Έγγραφο απαιτήσεων λογισμικού (SRS)

**Πληροφοριακό συστήματα διοδίων**

**αυτοκινητοδρόμων**

**ΕΚΔΟΣΗ 1.0.0 (final)**

# Εισαγωγή

Το παρόν έγγραφο αποτελεί την Προδιαγραφή Απαιτήσεων Λογισμικού (SRS) για την ανάπτυξη μιας ιστοεφαρμογής Διαχείρισης και Ανάλυσης διελεύσεων διοδίων. Η εφαρμογή σχεδιάζεται με σκοπό να προσφέρει στους χρήστες τη δυνατότητα αποτύπωσης των διοδίων σε πανελλαδικό χάρτη για όλους τους χρήστες, προβλέψεων με χρήση Μηχανικής Μάθησης για κάθε εταιρεία ξεχωριστά και αλγόριθμο απαλοιφής περιττών συναλλαγών μεταξύ των εταιρειών.

Η εισαγωγή παρέχει μια επισκόπηση των στόχων, του πεδίου εφαρμογής, των παραδοχών και των περιορισμών του συστήματος. Επίσης, αναλύει τις βασικές απαιτήσεις του λογισμικού και παρέχει ένα πλαίσιο για την κατανόηση του περιεχομένου και της δομής του εγγράφου.

### Σκοπός

Ο σκοπός του παρόντος εγγράφου είναι να παρέχει μια λεπτομερή περιγραφή των λειτουργικών και μη λειτουργικών απαιτήσεων της ιστοεφαρμογής. Επιδιώκει να διασφαλίσει ότι όλοι οι συμμετέχοντες στο έργο έχουν μια κοινή κατανόηση των στόχων και των προδιαγραφών του λογισμικού.

### 1.1.2 Πεδίο Εφαρμογής

Η ιστοεφαρμογή θα προσφέρει τις εξής βασικές λειτουργίες:

* + 1. Δυνατότητα σύνδεσης των εταιρειών με ειδικούς λογαριασμούς.
    2. Αποτύπωση διοδίων στον χάρτη με χρήση φίλτρων και περαιτέρω πληροφοριών για κάθε τοποθεσία.
    3. Προβλέψεις για τις διελεύσεις στα διόδια.
    4. Ανάλυσης στατιστικών δεδομένων για κάθε εταιρεία ξεχωριστά.
    5. Upload διελεύσεων και εκτέλεση αλγορίθμου απαλοιφής περιττών συναλλαγών.

### 1.1.3 Παραδοχές και Περιορισμοί

1. Αυτό το έγγραφο στηρίζεται στην υπόθεση ότι η ανάπτυξη της εφαρμογής θα γίνει με τη χρήση σύγχρονων τεχνολογιών διαδικτύου, εξασφαλίζοντας την προσβασιμότητά της από διάφορες συσκευές και προγράμματα περιήγησης. Επιπλέον, λαμβάνει υπόψη περιορισμούς όπως ο διαθέσιμος προϋπολογισμός, το χρονοδιάγραμμα του έργου και οι υπάρχοντες πόροι.

## 1.2 Διεπαφές (interfaces)

### 1.2.1 Διεπαφές με εξωτερικά συστήματα

## 1.2.1.1 Βάση Δεδομένων (Database System)

Το σύστημα χρησιμοποιεί **SQL βάση δεδομένων** για την αποθήκευση των δεδομένων.

### **Δεδομένα που αποθηκεύονται:**

* **Πληροφορίες Σταθμών Διοδίων** (όνομα, συντεταγμένες, εταιρεία).
* **Οικονομικές Συναλλαγές** (οφειλές μεταξύ εταιρειών).
* **Ιστορικά Δεδομένα Κυκλοφορίας** (διελεύσεις οχημάτων, ώρες αιχμής).
* **Δεδομένα Προβλέψεων ML** (αποτελέσματα προπονήσεων μοντέλων).
* **Διαχείριση Χρηστών** (λογαριασμοί, ρόλοι χρηστών)

## 1.2.1.2 Σύστημα Αυθεντικοποίησης (Authentication Server)

Το σύστημα απαιτεί οι χρήστες να συνδέονται μέσω **JWT Authentication**.

### **Διαδικασία Ταυτοποίησης:**

1. Ο χρήστης εισάγει τα διαπιστευτήριά του.
2. Το αίτημα αποστέλλεται στον **Authentication Server**.
3. Αν τα διαπιστευτήρια είναι σωστά, επιστρέφεται ένα **JWT Token**.
4. Το JWT χρησιμοποιείται για την ασφαλή πρόσβαση στα endpoints.

