Έγγραφο απαιτήσεων λογισμικού (SRS)

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

**TollTrax** – "Track, Manage, and Go"

# 1 Εισαγωγή

## 1.1 Εισαγωγή: Σκοπός του λογισμικού

Το συγκεκριμένο λογισμικό αποσκοπεί στη διαχείριση της διαλειτουργικότητας μεταξύ συστημάτων διοδίων διαφορετικών παρόχων, εξασφαλίζοντας την καταγραφή διελεύσεων, την αυτόματη εκκαθάριση οικονομικών οφειλών και την ανάλυση δεδομένων χρήσης. Το σύστημα υποστηρίζει τη λήψη δεδομένων από διαφορετικούς λειτουργούς διοδίων, την ασφαλή αποθήκευσή τους, τον υπολογισμό οικονομικών συμψηφισμών και τη διάθεση αναλύσεων σε ενδιαφερόμενα μέρη.

## 1.2 Διεπαφές (interfaces)

### 1.2.1 Διεπαφές με εξωτερικά συστήματα

Η διασύνδεση με εξωτερικά συστήματα πραγματοποιείται μέσω ενός REST API, το οποίο εξασφαλίζει τυποποιημένη επικοινωνία για την αποστολή και λήψη δεδομένων. Το API υποστηρίζει λειτουργίες όπως η επικύρωση πομποδεκτών, ο έλεγχος υπολοίπου, η καταγραφή διελεύσεων, και ο υπολογισμός οφειλών.

1.2.2 Διεπαφές με το χρήστη

Το λογισμικό διαθέτει δύο βασικές διεπαφές για αλληλεπίδραση με τον χρήστη, έπειτα από ταυτοποίηση. Η διαδικτυακή διεπαφή (WebUI) παρέχει ένα φιλικό περιβάλλον για την πρόσβαση σε λειτουργίες, όπως η διαχείριση δεδομένων, η εκτέλεση υπολογισμών, και η δημιουργία γραφημάτων και αναφορών. Η γραμμή εντολών (CLI) υποστηρίζει την επικοινωνία με το REST API, δίνοντας στους χρήστες τη δυνατότητα πρόσβασης σε δεδομένα και λειτουργίες μέσω εντολών.

# 2 Αναφορές - πηγές πληροφοριών

Αυτή η σελίδα είναι επίτηδες κενή.

# 3 Προδιαγραφές απαιτήσεων λογισμικού

## 3.1 Περιπτώσεις χρήσης

### 3.1.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: Admin Healthcheck

#### 3.1.1.1 Χρήστες (ρόλοι) που εμπλέκονται

Διαχειριστής (Admin): Πραγματοποιεί σύνδεση στο σύστημα και εκτελεί τη λειτουργία Healthcheck.

#### 3.1.1.2 Προϋποθέσεις εκτέλεσης

Ο Διαχειριστής έχει έγκυρα διαπιστευτήρια (username/password).

Το σύστημα υποστηρίζει την παραγωγή και επαλήθευση JWT token.

Υπάρχει πρόσβαση στο Admin Panel και στο αντίστοιχο κουμπί “Healthcheck”..

#### 3.1.1.3 Περιβάλλον εκτέλεσης

Frontend/Browser όπου ο Admin εισάγει τα στοιχεία του.

Backend που επικοινωνεί με το Database και πιστοποιεί τον χρήστη.

Admin Dashboard προσβάσιμο αφού επαληθευτεί το token.

#### 3.1.1.4 Δεδομένα εισόδου

Στοιχεία Σύνδεσης: username, password.

Αίτημα Healthcheck: Τα δεδομένα που ζητούνται να ελεγχθούν (π.χ. db\_connection, n\_stations, n\_tags, n\_passes).

#### 3.1.1.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Admin εισάγει τα διαπιστευτήρια (username/password) στο Login Frontend.

Το Login Frontend καλεί τη μέθοδο POST username and password προς το Backend.

Στο Backend εκτελείται η αναζήτηση στη βάση (Query executed) για τον συγκεκριμένο χρήστη:

* Αν δεν βρεθεί χρήστης: επιστρέφεται μήνυμα σφάλματος “incorrect credentials”.
* Αν βρεθεί χρήστης: παράγεται το JWT token.

To Backend επιστρέφει το JWT token και το Login Frontend το αποθηκεύει προσωρινά (store JWT token).

Το Login Frontend στέλνει αίτημα επαλήθευσης token στο Backend:

* Σε περίπτωση ληγμένου/μη έγκυρου token → εμφανίζεται μήνυμα “invalid or expired token”.
* Αν το token είναι έγκυρο → επιτρέπεται η πρόσβαση και γίνεται ανακατεύθυνση στο Analyst Dashboard.

