Έγγραφο απαιτήσεων λογισμικού (SRS)

TOLLSΕ21

# Εισαγωγή

## 1.1 Εισαγωγή: σκοπός του λογισμικού

Σκοπός του λογισμικού είναι η επίτευξη της διαλειτουργικότητας στα διόδια, δηλαδή η παροχή ενός πληροφοριακού συστήματος το οποίο θα προσφέρει δυνατότητα πρόσβασης σε όλους τους stakeholders και θα καθιστά εύκολες τις καταγραφές, εκκαθαρίσεις και συμψηφισμούς οφειλών διαφορετικών εταιριών διαχείρισης των διοδίων, ενώ παράλληλα θα επεξεργάζεται τα δεδομένα και θα παρέχει στατιστικά στοιχεία.

## 1.2 Διεπαφές (interfaces)

### 1.2.1 Διεπαφές με εξωτερικά συστήματα

### 1.2.2 Διεπαφές με το χρήστη

# Αναφορές - πηγές πληροφοριών

# Προδιαγραφές απαιτήσεων λογισμικού

## 3.1 Περιπτώσεις χρήσης

### 3.1.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: Λήψη και καταχώρηση δεδομένων διέλευσης

#### 3.1.1.1 Χρήστες (ρόλοι) που εμπλέκονται

Εμπλέκονται οι λειτουργοί/χειριστές των σταθμών των διοδίων.

#### 3.1.1.2 Προϋποθέσεις εκτέλεσης

Πρέπει σε όλα τα σημεία διελεύσεων είναι διαθέσιμες οι λειτουργίες "provider", "balance" και "pass". Επιπρόσθετα, πρέπει να μπορούν αυτά τα δεδομένα να μεταφερθούν στη βάση δεδομένων μας, μέσω κάποιας σύνδεσης.

#### 3.1.1.3 Περιβάλλον εκτέλεσης

Η περίπτωση χρήσης εκτελείται στη διαδικτυακή διεπαφή που συνδέει τους λειτουργούς με τη βάση δεδομένων μας.

#### 3.1.1.4 Δεδομένα εισόδου

Τα δεδομένα εισόδου είναι αυτά που λαμβάνονται από τους χειριστές και η εγκυρότητά τους εξαρτάται από τον εξοπλισμό στους σταθμούς διέλευσης, καθώς επίσης και τις ενέργειες των χειριστών. Τα δεδομένα εξόδου ταυτίζονται με τα δεδομένα εισόδου, επομένως έχουν την ίδια εγκυρότητα.

#### 3.1.1.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

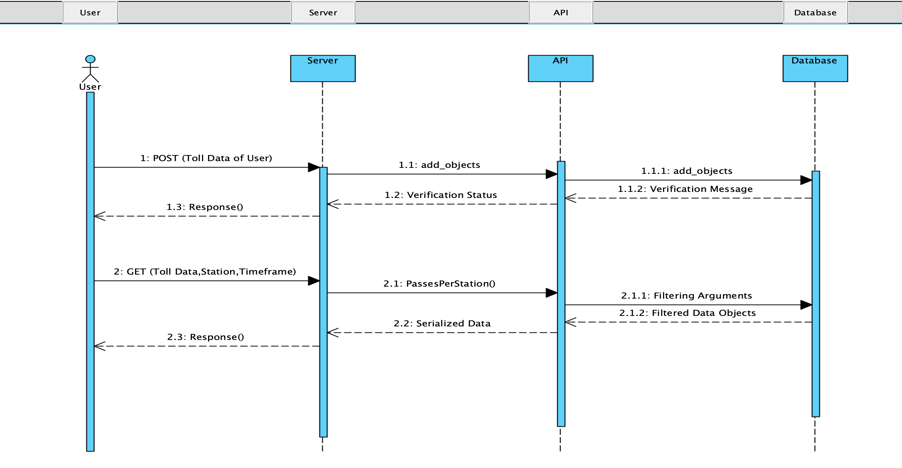
1. Ανά τακτά χρονικά διαστήματα, λαμβάνονται από τους σταθμούς διέλευσης τα δεδομένα των διελεύσεων που μεσολάβησαν από την προηγούμενη αποστολή. Αυτά περιλαμβάνουν την ταυτότητα του παρόχου του tag, το κόστος διέλευσης και τα δεδομένα γεγονότος διέλευσης.

2. Όλα τα παραπάνω αποθηκεύονται στη βάση δεδομένων.

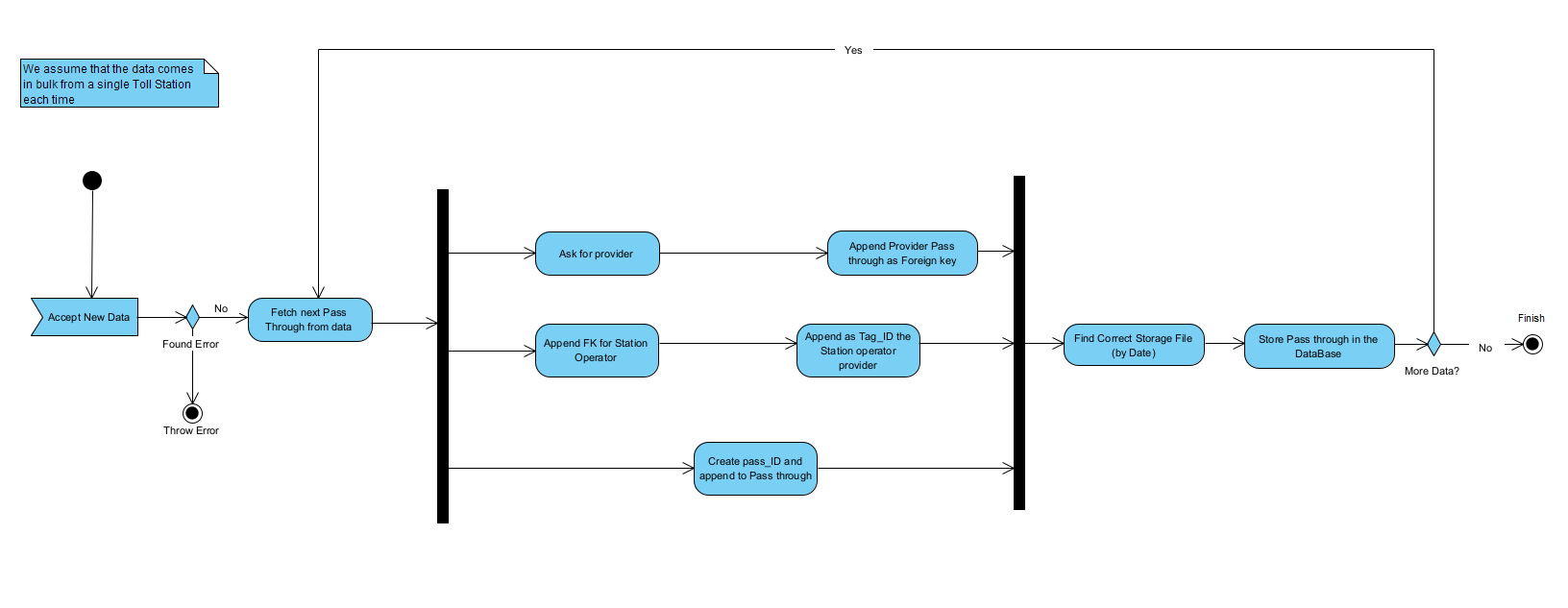
Προκύπτει απρόβλεπτη κατάσταση αν, τη στιγμή που ένας σταθμός διέλευσης πρέπει να στείλει τα δεδομένα που συνέλεξε, χαθεί η σύνδεση. Σε αυτή την περίπτωση, τα δεδομένα αυτά στέλνονται μαζί με τα επόμενα δεδομένα που θα συλλεχθούν στον επόμενο χρόνο αποστολής δεδομένων.

Υπάρχει η πιθανότητα σφάλματος αν καταγραφούν (από τα μηχανήματα) ή σταλούν (από τον χειριστή) λανθασμένα δεδομένα από κάποιον σταθμό. Το σφάλμα μπορεί να διορθωθεί αν τα δεδομένα αυτά προσδιοριστούν από τον σταθμό (ώρα αποστολής, περιεχόμενο) και δοθούν τα σωστά δεδομένα προς αντικατάστασή τους.

Διάγραμμα UML sequence:



Διάγραμμα UML activity:



#### 3.1.1.7 Δεδομένα εξόδου

#### 3.1.1.8 Παρατηρήσεις

### 3.1.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: Υπολογισμός και διαχείριση οφειλών

#### 3.1.2.1 Χρήστες (ρόλοι) που εμπλέκονται

Εμπλέκονται οι εταιρίες και οι τράπεζες.

#### 3.1.2.2 Προϋποθέσεις εκτέλεσης

Για να μπορέσει να εκτελεστεί η περίπτωση χρήσης, πρέπει πρώτα να υπάρχουν δεδομένα στη βάση (use case 3.1.1). Επιπλέον, πρέπει να έχει καταχωρηθεί στο σύστημα ο τραπεζικός λογαριασμός κάθε εταιρίας και ανά τακτά χρονικά διαστήματα οι εταιρίες να προσδιορίζουν την ημερομηνία και τον τρόπο (πληρωμή ολόκληρου ποσού, συμψηφισμός) πληρωμής των οφειλών τους.

#### 3.1.2.3 Περιβάλλον εκτέλεσης

Η περίπτωση χρήσης εκτελείται στη βάση δεδομένων (DBMS), όμως περιλαμβάνει και διεπαφές με τις τράπεζες. Επίσης, αλλαγές πραγματοποιούν και οι εταιρίες μέσω της διεπαφής χρήστη.

#### 3.1.2.4 Δεδομένα εισόδου

Τα δεδομένα εισόδου σχετικά με τον υπολογισμό των οφειλών λαμβάνονται από τη βάση δεδομένων και έχουν την εγκυρότητα που εξηγήθηκε στο 3.1.1. Για τις διάφορες επιλογές πληρωμών, χρειάζονται ενέργειες (επιπλέον δεδομένα εισόδου) από τις εταιρείες, η εγκυρότητα των οποίων εξαρτάται από τα μέλη τους που χειρίζονται την εφαρμογή. Τα δεδομένα εξόδου είναι έγκυρα, με την προϋπόθεση ότι είναι και τα δεδομένα εισόδου.

#### 3.1.2.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

1. Λαμβάνονται δεδομένα από τη βάση δεδομένων.

2. Αν ο πάροχος του tag διαφέρει από την εταιρία της οδού, αυξάνεται, με βάση τα υπόλοιπα δεδομένα, η οφειλή του παρόχου προς την εταιρία.

3. Στέλνονται ειδοποιήσεις σε όλες τις εμπλεκόμενες εταιρείες και αναμένεται απάντηση για τη διαχείριση των οφειλών.

3a. Συμψηφισμός των οφειλών. Οι οφειλές διαγράφονται και δημιουργείται μόνο μία καινούργια με τη διαφορά τους (αν δεν είναι μηδενική).

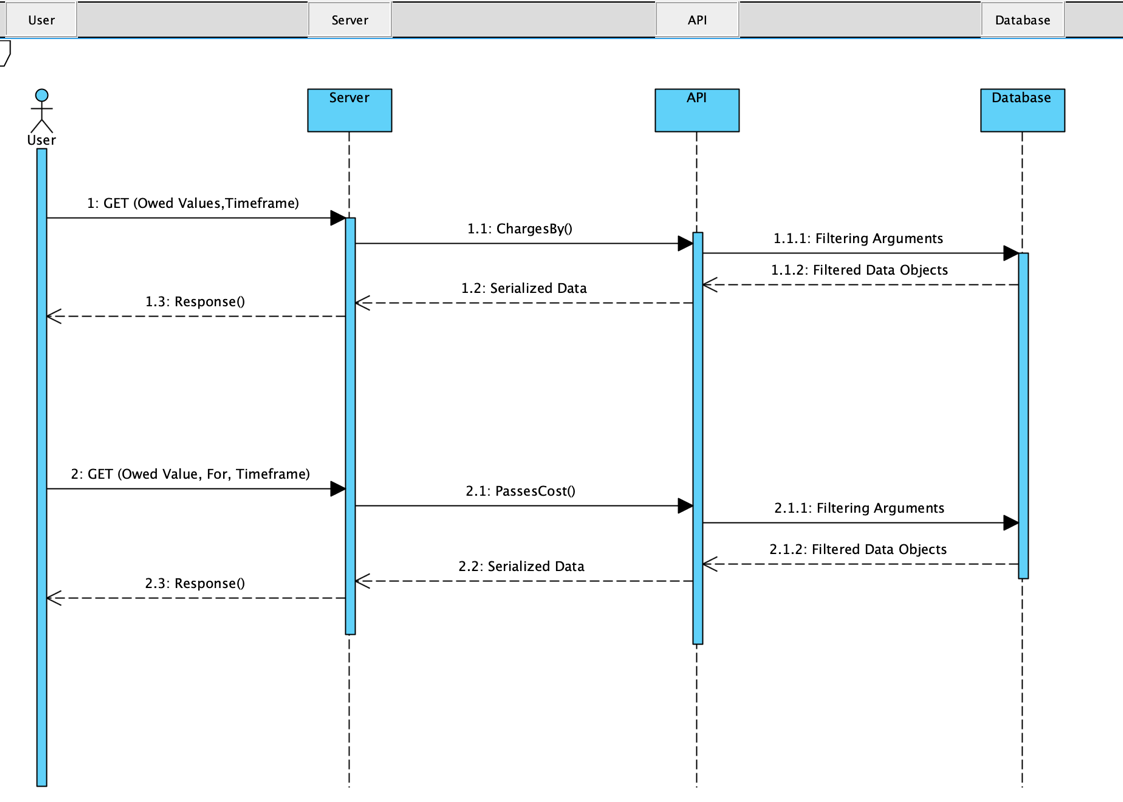
3b. Πληρώνεται μία οφειλή. Διαγράφεται από το σύστημα.

3c. Δεν πραγματοποιείται κάποια ενέργεια.

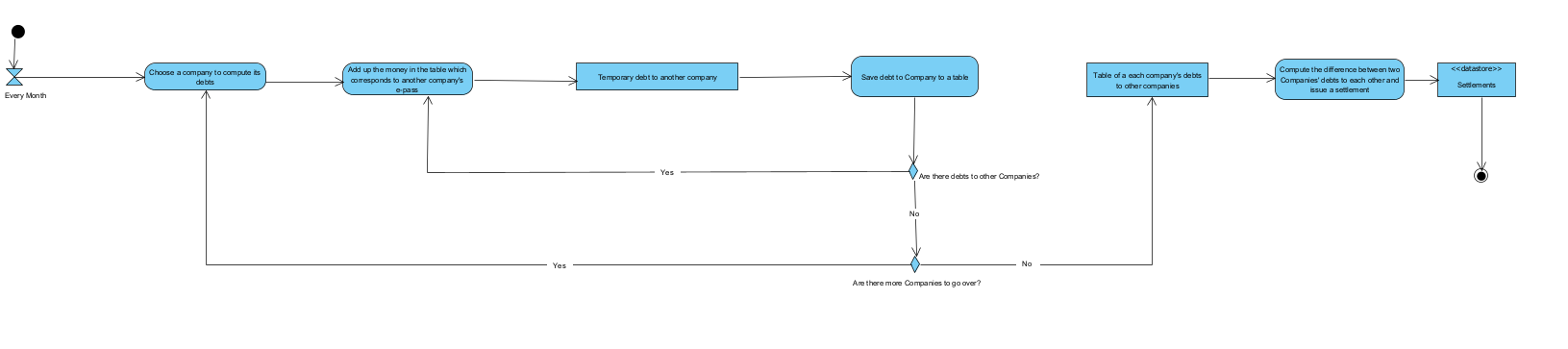
Σφάλμα:

Μπορεί κάποιος χρήστης της εφαρμογής να επιλέξει κατά λάθος μία ρύθμιση σχετική με τη διαχείριση των οφειλών. Αυτό λύνεται αν προστεθεί ένα χρονικό διάστημα στο οποίο κάθε ενέργεια μπορεί να αφαιρεθεί, πριν οριστικοποιηθούν οι αλλαγές.

Διάγραμμα UML sequence:



Διάγραμμα UML activity:



#### 3.1.2.7 Δεδομένα εξόδου

#### 3.1.2.8 Παρατηρήσεις

## 3.2 Απαιτήσεις επιδόσεων

Οι χρήστες θα θέλουν να δουν είτε τις οφειλές τους, είτε τα στατιστικά του μήνα. Θέλουμε και στις δύο περιπτώσεις τα δεδομένα/διαγράμματα αυτά να φορτώνονται σχετικά γρήγορα (να μην χρειάζεται πάνω από 1 λεπτό).

## 3.3 Απαιτήσεις οργάνωσης δεδομένων

### 3.3.1 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα

Η εκάστοτε εταιρία θα πρέπει να έχει πρόσβαση στις οφειλές της, καθώς και στις οφειλές άλλων εταιριών που την αφορούν. Όλοι οι χρήστες θα πρέπει να έχουν πρόσβαση στα στατιστικά.

Κανείς (εκτός από εμάς) δεν χρειάζεται και δεν πρέπει να έχει πρόσβαση στα επιμέρους στοιχεία.

## 3.5 Λοιπές απαιτήσεις

### 3.5.1 Απαιτήσεις διαθεσιμότητας λογισμικού

Οι χρήστες θα πρέπει να έχουν πρόσβαση στην εφαρμογή οποιαδήποτε στιγμή της ημέρας.

### 3.5.2 Απαιτήσεις ασφάλειας

Θα πρέπει να υπάρχουν τα κατάλληλα πρωτόκολλα ώστε να μην γίνεται υποκλοπή των δεδομένων της βάσης. Επίσης, τα στελέχη της εταιρίας που διαχειρίζονται τις πληρωμές θα πρέπει να ολοκληρώνουν μια διαδικασία ταυτοποίησης πριν κάνουν σημαντικές αλλαγές.