

# **PayWay**

## **Software Requirements Specification**

Version 2.0

## Revision History

Date	Version	Description	Author
01/12/2024	1.0	Αρχική σύνθεση του Εγγράφου Προδιαγραφών Λογισμικού	Αναστασία Ασκούνη Νικόλαος Βέρρας Αγγελική Ζέρβα Δημήτριος Καμπανάκης
12/02/2025	2.0	Τελική σύνθεση του Εγγράφου Προδιαγραφών Λογισμικού	Αναστασία Ασκούνη Νικόλαος Βέρρας Αγγελική Ζέρβα Δημήτριος Καμπανάκης

# Table of Contents

Table of Contents .....	3
Introduction.....	4
References .....	4
Overall Description.....	4
Problem Statement .....	4
Objectives .....	4
Constraints .....	5
Functional Requirements .....	6
Use Cases .....	6
Use Case Diagram .....	6
User Requirements.....	9
User Interfaces: .....	9
Technical Requirements.....	10
Performance .....	10
Scalability .....	10
Security .....	10
Maintainability .....	10
Usability.....	11
Availability .....	11
Hardware Requirements.....	11
Network.....	11
Server .....	11
Client Computers .....	11
Deployment Requirements.....	12
Deployment Diagram.....	12
Nodes .....	12
Components .....	13
Artifacts.....	14

## Introduction

Το παρόν έγγραφο περιέχει και αναλύει τις Προδιαγραφές Απαιτήσεων του Λογισμικού Payway. Συντάχθηκε με σκοπό να παρέχει λεπτομερή και δομημένη τεκμηρίωση του λογισμικού, καλύπτοντας τόσο τη φάση ανάπτυξης όσο και τη συντήρηση του κατά τη διάρκεια του κύκλου ζωής του.

Κύριους στόχους του εγγράφου αποτελούν:

- Παροχή αναλυτικών και επακριβών περιγραφών της λειτουργικής συμπεριφοράς του συστήματος.
- Καθορισμός των προδιαγραφών απόδοσης και ποιότητας που πρέπει να πληρεί το σύστημα.
- Κοινή αναφορά για την διασφάλιση αμοιβαίας κατανόησης μεταξύ των εμπλεκομένων μερών και της ομάδας ανάπτυξης.
- Ύπαρξη βασικής συμφωνίας για την σχεδίαση, ανάπτυξη και επαλήθευση της ορθής επιτέλεσης των συγκεκριμένων λειτουργιών.

## References

Για τη σύνταξη του τρέχοντος εγγράφου απαιτήσεων λογισμικού χρησιμοποιήθηκε το παρακάτω πρότυπο:

**IEEE Guide to Software Requirements Specifications (IEEE 29148:2011).**

## Overall Description

### Problem Statement

Στους σύγχρονους αυτοκινητοδρόμους πλήθος σταθμών διοδίων ανήκουν στο ίδιο οδικό δίκτυο, με αυτούς να αποτελούν ιδιοκτησία διαφορετικών παρόχων. Καθώς τα οχήματα διέρχονται από τους διάφορους σταθμούς, ανάλογα με τον πάροχο στον οποίο ανήκει ο πομποδέκτης τους, δημιουργούνται οι αντίστοιχες οφειλές και προκύπτει το ζήτημα της διαλειτουργικότητας των διοδίων. Ζητούμενη και επιθυμητή γίνεται η ύπαρξη ένα συστήματος το οποίο θα μπορεί να συμψηφίζει τις οφειλές μεταξύ των λειτουργών των διοδίων με ενιαίο και αποτελεσματικό τρόπο.

