Έγγραφο απαιτήσεων λογισμικού (SRS)

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

*ΣΗΜΕΙΩΣΗ: Το στυλ κειμένου (style) "Description" χρησιμοποιείται για να σας δώσει σύντομες οδηγίες για κάθε ενότητα. Το κείμενο που θα συμπληρώσετε πρέπει να γραφεί με το στυλ "Normal"*

EasyPass

# Εισαγωγή

## 1.1 Εισαγωγή: σκοπός του λογισμικού

*Οριοθέτηση του σκοπού του συστήματος*

Σκοπός του συστήματος EasyPass είναι η υλοποίηση της διαλειτουργικότητας στα διόδια.

Συγκεκριμένα, παρέχει στους οδηγούς, οι οποίοι έχουν προμηθευτεί πομπό από κάποιον από τους ιδιοκτήτες των αυτοκινητόδρομων, τη δυνατότητα να διέρχονται αυτόματα από τους σταθμούς διοδίων οποιουδήποτε ιδιοκτήτη. Το σύστημα EasyPass, δηλαδή, είναι υπεύθυνο για την συγκέντρωση και διαχείριση των οφειλών μεταξύ ιδιοκτητών, οι οποίες δημιουργούνται κατά τη διέλευση ενός οχήματος από σταθμό ιδιοκτήτη διαφορετικού από τον πάροχο του πομπού του οχήματος.

Οι οφειλές αυτές ομαδοποιούνται και η εκκαθάριση πραγματοποιείται σύμφωνα με παραμέτρους τις οποίες έχουν προσυμφωνήσει οι ενδιαφερόμενοι ιδιοκτήτες, σε συνεργασία με payment service provider(s), που αναλαμβάνουν την διεκπεραίωση των τραπεζικών συναλλαγών.

Επιπλέον, το σύστημα EasyPass παρέχει μια διαδικτυακή εφαρμογή, στην οποία οι ιδιοκτήτες μπορούν να παρακολουθήσουν την κατάσταση των οφειλών τους προς κάθε άλλο ιδιοκτήτη. Επιπροσθέτως, μέσω της εφαρμογής αυτής, το Υπουργείο Μεταφορών αποκτά πρόσβαση σε χρήσιμα στατιστικά δεδομένα σχετικά με την κίνηση στους εν λόγω αυτοκινητοδρόμους.

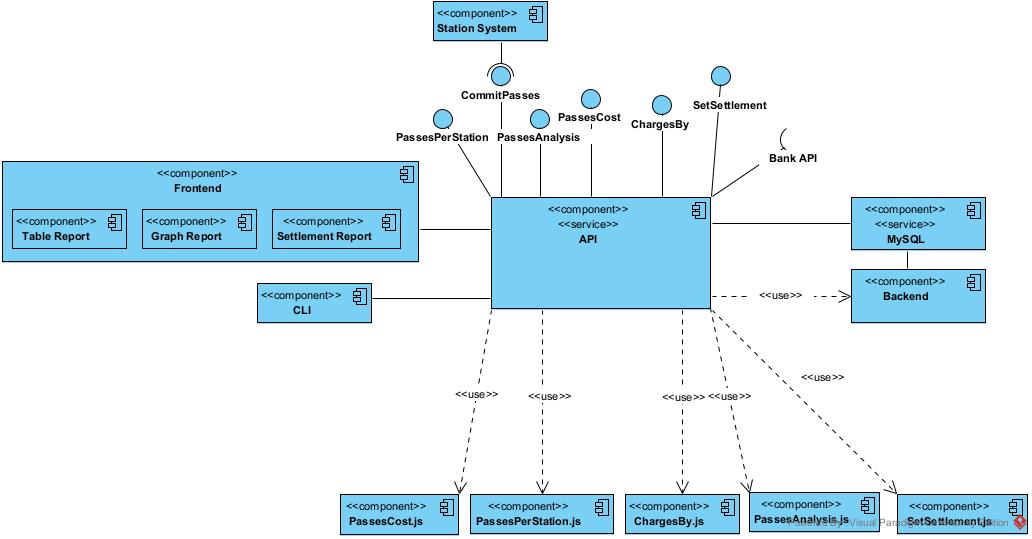
## 1.2 Διεπαφές (interfaces)

