Έγγραφο απαιτήσεων λογισμικού (SRS)

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

Πληροφοριακό σύστημα διαχείρισης φόρτισης ηλεκτρικών οχημάτων

# Εισαγωγή

## 1.1 Εισαγωγή: σκοπός του λογισμικού

Ο σκοπός του συστήματος που θα υλοποιήσουμε είναι η δημιουργία ενός πληροφοριακού συστήματος διαχείρισης της φόρτισης ηλεκτρικών οχημάτων, η οποία πραγματοποιείται σε δημόσιους ή ιδιωτικούς χώρους. Οι ιδιαίτερες απαιτήσεις των φορτιστών σε ηλεκτρική ισχύ σε συνδυασμό με τον μεγάλο απαιτούμενο χρόνο φόρτισης καθιστούν αναγκαίο τον σχεδιασμό ενός διαφορετικού συστήματος διαχείρισης της διαδικασίας της φόρτισης, από αυτόν των υγρών καυσίμων. Επιπλέον, η απελευθέρωση της αγοράς έχει οδηγήσει σε κατάργηση του σχήματος συνεργασίας ενός operator (διαχειριστής σταθμού) με μοναδικό πάροχο, ενώ έχουμε πληθώρα διαφορετικών κατασκευαστών οχημάτων. Το γεγονός αυτό επιβάλλει την προτυποποίηση των πρωτοκόλλων φόρτισης και των ηλεκτρικών συνδέσεων.

## 1.2 Διεπαφές (interfaces)

### 1.2.1 Διεπαφές με εξωτερικά συστήματα

### 1.2.2 Διεπαφές με το χρήστη

### 

*Εικόνα 1: Use Case (UML)*

# Αναφορές - πηγές πληροφοριών

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

*ΝΑ ΜΗΝ ΣΥΜΠΛΗΡΩΘΕΙ ΑΝ ΔΕΝ ΑΠΑΙΤΕΙΤΑΙ*

[*https://www.greenflux.com/ev-charging-software/*](https://www.greenflux.com/ev-charging-software/)

[*https://www.chargelab.co/*](https://www.chargelab.co/)

[*https://evbox.com/en/products/charging-management*](https://evbox.com/en/products/charging-management)

# Προδιαγραφές απαιτήσεων λογισμικού

## 3.1 Περιπτώσεις χρήσης

### 3.1.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: Φόρτιση Ηλεκτρικών Αυτοκινήτων

#### 3.1.1.1 Χρήστες (ρόλοι) που εμπλέκονται

Στην φόρτιση των ηλεκτρικών αυτοκινήτων εμπλέκονται οι παρακάτω ρόλοι:

* Κάτοχοι Ηλεκτρικών οχημάτων (EV Owners): Με σκοπό την τελική χρήση του Use case για φόρτιση του αυτοκινήτου.
* Διαχειριστές Σταθμών (Operators): Παροχή του χώρο και των συσκευών φόρτισης.
* Πάροχοι Ηλεκτρικής Ενέργειας (Energy Providers): Data για δημιουργία τιμοκατάλογου.
* Διανομείς Ηλεκτρικής Ενέργειας (Energy Distributors): Αξιοποίηση online δεδομένων για καλύτερη κατανομή στο grid.

#### 3.1.1.2 Προϋποθέσεις εκτέλεσης

Οι προϋποθέσεις εκτέλεσης είναι οι εξής:

* Συμβατότητα αυτοκινήτου και ύπαρξη κατάλληλου πρωτοκόλλου.
* Διαθέσιμη παροχή ενέργειας
* Payment transaction completed
* Διαθέσιμη θέση φόρτισης

#### 3.1.1.3 Περιβάλλον εκτέλεσης

Το περιβάλλον εκτέλεσης είναι στα charging points, μέσα σε μηχάνημα διεπαφής χρήστη.

Απαραίτητη είναι η σύνδεση με ένα DBMS που περιέχει τα στοιχεία για όλους τους τύπους αυτοκινήτων και των αντίστοιχων πρωτοκόλλων διαχείρισης, για τις θέσεις φόρτισης (charging points), τους συνδεδεμένους παρόχους και διανομείς ηλεκτρικής ενέργειας αλλά και τις συνεργαζόμενες τράπεζες.

