί η περίπτωση χρήσης

Πρέπει να υπάρχουν αρκετοί χρήστες της εφαρμογής ώστε να έχει νόημα η εξαγωγή γενικών στατιστικών και προβλέψεων για την κατανάλωση ενέργειας. Πρέπει επιπλέον οι ιδιοκτήτες των οχημάτων φόρτισης να έχουν συναινέσει στην ανώνυμη συλλογή στοιχείων που αφορούν την κατανάλωση ενέργειας από αυτούς καθώς επίσης και οι ιδιοκτήτες των σταθμών φόρτισης να έχουν αποδεχτεί την συλλογή στοιχείων που αφορούν την ενέργεια που καταναλώνεται στον σταθμό τους. Πρέπει επιπρόσθετα να έχουν καθοριστεί και διατυπωθεί τα απαραίτητα μαθηματικά μοντέλα τα οποία επιτρέπουν την παραγωγή σωστών στατιστικών στοιχείων και προβλέψεων. Τέλος, είναι απαραίτητο τα δεδομένα σχετικά με την κατανάλωση ενέργειας ανά φόρτιση να αποθηκεύονται για μεγάλα χρονικά διαστήματα στην βάση, προκειμένου να είναι δυνατή η εξαγωγή μακροχρόνιων στατιστικών και προβλέψεων.

#### 3.1.4.3Περιβάλλον εκτέλεσης

Αναφορά στο περιβάλλον στο οποίο εκτελείται η περίπτωση χρήσης. Πχ "διαδικτυακή διεπαφή χρήστη", "DBMS" κλπ

Το περιβάλλον εκτέλεσης είναι η διαδικτυακή επαφή χρήστη όπου θα παρουσιάζονται τα σχετικά με την ζήτηση και την κατανάλωση ενέργειας ανά σταθμό σε μορφή δεδομένων στην σελίδα ή διαγραμμάτων (barcharts/pies) , η βάση δεδομένων όπου αποθηκεύονται τα δεδομένα των φορτίσεων που αφορούν την κατανάλωση ενέργεια καθώς επίσης και το API μέσω του οποίου θα επιτυγχάνεται η επικοινωνία της διεπαφής χρήστη με την βάση δεδομένων.

#### 3.1.4.4Δεδομένα εισόδου

Καταγραφή δεδομένων εισόδου και εξόδου και συνθηκών εγκυρότητας αυτών.

Στα δεδομένα εισόδου συγκαταλέγονται:

* Toόνομα και ο κωδικός χρήστη μέσω του οποίου θα συνδέεται ο χρήστης (παραγωγός ενέργειας) στην εφαρμογή.
* H χρονική περίοδος, ο σταθμός φόρτισης ή η περιοχή για την οποία ο χρήστης επιθυμεί να δει δεδομένα και στατιστικά.
* Για να είναι τα τελευταία δεδομένα εισόδου έγκυρα θα πρέπει να έχουν πολύ συγκεκριμένη μορφή, αναγνωρίσιμη από το σύστημα, το οποίο θα επιτυγχάνεται με την παροχή σαφούς οδηγίας στην διεπαφή χρήστη σχετικά με την μορφή τους και έλεγχο ότι έχουν την επιθυμητή μορφή από τον χρήστη.

3.1.4.5Παράμετροι

Καταγραφή παραμέτρων και συνθηκών εγκυρότητας αυτών, εφόσον υπάρχουν παράμετροι.

-

#### 3.1.4.6Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Περιγραφή με κείμενο (Βήμα 1, Βήμα 2 κλπ) και διαγράμματα UML αλληλουχίας (Sequence) και δραστηριοτήτων (Activity). Περιλαμβάνεται η συμπεριφορά σε απρόβλεπτες καταστάσεις και σφάλματα (εναλλακτικές ροές).

**Βήμα 1:** Είσοδος στην αρχική σελίδα ως επισκέπτης

**Βήμα 2:** Επιλογή κουμπιού Sign in

**Βήμα 3:** Εισαγωγή username, password

**Βήμα 4:** Μεταφορά στην αρχική σελίδα και επιλογή μέσω Dropdown menu (σε navbar) εμφάνισης στατιστικών (“Energy demand forecasts”) ή απλής αναφοράς δεδομένων κατανάλωσης ενέργειας (“Energy Consumption”).