### Objectives

Το λογισμικό PayWay υλοποιεί την προαναφερθείσα διαλειτουργικότητα, προσφέροντας ταυτόχρονα υπηρεσίες διάθεσης και ανάλυσης δεδομένων που προκύπτουν από την καθημερινή χρήση των αυτοκινητοδρόμων. Το λογισμικό αποθηκεύει τα δεδομένα διέλευσης από τους σταθμούς διοδίων κάθε λειτουργού, υπολογίζει τους οικονομικούς συμψηφισμούς μεταξύ των λειτουργών, παρουσιάζει στατιστικά στοιχεία που αφορούν τους σταθμούς στους λειτουργούς, αλλά και σε τρίτους, όπως το Υπουργείο Μεταφορών, και ενημερώνει κατάλληλα τις δομές του σε περίπτωση εξωτερικής κάλυψης του οφειλόμενου ποσού από κάποιον λειτουργό.

## **Constraints**

### **1. Platform Compatibility**

Η λειτουργία του λογισμικού PayWay έχει βελτιστοποιηθεί για χρήση σε web browsers desktop περιβαλλόντων και αρχιτεκτονικών, χωρίς περιορισμό λειτουργικού συστήματος. Το λογισμικό δεν έχει ελεγχθεί για χρήση σε φορητές συσκευές και λειτουργικά συστήματα (iOS, Android) και ενδέχεται να παρουσιάζει μη βέλτιστη συμπεριφορά.

### **2. Scalability**

Το σύστημα έχει σχεδιαστεί να διαχειρίζεται μεγάλο αριθμό διελεύσεων, σταθμών και δεδομένων πληρωμών, παρ' όλ' αυτά, η πρώτη έκδοση ενδέχεται να παρουσιάζει μη αναμενόμενη συμπεριφορά με διαχείριση υπέρογκων δεδομένων.

### **3. Internet Dependency**

Το PayWay απαιτεί σταθερή πρόσβαση στο διαδίκτυο για την επικοινωνία της βάσης δεδομένων με τον server της υλοποίησης και την επικοινωνία με την frontend υλοποίηση. Σε αντίθετη περίπτωση, το λογισμικό αδυνατεί να επεξεργαστεί οποιοδήποτε αίτημα.