## 1.2.1.3 Ενσωμάτωση Με Εξωτερικούς Χάρτες (Leaflet.js / OpenStreetMap)

Το σύστημα χρησιμοποιεί **διαδραστικό χάρτη** για την προβολή διοδίων.

| **Υπηρεσία** | **Περιγραφή** |
| --- | --- |
| OpenStreetMap | Χρήση δωρεάν χαρτογραφικών δεδομένων για εμφάνιση διοδίων |
| Leaflet.js API | Χειρισμός διαδραστικών markers για κάθε σταθμό διοδίων |

### 1.2.2 Διεπαφές με το χρήστη

Η εφαρμογή μας αποτελείται από δύο κύρια υποσυστήματα, που επιτρέπουν στα εμπλεκόμενα μέλη να αλληλεπιδρούν με το σύστημα: ένα Command Line Interface (CLI) και ένα Frontend Web Application. Κάθε μία από αυτές τις διεπαφές υποστηρίζει μια ποικιλία λειτουργιών επεξεργασίας και προβολής των δεδομένων που παρέχονται από τον κύριο server του λογισμικού. Επιπλέον, τρίτες εφαρμογές έχουν τη δυνατότητα να αξιοποιήσουν απευθείας τις λειτουργίες του API μας, στέλνοντας αιτήματα στα αντίστοιχα endpoints.

# Αναφορές - πηγές πληροφοριών

# Προδιαγραφές απαιτήσεων λογισμικού

## 3.1 Περιπτώσεις χρήσης

Ο παρών ιστότοπος έχει 3 περιπτώσεις χρήσης, όπως περιγράφηκε παραπάνω. Σύνδεση εταιρειών, upload διελεύσεων και απαλοιφή συναλλαγών, προβλέψεις με ML, αποτύπωση διοδίων στο χάρτη με χρήση φίλτρων και παρουσίαση πληροφοριών για αυτά.

### 3.1.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: Διαχείριση Οφειλών μεταξύ Εταιρειών Διοδίων

#### 3.1.1.1 Χρήστες (ρόλοι) που εμπλέκονται

* Χρήστης (Εγγεγραμμένος χρήστης)
* Διαχειριστής συστήματος (Admin, για ακύρωση οφειλών)

#### 3.1.1.2 Προϋποθέσεις εκτέλεσης

* Ο χρήστης πρέπει να είναι συνδεδεμένος στο σύστημα.
* Η βάση δεδομένων πρέπει να περιέχει ενημερωμένες πληροφορίες για οφειλές.
* Ο διακομιστής API πρέπει να είναι λειτουργικός και να παρέχει δεδομένα.

#### 3.1.1.3 Περιβάλλον εκτέλεσης

* + Διαδικτυακή εφαρμογή βασισμένη σε React.js.
  + Επικοινωνία με REST API για ανάκτηση και διαχείριση δεδομένων.
  + Συμβατότητα με μοντέρνους browsers (Chrome, Firefox, Edge, Safari).

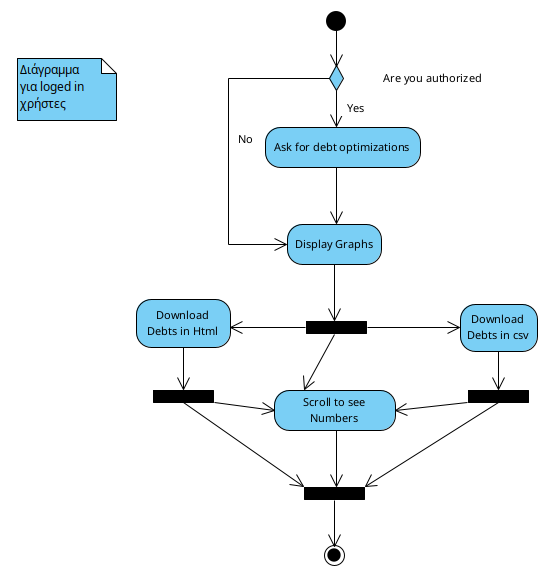
#### 3.1.1.4 Δεδομένα εισόδου

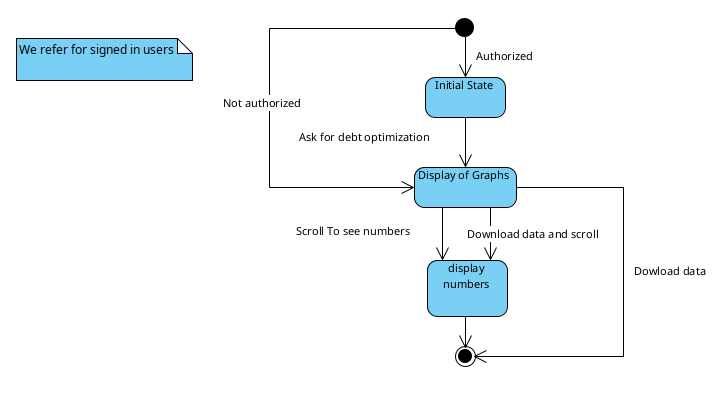
* Token ταυτοποίησης χρήστη (JWT).
* Αίτημα για βελτιστοποίηση οφειλών μέσω API.
* Αίτημα για βελτιστοποίηση οφειλών (μόνο για διαχειριστή).
* Επιλογή μορφής αρχείου για λήψη δεδομένων (CSV ή HTML).