#### 3.1.1.4 Δεδομένα εισόδου

Δεδομένα Εισόδου:

* Ηλεκτρικό όχημα (EV) (Με προϋπόθεση συμβατότητας)
* Στοιχεία πληρωμής (Προϋπόθεση το Bank Verification )
* Online data από την βάση δεδομένων

#### 3.1.1.5 Παράμετροι

Παράμετροι για το EV: Μοντέλο οχήματος, Ιδιοκτήτης, Χωρητικότητα καυσίμου

Παράμετροι για τα στοιχεία πληρωμής: Συνεργαζόμενη τράπεζα, Στοιχεία λογαριασμού του

#### 3.1.1.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

*Βήμα 1:* Ο χρήστης (EV Owner) ζητάει να ξεκινήσει η διαδικασία φόρτισης του οχήματός του.

*Βήμα 2:* Το Charging Point Machine προσδιορίζει το είδος και τον τύπο του οχήματος.

*Βήμα 3:* Το Charging Point Machine ζητάει απ’ το Backend το πρωτόκολλο για τον τύπο οχήματος.

*Βήμα 4:* To Backend επιστρέφει στο Charging Point Machine το πρωτόκολλο (ή ότι δεν υπάρχει και απορρίπτεται το όχημα).

*Βήμα 5:* Το Charging Point Machine ζητάει απ’ το Backend τα προγράμματα χρέωσης.

*Βήμα 6:* Το Backend επιστρέφει τα προγράμματα χρέωσης στο Charging Point Machine.

*Βήμα 7:* Το Charging Point Machine παραθέτει στον χρήστη τα προγράμματα χρέωσης.

*Βήμα 8:* Το Charging Point Machine ζητάει από τον χρήστη να επιλέξει ένα από τα προγράμματα χρέωσης που του έδειξε.

*Βήμα 9:* Ο χρήστης επιλέγει ένα από τα προγράμματα.

*Βήμα 10:* Το Charging Point Machine υπολογίζει το συνολικό εκτιμώμενο κόστος της φόρτισης με βάση το costing plan που επιλέχθηκε.

*Βήμα 10:* Το Charging Point Machine ζητά από το χρήστη να διαλέξει μέθοδο πληρωμής (ανάμεσα σε κάρτα και μετρητά).

*Βήμα 11:* Ο χρήστης επιλέγει μέθοδο πληρωμής.

*Βήμα 12.α:* Το Charging Point Machine ζητά απ’ το Bank API επιβεβαίωση για την πληρωμή με κάρτα.

*Βήμα 12.α.1.α:* Το Bank API επιβεβαιώνει την κάρτα και το επιστρέφει στο Charging Point Machine.

*Βήμα 12.α.1.β:* Το Charging Point Machine επιβεβαιώνει την μέθοδο πληρωμής στον χρήστη.

*Βήμα 12.α.2.α:* Το Bank API απορρίπτει την κάρτα και το επιστρέφει στο Charging Point Machine.

*Βήμα 12.α.2.β:* Ξαναπάμε στο *Βήμα 10.*

*Βήμα 12.β.1:* Το Charging Point Machine επιβεβαιώνει ότι εισέπραξε το ολικό κόστος.

*Βήμα 12.β.2:* Το Charging Point Machine επιβεβαιώνει την μέθοδο πληρωμής στον χρήστη.

*Βήμα 13:* Το Charging Point Machine παρέχει ρεύμα στο όχημα.

*Βήμα 14:* Έχουμε υπολογισμό του πραγματικού χρόνου και κόστους. Σε περίπτωση που το πραγματικό κόστος είναι διαφορετικό από το ποσό που πληρώθηκε παίρνουμε τις κατάλληλες ενέργειες επιστροφής, χρέωσης.

*Βήμα 15:* Το Charging Point Machine αναλύει τα δεδομένα της φόρτισης, τα αποθηκεύει στην βάση δεδομένων για την χρήση σε έκδοση περιοδικού λογαριασμού και τα παρουσιάζει στον EV Owner.