**Βήμα 5:**Μόνο στην περίπτωση που επιλεγούν τα δεδομένα κατανάλωσης ενέργειας εμφανίζονται δύο καινούριες επιλογές σε υπομενού, οι οποίες καθορίζουν ανά τι θα εμφανιστεί το αποτελέσμα. Υπάρχουν οι επιλογές "Per station" που θα εμφανίσει την κατανάλωση ενέργειας ανά σταθμό για μια περίοδο ή “Per District  ” που θα την εμφανίσει ανά περιοχή και περίοδο

**Βήμα 6:** Προσδιορισμός των περιορισμών της πρόβλεψης ή των δεδομένων που ο χρήστης επιθυμεί να δει (προσδιορισμός χρονικής περιόδου, ονόματος/τοποθεσίας σταθμού φόρτισης, γεωγραφικής περιοχής)

**Βήμα 7:** Πάτημα κουμπιού “Show Results” που εμφανίζει τα ζητούμενα δεδομένα - διαγράμματα στην ίδια σελίδα

**Βήμα 8:** Ο χρήστης έχει την δυνατότητα αν επιθυμεί να αλλάξει τους περιορισμούς και να δει το διάγραμμα που επέλεξε για διαφορετικές συνθήκες, πχ για διαφορετική χρονική περίοδο. (Επανάληψη Βημάτων 6-9 όσες φορές ο χρήστης επιθυμεί)

**Βήμα 9:** Πάτημα κουμπιού “Return to home page” και επιστροφή στην αρχική σελίδα του χρήστη

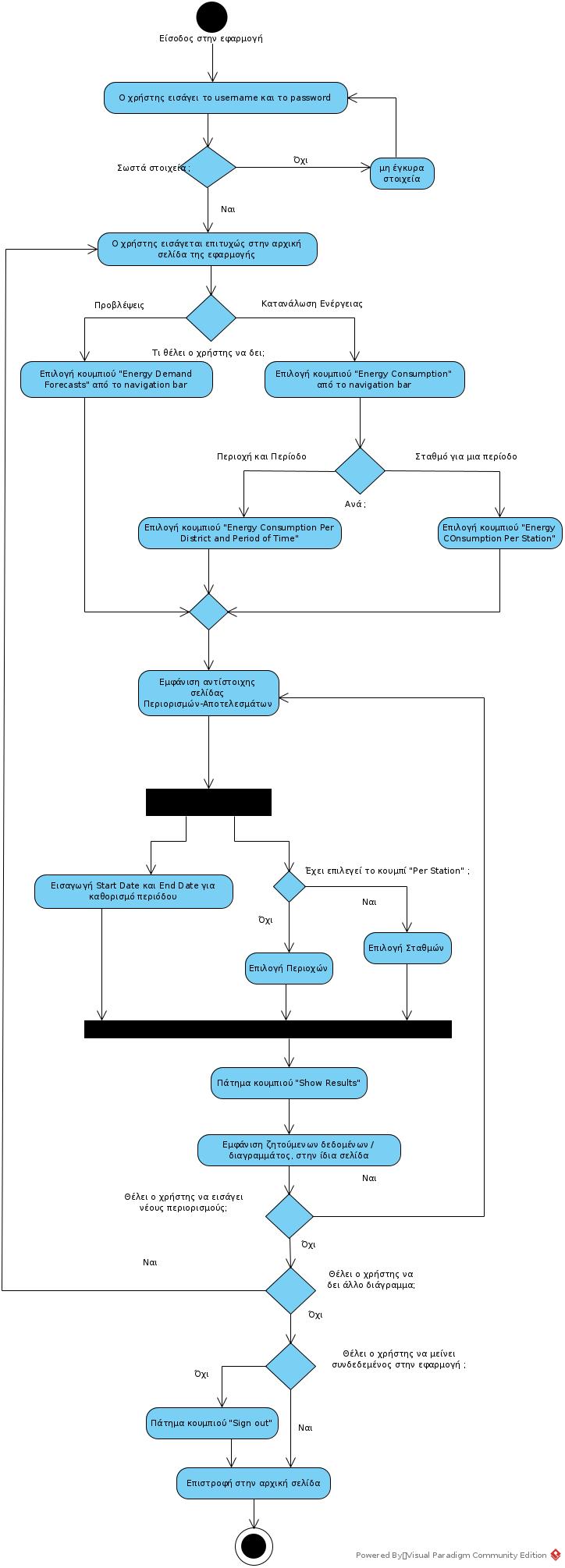
**Βήμα 10:** Ο χρήστης ενδέχεται να θέλει να δει κάποιο διαφορετικό διάγραμμα, σε αυτή την περίπτωση έχουμε επανάληψη Βημάτων 4-11