#### 3.1.1.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

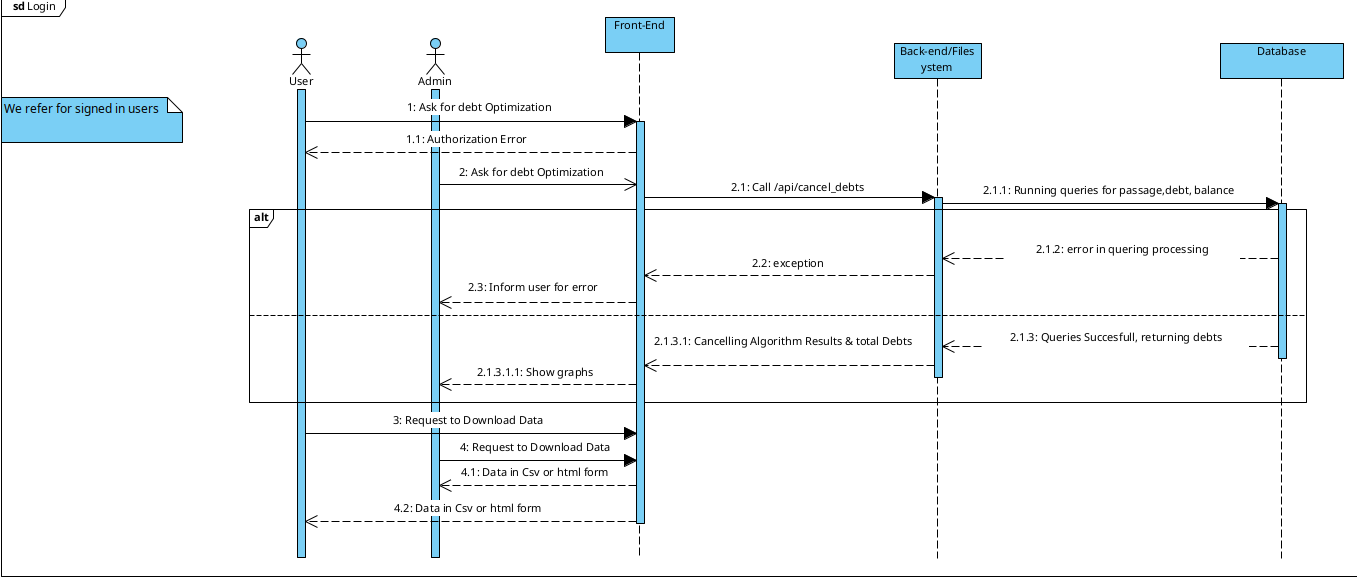
1. Ο χρήστης συνδέεται και ελέγχεται η ταυτότητά του μέσω JWT.
2. Το σύστημα ανακτά τα δεδομένα οφειλών από το API.
3. Οι αρχικές και βελτιστοποιημένες οφειλές εμφανίζονται σε μορφή HTML.
4. Ο χρήστης μπορεί να:
   1. **Πραγματοποιήσει λήψη δεδομένων** σε μορφή HTML ή CSV.
   2. **Κάνει scroll για να δει τις λεπτομέρειες των αριθμητικών δεδομένων.**
5. Αν ο χρήστης είναι **Admin**, μπορεί να ακυρώσει οφειλές.
6. Εφόσον ολοκληρωθεί η ακύρωση, ενημερώνεται η σελίδα με τα νέα δεδομένα.
7. Αν προκύψει σφάλμα (π.χ. μη εξουσιοδοτημένο αίτημα), εμφανίζεται μήνυμα στον χρήστη.

Activity Diagram:



State:

## Sequence:



#### 3.1.1.7 Δεδομένα εξόδου

Δεδομένα που επιστρέφονται από το API και προβάλλονται στον χρήστη:

* JSON με πληροφορίες οφειλών (αρχικές και βελτιστοποιημένες τιμές).
* HTML με δομημένη αναπαράσταση των δεδομένων.
* CSV αρχείο με τους οφειλέτες, τους δανειστές και τα ποσά.

#### 3.1.1.8 Παρατηρήσεις

* Οι χρήστες που δεν είναι διαχειριστές δεν μπορούν να ακυρώσουν οφειλές.
* Να υπάρχει fallback σε περίπτωση αποτυχίας του API (π.χ., μήνυμα σφάλματος).
* Να γίνεται αυτόματος έλεγχος λήξης του JWT και ανακατεύθυνση στη σελίδα σύνδεσης αν χρειαστεί.
* Η διαδικασία ακύρωσης οφειλών πρέπει να εκτελείται μόνο μέσω **PATCH** request για την ασφάλεια των δεδομένων.

### 3.1.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: Πλοήγηση και Ανάλυση Χάρτη Διοδίων

### 3.1.2.1 Χρήστες (Ρόλοι) που Εμπλέκονται

* Χρήστης (Επισκέπτης/Εγγεγραμμένος/Διαχειριστης)

### 3.1.2.2 Προϋποθέσεις Εκτέλεσης

Για την εκτέλεση αυτής της περίπτωσης χρήσης πρέπει να πληρούνται οι εξής προϋποθέσεις:

1. Ο χρήστης έχει πρόσβαση στο διαδίκτυο.
2. Η βάση δεδομένων περιέχει ενημερωμένες πληροφορίες για τα διόδια.
3. Ο διακομιστής API είναι λειτουργικός και μπορεί να επιστρέψει δεδομένα.