#### 3.1.1.7 Δεδομένα εξόδου

Δεδομένα Εξόδου:

* Ποσοστό φόρτισης
* Συνολικό κόστος
* Κατανάλωση ενέργειας
* Χρόνος φόρτισης

### 3.1.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: Εύρεση Πλησιέστερων Σταθμών Φόρτισης και Χρόνου Αναμονής.

#### 3.1.2.1 Χρήστες (ρόλοι) που εμπλέκονται

Εμπλέκονται οι παρακάτω:

* Κάτοχοι Ηλεκτρικών οχημάτων (EV Owners)
* Διαχειριστές Σταθμών (Operators)

#### 3.1.2.2 Προϋποθέσεις εκτέλεσης

* Ενεργοποιημένη η τοποθεσία του EV Owner
* Να περιέχει η βάση δεδομένων μας τις τοποθεσίες των σταθμών φόρτισης εντός της περιοχής του EV Owner
* Συμβατότητα αυτοκινήτου με τον κάθε σταθμό και ύπαρξη κατάλληλου πρωτοκόλλου

#### 3.1.2.3 Περιβάλλον εκτέλεσης

Διαδικτυακή διεπαφή χρήστη - Εφαρμογή: Ο χρήστης (EV Owner) θα βλέπει σε ένα χάρτη τους πλησιέστερους σταθμούς σε αυτόν και τη διαθεσιμότητά τους. Αν κάποιος δεν είναι διαθέσιμος, θα φαίνεται ο εκτιμώμενος χρόνος μέχρι να γίνει.

Απαιτεί σύνδεση σε DBMS που έχει πληροφορίες για όλους τους σταθμούς.

#### 3.1.2.4 Δεδομένα εισόδου

* Η σύγχρονη τοποθεσία του EV owner
* Η σύγχρονη τοποθεσία των σταθμών
* Ο εκτιμώμενος χρόνος απελευθέρωσης charging point σε κάθε σταθμό
* login credentials με προϋπόθεση να είναι έγκυρα και να βρίσκονται στην βάση μας

#### 3.1.2.5 Παράμετροι

* login credentials: (username, password)

#### 3.1.2.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

**Login στο App:**

*Βήμα 1:* Ο χρήστης (EV Owner) ζητάει να συνδεθεί.

*Βήμα 2:* Το app ζητάει το username και το password του χρήστη.

*Βήμα 3:* Ο χρήστης απαντάει παρέχοντας το username και το password.

*Βήμα 4:* Το app τα αποδέχεται και συνεχίζεται η πλοήγηση ή τα απορρίπτει και ο χρήστης ξαναπηγαίνει στο login.

**Αναζήτηση Πλησιέστερων Σταθμών Φόρτισης και Χρόνου Αναμονής:**

*Βήμα 1:* Ο χρήστης ζητάει τους πλησιέστερους σταθμούς φόρτισης και την κατάστασή τους.

*Βήμα 2:* Το app ζητάει την τοποθεσία του χρήστη.

*Βήμα 3:* Ο χρηστής παρέχει την τοποθεσία του.

*Βήμα 4:* Το app ζητάει απ’ το map τον χάρτη της περιοχής που βρίσκεται ο χρήστης.

*Βήμα 5:* Επιστρέφεται στο app ο ζητούμενος χάρτης.

*Βήμα 6:* Το app ζητάει απ’ το Operators Database όλους τους σταθμούς που περιλαμβάνονται σε αυτό τον χάρτη.

*Βήμα 7:* Το Operators Database επιστρέφει τους ζητούμενους σταθμούς.

*Βήμα 8:* Το app ζητάει απ’ το Operators Database την κατάσταση (availability state) των σταθμών αυτών.

*Βήμα 9:* Το Operators Database επιστρέφει την κατάσταση των σταθμών.

*Βήμα 10:* Το app ζητάει απ’ το Operators Database τον εκτιμώμενο εναπομείναντα χρόνο απελευθέρωσης για τους σταθμούς που ήταν unavailable.

*Βήμα 10:* Το Operators Database επιστρέφει αυτούς τους χρόνους.

*Βήμα 11:* Το App παραθέτει στον χρήστη ως έξοδο τον χάρτη της περιοχής με τους σταθμούς και το availability state τους, καθώς και τους χρόνους αναμονής, όπου χρειάζονται.

#### 3.1.2.7 Δεδομένα εξόδου

* Ηλεκτρονικός χάρτης της περιοχής
* Η τοποθεσία των συμβατών με το όχημά μας σταθμών πάνω στον χάρτη
* Η κατάσταση κάθε σταθμού : Αν δεν είναι διαθέσιμος να φαίνεται ο εκτιμώμενος χρόνος απελευθέρωσης charging point σε κάθε σταθμό.