### 1.2.1 Διεπαφές με εξωτερικά συστήματα

*Καταγραφή διεπαφών με εξωτερικά συστήματα και λογισμικό. Αναφορά σε πρότυπα ανταλλαγής δεδομένων και κλήσης υπηρεσιών. Ενσωμάτωση διαγραμμάτων UML component.*

*ΝΑ ΜΗΝ ΣΥΜΠΛΗΡΩΘΕΊ ΑΝ ΔΕΝ ΑΠΑΙΤΕΙΤΑΙ*

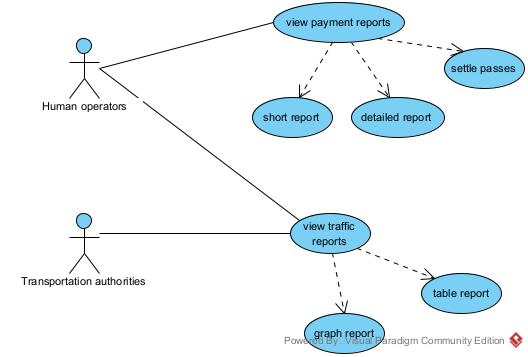
Στο τέλος κάθε ημέρας , το σύστημα κάθε σταθμού διοδίων (Station System) ενημερώνει το σύστημα μας με τη λίστα των διελεύσεων της ημέρας, χρησιμοποιώντας το αντίστοιχο endpoint (CommitPasses) του API.



### 1.2.2 Διεπαφές με το χρήστη

*Προδιαγραφή διεπαφών με το χρήστη, εφόσον υπάρχουν. Χρήση εργαλείων τύπου wireframes ή άλλων.*

*ΝΑ ΜΗΝ ΣΥΜΠΛΗΡΩΘΕΙ ΑΝ ΔΕΝ ΑΠΑΙΤΕΙΤΑΙ*

**

# Αναφορές - πηγές πληροφοριών

*Αναφορά σε πηγές πληροφοριών στο μέτρο της αναγκαιότητας για την κατανόηση του συστήματος*

*ΝΑ ΜΗΝ ΣΥΜΠΛΗΡΩΘΕΙ ΑΝ ΔΕΝ ΑΠΑΙΤΕΙΤΑΙ*

# Προδιαγραφές απαιτήσεων λογισμικού

## 3.1 Περιπτώσεις χρήσης

*Λεπτομερής προδιαγραφή των λειτουργιών του λογισμικού σε επίπεδο περιπτώσεων χρήσης. Ο αριθμός των περιπτώσεων χρήσης ανάλογα με τον αριθμό των μελών της ομάδας σύμφωνα με την εκφώνηση.*

### 3.1.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: Καταγραφή νέας χρέωσης

#### 3.1.1.1 Χρήστες (ρόλοι) που εμπλέκονται

*Αναφορά στους ρόλους που αφορά η περίπτωση χρήσης*

Οι ρόλοι που εμπλέκονται είναι οι εξής:

* σύστημα διέλευσης του σταθμού

#### 3.1.1.2 Προϋποθέσεις εκτέλεσης

*Καταγραφή των συνθηκών που πρέπει να ισχύουν ώστε να μπορεί να εκτελεστεί η περίπτωση χρήσης*

* Επιτυχής αναγνώριση του tag
* Σύνδεση στο διαδίκτυο

#### 3.1.1.3 Περιβάλλον εκτέλεσης

*Αναφορά στο περιβάλλον στο οποίο εκτελείται η περίπτωση χρήσης. Πχ "διαδικτυακή διεπαφή χρήστη", "DBMS" κλπ*

Η περίπτωση χρήσης εκτελείται σε υπολογιστή του σταθμού διοδίων. Τα δεδομένα που παράγονται αποθηκεύονται σε πρώτη φάση σε τοπική βάση δεδομένων και ανά τακτά χρονικά διαστήματα αποστέλλονται στην κεντρική βάση δεδομένων.