### 3.1.2.3 Περιβάλλον Εκτέλεσης

1. Διαδικτυακή εφαρμογή με διαδραστικό χάρτη μέσω Leaflet.js.
2. Επικοινωνία με REST API για λήψη δεδομένων διοδίων.
3. Συμβατότητα με μοντέρνους browsers (Chrome, Firefox, Edge, Safari).

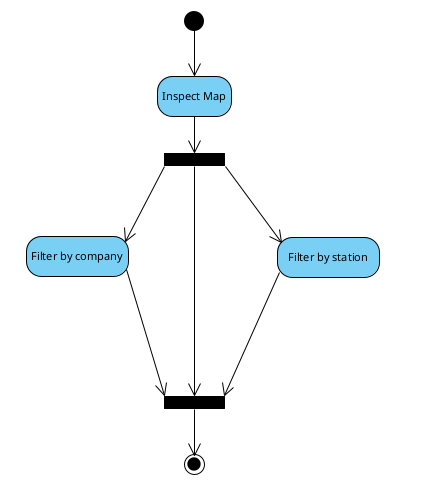
### 3.1.2.4 Δεδομένα Εισόδου

* Φίλτρο εταιρείας διαχείρισης διοδίων (Dropdown με επιλογές).
* Αναζήτηση διοδίου βάσει ονόματος (Text input).
* Συντεταγμένες γεωγραφικής θέσης (από API).
* Επιλογή σημείου στο χάρτη (κλικ σε marker).

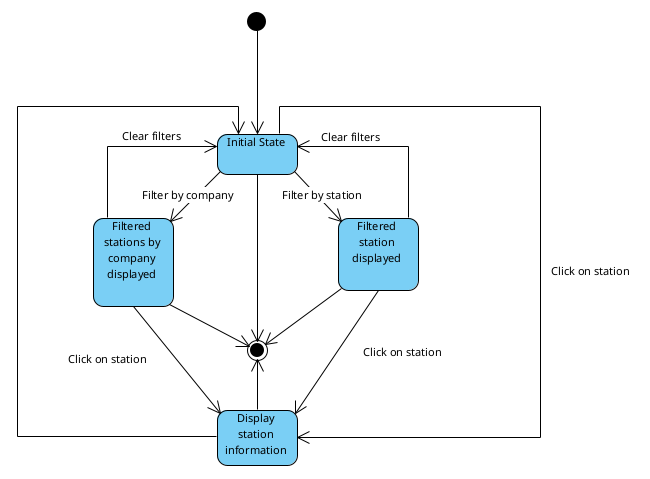
### 3.1.2.5 Αλληλουχία Ενεργειών - Επιθυμητή Συμπεριφορά

1. Ο χρήστης φορτώνει τη σελίδα του χάρτη.
2. Η εφαρμογή ανακτά τα δεδομένα διοδίων μέσω API.
3. Τα διόδια προβάλλονται στο χάρτη με markers.
4. Ο χρήστης μπορεί να:
   1. Φιλτράρει τα διόδια βάσει εταιρείας.
   2. Αναζητήσει διόδια βάσει ονόματος.
   3. Κλικάρει σε marker για πληροφορίες.
5. Όταν ανοίγει το popup πληροφοριών, εμφανίζονται:
   1. Όνομα διοδίου.
   2. Εταιρεία διαχείρισης.
   3. Email επικοινωνίας.
   4. Τιμοκατάλογος.
6. Ο χρήστης μπορεί να κλείσει το popup ή να αλλάξει φίλτρα.
7. Σε περίπτωση σφάλματος φόρτωσης δεδομένων, εμφανίζεται μήνυμα σφάλματος.
8. Ο χρήστης μπορεί να επιστρέψει στην αρχική κατάσταση (reset filters).

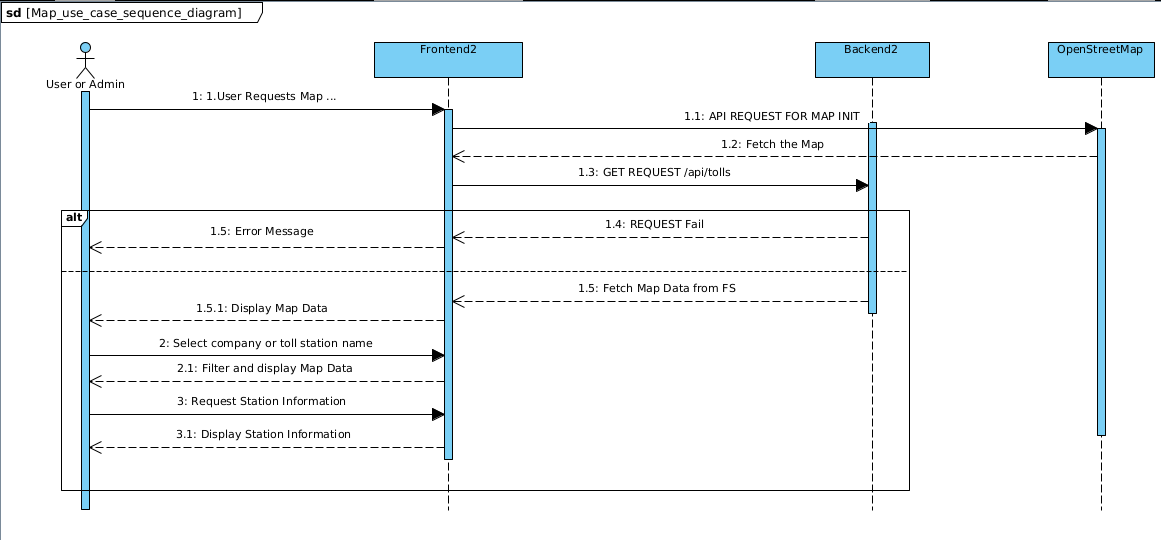
Activity:



State:



## Sequence:



### 3.1.2.7 Δεδομένα Εξόδου

* Δεδομένα που επιστρέφονται από το API και προβάλλονται στο χρήστη:
  + JSON με πληροφορίες διοδίων (όνομα, συντεταγμένες, εταιρεία, email, τιμές).
  + Γραφικά στοιχεία Leaflet.js (markers, popups).

### 3.1.2.8 Παρατηρήσεις

* Πρέπει να εξασφαλιστεί ότι η εφαρμογή λειτουργεί σωστά σε κινητές συσκευές.
* Οι markers πρέπει να έχουν διαφορετικά χρώματα για καλύτερη διάκριση.
* Να υπάρχει fallback σε περίπτωση αποτυχίας του API (π.χ., στατική λίστα διοδίων).
* Ενσωμάτωση caching για αποφυγή περιττών αιτήσεων στο API.

### 3.1.3 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 3: ( Προβλέψεις με ML)

### 3.1.3.1 Χρήστες (Ρόλοι) που Εμπλέκονται

* Εγγεγραμμένος Χρήστης
* Διαχειριστής Συστήματος (Admin, με δυνατότητα εκπαίδευσης μοντέλου)

### 3.1.3.2 Προϋποθέσεις Εκτέλεσης

* Ο χρήστης πρέπει να είναι συνδεδεμένος στο σύστημα μέσω JWT.
* Η βάση δεδομένων πρέπει να περιέχει ιστορικά δεδομένα κυκλοφορίας.
* Το API πρέπει να είναι διαθέσιμο για να επιστρέφει προβλέψεις.
* Ο διαχειριστής πρέπει να έχει δικαιώματα για εκπαίδευση μοντέλων.

### 3.1.3.3 Περιβάλλον Εκτέλεσης

* Διαδικτυακή εφαρμογή βασισμένη σε React.js.
* REST API για την επικοινωνία με τον διακομιστή.
* Υποστήριξη σε μοντέρνους browsers (Chrome, Firefox, Edge, Safari).
* Χρήση βιβλιοθήκης Recharts για την οπτικοποίηση των αποτελεσμάτων.

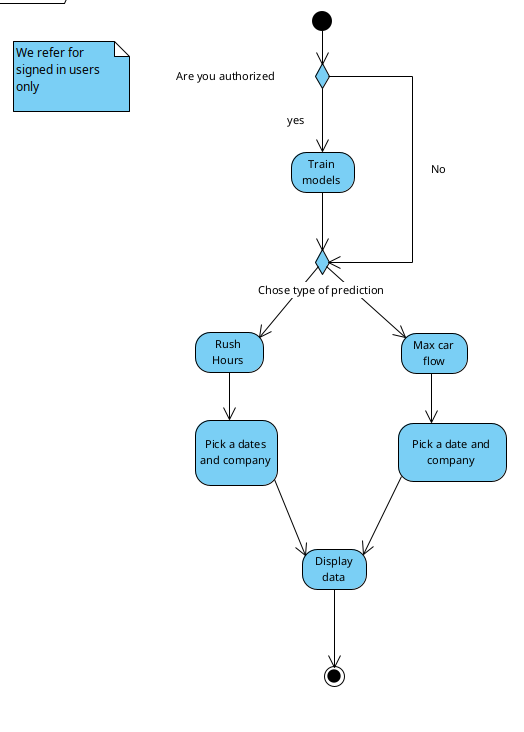
### 3.1.3.4 Δεδομένα Εισόδου

* Token ταυτοποίησης χρήστη (JWT).
* Επιλογή είδους πρόβλεψης:
  + Πρόβλεψη Ώρας Αιχμής.
  + Πρόβλεψη Μέγιστης Ροής Οχημάτων.
* Επιλογή εταιρείας διαχείρισης των διοδίων.
* Επιλογή ημερομηνιών:
  + Για πρόβλεψη ώρας αιχμής: Εύρος ημερομηνιών.
  + Για πρόβλεψη μέγιστης ροής: Μία ημερομηνία.
* Ο διαχειριστής μπορεί να ξεκινήσει την εκπαίδευση του μοντέλου.