**Βήμα 11:** Προαιρετική αποσύνδεση από την εφαρμογή, με πάτημα του κουμπιού “Sign out”

**Διαγράμματα UML:**

* Statistics for energy producer activity diagram:

#### 



* Statistics for energy producer sequence diagram:

#### 

#### 3.1.4.7Δεδομένα εξόδου

Διαγράμματα UML αλληλουχίας για την παραγωγή δεδομένων εξόδου. Ως δεδομένα εξόδου νοούνται όλα τα δεδομένα του συστήματος τα οποία δημιουργούνται ή μεταβάλλονται κατά την εκτέλεση (αν υπάρχουν τέτοια)

Στα δεδομένα εξόδου συγκαταλέγονται τα εξής:

* Στατιστικά δεδομένα
* Δεδομένα ανά σταθμό, χρονική περίοδο, περιοχή σε μορφή JSON
* Προβλέψεις ζήτησης ενέργειας ανά περιοχή και χρονική περίοδο

#### 3.1.4.8Παρατηρήσεις

Ο,τι δεν εντάσσεται στα προηγούμενα, εφόσον υπάρχει

-

### 3.1.5ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 5: **Έλεγχος απόδοσης οχημάτων από κατασκευαστές**

#### 3.1.5.1Χρήστες (ρόλοι) που εμπλέκονται

Αναφορά στους ρόλους που αφορά η περίπτωση χρήσης

Οι εμπλεκόμενοι χρήστες είναι οι κατασκευαστές των οχημάτων οι οποίοι επιθυμούν να αποκτήσουν πρόσβαση σε δεδομένα που αφορούν τα μοντέλα τους τα οποία κυκλοφορούν. Σε ένα δεύτερο επίπεδο εμπλέκονται επιπλέον οι ιδιοκτήτες των οχημάτων καθώς τα δεδομένα που παρέχονται στους κατασκευαστές προέρχονται από αυτούς.

#### 3.1.5.2Προϋποθέσεις εκτέλεσης

Καταγραφή των συνθηκών που πρέπει να ισχύουν ώστε να μπορεί να εκτελεστεί η περίπτωση χρήσης

Για να μπορέσει να επιτευχθεί η ζητούμενη χρήση, θα πρέπει ο χρήστης να έχει εισέλθει στον λογαριασμό του χρησιμοποιώντας τα ανάλογα αναγνωριστικά του (username/password), να υπάρχουν μοντέλα της συγκεκριμένης εταιρίας τα οποία χρησιμοποιούνται από χρήστες της εφαρμογής καθώς και να έχει περάσει ένα αρχικό χρονικό διάστημα από την έναρξη συλλογής των δεδομένων ώστε να υπάρχουν αρκετά για να έχει κάποια υπόσταση οποιαδήποτε στατιστική πληροφορία.

#### 3.1.5.3Περιβάλλον εκτέλεσης

Αναφορά στο περιβάλλον στο οποίο εκτελείται η περίπτωση χρήσης. Πχ "διαδικτυακή διεπαφή χρήστη", "DBMS" κλπ

Στο περιβάλλον εκτέλεσης περιλαμβάνονται:

* Η διαδικτυακή διεπαφή μέσω της οποίας ο χρήστης/κατασκευαστής θα εισέρχεται στο σύστημα
* Η βάση δεδομένων στην οποία βρίσκονται αποθηκευμένα τα δεδομένα τα οποία πρέπει ανακτηθούν
* Το API για την επικοινωνία μεταξύ της διεπαφής χρήστη και της βάσης δεδομένων.

#### 3.1.5.4Δεδομένα εισόδου

Καταγραφή δεδομένων εισόδου και εξόδου και συνθηκών εγκυρότητας αυτών.

Στα δεδομένα εισόδου συγκαταλέγονται:

* Tα αναγνωριστικά που θα χρησιμοποιήσει ο χρήστης για την είσοδό τους στην εφαρμογή(username/password)
* Το αντίστοιχο κουμπί/το αντίστοιχο πεδίο που θα πατήσει/συμπληρώσει στην διεπαφή του χρήστη για να πλοηγηθεί στα διάφορα δεδομένα που του παρέχονται.

