**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**

**ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**

Μάθημα: Τεχνολογία Λογισμικού (7ο εξάμηνο) ακ. έτος: 2024-2025

Έγγραφο απαιτήσεων λογισμικού (SRS)

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

**Easygo**

Θέμα Εργασίας

“Διαλειτουργικότητα στα πληροφοριακά συστήματα διοδίων αυτοκινητοδρόμων με διαφορετικά συστήματα αυτόματης διέλευσης - Ανάλυση δεδομένων διελεύσεων από διαφορετικές οπτικές γωνίες”

**Ομάδα: Softeng24-2**

Βιτζηλαίος Γιώργος 03116672

Παπαδάκος Αλέξανδρος 03118860

**Περιεχόμενα**

[1. Εισαγωγή 4](#__RefHeading___Toc31112_606882973)

[1.1 Εισαγωγή: σκοπός του λογισμικού 4](#__RefHeading___Toc16967_2998641135)

[1.2 Διεπαφές (interfaces) 4](#__RefHeading___Toc16969_2998641135)

[1.2.1 Διεπαφές με εξωτερικά συστήματα 4](#__RefHeading___Toc16971_2998641135)

[1.2.2 Διεπαφές με το χρήστη 6](#__RefHeading___Toc16973_2998641135)

[2. Αναφορές - πηγές πληροφοριών 6](#__RefHeading___Toc16975_2998641135)

[3. Προδιαγραφές απαιτήσεων λογισμικού 6](#__RefHeading___Toc16977_2998641135)

[3.1 Περιπτώσεις χρήσης 6](#__RefHeading___Toc16979_2998641135)

[3.1.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: (Log in/Sign up) 7](#__RefHeading___Toc16981_2998641135)

[3.1.1.1 Χρήστες (ρόλοι) που εμπλέκονται 7](#__RefHeading___Toc16983_2998641135)

[3.1.1.2 Προϋποθέσεις εκτέλεσης 7](#__RefHeading___Toc16985_2998641135)

[3.1.1.3 Περιβάλλον εκτέλεσης 8](#__RefHeading___Toc16987_2998641135)

[3.1.1.4 Δεδομένα εισόδου 8](#__RefHeading___Toc16989_2998641135)

[3.1.1.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά 8](#__RefHeading___Toc16991_2998641135)

[3.1.1.6 Δεδομένα εξόδου 10](#__RefHeading___Toc16993_2998641135)

[3.1.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: (Operator Balance) 10](#__RefHeading___Toc16997_2998641135)

[3.1.2.1 Χρήστες (ρόλοι) που εμπλέκονται 10](#__RefHeading___Toc16999_2998641135)

[3.1.2.2 Προϋποθέσεις εκτέλεσης 10](#__RefHeading___Toc17001_2998641135)

[3.1.2.3 Περιβάλλον εκτέλεσης 11](#__RefHeading___Toc17003_2998641135)

[3.1.2.4 Δεδομένα εισόδου 11](#__RefHeading___Toc17005_2998641135)

[3.1.2.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά 11](#__RefHeading___Toc17007_2998641135)

[3.1.2.6 Δεδομένα εξόδου 13](#__RefHeading___Toc17009_2998641135)

[3.1.2.7 Παρατηρήσεις 13](#__RefHeading___Toc17027_29986411351)

[3.1.3 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 3: (Load the Target) 13](#__RefHeading___Toc17013_2998641135)

[3.1.3.1 Χρήστες (ρόλοι) που εμπλέκονται 13](#__RefHeading___Toc17015_2998641135)

[3.1.3.2 Προϋποθέσεις εκτέλεσης 13](#__RefHeading___Toc17017_2998641135)

[3.1.3.3 Περιβάλλον εκτέλεσης 13](#__RefHeading___Toc17019_2998641135)

[3.1.3.4 Δεδομένα εισόδου 13](#__RefHeading___Toc17021_2998641135)

[3.1.3.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά 13](#__RefHeading___Toc17023_2998641135)

[3.1.3.6 Δεδομένα εξόδου 15](#__RefHeading___Toc17395_2998641135)

[3.1.3.7 Παρατηρήσεις 15](#__RefHeading___Toc17027_2998641135)

[3.1.4 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 4: (Data Extraction) 16](#__RefHeading___Toc17029_2998641135)

[3.1.4.1 Χρήστες (ρόλοι) που εμπλέκονται 16](#__RefHeading___Toc17031_2998641135)

[3.1.4.2 Προϋποθέσεις εκτέλεσης 16](#__RefHeading___Toc17033_2998641135)

[3.1.4.3 Περιβάλλον εκτέλεσης 16](#__RefHeading___Toc17035_2998641135)

[3.1.4.4 Δεδομένα εισόδου 16](#__RefHeading___Toc17037_2998641135)

[3.1.4.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά 16](#__RefHeading___Toc17039_2998641135)

[3.1.4.6 Δεδομένα εξόδου 18](#__RefHeading___Toc17397_2998641135)

[3.1.4.7 Παρατηρήσεις 18](#__RefHeading___Toc17043_2998641135)

[3.2 Απαιτήσεις επιδόσεων 18](#__RefHeading___Toc17045_2998641135)

[3.3 Απαιτήσεις οργάνωσης δεδομένων 18](#__RefHeading___Toc17047_2998641135)

[3.3.1 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα 18](#__RefHeading___Toc17049_2998641135)

[3.4 Λοιπές απαιτήσεις 19](#__RefHeading___Toc17051_2998641135)

[3.4.1 Απαιτήσεις διαθεσιμότητας λογισμικού 19](#__RefHeading___Toc17053_2998641135)

[3.4.2 Απαιτήσεις ασφάλειας 19](#__RefHeading___Toc17055_2998641135)

# **Εισαγωγή**

**Το παρόν έγγραφο αποτελεί την Προδιαγραφή Απαιτήσεων Λογισμικού (SRS) για την ανάπτυξη μιας ιστοεφαρμογής διαλειτουργικότητας σε πληροφοριακά συστήματα διοδίων σε αυτοκινητόδρομους.** Η εφαρμογή σχεδιάζεται με σκοπό να προσφέρει την δυνατότητα στους ιδιοκτήτες των αυτοκινητόδρομων να ελέγχουν και να εξωφλούν τυχόν οφειλές που δημιουργούνται μεταξύ τους. Ακόμα θα δίνει την δυνατότητα σε τρίτους φορείς (π.χ. το Υπουργείο Συγκοινωνιών) να μπορούν να αντλούν δεδομένα και στατιστικά στοιχεία από όλους τους αυτοκινητόδρομους και στους χρήστες των αυτοκινητόδρομων να επαναφορτίζουν την κάρτα τους.

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.

## **1.1 Εισαγωγή: σκοπός του λογισμικού**

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

## **1.2 Διεπαφές (interfaces)**

Σε ένα έργο τεχνολογίας λογισμικού, interfaces (διεπαφές) **είναι τα σημεία επικοινωνίας του λογισμικού μας με εξωτερικά συστήματα και άλλο λογισμικό**. Οι διεπαφές καθορίζουν τον τρόπο με τον οποίο τα μέρη του λογισμικού αλληλεπιδρούν μεταξύ τους, π.χ., μέσω μεθόδων, δεδομένων και πρωτοκόλλων. Διευκολύνουν τη συνεργασία μεταξύ διαφορετικών λειτουργικών τμημάτων, κάνοντας το σύστημα πιο ευέλικτο και επεκτάσιμο.

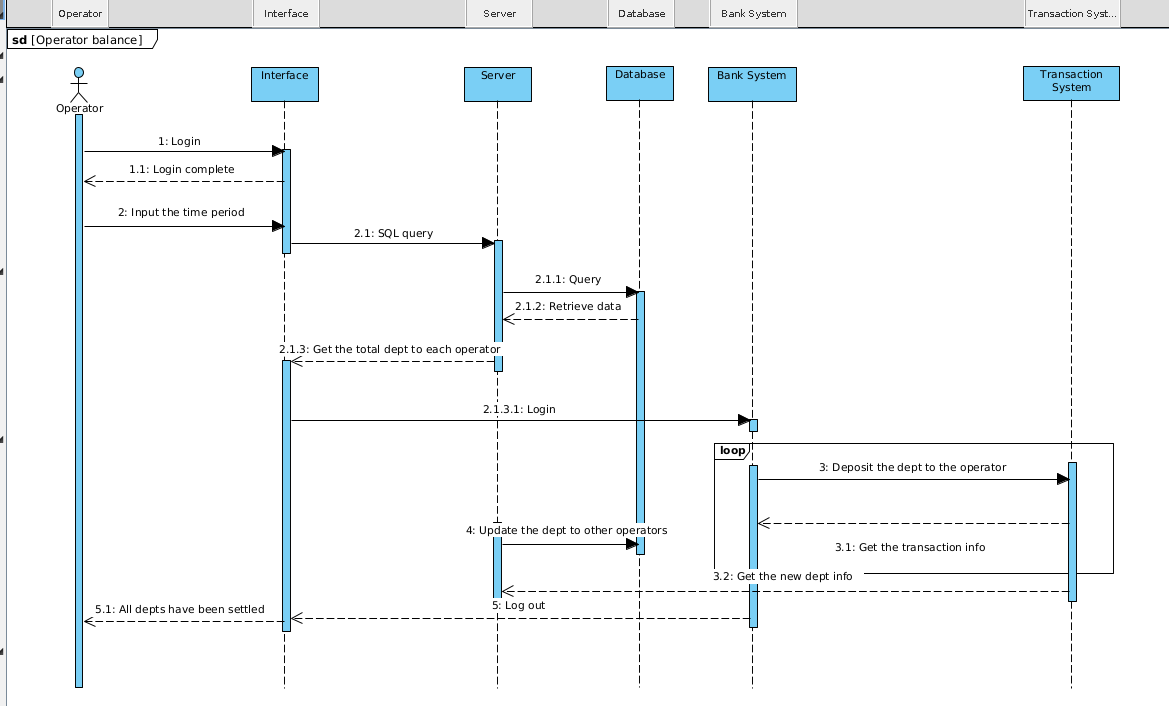
### **1.2.1 Διεπαφές με εξωτερικά συστήματα**

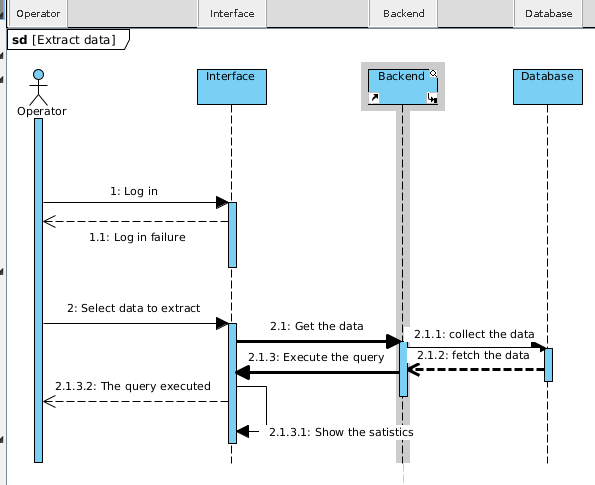
Η δική μας εφαρμογή,όπως είναι λογικό, έχει διεπαφές με εξωτερικά συστήματα, αφού πρόκειται για εφαρμογή διαλειτουργικότητας σε πληροφοριακά συστήματα δεδομένων και είναι αναγκαία η συνεχής ανταλλαγή δεδομένων μεταξύ αυτών των συστημάτων. **Πιο συγκεκριμένα, η εφαρμογή μας έχει διεπαφές με:**

* **Τα ηλεκτρονικά συστήματα διέλευσης διοδίων των εφτά οδών.** Για να μπορεί να συλλέγει τα δεδομένα από τις διελεύσεις των οχημάτων σε κάθε οδό και να υπολογίζει τις οφειλές που δημιουργούνται μεταξύ εταιριών διαχείρησης διοδίων. Οι οφειλές προς αλλήλους συμψηφίζονται και καταβάλλεται η διαφορά.
* **Τα τραπεζικά συστήματα των λειτουργών των οδών.** Για να μπορεί να αποστέλει τις χρηματικές διαφορές ανάμεσα στους οδούς σε κάθε τράπεζα και να καταβάλονται οι οφειλές.
* **Το Υπουργείο Μεταφορών.** Για να αποστέλουμε τα δεδομένα και να αναλύονται προκειμένου να εξαχθούν συμπεράσματα σχετικά με τη χρήση των αυτοκινητόδρομων, την συμπεριφορά των οδηγών, τις αποσβέσεις επενδύσεων, προβλέψεις μεγεθών κ.α.

**Τα παραπάνω υλοποιούνται με το πρότυπο ανταλλαγής δεδομένων REST(Representational State Transfer).** Το REST είναι ένα από τα πιο δημοφιλή πρότυπα για την κατασκευή API. Βασίζεται σε αρχές του HTTP και χρησιμοποιεί τις γνωστές μεθόδους του (GET, POST, PUT, DELETE) για τη διαχείριση των πόρων. Χρησιμοποιείται κυρίως για την επικοινωνία μεταξύ client και server.

Για να γίνουν πιο κατανοητά τα παραπάνω,παραθέτουμε τα UML component για την διεπαφή με τις τράπεζες αλλά και με το Υπουργείο Μεταφορών αντίστοιχα.





### **1.2.2 Διεπαφές με το χρήστη**

Η εφαρμογή που έχουμε δημιουργήσει δίνει την δυνατότητα στους χρήστες να επικοινωνούν με το συστήμα και να αντλούν χρήσιμες πληροφορίες. **Στο περιβάλλον της εφαρμογής(User Interface) θα δίνεται η δυνατότητα σε μία από τις καρτέλες πλοήγησης, να επιλέγει ο χρήστης ποιες πληροφορίες χρειάζεται να αντλήσει από το σύστημα μας.** Σύμφωνα με την επιλογή του, το API της εφαρμογής θα εμφανίζει οπτικοποιημένα, τα δεδομένα που ζήτησε. Ανάλογα τον τύπο του χρήστη της εφαρμογής, υπάρχει η δυνατότητα:

* Να εμφανίζεται το διαθέσιμο υπόλοιπο του εκάστοτε target για κάθε οδηγό.
* Να εμφανίζονται σε κάθε εταιρία οι οφειλές από την διέλευση πελατών της σε άλλους οδούς ή τα κέρδη από την διέλευση στην οδό της από πελάτες άλλων εταιριών, τον εκάστοτε μήνα. Σε περίπτωση οφειλών θα δίνεται η δυνατότητα εξόφλησης τους.
* Να εμφανίζεται πίνακας με γενικότερα στατιστικά και δεδομένα από την διέλευση οχημάτων από τα διόδια.
* Να εμφανίζονται τα στοιχεία όλων των οδών με διόδια και αντίστοιχα links στις ιστοσελίδες τους.

Τα παραπάνω θα οπτικοποιηθούν μέσω βιβλιοθηκών της python που δίνουν την δυνατότητα για την παρουσίαση στατιστικών στοιχείων και δεδομένων στους χρήστες μέσω διάφορων διαγραμμάτων. Κάποια τέτοια frameworks που θα χρησιμοποιήσουμε είναι το Matplotlib και το Seaborn.

# **Αναφορές - πηγές πληροφοριών**

Για να γίνει πράξη η εφαρμογή μας, είναι αναγκαίο να έχουμε την πρόσβαση σε κάποια δεδομένα και πηγές πληροφοριών, που μπορούν να μας εξασφαλίσουν την **σωστή λειτουργία** της εφαρμογής μας. Τέτοια πηγή πληροφοριών είναι:

* Τα συστήματα των εκάστοτε οδών. Για να μπορεί το σύστημα μας, να συλλέγει τα δεδομένα από τις διελεύσεις των οχημάτων σε κάθε οδό και να υπολογίζει τις οφειλές που δημιουργούνται μεταξύ εταιριών διαχείρησης διοδίων.

Η δική μας ιστοεφαρμογής διαλειτουργικότητας σε πληροφοριακά συστήματα διοδίων σε αυτοκινητόδρομους, πραγματοποιείται στα πλαίσια του μαθήματος “Τεχνολογίας Λογισμικού” της σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Ε.Μ.Π. Για αυτό το λόγο τα δεδομένα **δεν είναι πραγματικά,** έχουν δωθεί αιθαίρετα, καθαρά για ακαδημαικούς λόγους, για να μπορέσει η εφαμοργή μας να λειτουργίσει.

# **Προδιαγραφές απαιτήσεων λογισμικού**

Οι προδιαγραφές απαιτήσεων λογισμικού αφορούν την λεπτομερής προδιαγραφή των λειτουργιών του λογισμικού μας σε επίπεδο περιπτώσεων χρήσης. Παρακάτω αναλύουμε τόσο τις περιπτώσεις χρήσης της εφαρμογής μας (Use Cases), όσο και τις λειτουργικές απαιτήσεις αυτών των περιπτώσεων.

## **3.1 Περιπτώσεις χρήσης**

Το σύστημα μας έχει τέσσερις περιπτώσεις χρήσης (Use Cases). Πιο συγκεκριμένα δίνει την δυνατότητα:

* Σε όλους τους χρήστες (κατόχους διοδίων διέλευσης – κατόχους target) για δημιουργία λογαριασμού με Username και Password, για authentication χρήστη.
* Στους κατόχους διοδίων διέλευσης των οδών να βλέπουν τις οφειλές μεταξύ τους.
* Στους κατόχους target για τα εκάστοτε οχήματα την δυνατότητα φόρτισης τους.
* Στους κατόχους διοδίων διέλευσης την δυνατότητα να βλέπουν στατιστικά στοιχεία και να τα εξάγουν σε κάποιο αρχείο excel.

Τα παραπάνω θα υλοποιηθούν σε ένα εύχριστο και φιλικά προσκύμενο περιβάλλον (front – end), όπου κάθε χρήστης θα έχει την δυνατότητα να εκτελέσει τα παραπάνω use cases μέσω διαφορετικών καρτέλων. Σε όλους τους χρήστες θα υπάρχει η main καρτέλα πλοήγησης της εφαρμογής όπου ο χρήστης θα μπορεί να δει τα προσωπικά του στοιχεία, διάφορες χρήσιμες πληροφορίες για την εφαρμογή και τις δυνατότητες της.

Για να γίνουν πράξη τα παραπάνω, θα υλοποιηθούν διάφορες λειτουργικές απαιτήσεις, απαραίτητες για την ορθή χρήση της εφαρμογής. Πιο συγκεκριμένα:

* Τα διάφορα στοιχεία χρήστη (Username και Password) θα αποθηκεύονται στη βάση δεδομένων της εφαρμογής.
* Για να εμφανίζονται σωστά οι οφειλές μεταξύ των διάφορων οδών χρειάζεται να υλοποιηθεί στο επίπεδο του back-end, τόσο η σωστή διεπαφή με τα συστήματα των οδών διεύλευσης διοδίων, όσο και η διεπαφή με τις τράπεζες ώστε να ενημερώνεται το σύστημα όταν εξοφλείται κάποια οφειλή.
* Η αποθήκευση των στοιχείων target των καρτών του εκάστοτε χρήστη στην βάση δεδομένων αλλά και όταν ένα target επαναφορτίζεται, να εξασφαλίζεται τόσο η ενημέρωση του συστήματος μας από την εκάστοτε εταιρία που ανήκει το target όσο και των άλλων συστημάτων των οδών διέλευσης διοδίων από το δικό μας σύστημα.
* Η δυνατότητα να εξάγονται ομαδοποιημένα και επεξεργασμένα δεδομένα από την βάση, να μπορεί το Υπουργείο να έχει πρόσβαση σε αυτά μέσω μίας διεπαφής με το σύστημα μας.

### **3.1.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: (Log in/Sign up)**

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

Στο συγκεκριμένο Use Case οι ρόλοι που εμπλέκονται είναι οι χρήστες της εφαρμογής οι οποίοι μπορούν είτε να εισέλθουν μέσω ενός authentication στην εφαρμογή, είτε να εγγραφούν αντίστοιχα. Χρήστες της εφαρμογής μπορούν να είναι είτε οι κάτοχοι target διέλευσης διοδίων, είτε οι ιδιοκτήτες των διοδίων αυτών.

#### **3.1.1.2 Προϋποθέσεις εκτέλεσης**

Για να μπορέσει να εκτελεστεί το παραπάνω Use Case, θα πρέπει στην περίπτωση εισόδου του χρήστη στην εφαρμογή, να πληκτρολογήσει το σωστό Username και το σωστό Password, με τα οποία έχει κάνει εγγραφή. Στην περίπτωση που ο χρήστης θέλει να κάνει Sign up στην εφαρμογή μας, για να εκτελεστεί το παραπάνω Use Case θα πρέπει να δώσει Όνομα Χρήστη, καινούριο Κωδικό Πρόσβασης και τον αριθμό του Target του για να πιστοποιεί ότι είναι εγγεγραμμένος σε κάποια από τα συστήματα των αντίστοιχων εταιριών-οδών διέλευσης διοδίων. Οι ιδιοκτήτες διοδίων διέλευσεις σε οδούς εγγράφονται όλοι αυτόματα από τον super-admin του συστήματος. Σε περίπτωση που θέλει κάποια καινούρια εταιρία να εγγραφτεί στο σύστημα μας, αυτό γίνεται μόνο με την επικοινωνία του ενδιαφερόμενου με τον super-admin της εφαρμογής.

#### **3.1.1.3 Περιβάλλον εκτέλεσης**

Το περιβάλλον στο οποίο εκτελείται η συγκεκριμένη περίπτωση χρήσης, είναι η διαδικτυακή διεπαφή με τον χρήστη. Αφορά το πρώτο κομμάτι του front-end, όπου ο χρήστης ανοίγοντας την εφαρμογή καλείται να συμπληρώσει τα στοιχεία του είτε για να συνδεθεί στην εφαρμογή είτε για να εγγραφή.

#### **3.1.1.4 Δεδομένα εισόδου**

Στην περίπτωση που ο χρήστης – κάτοχος target θέλει να πραγματοποιήσει εγγραφή στο σύστημα, τα δεδομένα εισόδου είναι το Username και το Password που θα καταχωρύσει, τον αριθμό του target του. Σε περίπτωση εισόδου στην εφαρμογή απαιτείται μόνο το Username και το Password ως δεδομένα εισόδου.

#### **3.1.1.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά**

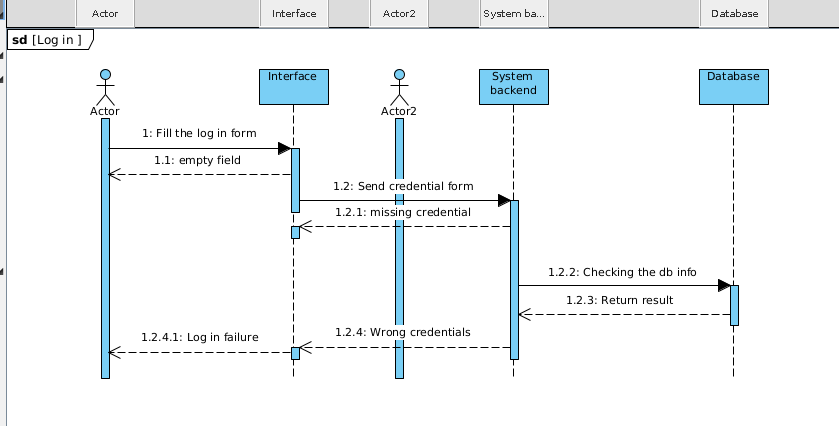
Όπως φαίνεται και παρά κάτω, στα αντίστοιχα UML διαγράμματα δραστηριοτήτων και αλληλουχίας, η επιθυμητή συμπεριφορά του Use Case είναι η εξής:

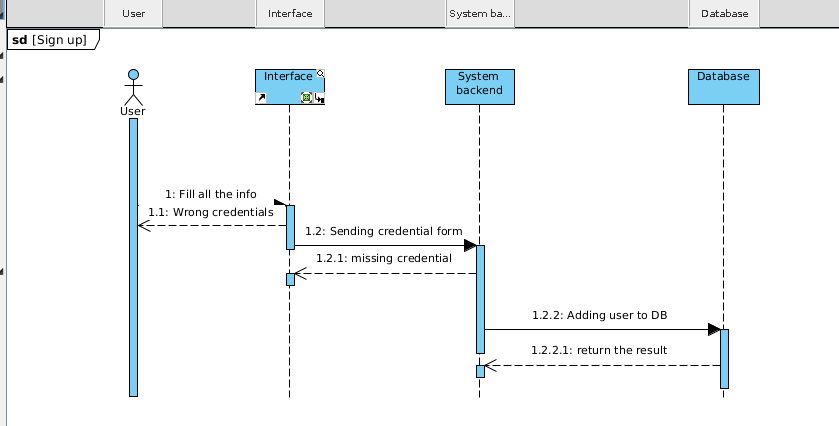
* Ο χρήστης πληκτρολογεί το σωστό Username και το σωστό Password, για να συνδεθεί στην εφαρμογή.
* Η εφαρμογή μέσω του API και του αντίστοιχου endpoint, στέλνει τις πληροφορίες αυτές στην βάση δεδομένων.
* Μέσω του querry στην βάση δεδομένων, επαληθεύουμε εάν ο χρήστης είναι όντως εγγεγραμένος και εάν ο κωδικός που έβαλε είναι σωστός.
* Εάν τα στοιχεία είναι σωστά, τότε μέσω του API ο χρήστης οδηγείται στην main σελίδα της εφαρμογής και να συνεχίσει την πλοήγηση του με βάση τις ανάγκες του.
* Εάν τα στοιχεία είναι λανθασμένα, τότε μέσω του API, εμφανίζεται ένα μήνυμα λάθους και η δυνατότητα στον χρήστη να προσπαθήσει ξανά να βάλει τα σωστά στοιχεία.

Σε περίπτωση που ο χρήστης επιλέξει την επιλογή της εγγραφής στην εφαρμογή, τότε:

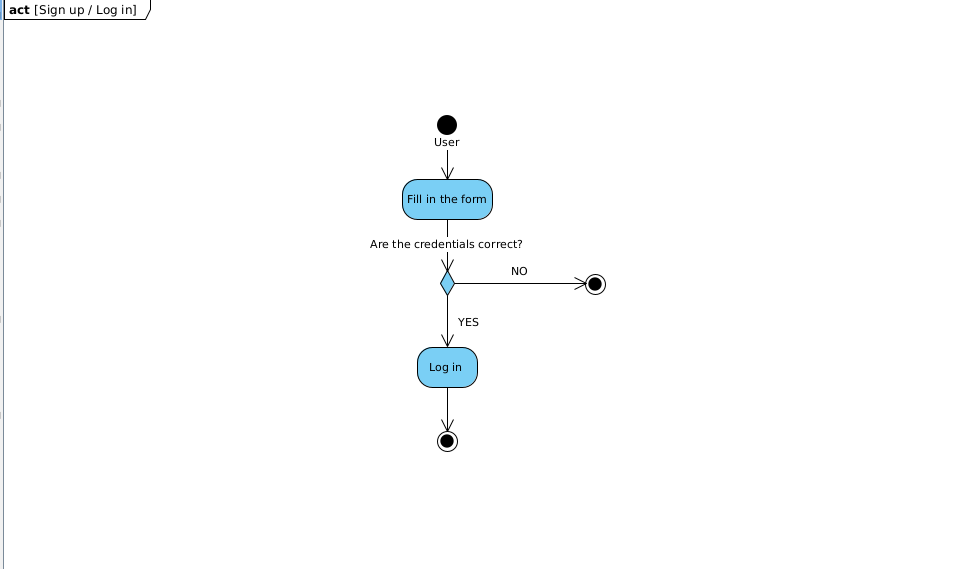
* Μαζί με τα στοιχεία του λογαριασμού του στην εφαρμογή (Username και Password) οφείλει να πληκτρολογήσει και το σωστό αριθμό Target.
* Η εφαρμογή μέσω του API και του αντίστοιχου endpoint, επικοινωνεί μέσω ενός interface με το αντίστοιχο σύστημα διέλευσης διοδίων της εκάστοτε οδού.
* Σε περίπτωση που ο χρήστης έχει Target, αποθηκεύει τις πληροφορίες αυτές στην βάση δεδομένων και μέσω του API ο χρήστης οδηγείται στην main σελίδα της εφαρμογής και να συνεχίσει την πλοήγηση του με βάση τις ανάγκες του.
* Σε περίπτωση που ο χρήστης πληκτρολογήσει λάθος αριθμό Target,τότε μέσω του API, εμφανίζεται ένα μήνυμα λάθους και η δυνατότητα στον χρήστη να προσπαθήσει ξανά να βάλει τα σωστά στοιχεία.

Παραθέτουμε τα UML διαγράμματα δραστηριοτήτων και αλληλουχίας.

* Διάγραμμα UML αλληλουχίας για Log in:
* Διάγραμμα UML αλληλουχίας για Sign up:



* Διάγραμμα UML δραστηριοτήτων για Log in/Sign up:



#### **3.1.1.6 Δεδομένα εξόδου**

Τέλος, δεδομένα εξόδου στο συγκεκριμένο Use Case, έχουμε όταν το Log in ή το Sign Up, αποτύγχουν. Σε αυτή την περίπτωση εμφανίζεται στην οθόνη του χρήστη ένα μήνυμα σφάλματος και η δυνατότητα να ξανά βάλει δεδομένα και να ξανά εκτελέσει το συγκεκριμένο Use Case. Σε περίπτωση επιτυγχημένης προσπάθιας Log in ή Sign up, τότε το API φορτώνει την main σελίδα της εφαρμογής και εμφανίζει ένα μήνυμα επιτυγχημένης εισόδου ή εγγραφής.

### **3.1.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: (Operator Balance)**

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

Στο συγκεκριμένο Use Case οι ρόλοι που εμπλέκονται είναι οι ιδιοκτήτες των διοδίων διέλευσης της εκάστοτε οδούς, που έχουν την δυνατότητα μέσω της εφαρμογής μας να βλέπουν τις οφειλές που έχουν μεταξύ τους.

#### **3.1.2.2 Προϋποθέσεις εκτέλεσης**

Για να μπορέσει να εκτελεστεί η παραπάνω περίπτωση χρήσης θα πρέπει να συνδεθεί ένας από τους λειτουργούς οδών με διόδια και να επιλέξει την αντίστοιχη καρτέλα που αναφέρεται στις οφειλές μεταξύ των εκάστοτε εταιριών.

#### **3.1.2.3 Περιβάλλον εκτέλεσης**

Το περιβάλλον στο οποίο εκτελείται η συγκεκριμένη περίπτωση χρήσης, είναι η διαδικτυακή διεπαφή με τον χρήστη. Αφορά το κομμάτι του front-end, όπου ο ιδιοκτήτης μίας εκ των εφτά εταιριών, αφού συνδεθεί στην εφαρμογή, επιλέγει την καρτέλα όπου εμφανίζονται οι οφειλές κατά αλλήλους. Στην συγκεκριμένη καρτέλα, ο χρήστης μπορεί να δεί τα ονόματα των άλλων έξι οδών με διόδια και από κάτω τις οφειλές που έχουν ή τις οφειλές που έχει αυτός με τις αντίστοιχες εταιρίες. Αν χρωστάει ή του χρωστάνε λεφτά φαίνεται από το +(του χρωστάνε) και το -(χρωστάει). Ακόμα δίπλα φαίνεται η τελευταία ημερομηνία όπου άλλαξε το συγκεκριμένο ποσό.

#### **3.1.2.4 Δεδομένα εισόδου**

Τα δεδομένα εισόδου για την συγκεκριμένη περίπτωση χρήσης είναι η διαφορά των οφειλών μεταξύ των εκάστοτε εταιρειών διοδίων. Αυτά τα δεδομένα μας τα εξασφαλίζει η διεπαφή που έχει το σύστημα μας με τα αντίστοιχα των εφτά εταιριών με διόδια καθώς και η επιλογή της χρονικής περιόδου για την οποία ενδιαφέρεται ο χρήστης.

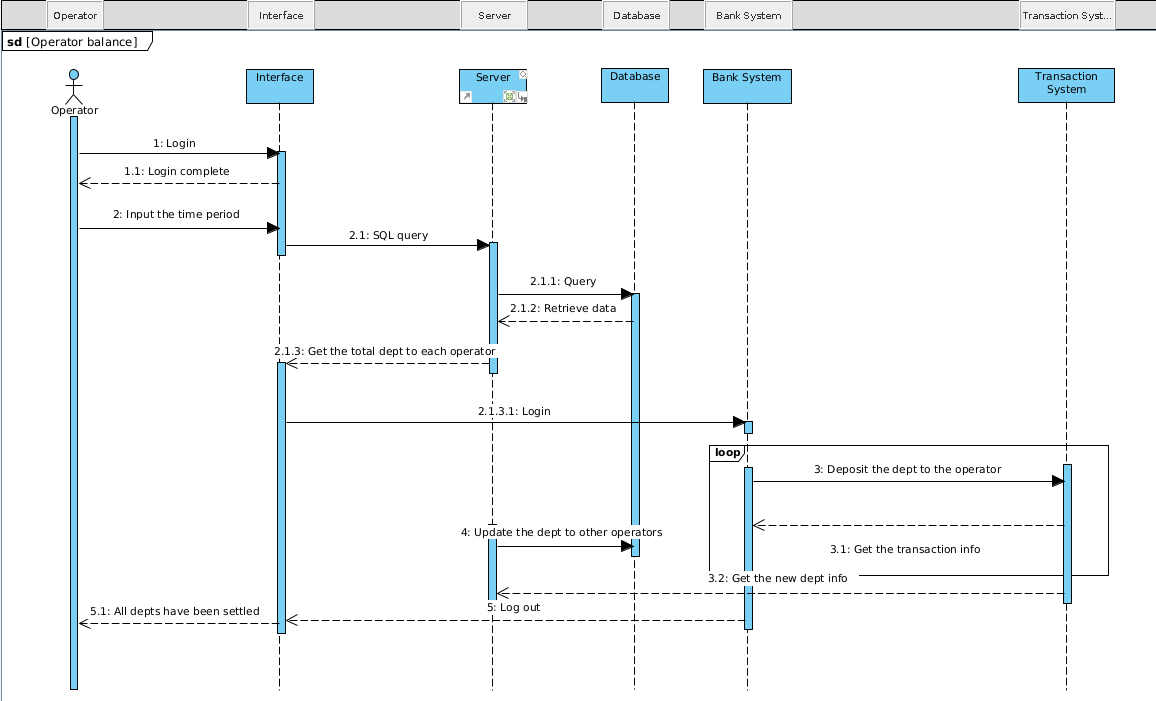
#### **3.1.2.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά**

Στη συγκεκριμένη περίπτωση χρήσης(Operator Balance) η αλληλουχία ενεργειών που θα ακολουθηθεί είναι η εξής:

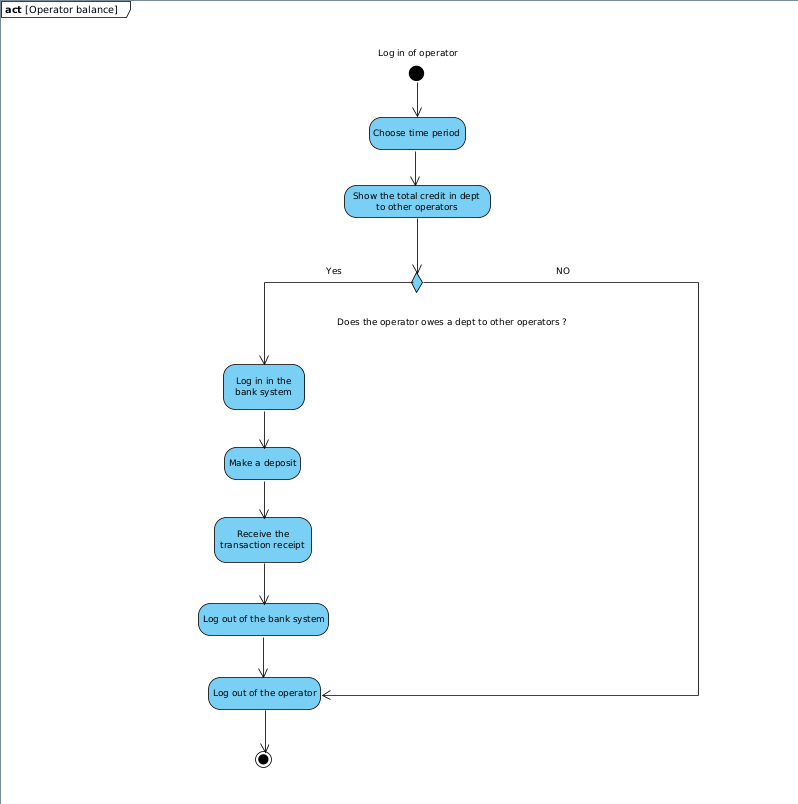
* Ο χρήστης-ιδιοκτήτης εταιρίας αφου συνδεθεί στην εφαρμογή μας επιλέγει την αντίστοιχη καρτέλα για το Opearator Balance.
* Το API της εφαρμογής μας μέσω της ενέργειας POST από το back-end εμφανίζει την καρτέλα που περιγράψαμε στην ενότητα “Περιβάλλον εκτέλεσης”
* Για να εμφανιστούν τα σωστά νούμερα με τις οφειλές για κάθε μία από τις υπόλοιπες έξι εταιρίες διοδίων, το back-end μέσω μίας διεπαφής έχει φροντίσει πρίν από κάποια μικρό χρονικό διάστημα(πχ μία μέρα) να επικοινωνήσει με τα συστήματα των εκάστοτε οδών και να αντλήσει τις πληροφορίες που χρειάζεται και να τις αποθηκεύσει στην βάση δεδομένων μας. Έτσι επιτυγχάνεται η σωστή εμφάνιση των οφειλών ανά τράπεζα και το Use Case επιτυγχάνει.
* Χρειάζεται προσοχή όταν τέλος του μήνα,οι τράπεζες εξοφλούν τις εκάστοτε οφειλές ανά εταιρία. Σε αυτή την περίπτωση, θα πρέπει το σύστημα μας να σβήσει από την βάση δεδομένων, όποια πληροφορία έχει κρατήσει από διελεύσεις target σε αλλήλους(και όχι στην δρόμο της εταιρίας που ανήκει) και να μηδενίσει τα πεδία με τις οφειλές. Η σωστή και έγκαρη ενημέρωση του συστήματος μας, επιτυγχάνεται από την διεπαφή του back-end με τις τράπεζες και με το κατάλληλο trigger όταν αυτές στείλουν μήνυμα ότι εξόφλησαν το ποσό.

Για να κατανοηθούν καλύτερα τα παραπάνω παραθέτουμε τα UML διαγράμματα δραστηριοτήτων(Activity) και αλληλουχίας(Sequence) της εφαρμογής μας.

* Διάγραμμα UML αλληλουχίας για Operator Balance:



* Διάγραμμα UML δραστηριοτήτων για Operator Balance:



#### **3.1.2.6 Δεδομένα εξόδου**

Τα δεδομένα εξόδου, είναι οι οφειλές που προκύπτουν μεταξύ των εταιριών και εμφανίζονται στο front-end της εφαρμογής μας.

#### **3.1.2.7 Παρατηρήσεις**

Είναι αναγκαίο να επισημάνουμε ότι η συγκεκριμένη εφαρμογή υλοποιείται στα πλαίσια του μαθήματος Τεχνολογίας Λογισμικού της Σχολής ΗΜΜΥ του ΕΜΠ. Επομένως, είναι αδύνατον να μπορέσει το σύστημα μας να επικοινωνήσει μέσω κάποιας διεπαφής με τα πραγματικά συστήματα των εφτά εταιριών. Για αυτό το λόγο το παραπάνω κομμάτι θα υλοποιηθεί μέσω προσομοίωσης στο back-end. Δηλαδή θα φορτώσουμε εμείς την βάση δεδομένων με συγκεκριμένα δεδομένα διέλευσης οχημάτων σε κάθε οδό.

### **3.1.3 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 3: (Load the Target)**

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

Στο συγκεκριμένο Use Case οι ρόλοι που εμπλέκονται είναι οι χρήστες που κατέχουν target και τους δίνεται η δυνατότητα μέσω της εφαρμογής μας να το φορτίσουν.

#### **3.1.3.2 Προϋποθέσεις εκτέλεσης**

Για να εκτελεστεί η παραπάνω περίπτωση χρήσης (Use Case), θα πρέπει να συνδεθεί ένας χρήστης που έχει στην κατοχή του target, να επιλέξει την καρτέλα η οποία αναφέρεται στην επαναφόρτιση του target, να τοποθετήσει τα στοιχεία του και να πατήσει το κουμπί επαναφόρτισης.

#### **3.1.3.3 Περιβάλλον εκτέλεσης**

Το περιβάλλον στο οποίο εκτελείται η συγκεκριμένη περίπτωση χρήσης, είναι η διαδικτυακή διεπαφή με τον χρήστη. Όπως αναφέραμε και παραπάνω, ο χρήστης αφού συνδεθεί, επιλέγει την καρτέλα επαναφόρτισης target. Όταν επιλέξει την καρτέλα εμφανίζονται πεδία προς συμπλήρωση όπως το ποσό που θέλει να καταχωρήσει στο target, τα στοιχεία του target, τα στοιχεία της τραπεζικής του κάρτας κ.α. Ακριβώς από κάτω υπάρχει το κουμπί επαναφόρτιση όπου στέλνει αίτημα για επαναφόρτιση του target.

#### **3.1.3.4 Δεδομένα εισόδου**

Τα δεδομένα εισόδου είναι τα πεδία που θα συμπληρώσει ο χρήστης στο περιβάλλον εκτέλεσης. Τα στοιχεία του target, τα στοιχεία της τραπεζικής του κάρτας και το ποσό που θέλει να βάλει στο target.

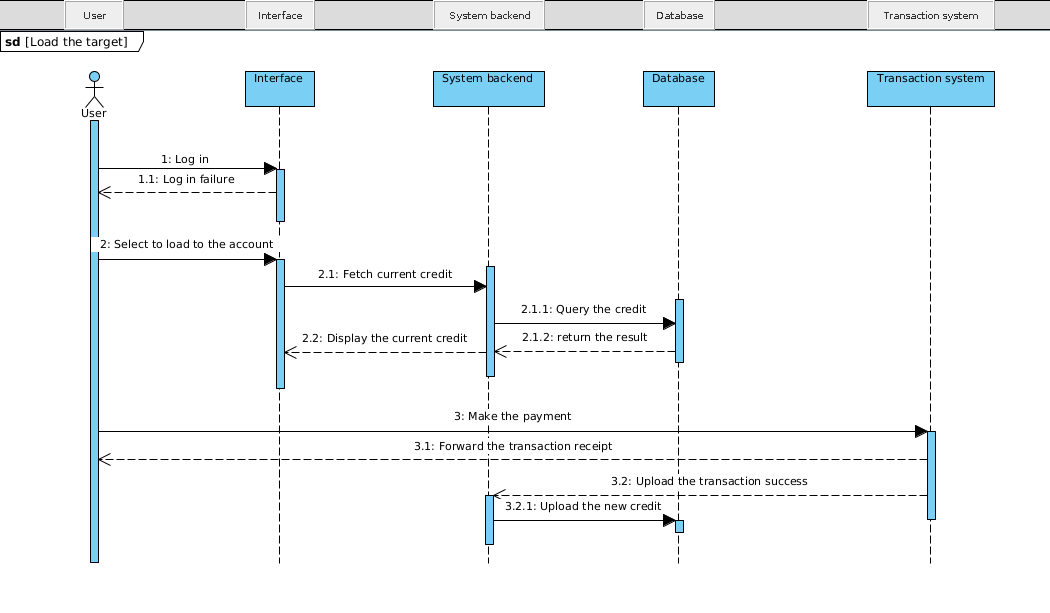
#### **3.1.3.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά**

Η αλληλουχία ενεργείων που θα ακολουθηθεί από το Use Case είναι:

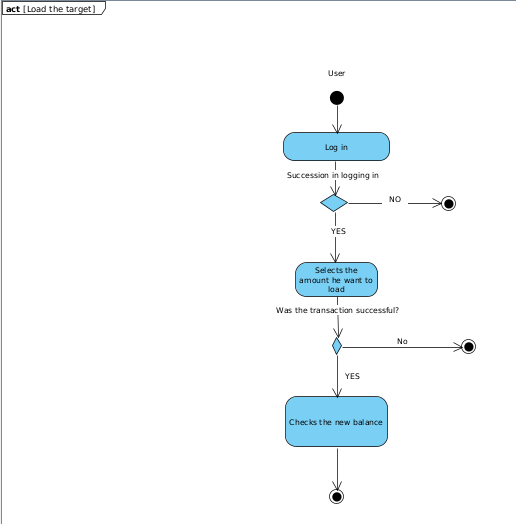
* Ο χρήστης- ιδιοκτήτης target συνδέεται στην εφαρμογή και ανοίγει την καρτέλα που περιγράψαμε παραπάνω
* Συμπληρώνει τα στοιχεία του και πατάει το κουμπί επαναφόρτιση
* Το API με την ενέργεια GET/PUT συλλέγει τα στοιχεία και στέλνει αίτημα στην αντίστοιχη τράπεζα του χρήστη για να γίνει η χρηματική συναλλαγή με την αντίστοιχη τράπεζα της εταιρίας.
* Μέσω της διεπαφής παίρνει ένα μήνυμα από την τράπεζα. Αν το μήνυμα είναι ότι η συναλλαγή επιτεύχθηκε, τότε το API εμφανίζει ένα μήνυμα ”Επιτυχία Συναλλαγής” στο front-end της εφαρμογής. Αντίστοιχα ενημερώνεται η βάση δεδομένων με το καινούριο υπόλοιπο.
* Αν η συναλλαγή απορριφθεί, τότε το API εμφανίζει ένα μήνυμα σφάλματος στο front-end της εφαρμογής.

Για να κατανοηθούν καλύτερα τα παραπάνω παραθέτουμε τα UML διαγράμματα δραστηριοτήτων(Activity) και αλληλουχίας(Sequence) της εφαρμογής μας.

* Διάγραμμα UML αλληλουχίας για Load the Target:



* Διάγραμμα UML δραστηριοτήτων για Load the Target:



#### **3.1.3.6 Δεδομένα εξόδου**

Τα δεδομένα εξόδου είναι το μήνυμα είτε επιτυχίας είτε σφάλματος της συναλλαγής. Αντίστοιχα στην main σελίδα της εφαρμογής όπου εμφανίζονται τα στοιχεία χρήστη, ανανεώνεται το υπόλοιπο του target εάν επιτευχθεί η συναλλαγή.

#### **3.1.3.7 Παρατηρήσεις**

Είναι αναγκαίο να επισημάνουμε ότι η συγκεκριμένη εφαρμογή υλοποιείται στα πλαίσια του μαθήματος Τεχνολογίας Λογισμικού της Σχολής ΗΜΜΥ του ΕΜΠ. Επομένως, είναι αδύνατον να μπορέσει το σύστημα μας να επικοινωνήσει μέσω κάποιας διεπαφής με πραγματικά συστήματα τράπεζας. Για αυτό το λόγο το παραπάνω κομμάτι θα υλοποιηθεί μέσω προσομοίωσης στο back-end. Δηλαδή θα θεωρούμε ότι όλες οι κάρτες είναι αποδεκτές και με άπειρο αριθμό χρημάτων.

### **3.1.4 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 4: (Data Extraction)**

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

Στην περίπτωση χρήσης (Data Extraction) οι χρήστες που εμπλέκονται είναι και οι ιδιοκτήτες των εταιριών διοδίων.

#### **3.1.4.2 Προϋποθέσεις εκτέλεσης**

Η προϋπόθεση εκτέλεσης της παραπάνω περίπτωσης χρήσης είναι να συνδεθεί οποιοσδήποτε χρήστης στην εφαρμογή μας και να επιλέξει την καρτέλα όπου εμφανίζονται διάφορα στατιστικά στοιχεία από τις διελεύσεις των target στα διόδια όλων των εταιριών.

#### **3.1.4.3 Περιβάλλον εκτέλεσης**

Το περιβάλλον στο οποίο εκτελείται η συγκεκριμένη περίπτωση χρήσης, είναι η διαδικτυακή διεπαφή με τον χρήστη. Ο χρήστης αφού συνδεθεί, επιλέγει την καρτέλα που εμφανίζει τα διάφορα στοιχεία από τις διελεύσεις στις εφτά οδούς.

#### **3.1.4.4 Δεδομένα εισόδου**

Δεδομένα εισόδου για το συγκεκριμένο Use Case είναι όλα τα δεδομένα από τις διελεύσεις των target από τις εφτά οδούς με διόδια οι οποίες είναι αποθηκευμένες στη βάση δεδομένων της εφαρμογής και συλλέγονται από τα αντίστοιχα συστήματα των οδών.

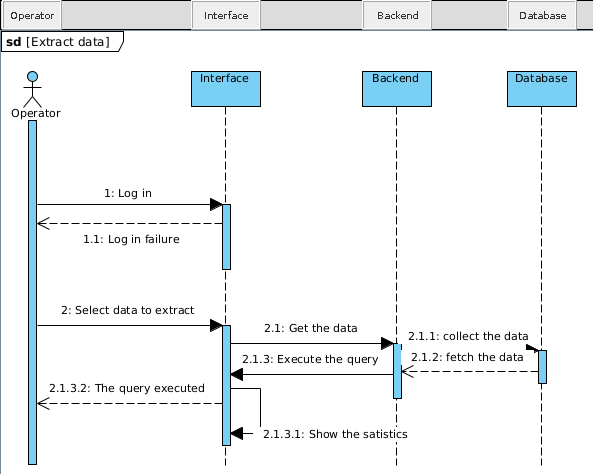
#### **3.1.4.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά**

Η αλληλουχία ενεργειών που ακολουθεί η περίπτωση χρήσης (Data Extraction) είναι η εξής:

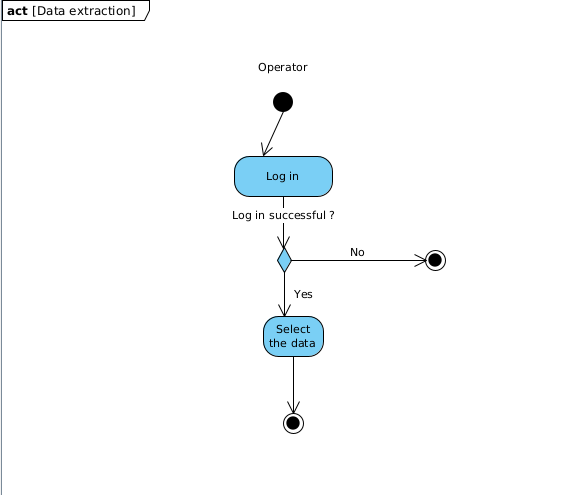
* Ο εκάστοτε χρήστης συνδέεται στην εφαρμογή.
* Επιλέγει την καρτέλα εξαγωγής δεδομένων. Το API επικοινωνεί με την βάση δεδομένων και αποστέλει μέσω διάφορων querries στο front-end της εφαρμογής στοιχεία από τις διελεύσεις.

Για να κατανοηθούν καλύτερα τα παραπάνω παραθέτουμε τα UML διαγράμματα δραστηριοτήτων(Activity) και αλληλουχίας(Sequence) της εφαρμογής μας.

* Διάγραμμα UML αλληλουχίας για Data Extraction:



* Διάγραμμα UML δραστηριοτήτων για Data Extraction:



#### **3.1.4.6 Δεδομένα εξόδου**

Δεδομένα εξόδου είναι τα δεδομένα από τις διελεύσεις,στατιστικά στοιχεία και διάφορα διαγράμματα τα οποία αντλούνται από την βάση δεδομένων και εμφανίζονται στο front-end της εφαρμογής.

#### **3.1.4.7 Παρατηρήσεις**

Είναι αναγκαίο να επισημάνουμε ότι η συγκεκριμένη εφαρμογή υλοποιείται στα πλαίσια του μαθήματος Τεχνολογίας Λογισμικού της Σχολής ΗΜΜΥ του ΕΜΠ. Επομένως, είναι αδύνατον να μπορέσει το σύστημα μας να επικοινωνήσει μέσω κάποιας διεπαφής με τα πραγματικά συστήματα των εφτά εταιριών. Για αυτό το λόγο το παραπάνω κομμάτι θα υλοποιηθεί μέσω προσομοίωσης στο back-end. Δηλαδή θα φορτώσουμε εμείς την βάση δεδομένων με συγκεκριμένα δεδομένα διέλευσης οχημάτων σε κάθε οδό και από εκεί θα εξάγουμε τα δεδομένα προς την εφαρμογή.

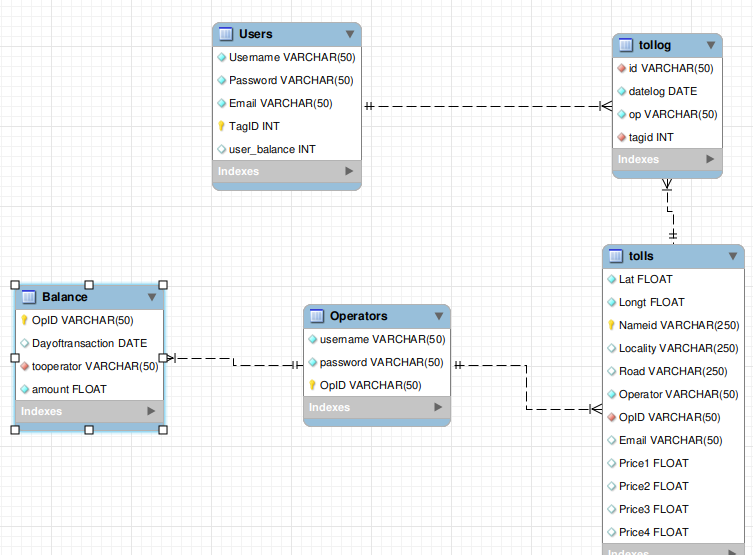
## **3.2 Απαιτήσεις επιδόσεων**

Δεδομένου ότι η εφαρμογή μας δημιουργείται στα πλαίσια μαθήματος όπως έχουμε αναφέρει παραπάνω, τα δεδομένα που εισάγουμε είναι λίγα και εισάγονται για να τεκμηριώσουν την ορθότητα της εφαρμογής, οι χρήστες είναι εφτά για κάθε οδό και δημιουργούμε κάποιους σχετικά λίγους (της τάξεως διψήφιου αριθμού) για ιδιοκτήτες target. Αντίστοιχα οι λειτουργικές απαιτήσεις και τα Use Cases είναι λίγα. Για παράδειγμα τα Use Cases είναι 4. Για αυτό το λόγο, αφού το φορτίο του λογισμικού είναι πολύ μικρό, η επιθυμητή επίδοση του συστήματος μας είναι σε όλες τις καταστάσεις να ανταποκρίνεται σε κλάσματα του δευτερολέπτου.

## **3.3 Απαιτήσεις οργάνωσης δεδομένων**

### **3.3.1 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα**

´Οσον αφορά τις απαιτήσεις και τους περιορισμούς της βάσης δεδομένων, αυτή καθορίζεται από τα δεδομένα και τους εμπλεκόμενους (Stakeholders) του συστήματος μας. Δηλαδή αυτή θα πρέπει να αποθηκεύει σε μία οντότητα τις ανά σύστημα εταιρίες με attributes τα στοιχεία διελεύσεων, σε άλλη οντότητα τα στοιχεία των τραπεζών, σε άλλες οντότητα τα στοιχεία των χρηστών της δικής μας εφαρμογής κλπ. Για την καλύτερη κατανόηση παραθέτουμε το ER διάγραμμα της Βάσης Δεδομένων μας.



## **3.4 Λοιπές απαιτήσεις**

### **3.4.1 Απαιτήσεις διαθεσιμότητας λογισμικού**

Για τις απαιτήσεις της διαθεσιμότητας του λογισμικού, δεδομένου ότι το λογισμικό δημιουργείται για ακαδημαϊκούς λόγους, η διαθεσιμότητα του θα είναι ελεύθερη στο github για οποιοδήποτε θελήσει μετά το πέρας του τρέχοντος ακαδημαϊκού εξαμήνου. Το λογισμικό ακολουθώντας τα βήματα που αναφέρονται στο github, μπορεί να εγκατασταθεί και να εκτελεστεί σε local επίπεδο σε οποιοδήποτε server.

### **3.4.2 Απαιτήσεις ασφάλειας**

Απαιτήσεις για την ασφάλεια του λογισμικού μας δεν έχουν υλοποιηθεί ακόμη, παρόλα αυτά θα υπάρξει μερίμνα στην επόμενη φάση ανάπτυξης του λογισμικού μας, κυρίως ως προς την ασφάλεια κατά την εγγραφή και είσοδο των χρηστών και κατά την κλήση διάφορων λειτουργιών του API. Όπως προαναφέρθηκε το λογισμικό δημιουργείται για ακαδημαϊκούς λόγους και στα πλαίσια μαθήματος της Σχολής ΗΜΜΥ. Το λογισμικό θα είναι open source για οποιοδήποτε θελήσει να το εγκαταστήσει και είτα να το αλλάξει είτε να το επεκτείνει.