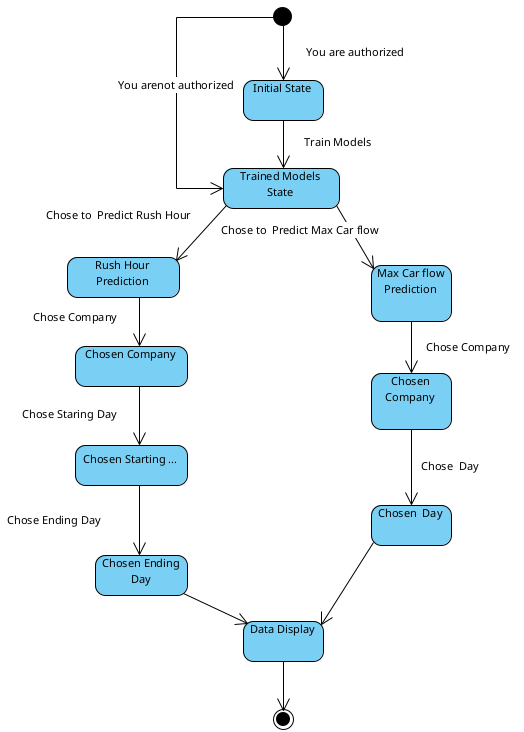
### 3.1.3.5 Αλληλουχία Ενεργειών - Επιθυμητή Συμπεριφορά

1. **Ο χρήστης συνδέεται** και ελέγχεται η ταυτότητά του μέσω JWT.
2. Το σύστημα ελέγχει αν ο χρήστης είναι **διαχειριστής**:
   1. Αν ναι, εμφανίζεται η επιλογή για εκπαίδευση μοντέλων.
3. Ο χρήστης επιλέγει **τον τύπο πρόβλεψης**:
   1. **Ώρα αιχμής**: Επιλογή ημερομηνιών και εταιρείας.
   2. **Μέγιστη ροή οχημάτων**: Επιλογή ημερομηνίας και εταιρείας.
4. Ο χρήστης υποβάλλει το αίτημα και το σύστημα:
   1. Καλεί το API για να ανακτήσει τα δεδομένα.
   2. Αντιστοιχίζει τα δεδομένα με τα αντίστοιχα ονόματα σταθμών διοδίων.
   3. Φιλτράρει τα δεδομένα ώστε να εμφανίζονται μόνο μοναδικές εγγραφές.
5. Το σύστημα εμφανίζει τα αποτελέσματα:
   1. Πίνακας με τις ημερομηνίες και τις προβλέψεις.
   2. Διαδραστικό γράφημα με τις προβλέψεις κυκλοφορίας.
6. **Σε περίπτωση σφάλματος**, εμφανίζεται κατάλληλο μήνυμα στον χρήστη.

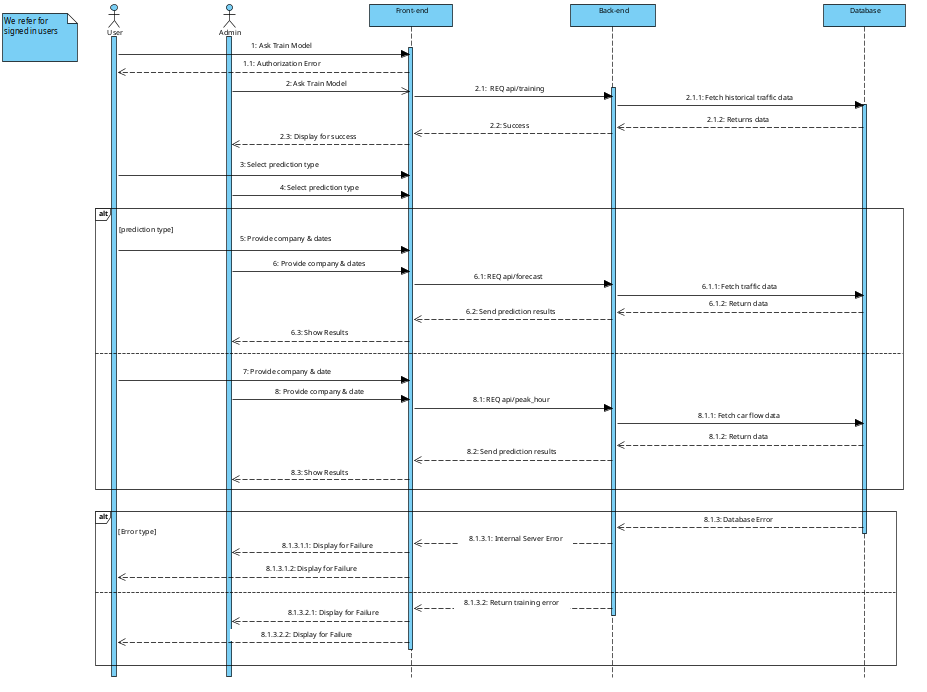
Activity Diagram:



State:



Sequence:



### 3.1.3.7 Δεδομένα Εξόδου

* Δεδομένα πρόβλεψης που επιστρέφονται από το API:
  + Για την **ώρα αιχμής**:
    - Ημερομηνία, προβλεπόμενη ώρα αιχμής.
  + Για τη **μέγιστη ροή οχημάτων**:
    - Όνομα σταθμού διοδίων, προβλεπόμενος αριθμός οχημάτων.
* Διαδραστικό γράφημα που απεικονίζει τις προβλέψεις.