#### 3.1.5.5Παράμετροι

Καταγραφή παραμέτρων και συνθηκών εγκυρότητας αυτών, εφόσον υπάρχουν παράμετροι.

-

#### 3.1.5.6Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Περιγραφή με κείμενο (Βήμα 1, Βήμα 2 κλπ) και διαγράμματα UML αλληλουχίας (Sequence) και δραστηριοτήτων (Activity). Περιλαμβάνεται η συμπεριφορά σε απρόβλεπτες καταστάσεις και σφάλματα (εναλλακτικές ροές).

**Βήμα 1:** Είσοδος στην αρχική σελίδα ως επισκέπτης

**Βήμα 2:** Επιλογή κουμπιού για Sign in

**Βήμα 3:**Εισαγωγή username, password

**Βήμα 4:** Μεταφορά στην αρχική σελίδα και επιλογή μέσω Dropdown menu (σε navbar) εμφάνισης δεδομένων φόρτισης (“Charging sessions report”), στατιστικών μέσου ενεργειακού κόστους ανά χιλιόμετρο (“Mean energy cost”) ή αναφοράς κατανάλωσης ενέργειας (“Energy consumtption report”).

**Βήμα 5:**Μόνο στην περίπτωση που επιλεγούν τα στατιστικά μέσου ενεργειακού κόστους ανά χιλιόμετρο  εμφανίζονται δύο καινούριες επιλογές σε υπομενού, οι οποίες θα καθορίζουν αν το τα δεδομένα θα αφορούν το μέσο ενεργειακό κόστος ανά χιλιόμετρο για τα αυτοκίνητα κάποιου μοντέλου του συγκεκριμένου κατασκευαστή (επιλογή “For your car models”), ή στατιστικά μέσου ενεργειακού κόστους ανά χιλιόμετρο ανά κατασκευστή (επιλογή “General statistics”).

Στα βήματα 4 και 5 μπορεί ο χρήστης πατώντας σε αντίστοιχη ετικέτα στο υπομενού να μεταφερθεί σε υποσελίδες, όπου προσδιορίζονται ξανά οι επιλογές που διαθέτει ως προς τα δεδομένα που επιθυμεί να δει και θα μπορεί να επιλέξει από εκεί αντί να επιλέξει απευθείας από το υπομενού.

**Βήμα 6:** Προσδιορισμός παραμέτρων για την εμφάνιση των ζητούμενων δεδομένων π.χ. προσδιορισμός χρονικής περιόδου, μοντέλου οχήματος, γεωγραφικής περιοχής και

Βήμα 7 : Πάτημα κουμπιού για εμφάνιση αποτελεσμάτων

**Βήμα 8 :**Προαιρετική δυνατότητα επιλογής κουμπιού για εξαγωγή των στατιστικών/δεδομένων από την εφαρμογή(“Extract data”) σε csv μορφή.