### 3.1.3 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 3: Διαχείριση Ενέργειας και Πλάνο Κατανάλωσης

#### 3.1.3.1 Χρήστες (ρόλοι) που εμπλέκονται

Για την διαχείριση κατανάλωσης ενέργειας εμπλέκονται οι παρακάτω ρόλοι:

* Διαχειριστές Σταθμών (Operators): Δημιουργία και δυνατότητα παρουσίασης δεδομένων. Επιλογή διαφορετικού πλάνου και provider
* Πάροχοι Ηλεκτρικής Ενέργειας (Energy Providers): Δημιουργία πλάνου κοστολόγησης και κατάλληλης διανομής ενέργειας.

#### 3.1.3.2 Προϋποθέσεις εκτέλεσης

* Χώρος στην βάση δεδομένων και στον δίσκο υπολογιστή
* Σωστή και επαρκή καταγραφή δεδομένων

#### 3.1.3.3 Περιβάλλον εκτέλεσης

* Διαδικτυακή διεπαφή για παρουσίαση δεδομένων ανάλυσης
* DBMS
* Εφαρμογή Βελτιστοποίησης smart grid

#### 3.1.3.4 Δεδομένα εισόδου

Άντληση από την βάση δεδομένων:

* Δεδομένα κατανάλωσης
* Δεδομένα υπάρχουσας κατάστασης του smart grid
* login credentials

#### 3.1.3.5 Παράμετροι

* Κατανάλωση: Ενέργεια κατανάλωσης, timestamp, charging point, charging station, ρυθμός κατανάλωσης, user.
* SmartGrid: τρέχον κατανάλωση (timestamp)
* login credentials: username, password (valid user)

#### 3.1.3.6 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

LogIn

*Βήμα 1:* Operator Request Login σε ένα WebApi.

*Βήμα 2:* Το WebAPI ζητάει τα credetetianl του operator.

*Βήμα 3:* Του παρέχει τα δεδομένα (username και password).

*Βήμα 4:* Το WebAPI ζητάει από ένα ValidUserAPI την επιβεβαίωση.

*Βήμα 5:* Ο operator δρομολογήσετε στην welcome page (ή ξανά στο login σε περίπτωση μη αποδοχής.

Παρουσίαση δεδομένων κατανάλωσης

*Βήμα 1:* Ο operator ζητάει από το webAPI να του δείξει τα δεδομένα

*Βήμα 2:* Το webAPI ζητάει από το businessLogicAPI τα δεδομένα.

*Βήμα 3:* Το businessLogicAPI ζητάει τα δεδομένα από την βάση και από το SmartGridAPI που έχει την κατάσταση του δικτύου.

*Βήμα 4:* Το businessLogicAPI δημιουργεί το επιχειρηματικό μοντέλο.

*Βήμα 5:* Δημιουργούνται τα διαγράμματα για απεικόνιση των δεδομένων.

*Βήμα 6:* Και τα παρέχει στο webAPI για προβολή.

Εύρεση καλύτερου πλάνου κατανάλωσης με βάση τους παρόχους

*Βήμα 1:* Ο operator ζητάει από το webAPI το καλύτερο πλάνο διαχείρισης της ενέργειας.

*Βήμα 2:* Το οποίο ζητά από το BusinessLogicAPI το καλύτερο πλάνο.

*Βήμα 3:* Το οποίο συλλέγει δεδομένα από την βάση δεδομένων και το smart grid.

*Βήμα 4:* Βρίσκει την καλύτερη δυνατή παροχή ενέργειας, με βάση την τιμοκαταλόγηση, την κατανάλωση και την κατάσταση του smart grid.

*Βήμα 5:* Το BusinessLogic ζητάει επιβεβαίωση του πλάνου από τον energy provider. Δέχεται ένα εναλλακτικό πλάνο σε περίπτωση που οι providers επιθυμούν και το ελέγχει.

*Βήμα 6:* Δίνει το πλάνο στο smartGridAPI για μία προσομοίωση και αξιολόγηση.

*Βήμα 7:* Επιστρέφει τα δεδομένα στο WebAPI το οποίο τα παρουσιάζει στον χρήστη.