#### 3.1.1.4 Δεδομένα εισόδου

*Καταγραφή δεδομένων εισόδου και εξόδου και συνθηκών εγκυρότητας αυτών.*

Οι συνθήκες εγκυρότητας για κάθε δεδομένο είναι ο τύπος δεδομένων που χρησιμοποιούμε για το αντίστοιχο πεδίο στη βάση.

* tagID
* vehicle\_ref
* timestamp
* charge

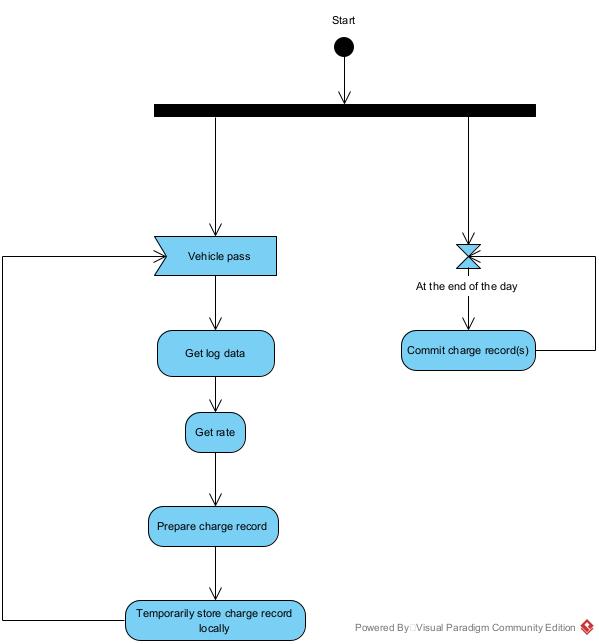
#### 3.1.1.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

*Περιγραφή με κείμενο (Βήμα 1, Βήμα 2 κλπ) και διαγράμματα UML αλληλουχίας (Sequence) και δραστηριοτήτων (Activity). Περιλαμβάνεται η συμπεριφορά σε απρόβλεπτες καταστάσεις και σφάλματα (εναλλακτικές ροές).*

Όταν εκκινεί το σύστημα, αρχίζουν να εκτελούνται ταυτόχρονα δύο νήματα. Το πρώτο (δεξί στο σχήμα), απλώς υποβάλει τα αποθηκευμένα charge records στο τέλος κάθε ημέρας.

Το δεύτερο νήμα ακολουθεί αυτά τα βήματα:

1. Περιμένει μέχρι να έρθει σήμα από το σύστημα του σταθμού ότι έφτασε όχημα στα διόδια
2. Λαμβάνει τις πληροφορίες της διέλευσης από το σύστημα του σταθμού
3. Αν ο πάροχος του tag ταυτίζεται με τον πάροχο του σταθμού, επιστρέφει στο Βήμα 1
4. Αλλιώς, λαμβάνει το κόστος διέλευσης από το σύστημα του σταθμού
5. Ετοιμάζει το charge record από τα δεδομένα που έχει
6. Αποθηκεύει προσωρινά το charge record και επιστρέφει στο Βήμα 1



#### 3.1.1.7 Δεδομένα εξόδου

*Διαγράμματα UML αλληλουχίας για την παραγωγή δεδομένων εξόδου. Ως δεδομένα εξόδου νοούνται όλα τα δεδομένα του συστήματος τα οποία δημιουργούνται ή μεταβάλλονται κατά την εκτέλεση (αν υπάρχουν τέτοια)*

Τα δεδομένα εξόδου σε αυτή τη περίπτωση είναι τα δεδομένα που δημιουργούμε και στέλνουμε στη βάση, δηλαδή τα charge records.

#### 3.1.1.8 Παρατηρήσεις

*Ο,τι δεν εντάσσεται στα προηγούμενα, εφόσον υπάρχει*

### 3.1.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: Έλεγχος κίνησης

#### 3.1.2.1 Χρήστες (ρόλοι) που εμπλέκονται

*Αναφορά στους ρόλους που αφορά η περίπτωση χρήσης*

Η περίπτωση χρήσης αφορά τις αρχές που επιβλέπουν την κίνηση στους αυτοκινητόδρομους (Υπουργείο Μεταφορών), καθώς και τους διαχειριστές των αυτοκινητόδρομων , αν θέλουν να συλλέξουν στατιστικά στοιχεία για την κίνηση των οχημάτων σε αυτούς.

#### 3.1.2.2 Προϋποθέσεις εκτέλεσης

*Καταγραφή των συνθηκών που πρέπει να ισχύουν ώστε να μπορεί να εκτελεστεί η περίπτωση χρήσης*

Το σύστημα να βρίσκεται σε κατάσταση λειτουργίας και να είναι συνδεδεμένο με το δίκτυο.

#### 3.1.2.3 Περιβάλλον εκτέλεσης

*Αναφορά στο περιβάλλον στο οποίο εκτελείται η περίπτωση χρήσης. Πχ "διαδικτυακή διεπαφή χρήστη", "DBMS" κλπ*

Η εκκαθάριση των οφειλών θα γίνεται μέσω διαδικτυακής εφαρμογής.

#### 3.1.2.4 Δεδομένα εισόδου

*Καταγραφή δεδομένων εισόδου και εξόδου και συνθηκών εγκυρότητας αυτών.*

1. Επιλογή ανάμεσα σε :Table Report (Απεικόνιση των δεδομένων σε πίνακα) και Graph Report (απεικόνιση των δεδομένων σε γράφημα)
2. dateFrom : Ημερομηνία έναρξης της κίνησης (πρέπει να είναι έγκυρη ημερομηνία)
3. dateTo: Ημερομηνία λήξης της κίνησης (πρέπει να είναι έγκυρη ημερομηνία και αργότερα από το dateFrom)

Στην περίπτωση του Table Report:

1. operatorID1: Πρώτος operator (πρέπει να είναι έγκυρο όνομα operator)
2. operatorID2: Δεύτερος operator(πρέπει να είναι έγκυρο όνομα operator)

Στην περίπτωση του Graph Report:

1. stationID: ID του σταθμού για τον οποίο αναζητούμε τα passes (πρέπει να είναι έγκυρος κωδικός σταθμού)

#### 3.1.2.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

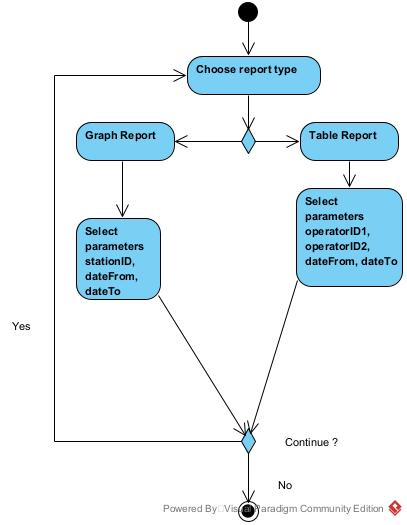
*Περιγραφή με κείμενο (Βήμα 1, Βήμα 2 κλπ) και διαγράμματα UML αλληλουχίας (Sequence) και δραστηριοτήτων (Activity). Περιλαμβάνεται η συμπεριφορά σε απρόβλεπτες καταστάσεις και σφάλματα (εναλλακτικές ροές).*

Βήμα 1:Επιλογή τύπου προβολής αναφοράς (table report ή graph report)

Βήμα 2*:*Επιλογή των αντίστοιχων παραμέτρων (σταθμού ή operators και ημερομηνίας)

Βήμα 3:Προβολή αναφοράς

Βήμα 4:Επιλογή συνέχειας ή εξόδου από την εφαρμογή.

****

#### 3.1.2.7 Δεδομένα εξόδου

*Διαγράμματα UML αλληλουχίας για την παραγωγή δεδομένων εξόδου. Ως δεδομένα εξόδου νοούνται όλα τα δεδομένα του συστήματος τα οποία δημιουργούνται ή μεταβάλλονται κατά την εκτέλεση (αν υπάρχουν τέτοια)*

Στην περίπτωση του Table Report, έξοδος είναι μία αναφορά με όλα τα passes που πραγματοποιήθηκαν από οχήματα με tag του πρώτου operator σε σταθμούς του δεύτερου operator , ανάμεσα στις δοσμένες ημερομηνίες, σε μορφή πίνακα .   
Στην περίπτωση του Graph Report , έξοδος είναι ένα γράφημα που παρουσιάζει πόσα οχήματα με tag του κάθε αυτοκινητόδρομου πέρασαν από τον δοσμένο σταθμό ανάμεσα στις δοσμένες ημερομηνίες.

#### 3.1.2.8 Παρατηρήσεις

*Ο,τι δεν εντάσσεται στα προηγούμενα, εφόσον υπάρχει*

### 3.1.3 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 3: Προβολή αναφορών

#### 3.1.3.1 Χρήστες (ρόλοι) που εμπλέκονται

*Αναφορά στους ρόλους που αφορά η περίπτωση χρήσης*

Η περίπτωση χρήσης αφορά τους διαχειριστές των αυτοκινητόδρομων.

#### 3.1.3.2 Προϋποθέσεις εκτέλεσης

*Καταγραφή των συνθηκών που πρέπει να ισχύουν ώστε να μπορεί να εκτελεστεί η περίπτωση χρήσης*

Το σύστημα να είναι σε λειτουργία και συνδεδεμένο με το δίκτυο.

#### 3.1.3.3 Περιβάλλον εκτέλεσης

*Αναφορά στο περιβάλλον στο οποίο εκτελείται η περίπτωση χρήσης. Πχ "διαδικτυακή διεπαφή χρήστη", "DBMS" κλπ*

Η περίπτωση χρήσης θα εκτελείται σε περιβάλλον διαδικτυακής εφαρμογής.

#### 3.1.3.4 Δεδομένα εισόδου

*Καταγραφή δεδομένων εισόδου και εξόδου και συνθηκών εγκυρότητας αυτών.*

1. operatorID1: Πρώτος operator (πρέπει να είναι έγκυρο όνομα operator)
2. operatorID2: Δεύτερος operator(πρέπει να είναι έγκυρο όνομα operator)
3. dateFrom: Ημερομηνία έναρξης της κίνησης (πρέπει να είναι έγκυρη ημερομηνία)
4. dateTo: Ημερομηνία λήξης της κίνησης (πρέπει να είναι έγκυρη ημερομηνία και αργότερα από το dateFrom)
5. Επιλογή ανάμεσα σε : short report (σύντομη αναφορά) , detailed report (αναλυτική αναφορά) , settle passes .

#### 3.1.3.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

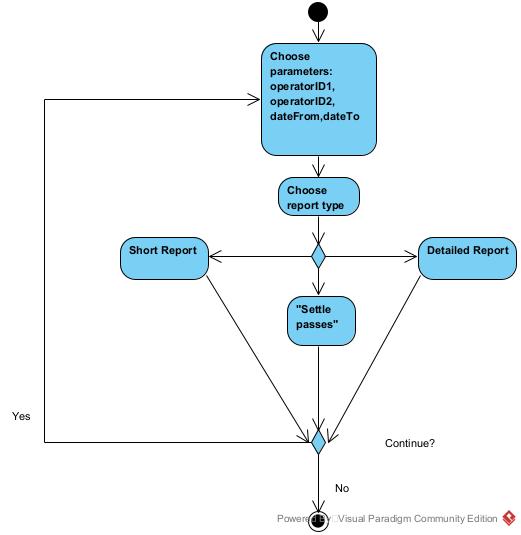
*Περιγραφή με κείμενο (Βήμα 1, Βήμα 2 κλπ) και διαγράμματα UML αλληλουχίας (Sequence) και δραστηριοτήτων (Activity). Περιλαμβάνεται η συμπεριφορά σε απρόβλεπτες καταστάσεις και σφάλματα (εναλλακτικές ροές).*

Βήμα 1*:*Επιλογή παραμέτρων (operators και το χρονικό διάστημα για το οποίο θα γίνει η αναφορά)

Βήμα 2:Επιλογή τύπου προβολής αναφοράς (short ή detailed) ή “settle passes”

Βήμα 3:Προβολή αναφοράς ή σήμανση των passes ως settled στη βάση.

Βήμα 4:Επιλογή συνέχειας ή εξόδου από την εφαρμογή.

****

#### 3.1.3.7 Δεδομένα εξόδου

*Διαγράμματα UML αλληλουχίας για την παραγωγή δεδομένων εξόδου. Ως δεδομένα εξόδου νοούνται όλα τα δεδομένα του συστήματος τα οποία δημιουργούνται ή μεταβάλλονται κατά την εκτέλεση (αν υπάρχουν τέτοια)*

Στην περίπτωση του short report: μία απλή αναφορά με το συνολικό ποσό που οφείλεται μεταξύ των δύο operators μεταξύ των δοσμένων ημερομηνιών.   
Στην περίπτωση του detailed report: μία εκτεταμένη αναφορά με όλα τα passes που πραγματοποιήθηκαν σε σταθμούς του πρώτου operator από οχήματα με tags του δεύτερου operator , και αντίστροφα, μεταξύ των δοσμένων ημερομηνιών, καθώς και υπολογισμός του συνολικού ποσού που οφείλεται.

Στην περίπτωση “settle passes”: αλλάζει η κατάσταση των passes στη βάση, καθώς τα θεωρούμε ως settled. Η συγκεκριμένη λειτουργική απαίτηση θεωρούμε ότι θα πραγματοποιείται αυτόματα ανά συγκεκριμένο χρονικό διάστημα , αλλά παρουσιάζεται εδώ για λόγους επίδειξης.

#### 3.1.3.8 Παρατηρήσεις

*Ο,τι δεν εντάσσεται στα προηγούμενα, εφόσον υπάρχει*

## 3.2 Απαιτήσεις επιδόσεων

*Ποσοτική τεκμηρίωση μέτρων και κριτηρίων επιθυμητών επιδόσεων με αναφορά στα ποσοτικά χαρακτηριστικά εισόδων και φορτίου του λογισμικού.*

## Σημαντικά κριτήρια επιδόσεων είναι τα ακόλουθα:

* Χαμηλός χρόνος απόκρισης κατά την ανάκληση δεδομένων στην διαδικτυακή εφαρμογή.
* Δυνατότητα αποθήκευσης αρκετών δεδομένων

## 3.3 Απαιτήσεις οργάνωσης δεδομένων

### 3.3.1 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα

*Απαιτήσεις πρόσβασης και περιορισμοί. Αναφορά στο διάγραμμα οντοτήτων-συσχετίσεων.*

Οι operators έχουν πρόσβαση μόνο στα δεδομένα που τους αφορούν άμεσα, δηλαδή στα passes που έγιναν σε δικό τους σταθμό ή από δικό τους tag.

Οι transportation authorities έχουν πρόσβαση στα passes, μέσω των λειτουργιών graph report και table report.

Οι payment service providers δεν έχουν καμία πρόσβαση στα δεδομένα.

## 3.5 Λοιπές απαιτήσεις

### 3.5.1 Απαιτήσεις διαθεσιμότητας λογισμικού

*Τεκμηρίωση απαιτήσεων διαθεσιμότητας*

Το σύστημά μας θα πρέπει να έχει διαθεσιμότητα τουλάχιστον 80% το μήνα, ώστε να πληροί τις απαιτήσεις των ιδιοκτητών και του Υπουργείου Μεταφορών.

### 3.5.2 Απαιτήσεις ασφάλειας

*Τεκμηρίωση απαιτήσεων ασφαλείας*

* Θα πρέπει να υποστηρίζεται το πρωτόκολλο HTTPS για όλες τις διεπαφές.
* Κρυπτογράφηση προσωπικών δεδομένων των χρηστών.