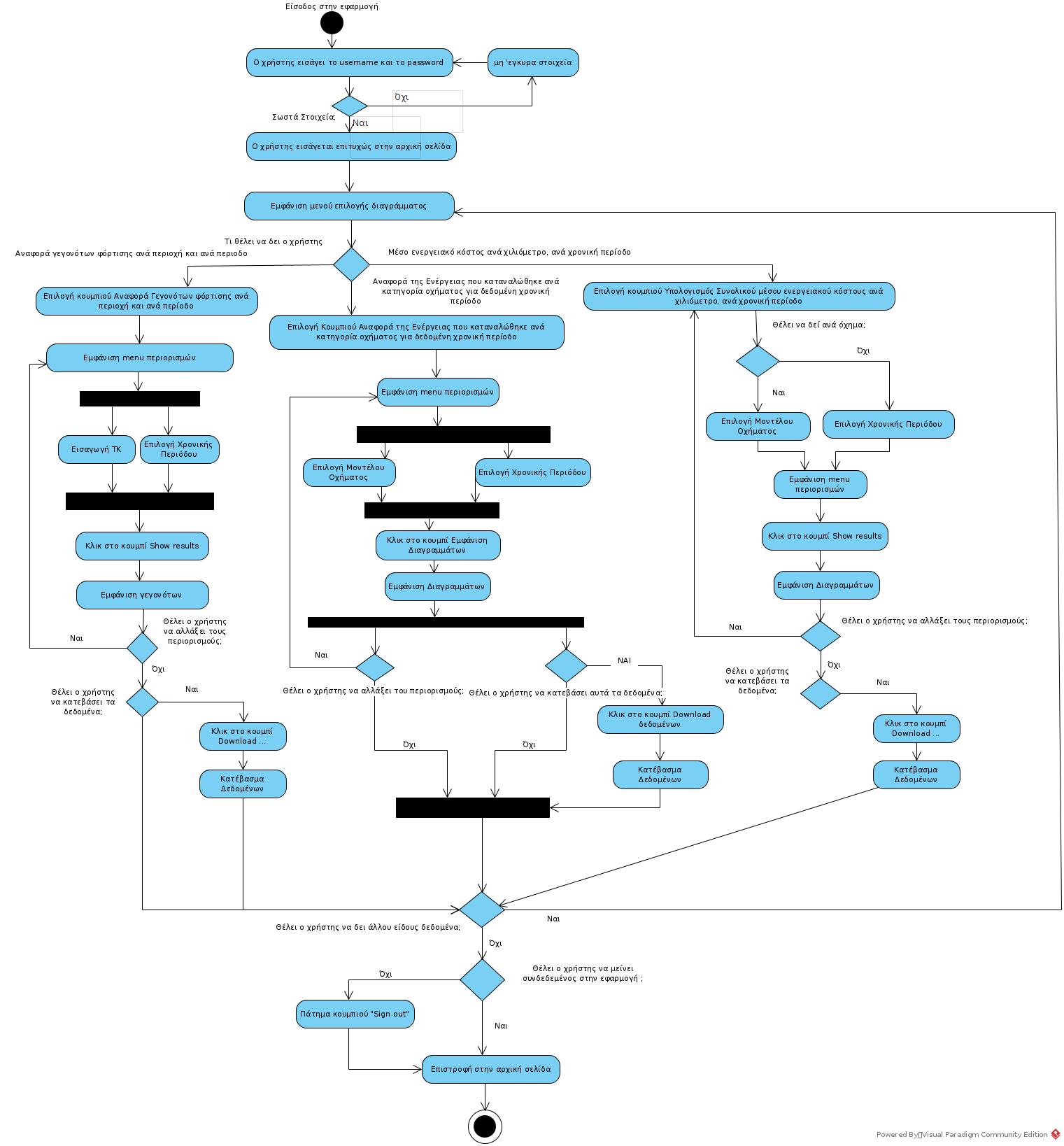
**Βήμα 9** : Ο χρήστης μπορεί να αλλάξει τις παραμέτρους για εμφάνιση των ζητούμενων δεδομένων και να δει τα δεδομένα για τις νέες παραμέτρους στην ίδια σελίδα (επανάληψη βημάτων 6,7,8) όσες φορές το επιθυμεί.

**Βήμα 10:** Πάτημα κουμπιού “Home” στο navabar του ιστοτόπου και επιστροφή στην αρχική σελίδα του χρήστη ή επιλογή εμφάνισης νέας κατηγορίας δεδομένων μέσω του navbar.