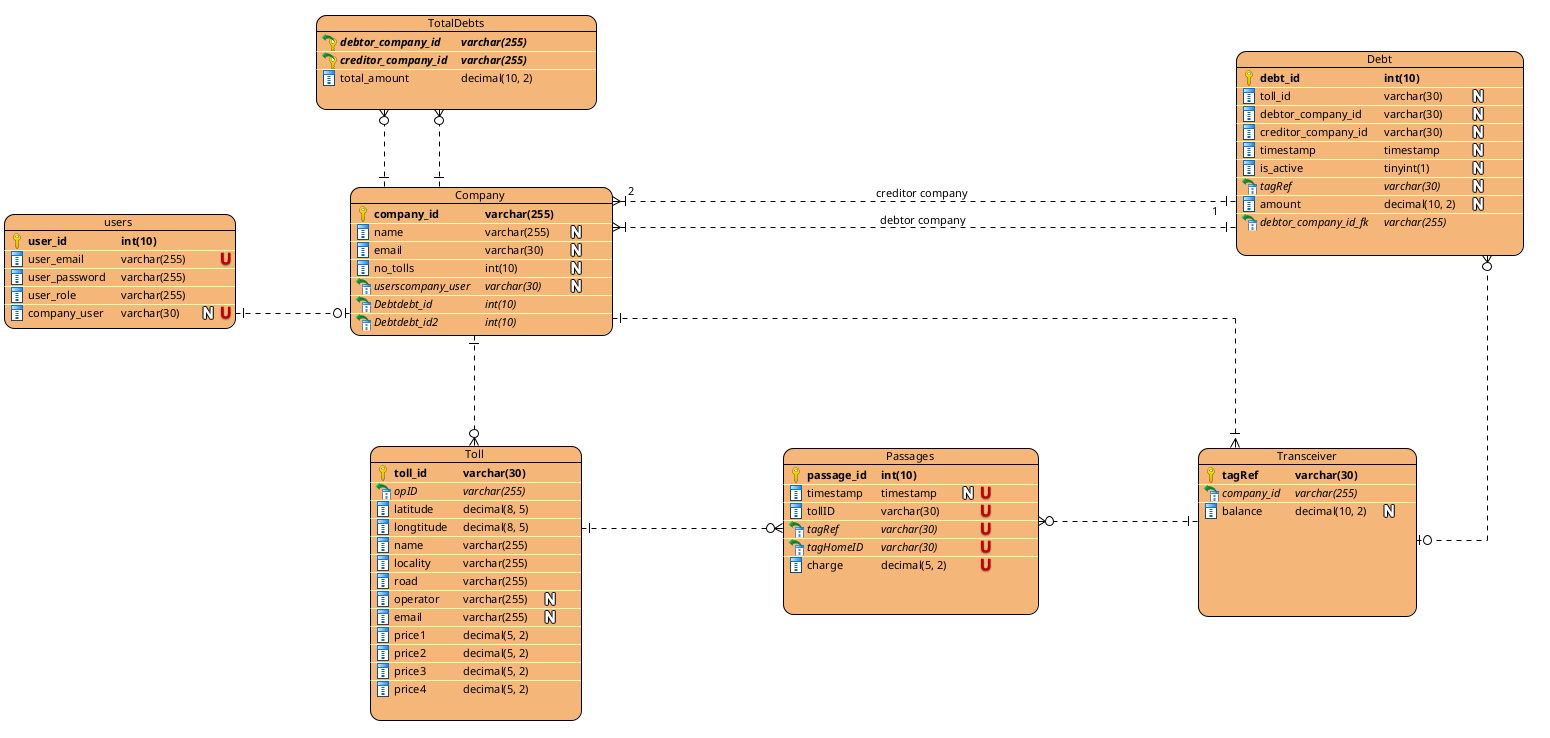
### 3.1.3.8 Παρατηρήσεις

* Οι μη εξουσιοδοτημένοι χρήστες δεν μπορούν να εκπαιδεύσουν μοντέλα.
* Σε περίπτωση αποτυχίας API, εμφανίζεται μήνυμα σφάλματος.
* Ο έλεγχος λήξης του JWT γίνεται αυτόματα.
* Οι προβλέψεις βασίζονται σε ιστορικά δεδομένα, οπότε μπορεί να έχουν μεταβλητή ακρίβεια.