#### 3.1.3.7 Δεδομένα εξόδου

* Δεδομένα κατανάλωσης
* Δεδομένα του Grid
* Δεδομένα βέλτιστης παροχής ενέργειας

## 3.2 Απαιτήσεις επιδόσεων

* Χαμηλή πιθανότητα δυσλειτουργίας
* Φιλικό προς τον χρήστη
* Ταχύτητα/Αποκρισιμότητα των διεπαφών
* Συμβατό με πολλές πλατφόρμες(Web,Android κλπ)
* Επεκτάσιμο/αναβαθμίσιμο
* Ασφάλεια από επιθέσεις
* Οθόνη αφής στα Charging Points
* Ακρίβεια στις μετρήσεις
* Αξιοπιστία
* Συνέπεια των δεδομένων σε κάθε σημείο
* Διαδραστικό
* Εύχρηστο σε κάθε είδους συσκευή (mobile, desktop κτλ)
* Ευέλικτο για μελλοντικές αναβαθμίσεις
* Δυναμικό (real time επεξεργασία των δεδομένων)
* Να μπορεί να χρησιμοποιείται από διαφορετικούς παρόχους
* Δυνατότητα διαχείρισης μεγάλου όγκου δεδομένων
* Ασύρματη συλλογή δεδομένων από οχήματα
* Μικρές απαιτήσεις σε hardware, Cloud Computing

## 3.3 Απαιτήσεις οργάνωσης δεδομένων

### 3.3.1 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα

Διαφορετικά Συστήματα διαχείρισης χρηστών για διαφορετικούς ρόλους. Access levels σε επίπεδο EV owner, Operator, Providers, Distributors. Ο κάθε χρήστης έχει διαφορετικό access permission στα δεδομένα.

3.4 Περιορισμοί σχεδίασης

Οι περιορισμοί θα προκύψουν σε μεγαλύτερο επίπεδο κατα την διαδικασία ανάπτυξης και ενασχόλησης με τα πιο συγκεκριμένα frameworks που θα χρησιμοποιηθούν. Η εφαρμογη θα αναπτυχθεί σε Java, HTML, CSS (bootstrap), θα γίνει χρήση της MySQL για database και Apache server. Για build automation θα χρησιμοποιήσουμε το Gradle καθώς και για τον έλεγχο το spock. Ως content management system χρησιμοποιήσουμε το git.

## 3.5 Λοιπές απαιτήσεις

### 3.5.1 Απαιτήσεις διαθεσιμότητας λογισμικού

Βασική απαίτηση είναι η συνεχής λειτουργία του συστήματος μας 24/7. Μέγιστη αποδεκτή απόκλιση από την κανονική λειτουργία είναι 30 λεπτά/Χρόνο. Με αυτόν τον τρόπο θα μπορούμε να διασφαλίσουμε την καθολική λειτουργία του συστήματος μας και την καλύτερη εξυπηρέτηση όλων των εμπλεκόμενων χρηστών.

### 3.5.2 Απαιτήσεις ασφάλειας

Η διαδικτυακή µας πλατφόρµα χρησιµοποιεί το πρωτόκολλο HTTPS, ώστε να σιγουρευτούµε πως τα δεδοµένα των χρηστών προστατεύονται από επιθέσεις. Επίσης, τα δεδοµένα της βάσης δεδοµένων απαιτούµε να είναι κρυφά προς τον έξω κόσµο και να µπορούν µόνο οι διαχειριστές να έχουν πρόσβαση σε αυτά. Ως επιπλέον θυρίδα ασφαλείας, χρησιµοποιούµε ταυτοποίηση µε hash authentication, ώστε να βεβαιωθούµε ότι δεν µπορεί να γίνει reverse engineering σε αυτή την τεχνική.

### 3.5.3 Απαιτήσεις συντήρησης

Σκοπός είναι να εφαρµόζονται συχνά εργασίες συντήρησης στους κύριους διακοµιστές της εφαρµογής µας, όπου θα χρησιµοποιούνται backup διακοµιστές, ώστε όντως να εξασφαλίζουµε την συνεχή και καθολική εξυπηρέτηση των χρηστών. Ταυτόχρονα, θα αναβαθμίζεται συνεχώς τόσο το λογισµικό όσο και το DBMS µας, ώστε να είμαστε συνεπείς στις απαιτήσεις της αγοράς.