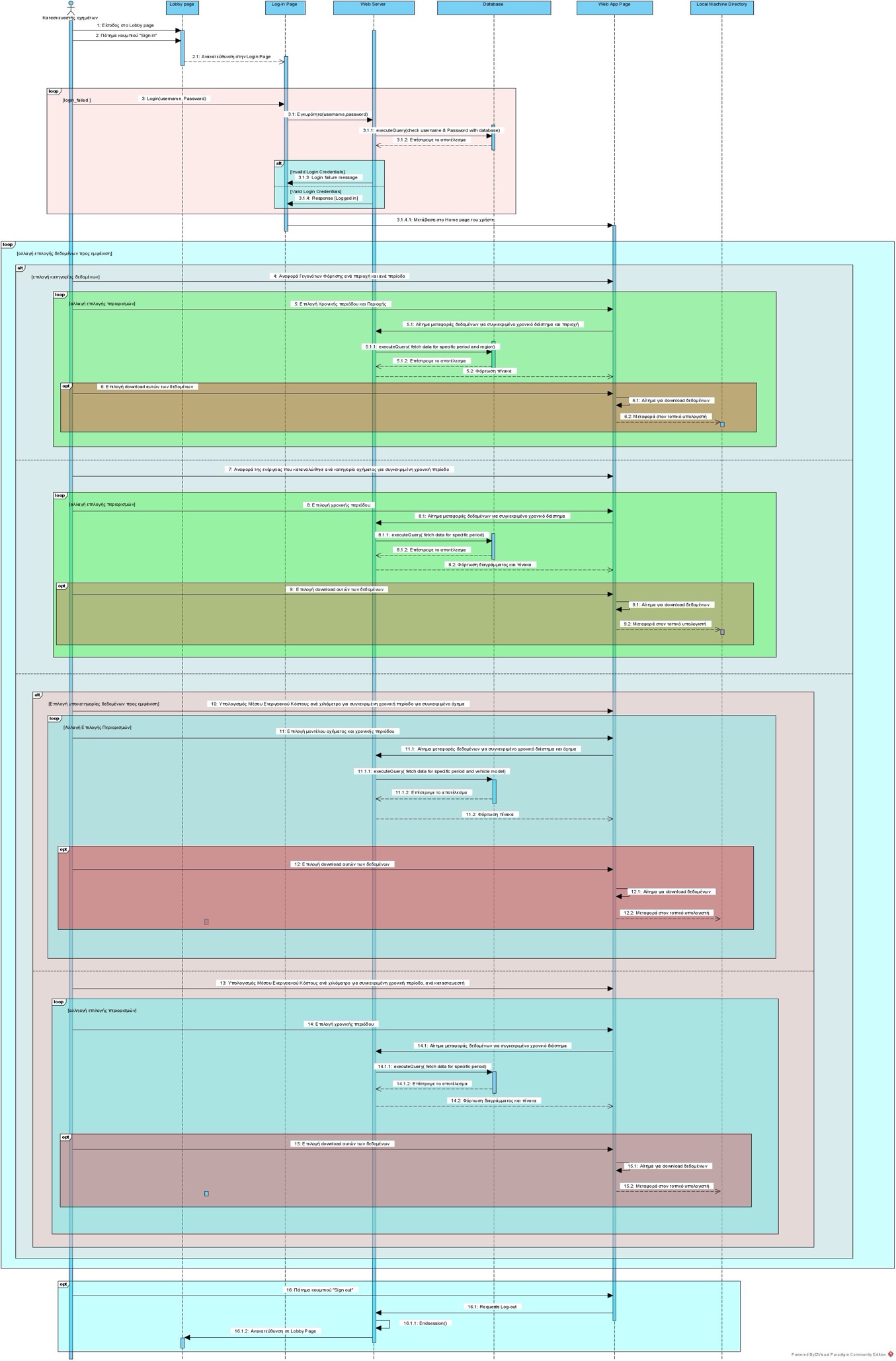
**Βήμα 11:** Αποσύνδεση από την εφαρμογή, αν ο χρήστης το επιθυμεί

**Διαγράμματα UML:**

* Statistics for manufacturer activity diagram:

**

* Statistics for manufacturer sequence diagram:

**

#### 3.1.5.7Δεδομένα εξόδου

Διαγράμματα UML αλληλουχίας για την παραγωγή δεδομένων εξόδου. Ως δεδομένα εξόδου νοούνται όλα τα δεδομένα του συστήματος τα οποία δημιουργούνται ή μεταβάλλονται κατά την εκτέλεση (αν υπάρχουν τέτοια)

Στα δεδομένα εξόδου συγκαταλέγονται τα εξής:

* Oι πληροφορίες οι οποίες θα εμφανιστούν σε μορφή row charts/πινάκων στην διεπαφή του χρήστη
* Tο αντίστοιχο έγγραφο το οποίο θα μπορεί να εξάγει ο χρήστης με τα δεδομένα σε μορφή CSV από τη βάση.

#### 3.1.5.8Παρατηρήσεις

Ο,τι δεν εντάσσεται στα προηγούμενα, εφόσον υπάρχει

-

## 3.2 Απαιτήσεις επιδόσεων

Ποσοτική τεκμηρίωση μέτρων και κριτηρίων επιθυμητών επιδόσεων με αναφορά στα ποσοτικά χαρακτηριστικά εισόδων και φορτίου του λογισμικού.