Ο Admin επιλέγει στο Analyst Dashboard το Admin Panel και πατάει το κουμπί Healthcheck.

Γεννάται αίτημα Request db\_connection, n\_stations, n\_tags, n\_passes προς το Backend.

Το Backend κάνει τους απαραίτητους ελέγχους (OK Received) και ενημερώνει ότι επαλήθευσε επιτυχώς τα δεδομένα .

Τελικά εμφανίζεται στον Admin μήνυμα με τα αποτελέσματα (π.χ. “Display message with results”).

#### 3.1.1.7 Δεδομένα εξόδου

Μήνυμα επιτυχίας Healthcheck:

Μήνυμα σφάλματος: “incorrect credentials” ή “invalid/expired token” (σε περίπτωση αποτυχημένης σύνδεσης ή επαλήθευσης).

#### 3.1.1.8 Παρατηρήσεις

Αν ο Admin δεν έχει έγκυρο token, θα πρέπει να ξανακάνει login.

Τα στοιχεία που επιστρέφονται ως αποτέλεσμα του Healthcheck μπορεί να διαφέρουν ανάλογα με την υλοποίηση (π.χ. status σύνδεσης βάσης, αριθμός σταθμών, tags, passes κλπ.).

### 3.1.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: **Admin Add Passes**

#### 3.1.2.1 Χρήστες (ρόλοι) που εμπλέκονται

Διαχειριστής (Admin): Εκτελεί τη σύνδεση και επιλέγει να προσθέσει νέα passes.

#### 3.1.2.2 Προϋποθέσεις εκτέλεσης

Ο Admin έχει έγκυρα διαπιστευτήρια και ενεργό/έγκυρο JWT token.

Διαθέτει δικαιώματα διαχείρισης (πρόσβαση στο Admin Panel).

#### 3.1.2.3 Περιβάλλον εκτέλεσης

Web/Frontend περιβάλλον για την εισαγωγή δεδομένων και την προβολή αποτελεσμάτων.

Backend και Database που διαχειρίζονται τη λειτουργικότητα προσθήκης passes.

#### 3.1.2.4 Δεδομένα Εισόδου

Στοιχεία Σύνδεσης: username, password.

Πληροφορίες για τα νέα Passes (π.χ. αριθμός, χαρακτηριστικά).

#### 3.1.2.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Admin εισάγει διαπιστευτήρια στο Frontend.

Αποστέλλονται στο Backend (Post username and password).

Το Backend ελέγχει αν υπάρχει user record:

* Αν όχι → “incorrect credentials”.
* Αν ναι → παράγεται JWT token.

Επαλήθευση JWT token από το Backend (invalid ή valid).

Σε περίπτωση valid token → ανακατεύθυνση στο Analyst Dashboard.

Ο Admin επιλέγει Admin Panel και πατάει “Add Passes”.

Δημιουργείται αίτημα Add Passes προς το Backend.

Το Backend εκτελεί την καταχώριση (OK Added).

Εμφανίζεται μήνυμα επιτυχίας: “Passes added successfully. New passes: n\_newpasses”.

#### 3.1.2.6 Δεδομένα Εξόδου

Μήνυμα επιτυχίας με αναφορά στον αριθμό των νέων passes.

Ενημέρωση σφάλματος/αποτυχίας αν δεν προστεθούν τα passes ή αν το token είναι ληγμένο.

#### 

### 3.1.2.8 Παρατηρήσεις

* Το σύστημα ενδέχεται να επιστρέφει λεπτομερέστερη πληροφόρηση για τα passes (π.χ. IDs που δημιουργήθηκαν).
* Αν το token λήξει, ο Admin καλείται να συνδεθεί εκ νέου.

### 3.1.3 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 3: Admin Reset Passes

#### 3.1.3.1 Χρήστες (ρόλοι) που εμπλέκονται

Διαχειριστής (Admin): Μπορεί να εκτελέσει τη λειτουργία επαναφοράς (reset) όλων των passes.

#### 3.1.3.2 Προϋποθέσεις εκτέλεσης

* Έγκυρα διαπιστευτήρια και ενεργό JWT token.
* Δικαιώματα πρόσβασης στο Admin Panel.

#### 3.1.3.3 Περιβάλλον εκτέλεσης

Frontend (Login + Admin Panel).

Backend & Database, όπου τηρούνται τα δεδομένα των passes.

#### 3.1.3.4 Δεδομένα Εισόδου

Στοιχεία Σύνδεσης (username, password).

Εντολή Reset Passes: Δεν απαιτούνται συνήθως επιπρόσθετες παράμετροι, πέρα από το token που επαληθεύει τα δικαιώματα.

#### 3.1.3.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Ο Admin συνδέεται (αν δεν είναι ήδη συνδεδεμένος).

Το σύστημα ελέγχει την εγκυρότητα του token (invalid ή valid).

