Έγγραφο απαιτήσεων λογισμικού (SRS)

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

ΔΙΑΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ ΔΙΟΔΙΩΝ

# Εισαγωγή

## 1.1 Εισαγωγή: σκοπός του λογισμικού

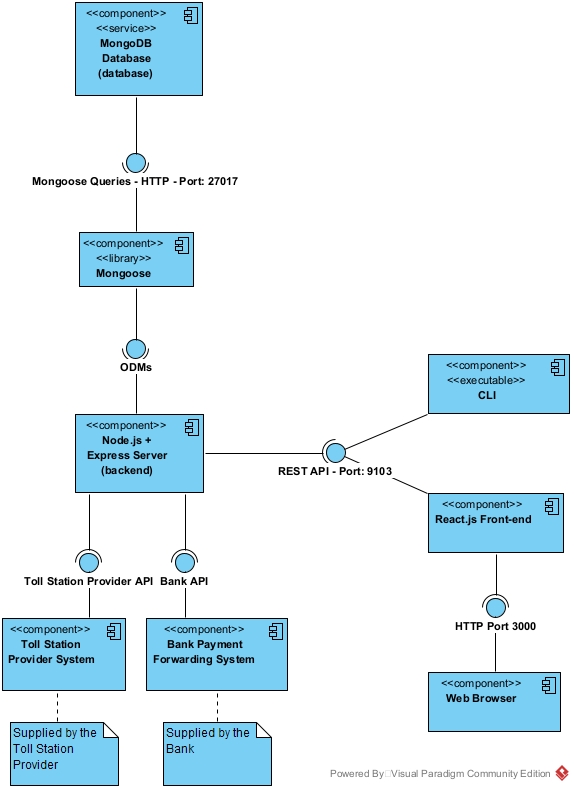
Ο σκοπός του λογισμικού είναι η διαλειτουργικότητα του συστήματος των διοδίων διάφορων αυτοκινητοδρόμων της Ελλάδας. Το σύστημα μας θα είναι υπεύθυνο για την παροχή της δυνατότητας σε έναν οδηγό που κατέχει έναν πομποδέκτη μίας συμμετέχουσας εταιρείας να μπορεί να διελεύσει από σταθμούς διοδίων άλλων εμπλεκόμενων εταιρειών με τον πομποδέκτη αυτό. Επίσης, θα διατηρείται αρχείο των οφειλών της κάθε εταιρείας προς τις υπόλοιπες και με βάση αυτό θα διενεργείται η διαδικασία υπολογισμού των οφειλών αυτών και η ενημέρωση σχετικά με την εξόφληση τους.

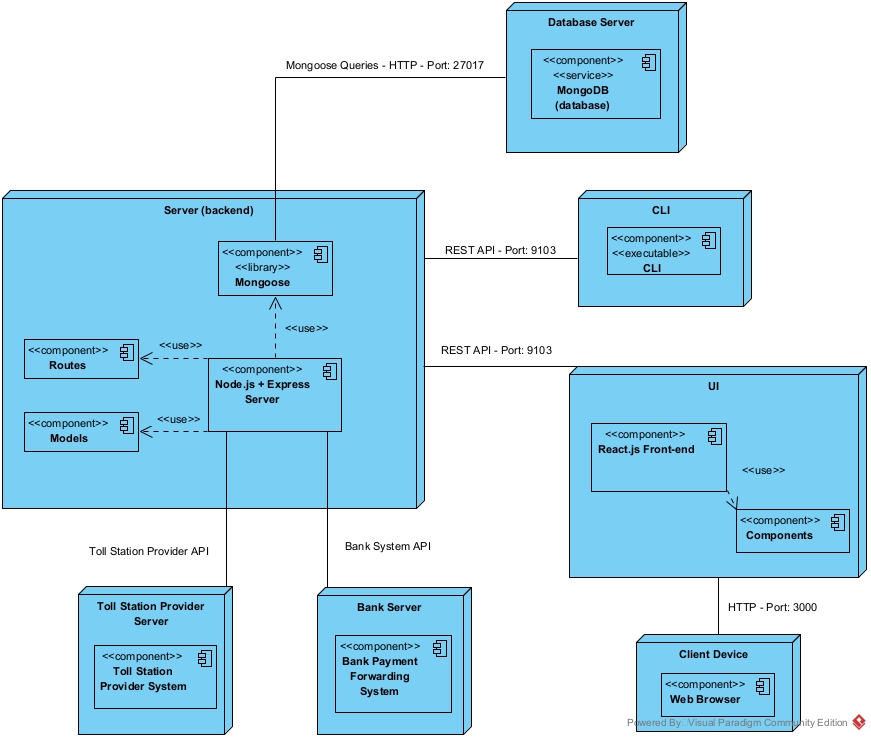
## 1.2 Διεπαφές (interfaces)

### 1.2.1 Διεπαφές με εξωτερικά συστήματα

* API κάθε εταιρείας λειτουργού αυτοκινητόδρομου που συμμετέχει στο έργο
* API των τραπεζών που ενημερώνουν για τις εξοφλήσεις των οφειλών

Ακολουθεί το διάγραμμα UML Component όλου του μέχρι τώρα συστήματος και στη συνέχεια φαίνεται και το UML Deployment διάγραμμα του λογισμικού:





### 1.2.2 Διεπαφές με το χρήστη

Η διεπαφή που θα υλοποιηθεί θα αντλεί δεδομένα από το σύστημα μας και θα τα οπτικοποιεί σύμφωνα με τις παραμέτρους που ορίζει κάθε φορά ο χρήστης του συστήματος.

# Αναφορές - πηγές πληροφοριών

N/A.

# Προδιαγραφές απαιτήσεων λογισμικού

## 3.1 Περιπτώσεις χρήσης

### 3.1.1 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 1: Απόφαση διέλευσης από σταθμό διοδίων

#### 3.1.1.1 Χρήστες (ρόλοι) που εμπλέκονται

* Οδηγός (κάτοχος tagID)
* Εταιρεία που της ανήκει το tag (tagProvider)
* Εταιρεία που της ανήκει ο σταθμός διοδίων (stationProvider)

#### 3.1.1.2 Προϋποθέσεις εκτέλεσης

* Σύνδεση στο διαδίκτυο
* Έγκυρο tagID που ανήκει σε κάποια συμμετέχουσα εταιρεία
* Πρόσβαση και ομαλή λειτουργία στα API των παρόχων

#### 3.1.1.3 Περιβάλλον εκτέλεσης

* Τοπικός υπολογιστής στο σταθμό διοδίων για έλεγχο και επεξεργασία

#### 3.1.1.4 Δεδομένα εισόδου

**Είσοδος**:

* tagID

**Έξοδος**:

* Pass για επίτρεψη διέλευσης ή No Pass για απαγόρευση
* Error σε περίπτωση σφάλματος

**Συνθήκες εγκυρότητας**:

* Το tagID να είναι έγκυρο, μοναδικό και να ανήκει σε ένα μόνο πάροχο

#### 3.1.1.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

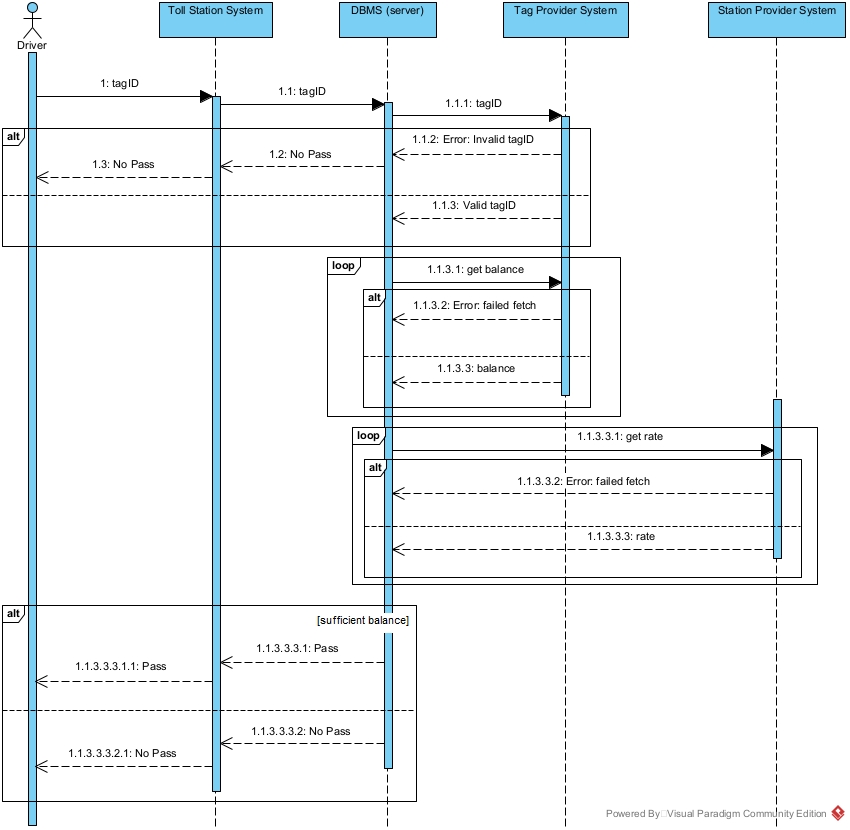
**Βήματα:**

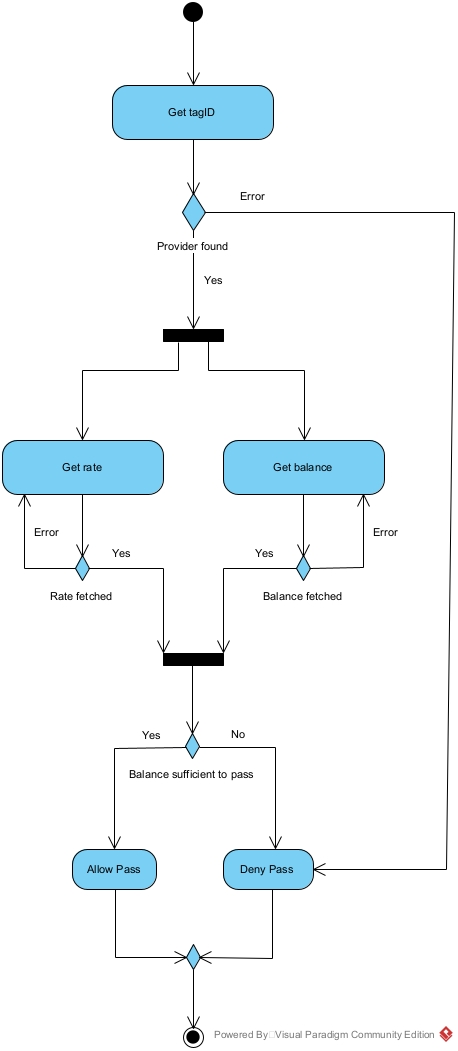
1. Λήψη της ταυτότητας του παρόχου του tagID (tagProvider)
2. Λήψη της τιμής διέλευσης (rate)
3. Λήψη του υπολοίπου του tagID από τον πάροχο (balance)
4. Σύγκριση υπολοίπου-τιμής και επίτρεψη ή απαγόρευση διέλευσης

**Διαχείριση σφαλμάτων (ανάλογα με το βήμα που παρουσιάζονται):**

1. Error: Μη έγκυρο tagID (δεν έχει σωστό format, δεν ανήκει σε κανέναν πάροχο, ανήκει σε πάνω από έναν παρόχους) Ενημέρωση για Error και απαγόρευση διέλευσης
2. Error: Αποτυχία λήψης της τιμής διέλευσης (απουσία σύνδεσης, σφάλμα του API του παρόχου) Ενημέρωση για Error και επανάληψη του βήματος
3. Error: Αποτυχία λήψης του υπολοίπου (απουσία σύνδεσης, σφάλμα του API του παρόχου) Ενημέρωση για Error και επανάληψη του βήματος

Ακολουθεί το UML Sequence διάγραμμα και το UML Activity διάγραμμα:





#### 3.1.1.6 Δεδομένα εξόδου

Ν/Α.

#### 3.1.1.7 Παρατηρήσεις

Ν/Α.

### 3.1.2 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 2: Προετοιμασία και καταγραφή οφειλής

#### 3.1.2.1 Χρήστες (ρόλοι) που εμπλέκονται

* Εταιρεία που της ανήκει το tagID του οδηγού (tagProvider)
* Εταιρεία που της ανήκει ο σταθμός διοδίων (stationProvider)

#### 3.1.2.2 Προϋποθέσεις εκτέλεσης

* Σύνδεση στο διαδίκτυο

#### 3.1.2.3 Περιβάλλον εκτέλεσης

* Τοπικός υπολογιστής στο σταθμό διοδίων για έλεγχο και επεξεργασία
* Κεντρικό DBMS του συστήματος μας για αποθήκευση

#### 3.1.2.4 Δεδομένα εισόδου

**Είσοδος**:

* passID (ID διέλευσης)
* timestamp (χρονική στιγμή διέλευσης σε UTC)
* stationRef (ID σταθμού διοδίων)
* vehicleRef (ID οχήματος)
* charge (ποσό οφειλής ίσο με rate)
* tagAbbr (συντομογραφία ονόματος παρόχου του tagID)
* home (true εάν οι πάροχοι είναι ίδιοι, ειδάλλως false)

**Έξοδος**:

* Pass record
* Error σε περίπτωση σφάλματος

#### 3.1.2.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

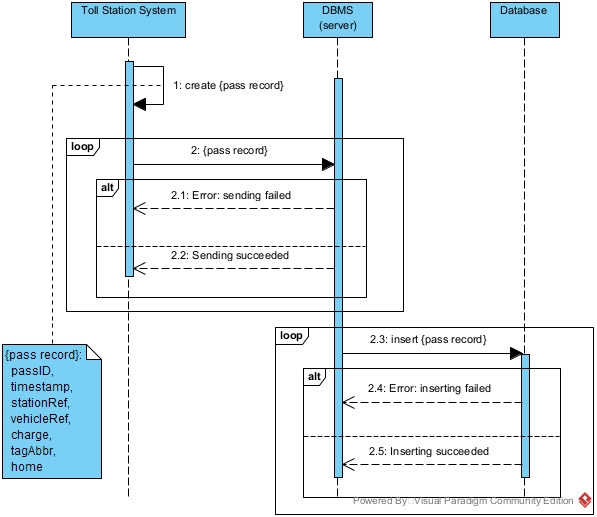
**Βήματα:**

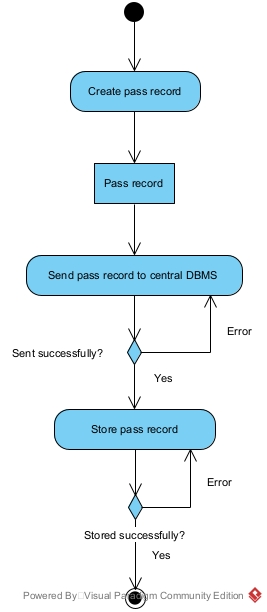
1. Σύνθεση της καταγραφής οφειλής
2. Αποστολή καταγραφής οφειλής στο κεντρικό DBMS (server)
3. Αποθήκευση της καταγραφής στη βάση δεδομένων

**Διαχείριση σφαλμάτων (ανάλογα με το βήμα που παρουσιάζονται):**

1. Error: Σφάλμα κατά τη μεταφορά μέσω του δικτύου Ενημέρωση για Error και επανάληψη του βήματος
2. Error: Σφάλμα κατά την αποθήκευση Ενημέρωση για Error και επανάληψη του βήματος

Ακολουθεί το UML Sequence διάγραμμα και το UML Activity διάγραμμα:





#### 3.1.2.6 Δεδομένα εξόδου

Δημιουργείται μια εγγραφή οφειλής (pass record) που περιέχει τα εξής δεδομένα:

* passID (ID διέλευσης)
* timestamp (χρονική στιγμή διέλευσης σε UTC)
* stationRef (ID σταθμού διοδίων)
* vehicleRef (ID οχήματος)
* charge (ποσό οφειλής ίσο με rate)
* tagAbbr (συντομογραφία ονόματος παρόχου του tagID)
* home (true εάν οι πάροχοι είναι ίδιοι, ειδάλλως false)

#### 3.1.2.7 Παρατηρήσεις

Ν/Α.

### 3.1.3 ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ 3: Υπολογισμός οφειλών προς μία εταιρεία

#### 3.1.3.1 Χρήστες (ρόλοι) που εμπλέκονται

* Εταιρεία διαχείρισης διοδίων για την οποία θα υπολογιστούν οι οφειλές (stationProvider)
* Όλες οι υπόλοιπες εταιρείες διαχείρισης διοδίων (tagProvider) που συμμετέχουν στο έργο και έχουν τουλάχιστον μία διέλευση στους σταθμούς της παραπάνω εταιρείας

#### 3.1.3.2 Προϋποθέσεις εκτέλεσης

* Ομαλή λειτουργία του κεντρικού DBMS, στο οποίο υπάρχουν καταγραφές των οφειλών μεταξύ των εταιρειών λόγω διελεύσεων

#### 3.1.3.3 Περιβάλλον εκτέλεσης

* Κεντρικό DBMS του συστήματος μας

#### 3.1.3.4 Δεδομένα εισόδου

**Είσοδος**:

* op\_ID (stationProvider)
* fromDate (ημερομηνία από την οποία ξεκινάει το χρονικό διάστημα του υπολογισμού οφειλών)
* toDate (ημερομηνία στην οποία τελειώνει το χρονικό διάστημα του υπολογισμού οφειλών)

**Έξοδος**:

* Έγγραφο με γενικές πληροφορίες του υπολογισμού για την εταιρία διαχείρισης διοδίων (stationProvider) και μία λίστα με το πλήθος των διελεύσεων και τα οφειλόμενα ποσά από κάθε άλλη εταιρεία (tagProvider) που έχει τουλάχιστον μία διέλευση στους σταθμούς της παραπάνω εταιρείας

**Συνθήκες εγκυρότητας:**

* Το op\_ID να αντιστοιχεί σε κάποιον stationProvider που συμμετέχει στο έργο.
* Το toDate πρέπει να είναι ημερολογιακά αργότερα από το fromDate.

#### 3.1.3.5 Αλληλουχία ενεργειών - επιθυμητή συμπεριφορά

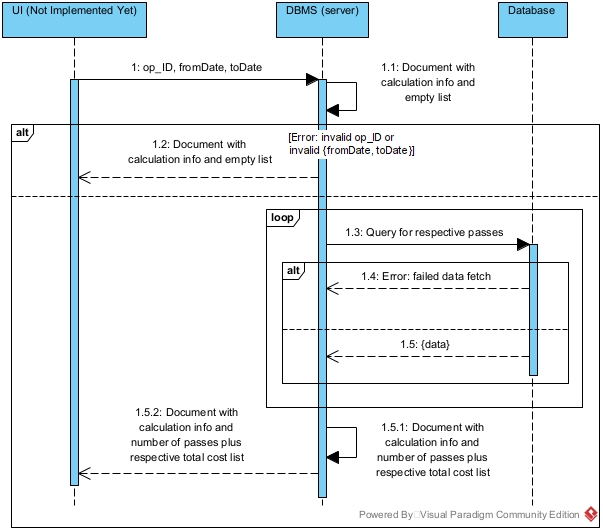
**Βήματα:**

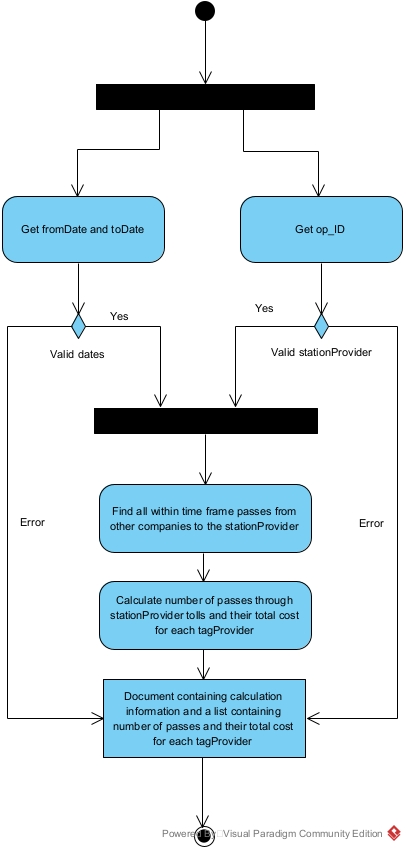
1. Λήψη του op\_ID (stationProvider) και των ημερομηνιών fromDate και toDate, οι οποίες αποτελούν τη χρονική περίοδο του υπολογισμού
2. Εύρεση όλων των χρονικά αποδεκτών, με βάση τις ημερομηνίες εισόδου, διελεύσεων, οι οποίες έγιναν στους σταθμούς της εταιρείας (stationProvider) από τις υπόλοιπες εταιρείες (tagProvider)
3. Εύρεση πλήθους διελεύσεων και συνολικού ποσού οφειλής για κάθε εταιρεία με μία τουλάχιστον διέλευση

**Διαχείριση σφαλμάτων (ανάλογα με το βήμα που παρουσιάζονται):**

1. Μη έγκυρο op\_ID ή μη έγκυρη χρονική περίοδος υπολογισμού (fromDate, toDate) Ενημέρωση για Error και εμφάνιση εγγράφου με τις γενικές πληροφορίες του υπολογισμού και μία άδεια λίστα
2. Σφάλμα κατά την αναζήτηση στη βάση δεδομένων για τις έγκυρες διελεύσεις στους σταθμούς της εταιρείας op\_ID (stationProvider)Ενημέρωση για Error και επανάληψη του βήματος

Ακολουθεί το UML Sequence διάγραμμα και το UML Activity διάγραμμα:





#### 3.1.3.6 Δεδομένα εξόδου

Δημιουργείται ένα έγγραφο με γενικές πληροφορίες του υπολογισμού για την εταιρία διαχείρισης διοδίων (stationProvider) και μία λίστα με το πλήθος των διελεύσεων και τα οφειλόμενα ποσά από κάθε άλλη εταιρεία (tagProvider) που έχει τουλάχιστον μία διέλευση στους σταθμούς της παραπάνω εταιρείας.

#### 3.1.3.7 Παρατηρήσεις

* Η εκτέλεση του συγκεκριμένου use case γίνεται για τις συμφωνημένες ημερομηνίες

## 3.2 Απαιτήσεις επιδόσεων

Εύλογο χρονικό διάστημα απόκρισης συστήματος στην αίτηση μιας διέλευσης:

* Σε κανονικές συνθήκες λειτουργίας του συστήματος, η ανώτατη αποδεκτή διάρκεια ορίζεται στα X δευτερόλεπτα (προτείνουμε ενδεικτική τιμή για το Χ τα 5 δευτερόλεπτα)
* Σε συνθήκες αυξημένου κυκλοφοριακού φόρτου, η ανώτατη αποδεκτή διάρκεια ορίζεται στα 2Χ δευτερόλεπτα
* Σε συνθήκες περιορισμένης παροχής δικτύου, η ανώτατη αποδεκτή διάρκεια ορίζεται στα 2Χ δευτερόλεπτα

## 3.3 Απαιτήσεις οργάνωσης δεδομένων

### 3.3.1 Απαιτήσεις και περιορισμοί πρόσβασης σε δεδομένα

**Απαιτήσεις:**

* Πρόσβαση στα δεδομένα των tags (tagID, tagProvider, balance) στα συστήματα των εταιρειών
* Πρόσβαση στα δεδομένα των τιμών διέλευσης (rate) στα συστήματα των εταιρειών

## 3.5 Λοιπές απαιτήσεις

### 3.5.1 Απαιτήσεις διαθεσιμότητας λογισμικού

* Το λογισμικό πρέπει να είναι διαθέσιμο τις ώρες λειτουργίας των αυτοκινητοδρόμων, δεδομένου ότι υπάρχει σύνδεση στο διαδίκτυο
* Το λογισμικό πρέπει να είναι διαθέσιμο τουλάχιστον για το 80% του μήνα, δηλαδή να έχει μέγιστο χρόνο εκτός λειτουργίας τις 7 ημέρες ανά μήνα
* Το λογισμικό εκτελεί λειτουργίες υπολογισμού οφειλών, εξόφλησης αυτών και έκδοσης αναλυτικών αναφορών σε συγκεκριμένες συμφωνημένες χρονικές στιγμές.
* Η διεπαφή χρήστη είναι διαθέσιμη πάντα

### 3.5.2 Απαιτήσεις ασφάλειας

* Αποθήκευση μόνο των απολύτως αναγκαίων δεδομένων για την προστασία του απορρήτου των εμπλεκόμενων μερών
* Κρυπτογράφηση των προσωπικών δεδομένων των εμπλεκόμενων μερών.
* Χρήση του πρωτοκόλλου HTTPS για τις διάφορες συνδέσεις μεταξύ των συστημάτων, για λόγους ασφάλειας