Καταρχάς είναι απαραίτητο το σύστημα να μπορεί να ανταποκρίνεται γρήγορα σε αιτήσεις της διεπαφής των ιδιοκτητών ,όπως η εμφάνιση των κοντινότερων διαθέσιμων σταθμών. Λαμβάνοντας αυτό υπόψιν, μας ενδιαφέρει ο χρόνος απόκρισης σε ms τόσο μεταξύ του API των σταθμών και του API που αφορά τους ιδιοκτήτες, καθώς και του τελευταίου με το UI του χρήστη. Επίσης, λόγω της αύξησης των χρηστών της εφαρμογής με την πάροδο του χρόνου, θεωρείται απαραίτητο η αύξηση των αποθηκευμένων δεδομένων στη βάση, να συνοδεύεται και από αντίστοιχη αύξηση των διαθέσιμων υπολογιστικών πόρων. Επιπλέον, είναι σημαντικός και ο χρόνος όπου γίνεται, σε επίπεδο Backend, κάποια επεξεργασία στα δεδομένα, εξαγωγή στατιστικών και επιστροφή στο user interface των εταιρειών. Ακόμα, δεδομένου ότι πολλοί ιδιοκτήτες ηλεκτρικών οχημάτων θα πρέπει να έχουν ταυτόχρονα πρόσβαση στην εφαρμογή αλλά και ότι το API μεγάλων σταθμών θα πρέπει να εξυπηρετεί αρκετούς χρήστες ταυτόχρονα, είναι σημαντικό να εξετάσουμε την απόδοση και τους χρόνους απόκρισης του API που αφορά τους ιδιοκτήτες, σε περιπτώσεις που πολλοί χρήστες είναι συνδεδεμένοι στο σύστημα, αλλά και το API των σταθμών και να βελτιώσουμε τους υπολογιστικούς πόρους αντίστοιχα.

## 3.3 Απαιτήσεις οργάνωσης δεδομένων

### 3.3.1 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα

Αρχικά για την πρόσβαση στα δεδομένα είναι απαραίτητο να υπάρχει σύνδεση στο internet προκειμένου να φορτώσει το UI αλλά και προκειμένου να επικοινωνεί αυτό κάθε φορά με το κατάλληλο API. Επιπλέον για την παρουσίαση σε χάρτη σταθμών που είναι κοντά στον χρήστη, απαιτείται GPS.

Όσον αφορά τους περιορισμούς, οι εταιρείες θα έχουν πρόσβαση σε δεδομένα ιδιοκτητών ηλεκτρικών οχημάτων μόνο εφόσον αυτοί έχουν δώσει τη συγκατάθεση τους. Επίσης κρίθηκε απαραίτητο καμία εταιρεία να μην μπορεί να δει δεδομένα που αφορούν άλλες εταιρείες και αντίστοιχα κανένας ιδιοκτήτης ηλεκτρικού οχήματος να μην έχει πρόσβαση σε προσωπικά δεδομένα άλλων χρηστών. Τέλος, για τη δημιουργία στατιστικών για τις εταιρείες είναι σημαντικό να είναι αποθηκευμένα δεδομένα που αφορούν επαρκές χρονικό διάστημα.

## 3.4 Περιορισμοί σχεδίασης

Λεπτομερής τεχνική τεκμηρίωση των περιορισμών σχεδίασης οι οποίοι επιβάλλονται από απαιτήσεις συμμόρφωσης σε πρότυπα, κανονισμούς, ή άλλους περιορισμούς του έργου. Περιλαμβάνεται η πολιτική ονοματολογίας οντοτήτων δεδομένων και πεδίων. Τέτοιοι περιορισμοί μπορεί να επιβάλλονται από τη χρήση βιβλιοθηκών, frameworks, περιβαλλόντων ανάπτυξης κλπ

* Θέλουμε το σύστημα να είναι συμβατό με το πρότυπο ΙΕΕΕ
* Έγινε χρήση της NodeJS για το back end
* Έγινε χρήση της ReactJS για το front end
* Έγινε χρήστη του Click για το Command Line
* Έγινε χρήση του Mocha, Chai για το testing του back end
* Έγινε χρήση των mock, unittest, pytest για το testing του cli
* Ένας περιορισμός επιβάλλεται από τη χρήση Restful API. Συγκεκριμένα το back-end πρέπει να φροντίσει ώστε να επιστρέφει πάντα μία πλήρη και κατανοητή αναπαράσταση των δεδομένων σε μορφοποίηση JSON. Επιπλέον, το User Interface, είναι επιφορτισμένο με τη ευθύνη να διαχειρίζεται σωστά τα JSON δεδομένα που λαμβάνει από το εκάστoτε API και να τα επιστρέφει σε κατανοητή μορφή στον χρήστη.
* Ένας ακόμη περιορισμός πηγάζει από την χρήση του αρχιτεκτονικού μοντέλου MVC(Model-view-Control) για το front-end. Συγκεκριμένα το User Interface οφείλει να αποτελείται από τρία διακριτά μέρη που οφείλουν να αλληλεπιδρούν ορθά μεταξύ τους. Τα τρία αυτά μέρη είναι το Model, το οποίο συνίσταται από τα δεδομένα της εφαρμογής και τους κανόνες για τη διαχείρισή τους, το View που συνίσταται από τα διάφορα αντικείμενα του User Interface, όπως εικόνες, input fields κλπ, αλλά και το Controller, το οποίο αποτελεί ουσιαστικά το διαμεσολαβητή των άλλων δύο μερών, καθώς διαχειρίζεται την αλληλεπίδραση του χρήστη με τα στοιχεία του View(πχ το κλικ σε ένα check box) και μεταφέρει δεδομένα από το View στο Model και αντίστροφα. Οφείλουμε να χωρίσουμε το UI στα τρία αυτά μέρη υλοποιώντας τους άνωθεν κανόνες.