Αν το token είναι έγκυρο, ο Admin μεταβαίνει στο Analyst Dashboard.

Ο Admin επιλέγει το Admin Panel και πατάει το κουμπί “Reset Passes”.

Αποστέλλεται αίτημα Clear Passes προς το Backend.

Το Backend διαγράφει/μηδενίζει τα υπάρχοντα passes (OK Cleared).

Προαιρετικά, αν υπάρχει διαδικασία προσθήκης default passes, μπορεί να υλοποιείται εκεί.

Το σύστημα επιστρέφει μήνυμα επιτυχίας: “Passes reset successfully”.

#### 3.1.3.6 Δεδομένα Εξόδου

* Επιβεβαίωση: “Passes reset successfully”.
* Μήνυμα σφάλματος σε περίπτωση μη έγκυρου token ή αποτυχίας reset.

### 3.1.4 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 4:: Admin Reset Stations

#### 3.1.4.1 Χρήστες (ρόλοι) που εμπλέκονται

Διαχειριστής (Admin): Εκτελεί τη λειτουργία επαναφοράς (reset) των σταθμών (stations).

#### 3.1.4.2 Προϋποθέσεις εκτέλεσης

* Ο Admin διαθέτει έγκυρο token (JWT) μετά από επιτυχή σύνδεση.
* Υπάρχει υποστηριζόμενη λειτουργία reset για τους σταθμούς στην εφαρμογή (π.χ. διαγραφή/προσθήκη default).

#### 3.1.4.3 Περιβάλλον εκτέλεσης

Web/Frontend (Login, Admin Panel).

Backend & Database, όπου αποθηκεύονται στοιχεία σταθμών (stations).

#### 3.1.4.4 Δεδομένα Εισόδου

Στοιχεία Σύνδεσης: username, password.

Εντολή Reset Stations: Παρέχεται από το Admin Panel (button “Reset Stations”).

#### 3.1.4.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Ο Admin κάνει login (αν δεν είναι ήδη συνδεδεμένος) → το Backend παράγει/επικυρώνει JWT token.

Αν το token είναι έγκυρο, ο Admin αποκτά πρόσβαση στο Analyst Dashboard.

Ο Admin επιλέγει Admin Panel και πατά “Reset Stations”.

Δημιουργείται αίτημα Clear Toll Stations προς το Backend.

Το Backend διαγράφει (clear) τους υπάρχοντες σταθμούς και επιστρέφει “OK Cleared”.

Το Backend μπορεί να προσθέσει εκ νέου προεπιλεγμένους σταθμούς (Add Toll Stations) → “OK Added”.

Εμφανίζεται μήνυμα επιβεβαίωσης “Stations reset successfully”.

#### 3.1.4.6 Δεδομένα Εξόδου

* Μήνυμα επιτυχίας: “Stations reset successfully”.
* Ειδοποίηση σφάλματος σε περίπτωση αποτυχίας ή μη έγκυρου token.

### 3.1.5 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 5: Operator Debt Settlement

#### 3.1.5.1 Χρήστες (ρόλοι) που εμπλέκονται

Operator (Χειριστής): Χρήστης του συστήματος ο οποίος συνδέεται και χειρίζεται τις λειτουργίες για τον υπολογισμό/προβολή των οφειλών (debt settlements).

#### 3.1.5.2 Προϋποθέσεις εκτέλεσης

* Ο Operator διαθέτει έγκυρα διαπιστευτήρια (username/password).
* Έχει δημιουργηθεί και υποστηρίζεται από το σύστημα η ροή με χρήση JWT token (login, επαλήθευση token).
* Υπάρχει πρόσβαση (δικαιώματα) στο Operator Dashboard.

#### 3.1.5.3 Περιβάλλον εκτέλεσης

Login Frontend (web-based) για την εισαγωγή διαπιστευτηρίων και την παρουσίαση των αποτελεσμάτων.

Backend που διαχειρίζεται τη λογική ελέγχου διαπιστευτηρίων, παραγωγής/επικύρωσης token και υπολογισμού οφειλών.

Database στην οποία υπάρχουν τα απαραίτητα δεδομένα χρηστών και οικονομικών συναλλαγών.

Operator Dashboard, στο οποίο γίνεται η παρουσίαση των οφειλών και η διαχείριση τους από τον Operator

#### 3.1.5.4 Δεδομένα Εισόδου

Στοιχεία Σύνδεσης: username, password.

Εντολή/Αίτημα Υπολογισμού Οφειλών: Ενεργοποιείται μέσω του κουμπιού “Refresh Settlements” (ή παρόμοια επιλογή) στο Operator Dashboard.

#### 3.1.5.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

(Operator) Εισάγει διαπιστευτήρια στο Login Frontend.