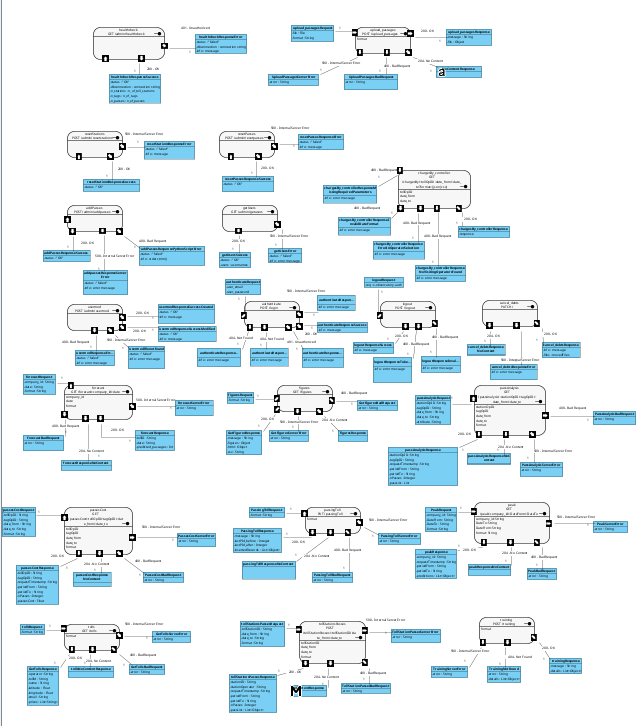
*-*

# Λοιπά διαγράμματα UML

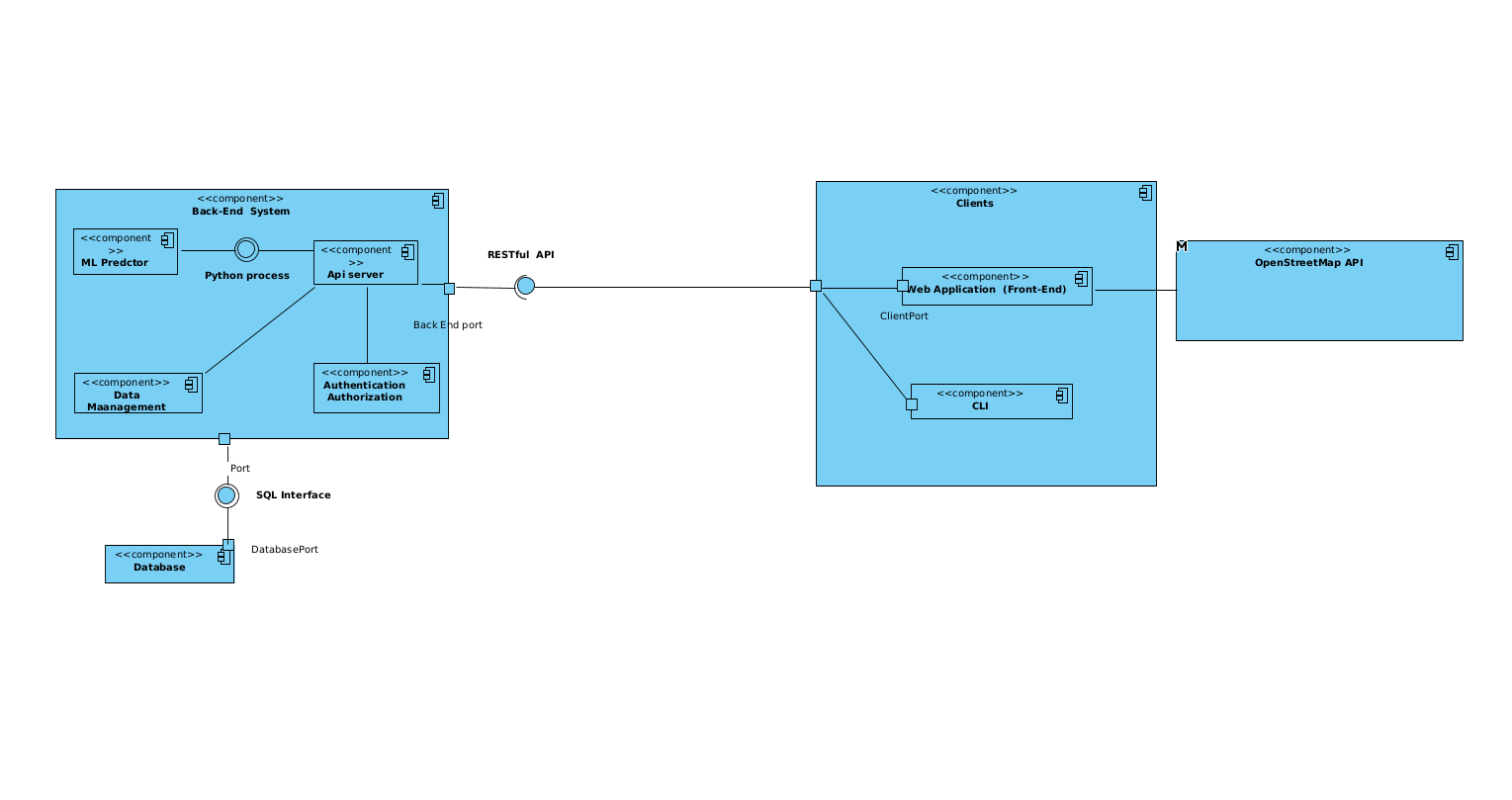
ER Diagram:



UML API:



Component Diagram:



Deployment Diagram:

