Έγγραφο απαιτήσεων λογισμικού (SRS)

Διαλειτουργικότητα στα διόδια αυτόματης διέλευσης

# 1. Εισαγωγή

## 1.1 Εισαγωγή: σκοπός του λογισμικού

Σκοπός του λογισμικού αυτού είναι η δημιουργία ενός συστήματος το οποίο θα είναι υπεύθυνο για τις οφειλές μεταξύ των εταιρειών διαχείρισης των αυτοκινητοδρόμων μετά τον συμψηφισμό τους ( ή μετά τον συμψηφισμό των επιμέρους χρεώσεων-οφειλών). Η εφαρμογή αυτή απευθύνεται στις εταιρείες διαχείρισης των αυτοκινητοδρόμων, οι οποίες μέσω της εφαρμογής μπορούν να έχουν πρόσβαση στα δεδομένα των πομποδεκτών αλλά και να έχουν την πλήρη εικόνα των οφειλών τους από και προς τις άλλες εταιρείες.

Οι εταιρείες αφού εισαχθούν στο σύστημα ως διαπιστευμένοι χρήστες έχουν τις εξής δυνατότητες:

* Να επιλέξουν κάποιον σταθμό και μία χρονική περίοδο και να δουν αναλυτική λίστα με όλες τις διελεύσεις από τον σταθμό αυτό στη συγκεκριμένη περίοδο,
* Να επιλέξουν δύο εταιρείες και μία χρονική περίοδο και να δουν αναλυτική λίστα με όλες τις διελεύσεις που έγιναν από σταθμούς της μιας εταιρείας με tag της άλλης στη συγκεκριμένη περίοδο,
* Να επιλέξουν δύο εταιρείες και μια χρονική περίοδο και να δουν τον αριθμό των διελεύσεων από σταθμούς της μιας εταιρείας με tag της άλλης, καθώς και το συνολικό κόστος οφειλής για αυτές τις διελεύσεις στη συγκεκριμένη περίοδο και
* Να επιλέξουν μια εταιρεία και μία χρονική περίοδο και να δουν τον αριθμό των διελεύσεων από σταθμούς αυτής της εταιρείας με tag όλων των υπόλοιπων εταιρειών, καθώς και το κόστος οφειλής που αντιστοιχεί στην καθεμία από αυτές για την περίοδο αυτή.

Εκτός των παραπάνω λειτουργιών θα υπάρχει και ένας διαχειριστής του συστήματος (administrator) ο οποίος θα μπορεί να ελέγχει τη σύνδεση του συστήματος με τη βάση δεδομένων, καθώς και να την αρχικοποιεί με συγκεκριμένα δεδομένα.

## 1.2 Διεπαφές (interfaces)

### 1.2.1 Διεπαφές με εξωτερικά συστήματα

* Διεπαφές με τραπεζικά συστήματα:

διενέργεια συναλλαγών μεταξύ των παρόχων (ολοκλήρωση πληρωμών των οφειλών)

διενέργεια συναλλαγών των παρόχων με τους χρήστες (φόρτιση tags)

* Διεπαφές με συστήματα διέλευσης και φόρτισης:

ανέπαφη και αυτόματη αναγνώριση των tags και σύνδεση με τη βάση δεδομένων

### 1.2.2 Διεπαφές με το χρήστη

* Frontend εφαρμογής:
  + - Χειρισμός του λογισμικού από εκπροσώπους των εταιρειών διοδίων, μέσω ιστοσελίδας σε κάποιο διακομιστή, για τις διάφορες λειτουργίες του λογισμικού στις οποίες συμμετέχει ο χρήστης-εταιρεία και από κάποιον διαχειριστή του συστήματος που θα έχει πρόσβαση σε διαχειριστικές λειτουργίες.
    - Χειρισμός του λογισμικού από τον διαχειριστή, μέσω ιστοσελίδας σε κάποιο διακομιστή, για έλεγχο και διατήρηση του συστήματος σε εύρυθμη λειτουργική κατάσταση.

* Command Line Interface :
  + - Χειρισμός του λογισμικού από εκπροσώπους των εταιρειών διοδίων, μέσω της χρήση ενός τερματικού, για τις διάφορες λειτουργίες του λογισμικού στις οποίες συμμετέχει ο χρήστης-εταιρεία και από κάποιον διαχειριστή του συστήματος που θα έχει πρόσβαση σε διαχειριστικές λειτουργίες.
    - Χειρισμός του λογισμικού από τον διαχειριστή, μέσω της χρήσης ενός τερματικού, για έλεγχο και διατήρηση του συστήματος σε εύρυθμη λειτουργική κατάσταση.

2. Αναφορές - πηγές πληροφοριών

Ν/Α

3. Προδιαγραφές απαιτήσεων λογισμικού

## 3.1 Περιπτώσεις χρήσης

### 3.1.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: Διέλευση οχήματος από διόδια

#### 3.1.1.1 Χρήστες (ρόλοι) που εμπλέκονται

* Ιδιοκτήτες οχημάτων
* Πάροχος πομποδέκτη
* Διόδια

#### 3.1.1.2 Προϋποθέσεις εκτέλεσης

* Ύπαρξη δεδομένων χρήστη στο σύστημα

#### 3.1.1.3 Περιβάλλον εκτέλεσης

* Back-end server
* Database server
* Command Line Interface
* Application Programmable Interface

#### 3.1.1.4 Δεδομένα εισόδου

Το σύστημα απαιτεί από τον χρήστη τις παρακάτω εισόδους :

Στοιχεία αυτοκινήτου

* Vehicle ID

Tag

* Tag ID

Διόδια

* Station ID

Ποσό πληρωμής

* Fee

Πότε συνέβη η διέλευση

* Τimestamp

#### 3.1.1.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

1. Σφάλμα σε περίπτωση μη επαρκούς χρηματικού υπολοίπου
2. Ενημέρωση χρηματικού υπολοίπου του tag έπειτα από επιτυχημένη διέλευση
3. Καταγραφή διέλευσης μεταξύ παρόχου και διοδίου

Ακολουθεί το Διάγραμμα Activity UML :

#### 

#### 3.1.1.7 Δεδομένα εξόδου

Το σύστημα επιστρέφει τις παρακάτω εξόδους.

Συνθήκες εγκυρότητας θεωρούνται παντού:

1. η ύπαρξη των δεδομένων του χρήστη στο σύστημα
2. επαρκές υπόλοιπο για πραγματοποίηση πληρωμής τέλους διοδίου

Στοιχεία αυτοκινήτου

* Vehicle ID

Tag

* Tag ID

Αποτέλεσμα πληρωμής εξαρτάται από το εάν υπάρχει επαρκές υπόλοιπο ή όχι)

Χρονική στιγμή διέλευσης (Timestamp)

### 

### 3.1.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: Υπολογισμός οφειλών μεταξύ εταιρειών διοδίων

#### 3.1.2.1 Χρήστες (ρόλοι) που εμπλέκονται

Εταιρείες διοδίων (operators)

#### 3.1.2.2 Προϋποθέσεις εκτέλεσης

Καταγεγραμμένες διελεύσεις οχημάτων και αντίστοιχων παρόχων (αριθμός διελεύσεων οχημάτων που ανήκουν σε έναν πάροχο και πέρασαν από άλλο πάροχο μέσα σε συγκεκριμένο χρονικό διάστημα και ποιοι είναι οι αντίστοιχοι πάροχοι)

#### 3.1.2.3 Περιβάλλον εκτέλεσης

* Front-end server
* Command Line Interface
* Back-end server
* Database server
* Application Programming Interface

#### 

#### 3.1.2.4 Δεδομένα εισόδου

Στοιχεία Παρόχων:

* operatorID1
* operatorID2

Ημερομηνίες χρονικής περιόδου οφειλών:

* dateFrom
* dateTo

Ποσό οφειλής:

* amount

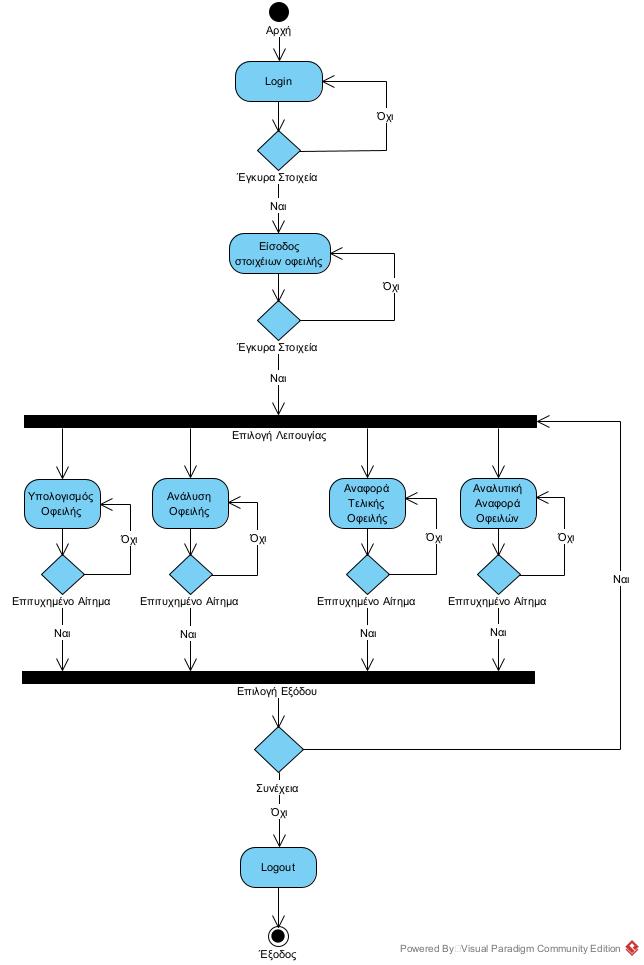
Κατεύθυνση οφειλής:

* operatorDebited
* operatorCredited

#### 3.1.2.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

Η επιθυμητή συμπεριφορά αποτελείται από τα παρακάτω βήματα:

* Είσοδος του παρόχου στο λογισμικό (operatorID, password)
* Επιλογή ενέργειας (υπολογισμός οφειλών)
* Εισαγωγή δεδομένων (επιθυμητό χρονικό διάστημα - ημερομηνίες)
* Έξοδος δεδομένων (οφειλή από ή προς κάθε πάροχο του δικτύου, ποσό οφειλής)

Ακολουθεί το Διάγραμμα Activity UML :

#### 3.1.2.7 Δεδομένα εξόδου

Το σύστημα επιστρέφει τις παρακάτω εξόδους:

Υπολογισμός οφειλής:

* amount
* operatorDebited
* operatorCredited

Ανάλυση Οφειλής:

* [chargeRecordList1]
* [chargeRecordList2]

Αναφορά Τελικής Οφειλής:

* amount
* direction

Αναλυτική Αναφορά Οφειλής:

* [chargeRecordList1]
* [chargeRecordList2]

Συνθήκες εγκυρότητας θεωρούνται:

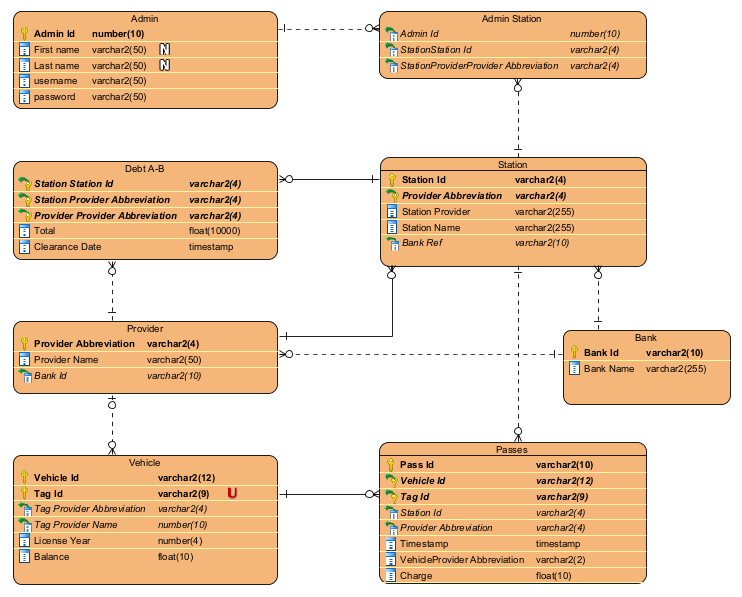
* Έγκυρα στοιχεία login
* Έγκυρα στοιχεία εισόδου (εταιρείες, ημερομηνίες)
* Επιτυχημένη επιλογή λειτουργίας

## 3.2 Απαιτήσεις επιδόσεων

Το λογισμικό θα πρέπει να ανταποκρίνεται και να εξυπηρετεί όλους τους χρήστες ταυτόχρονα ακόμα και σε ώρες αιχμής. Επομένως, χρειάζεται να είναι συνεχώς διαθέσιμο για ταυτόχρονη χρήση χρηστών με πλήθος ίσο με τις γραμμές διοδίων όλων των εταιρειών. Για το πρόβλημά μας θεωρούμε ότι έχουμε 500 τέτοιες θέσεις εξυπηρέτησης. Επιπλέον, το λογισμικό θα πρέπει να μπορεί να εξυπηρετεί ταυτόχρονα και requests από όλους τους διαθέσιμους παρόχους, ώστε να μπορούν να χρησιμοποιούν ταυτόχρονα το λογισμικό και αυτό να αποκρίνεται άμεσα. Θεωρούμε αριθμό 50 requests ανά δευτερόλεπτο για αυτόν τον σκοπό. Τέλος, οριοθετούμε την ολοκλήρωση κάθε λειτουργίας σε λιγότερο από 3 δευτερόλεπτα για τα requests των παρόχων όσο και για την απόκριση προς τους χρήστες (εμφάνιση δεδομένων εξόδων), ώστε να εξυπηρετούνται όλοι άμεσα.

## 3.3 Απαιτήσεις οργάνωσης δεδομένων

### 3.3.1 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα



## 3.4 Λοιπές απαιτήσεις

### 3.4.1 Απαιτήσεις διαθεσιμότητας λογισμικού

Το λογισμικό χρειάζεται να είναι διαθέσιμο οποιαδήποτε στιγμή

* Χρειάζονται αντίγραφα της βάσης δεδομένων και εναλλακτικοί servers για τη μη-διακοπή λειτουργίας του συστήματος σε καμία περίπτωση (π.χ. συντήρηση server)
* Συνέπεια στη δυνατότητα πρόσβασης των χρηστών
* Σύνδεση με πάροχο υπηρεσιών διαδικτύου για την πλήρη διαθεσιμότητα των λειτουργιών του συστήματος μέσω διαδικτύου
* Σύνδεση με πάροχο διασύνδεσης με τράπεζες για την εξυπηρέτηση των πληρωμών

### 3.4.2 Απαιτήσεις ασφάλειας

* Υποστήριξη πρωτοκόλλου HTTPS για όλες τις διεπαφές, μέσω self-signed certificate
* Ασφαλής προσπέλαση δεδομένων
* Κρυπτογράφηση κωδικών πρόσβασης και στοιχείων χρηστών
* Μη επίτρεψη πρόσβασης στα προσωπικά στοιχεία των χρηστών