Το Login Frontend αποστέλλει αίτημα POST username and password στο Backend για έλεγχο.

Στο Backend εκτελείται αναζήτηση στη Database για τον συγκεκριμένο χρήστη (Query executed):

* Αν δεν βρεθεί χρήστης, επιστρέφεται μήνυμα: “incorrect credentials”. Εμφανίζεται αντίστοιχο μήνυμα στο Frontend.
* Αν βρεθεί, το σύστημα παράγει JWT token.

Το Backend επιστρέφει το JWT token στο Login Frontend, όπου γίνεται store JWT token.

Ακολουθεί αίτημα request to verify token στο Backend:

* Αν το token είναι invalid ή expired, προβάλλεται μήνυμα “invalid or expired token”.
* Αν το token είναι valid, πραγματοποιείται ανακατεύθυνση στο Operator Dashboard (βήμα Redirect to Operator...).

Στο Operator Dashboard, ο Operator βλέπει τις διαθέσιμες λειτουργίες (π.χ. “Refresh Settlements”).

Ο Operator κλικάρει το κουμπί “Refresh Settlements” → αποστέλλεται αίτημα Calculate Debt… στο Backend.

Το Backend υπολογίζει τα σχετικά οικονομικά στοιχεία (OK Calculated) και τα επαληθεύει (Verified Debt...).

Επιστρέφονται τα αποτελέσματα στον Operator (π.χ. “Display Table with Debt…”), ώστε ο Operator να δει τις τρέχουσες οφειλές/ρυθμίσεις.

#### 3.1.5.6 Δεδομένα Εξόδου

* Μήνυμα σφάλματος (αν το token δεν είναι έγκυρο ή τα διαπιστευτήρια είναι λανθασμένα).
* Πίνακας Οφειλών ή συνοπτικές πληροφορίες σχετικά με το χρέος (Debt Settlements).
* Ενημέρωση “OK Calculated” στο Backend, που οδηγεί στην εμφάνιση των υπολογισμένων αποτελεσμάτων στο Frontend.

### 3.1.6 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 6: Operator Station Popularity

#### 3.1.6.1 Χρήστες (ρόλοι) που εμπλέκονται

Operator (Χειριστής): Χρησιμοποιεί το σύστημα με σκοπό τον υπολογισμό/επισκόπηση της “δημοφιλίας” σταθμών (stations).

#### 3.1.6.2 Προϋποθέσεις εκτέλεσης

* Ο Operator διαθέτει έγκυρα διαπιστευτήρια (username, password).
* Το σύστημα υποστηρίζει μηχανισμό JWT token για σύνδεση και επαλήθευση.
* Ο Operator έχει πρόσβαση στο Operator Dashboard μετά από επιτυχή σύνδεση.

#### 3.1.6.3 Περιβάλλον εκτέλεσης

Login Frontend (Web) για την εισαγωγή των διαπιστευτηρίων και την απεικόνιση των αποτελεσμάτων.

Backend που ελέγχει τις πληροφορίες σύνδεσης, παράγει/επαληθεύει token και υπολογίζει τη δημοφιλία των σταθμών.

Database που περιέχει τα απαιτούμενα δεδομένα (χρήστη, σταθμούς, κ.λπ.).

Operator Dashboard στο οποίο εμφανίζονται τα αποτελέσματα του υπολογισμού της δημοφιλίας των σταθμών.

#### 3.1.6.4 Δεδομένα Εισόδου

Στοιχεία Σύνδεσης: username, password.

Αίτημα Υπολογισμού Δημοφιλίας (από το κουμπί “Load Station Popularity” ή παρόμοια λειτουργία), το οποίο υποδεικνύει στο Backend να κάνει τους σχετικούς υπολογισμούς.

#### 3.1.6.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

(Operator) Εισάγει τα διαπιστευτήρια στο Login Frontend (username, password).

Το Login Frontend στέλνει αίτημα Post username and password προς το Backend.

Στο Backend εκτελείται αναζήτηση στη Database (Query executed) για την εξακρίβωση των στοιχείων του χρήστη:

* Αν δεν βρεθεί χρήστης: incorrect credentials επιστρέφεται, εμφανίζεται μήνυμα στο Frontend.
* Αν βρεθεί χρήστης: παράγεται JWT token.

Το Backend επιστρέφει το token στο Login Frontend, όπου γίνεται store JWT token.

Αποστέλλεται αίτημα request to verify token στο Backend:

* Αν το token είναι invalid/expired, εμφανίζεται μήνυμα “invalid or expired token”.
* Αν είναι valid, το σύστημα κάνει Redirect to Operator Dashboard.

Στο Operator Dashboard, ο Operator βλέπει τις διαθέσιμες επιλογές.

Ο Operator κάνει κλικ στο κουμπί “Load Station Popularity”.