## 3.5 Λοιπές απαιτήσεις

### 3.5.1 Απαιτήσεις διαθεσιμότητας λογισμικού

Το API και το UI που αφορούν τους ιδιοκτήτες του οχήματος πρέπει να είναι διαρκώς διαθέσιμα καθώς ένας ιδιοκτήτης οχήματος αναμένει οποιαδήποτε ώρα της μέρας να μπορεί να δει την κατάσταση του οχήματος και την πορεία φόρτισής του. Ωστόσο, το API και το UI που αφορούν εταιρείες κατασκευής και παραγωγής ενέργειας καθώς και τους σταθμούς, είναι σημαντικό να είναι διαθέσιμα κυρίως τις ώρες που οι εν λόγω επιχειρήσεις λειτουργούν (εργάσιμες μέρες και ώρες), οπότε ενδεχομένως να μπορούμε να πραγματοποιούμε αναβαθμίσεις τις νυχτερινές ώρες.

### 3.5.2 Απαιτήσεις ασφάλειας

Τεκμηρίωση απαιτήσεων ασφαλείας

Είναι αναγκαίο στα δεδομένα μία εταιρείας να έχει πρόσβαση μόνο η εταιρεία και στα δεδομένα ενός ιδιοκτήτη αυτοκινήτου να έχει πρόσβαση μόνο ο ιδιοκτήτης και σε περίπτωση που αποστέλλονται δεδομένα ιδιώτη προς εταιρεία θα πρέπει να στέλνονται κρυπτογραφημένα, άρα ανώνυμα και με τη συγκατάθεση του ιδιοκτήτη. Επιπλέον κάποιοι άλλοι περιορισμοί είναι το ασφαλές sing in , με αποθήκευση μόνο κατακερματισμένων- hashed κωδικών στη βάση. Επίσης, η ανταλλαγή δεδομένων γίνεται μέσω ασφαλούς σύνδεσης SSL.

### 3.5.3 Απαιτήσεις συντήρησης

Τεκμηρίωση απαιτήσεων συντήρησης

Είναι απαραίτητο να δημιουργηθούν αρχικά κάποια beta tests προκειμένου να ελεγχθεί η ορθή λειτουργία της εφαρμογής στο μεγαλύτερο μέρος της. Ενδεχομένως θα μπορούσε να κυκλοφορήσει αρχικά και μία beta edition για λίγους χρήστες, για τον ίδιο λόγο με τα προηγούμενα. Επιπλέον, επειδή οι εταιρείες θα χρειάζονται μεγάλο όγκο δεδομένων είναι σημαντικό να δημιουργούνται back-up αντίγραφα κάποιων δεδομένων προκειμένου να διασφαλιστεί ότι δεν θα απολεσθούν. Ταυτόχρονα, όταν κυκλοφορήσει η εφαρμογή είναι σημαντικό να δημιουργηθεί μία ομάδα εξυπηρέτησης χρηστών που θα μεταφέρει τα προβλήματα που εντοπίζουν οι χρήστες στους προγραμματιστές ή θα δύναται να προσφέρει εξειδικευμένη υποστήριξη σε οργανισμούς όπως πχ σε μία εταιρεία. Τέλος, θα γίνονται τακτικές αναβαθμίσεις του λογισμικού ώστε να διορθώνονται bugs που προκύπτουν καθώς και να είναι συμβατό με νέες εκδόσεις τεχνολογιών.