Δημιουργείται αίτημα Calculate Station Popularity προς το Backend, ώστε να υπολογιστεί η δημοφιλία των σταθμών με βάση τα δεδομένα της βάσης.

Το Backend ολοκληρώνει τον υπολογισμό (OK Calculated) και επιστρέφει τα αποτελέσματα (π.χ. Verified Station Popularity).

Το Frontend εμφανίζει στον Operator έναν πίνακα με την “Station Popularity” (π.χ. λίστα σταθμών, αριθμό διελεύσεων ή όποια κριτήρια ορίζουν τη δημοφιλία).

#### 3.1.6.6 Δεδομένα Εξόδου

* Μήνυμα επιτυχίας ή επιβεβαίωση ότι ο υπολογισμός ολοκληρώθηκε (“OK Calculated”, “Verified Station Popularity”).
* Πίνακας με Station Popularity (με κριτήρια κατάταξης/αξιολόγησης).
* Μήνυμα σφάλματος (αν δεν επαληθευτεί το token ή αν τα διαπιστευτήρια είναι λανθασμένα).

### 3.1.7 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 7: Operator Station Popularity

#### 3.1.7.1 Χρήστες (ρόλοι) που εμπλέκονται

Operator (Χειριστής): Συνδέεται στο σύστημα και εκτελεί τη λειτουργία υπολογισμού/προβολής κατάταξης οχημάτων (vehicle model rankings).

#### 3.1.7.2 Προϋποθέσεις εκτέλεσης

* Ο Operator διαθέτει έγκυρα διαπιστευτήρια (username/password).
* Υπάρχει υποστήριξη για τον μηχανισμό JWT token (έκδοση & επαλήθευση).
* Ο Operator έχει δικαίωμα πρόσβασης στο Operator Dashboard, όπου περιλαμβάνεται η συγκεκριμένη λειτουργία.

#### 3.1.7.3 Περιβάλλον εκτέλεσης

Login Frontend (ιστοσελίδα/εφαρμογή) για την εισαγωγή διαπιστευτηρίων.

Backend που ελέγχει τα στοιχεία σύνδεσης, επαληθεύει/παράγει το JWT token και υπολογίζει τα rankings των οχημάτων.

Database όπου αποθηκεύονται οι πληροφορίες για χρήστες, οχήματα κ.λπ.

Operator Dashboard για την παρουσίαση των αποτελεσμάτων στον χειριστή.

#### 3.1.7.4 Δεδομένα Εισόδου

Στοιχεία σύνδεσης: username, password.

Αίτημα Υπολογισμού Κατάταξης Οχημάτων (π.χ. “Load Vehicle Model Rankings”), που υποδεικνύει στο Backend να εκτελέσει την ανάλυση/υπολογισμό.

#### 3.1.7.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

1. Ο Operator πληκτρολογεί το username και το password στο Login Frontend.
2. Το Login Frontend στέλνει αίτημα POST username and password στο Backend για έλεγχο.
3. Στο Backend γίνεται έρευνα στη Database (Query executed) για τον συγκεκριμένο χρήστη:
   * Αν δεν βρεθεί χρήστης, επιστρέφεται “incorrect credentials” και εμφανίζεται μήνυμα σφάλματος.
   * Αν βρεθεί, δημιουργείται το JWT token.
4. Ο Backend επιστρέφει το token στο Login Frontend, όπου γίνεται store JWT token.
5. Ακολουθεί αίτημα request to verify token προς το Backend:
   * Αν το token είναι invalid/expired, εμφανίζεται μήνυμα “invalid or expired token”.
   * Αν το token είναι valid, γίνεται ανακατεύθυνση (Redirect to Operator...) στο Operator Dashboard.
6. Ο Operator εντός του Operator Dashboard επιλέγει το κουμπί “Load Vehicle Model Rankings”.
7. Δημιουργείται αίτημα Calculate Vehicle Model προς το Backend για να υπολογιστούν τα rankings.
8. Το Backend ολοκληρώνει τον υπολογισμό (OK Calculated) και επιστρέφει αναλυτικές πληροφορίες (π.χ. “Verified Vehicle Model”).
9. Το Frontend εμφανίζει πίνακα κατάταξης (Display Table with Vehicle Model), παρέχοντας στον Operator τις σχετικές πληροφορίες (π.χ. κατάταξη βάσει χρήσης, συχνότητας διελεύσεων κ.λπ.).

#### 3.1.7.6 Δεδομένα Εξόδου

* Μήνυμα σφάλματος: “incorrect credentials” ή “invalid/expired token” σε περίπτωση αποτυχημένης σύνδεσης ή μη έγκυρου token.
* Πίνακας Κατάταξης Οχημάτων: Με βάση τα υπολογισμένα στοιχεία (π.χ. πιο δημοφιλή/χρησιμοποιούμενα μοντέλα).
* Μήνυμα επιτυχίας για τον υπολογισμό (π.χ. “OK Calculated”, “Verified Vehicle Model”).

### 3.1.7.7 Απαιτήσεις διαθεσιμότητας λογισμικού

Το λογισμικό πρέπει να είναι διαθέσιμο 24/7 με ελάχιστους χρόνους διακοπής σε περιπτώσεις συντήρησης.

### 3.1.7.8 Απαιτήσεις ασφάλειας

Η πρόσβαση επιτρέπεται μόνο σε ταυτοποιημένους χρήστες μέσω έγκυρων διαπιστευτηρίων, ενώ όλες οι ενέργειες καταγράφονται για λόγους ασφαλείας. Τα δεδομένα των χρηστών προστατεύονται αυστηρά και διατίθενται μόνο σε εξουσιοδοτημένους φορείς. Επιπλέον, η επικοινωνία κρυπτογραφείται μέσω του πρωτοκόλλου HTTPS, μειώνοντας σημαντικά τους κινδύνους μη εξουσιοδοτημένης πρόσβασης.

#### 3.1.8 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 8 : PASSES

#### 3.1.8.1 Χρήστες που εμπλέκονται

* Normal\_User
* Operator
* Admin

#### 3.1.8.2 Προϋποθέσεις εκτέλεσης

* Ο χρήστης διαθέτει ήδη έγκυρο όνομα χρήστη και κωδικό πρόσβασης .
* Το σύστημα είναι σε λειτουργία και διαθέσιμο.
* Υπάρχει αποθηκευμένο τουλάχιστον ένα έγκυρο “user record” στη βάση δεδομένων.

#### 3.1.8.3 Περιβάλλον εκτέλεσης

* Web browser από την πλευρά του χρήστη.
* Web server/Backend όπου τρέχει η εφαρμογή.
* Βάση δεδομένων η οποία είναι προσβάσιμη από το Backend.

#### 3.1.8.4 Δεδομένα εισόδου

* Όνομα χρήστη : Συμβολοσειρά κειμένου.
* Κωδικός πρόσβασης :

#### 3.1.8.5 Αλληλουχία ενεργειών – επιθυμητή συμπεριφορά

1. Ο χρήστης (οποιοσδήποτε από τους ρόλους: Normal\_User//Operator/Admin) ανοίγει τη σελίδα σύνδεσης και εισάγει τα διαπιστευτήριά του (username, password).
2. Το Login Frontend αποστέλλει τα διαπιστευτήρια στο Backend
3. Το Backend ελέγχει τη βάση δεδομένων :
   1. Αν δεν βρεθεί αντιστοιχία, επιστρέφεται μήνυμα σφάλματος , και το Frontend εμφανίζει στον χρήστη μήνυμα “Λάθος όνομα χρήστη ή κωδικός πρόσβασης”.
   2. Αν βρεθεί αντιστοιχία , το Backend δημιουργεί ένα JWTtoken.
4. Το Frontend λαμβάνει το JWT token, το αποθηκεύει και κατόπιν στέλνει αίτημα για επικύρωση του token στο Backend .
5. Το Backend ελέγχει το token:
   1. Αν είναι λήγον ή μη έγκυρο , επιστρέφεται σφάλμα και το Frontend εμφανίζει σχετικό μήνυμα .
   2. Αν είναι έγκυρο , το σύστημα ανακατευθύνει τον χρήστη στη σελίδα “User Dashboard” .
6. Από τη σελίδα User Dashboard, ο χρήστης μπορεί να προχωρήσει σε διάφορες ενέργειες (π.χ. “Click on Total Station Button” / “Pass Analysis” / “Pass Cost”), οπότε ακολουθεί η ροή:
   1. Γίνεται GET request προς το Backend.
   2. Το Backend εκτελεί SQL query στη βάση .
   3. Επιστρέφει τα αποτελέσματα .
   4. Το Frontend λαμβάνει τα δεδομένα σε JSON.

#### 3.1.8.6 Δεδομένα εξόδου

* Μήνυμα σφάλματος σε περίπτωση λανθασμένων διαπιστευτηρίων ή μη έγκυρου token.
* JWT token σε περίπτωση επιτυχημένης σύνδεσης.
* JSON δεδομένα για την εμφάνιση στο Dashboard ύστερα από ενέργειες του χρήστη (π.χ. κλικ σε κουμπιά “Total Station”, “Pass Analysis”, “Pass Cost” κ.λπ.).

#### 3.1.8.7 Παρατηρήσεις

* Σε περίπτωση που το token λήξει ή καταστεί μη έγκυρο κατά τη διάρκεια της περιήγησης, ο χρήστης θα πρέπει να ανακατευθυνθεί στη σελίδα σύνδεσης ή να του εμφανιστεί σχετικό μήνυμα.
* Τα δεδομένα του χρήστη ενδέχεται να προσδιορίζουν ποιες λειτουργίες έχει δικαίωμα να εκτελέσει μέσα στο σύστημα.
* Το Frontend οφείλει να διαχειρίζεται με ασφάλεια το JWT token.

### 3.1.9 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 9: Διαδικασία Εισόδου Χρήστη και Εμφάνισης Στοιχείων

#### 3.1.9.1 Χρήστες (ρόλοι) που εμπλέκονται

* Τελικός Χρήστης: Πραγματοποιεί είσοδο στο σύστημα εισάγοντας διαπιστευτήρια (username, password).
* Login Frontend: Η διεπαφή (π.χ. web/mobile) μέσω της οποίας ο χρήστης εισάγει τα στοιχεία του.
* Backend: Ελέγχει τα διαπιστευτήρια, παράγει και επαληθεύει το JWT token, και παρέχει τις βασικές λειτουργίες ελέγχου.
* Database: Περιέχει τα στοιχεία των χρηστών (username, password) και τα δεδομένα για το υπόλοιπο και το ιστορικό του χρήστη.
* Operator Dashboard: Ενδέχεται να αποτελεί το περιβάλλον/σελίδα όπου εμφανίζονται τα αποτελέσματα (υπόλοιπο, ιστορικό κτλ.) αφού ο χρήστης συνδεθεί.

#### 3.1.9.2 Προϋποθέσεις εκτέλεσης

1. Ο χρήστης διαθέτει ήδη έγκυρα διαπιστευτήρια (username, password) στο σύστημα.
2. Το σύστημα (Frontend, Backend, Βάση Δεδομένων) είναι σε λειτουργία και επικοινωνούν κανονικά.
3. Υπάρχει διαθέσιμη σύνδεση δικτύου για την επικοινωνία με τον διακομιστή (backend).

#### 3.1.9.3 Περιβάλλον εκτέλεσης

* Η εφαρμογή εκτελείται σε περιβάλλον web ή mobile, όπου ο χρήστης έχει πρόσβαση στον Login Frontend.
* Ο Backend φιλοξενείται σε διακομιστή που έχει πρόσβαση στη βάση δεδομένων.

#### 3.1.9.4 Δεδομένα εισόδου

1. Διαπιστευτήρια Χρήστη:
   * Όνομα Χρήστη (Username)
   * Κωδικός Πρόσβασης (Password)
2. JWT Token (σε μεταγενέστερο στάδιο, όταν ο χρήστης έχει συνδεθεί επιτυχώς και κάνει επόμενες αιτήσεις).

#### 3.1.9.5 Αλληλουχία ενεργειών - Επιθυμητή συμπεριφορά

Με βάση το sequence diagram:

1. Ο Χρήστης εισάγει διαπιστευτήρια στο Login Frontend.
2. Το Login Frontend καλεί τη μέθοδο για έλεγχο (π.χ. POST /login) στο Backend μεταβιβάζοντας το username και το password.
3. Το Backend εκτελεί query στη Database για να ελέγξει αν υπάρχει ο συγκεκριμένος χρήστης:
   1. Εάν δεν βρεθεί εγγραφή χρήστη, επιστρέφεται μήνυμα σφάλματος «Λανθασμένα διαπιστευτήρια».
   2. Εάν βρεθεί, γίνεται επεξεργασία του κωδικού πρόσβασης και δημιουργείται JWT token.
4. Επιτυχής Σύνδεση:
   1. Το Backend επιστρέφει το JWT token στο Login Frontend.
   2. Το Frontend αποθηκεύει το token (π.χ. local storage, session storage) και στέλνει αίτημα για να επαληθεύσει το token (verify token).
   3. Το Backend επαληθεύει το token:
      * Αν το token είναι έγκυρο, επιστρέφει ένδειξη “Valid Token”.
      * Αν το token είναι μη έγκυρο ή ληγμένο, επιστρέφει ένδειξη “Invalid Token”.
5. Μη έγκυρο Token:
   1. Το Login Frontend προβάλλει μήνυμα «μη έγκυρο ή ληγμένο token» και ενδεχομένως ζητά εκ νέου σύνδεση.
6. Έγκυρο Token:
   1. Το Login Frontend ανακατευθύνει τον χρήστη στην επόμενη σελίδα/περιβάλλον (π.χ. Operator Dashboard).
7. Ανάκτηση Υπολοίπου & Ιστορικού (αφού ο χρήστης είναι συνδεδεμένος):
   1. Ο χρήστης επιλέγει “Get User Balance & History”.
   2. Το Frontend στέλνει αίτημα προς το Backend (μαζί με το αποθηκευμένο token).
   3. Το Backend επαληθεύει το token, υπολογίζει το υπόλοιπο και αντλεί το ιστορικό από τη Database.
   4. Το Backend επιστρέφει τα υπολογισμένα αποτελέσματα (Verified User Balance & History).
   5. Το Frontend εμφανίζει τα δεδομένα στον χρήστη (π.χ. στο Operator Dashboard)

#### 3.1.9.7 Δεδομένα εξόδου

1. Μήνυμα επιτυχούς ή μη επιτυχούς σύνδεσης (π.χ. «Login successful» ή «Invalid username/password»).
2. JWT token (σε περίπτωση επιτυχούς εισόδου).
3. Μηνύματα σφάλματος σε περίπτωση ληγμένου/μη έγκυρου token.
4. Πληροφορίες υπολοίπου και ιστορικού συναλλαγών του χρήστη (μετά τη σύνδεση).

#### 3.1.9.8 Παρατηρήσεις

* Το JWT token έχει ημερομηνία λήξης (expiry). Εάν λήξει, απαιτείται εκ νέου σύνδεση.
* Η εφαρμογή θα πρέπει να αποθηκεύει με ασφαλή τρόπο το token (π.χ. HTTPS, secure storage).
* Η ασφάλεια του κωδικού πρόσβασης διασφαλίζεται από το backend (hashing/salting).
* Σε σενάρια όπου αποτυγχάνει συχνά η επαλήθευση, ενδέχεται να εφαρμοστούν επιπλέον μέτρα ασφαλείας (π.χ. lockout).

## 3.2 Απαιτήσεις επιδόσεων

Το λογισμικό θα πρέπει να διαχειρίζεται την ταυτόχρονη διέλευση χρηστών από πολλαπλά σημεία διέλευσης, χωρίς να εξαντλούνται οι πόροι του και διατηρώντας ικανοποιητικούς χρόνους απόκρισης.

Η εκκαθάριση οφειλών μεταξύ εταιρειών να γίνεται εβδομαδιαία, χωρίς καθυστερήσεις.

Το σύστημα θα πρέπει να ανταποκρίνεται στις ανάγκες ανάλυσης μεγάλων όγκων δεδομένων με ακρίβεια, υποστηρίζοντας τους ενδιαφερόμενους φορείς σε πραγματικό χρόνο.

## 3.3 Απαιτήσεις οργάνωσης δεδομένων

### 3.3.1 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα

Το λογισμικό εφαρμόζει πολιτικές πρόσβασης, διασφαλίζοντας ότι κάθε χρήστης έχει πρόσβαση μόνο στα δεδομένα που σχετίζονται με τις αρμοδιότητές του. Οι οδηγοί μπορούν να δουν πληροφορίες που αφορούν τα δικά τους οχήματα, διελεύσεις, και πομποδέκτες, ενώ οι πάροχοι έχουν πρόσβαση στα δεδομένα των χρηστών τους, στους σταθμούς διοδίων που διαχειρίζονται, και στις οικονομικές συναλλαγές με άλλους παρόχους. Οι αναλυτές έχουν περιορισμένη πρόσβαση σε δεδομένα διελεύσεων και οχημάτων για την εξαγωγή στατιστικών, με όλες τις ενέργειες να καταγράφονται.

Η πρόσβαση στα δεδομένα βασίζεται σε συγκεκριμένες σχέσεις οντοτήτων:

* Κάθε χρήστης συνδέεται με πολλαπλά οχήματα, και κάθε όχημα μπορεί να διαθέτει έναν ή περισσότερους πομποδέκτες.
* Οι διελεύσεις συσχετίζονται με έναν πομποδέκτη και έναν σταθμό διοδίων, ενώ οι πάροχοι διαχειρίζονται πολλούς σταθμούς, διασφαλίζοντας τη λειτουργία του δικτύου.
* Οι οικονομικοί διακανονισμοί συνδέουν παρόχους, καταγράφοντας χρηματοοικονομικές συναλλαγές, ενώ οι αναλυτές έχουν εξειδικευμένη πρόσβαση σε οχήματα και διελεύσεις με σχέση 1:Ν.

Το σύστημα ενισχύει την ασφάλεια των δεδομένων με κρυπτογράφηση HTTPS και έλεγχο ταυτότητας χρηστών.

## 3.5 Λοιπές απαιτήσεις

### 3.5.1 Απαιτήσεις διαθεσιμότητας λογισμικού

Το λογισμικό πρέπει να είναι διαθέσιμο 24/7 με ελάχιστους χρόνους διακοπής σε περιπτώσεις συντήρησης.

### 3.5.2 Απαιτήσεις ασφάλειας

Η πρόσβαση επιτρέπεται μόνο σε ταυτοποιημένους χρήστες μέσω έγκυρων διαπιστευτηρίων.