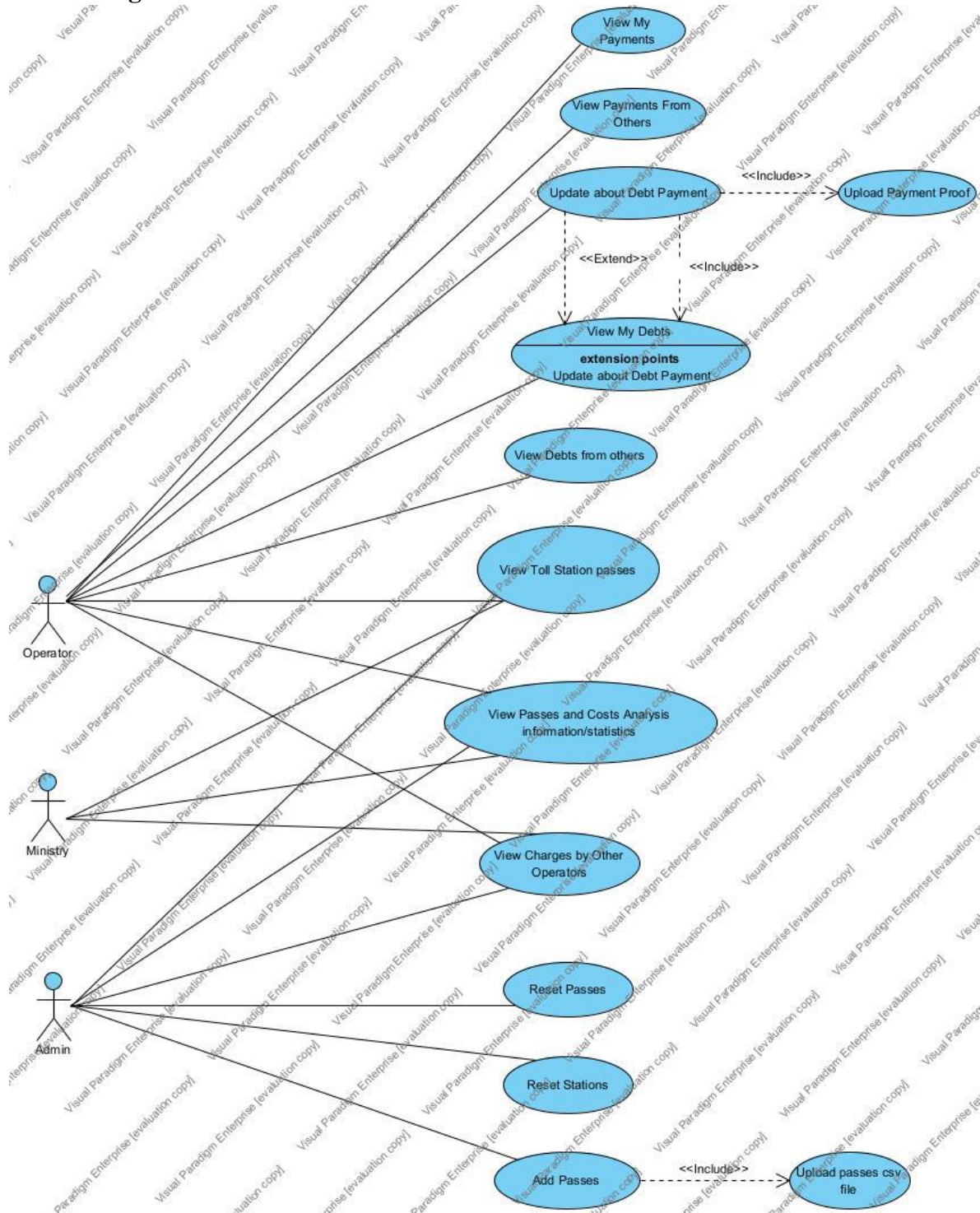
### **4. Data Dependency**

Η λειτουργία του λογισμικού και η ποιότητα των αποτελεσμάτων του εξαρτώνται άμεσα από τα δεδομένα που του προσφέρονται από τους λειτουργούς.

# Functional Requirements

## Use Cases

### Use Case Diagram



Use Case	Description
<b>Add Passes</b>	Αυτή η περίπτωση χρήσης επιτρέπει στον διαχειριστή να προσθέσει νέες διελεύσεις στη βάση δεδομένων.
<b>Update About Debt Payment</b>	Αυτή η περίπτωση χρήσης επιτρέπει στους λειτουργούς να επιλέξουν μια οφειλή από την ενότητα "My Debts" και να ανεβάσουν αποδεικτικό πληρωμής. Παρόλο που οι λειτουργοί δεν μπορούν να πραγματοποιήσουν την πληρωμή μέσω του συστήματος, έχουν τη δυνατότητα να υποβάλουν αποδεικτικό για εξόφληση πληρωμής ή μέρους της που έχει γίνει εξωτερικά.
<b>Reset Passes</b>	Αυτή η περίπτωση χρήσης επιτρέπει στον διαχειριστή να επαναφέρει και να διαγράψει όλες τις διελεύσεις από τη βάση δεδομένων.
<b>Reset Stations</b>	Αυτή η περίπτωση χρήσης επιτρέπει στον διαχειριστή να επαναφέρει τους σταθμούς διοδίων. Οι σταθμοί διαγράφονται και στη συνέχεια επανεισάγονται στη βάση δεδομένων.
<b>View Charges by Other Operators</b>	<p>Αυτή η περίπτωση χρήσης επιτρέπει στους λειτουργούς να προβάλλουν τις συνολικές χρεώσεις από άλλους λειτουργούς κατά τη διάρκεια επιλεγμένης χρονικής περιόδου. Επιπλέον, η εφαρμογή παρέχει στατιστικά γραφήματα για τη σύγκριση χρεώσεων από διαφορετικούς λειτουργούς.</p> <p>Εάν ο χρήστης είναι διαχειριστής, έχει πρόσβαση στις χρεώσεις όλων των λειτουργών προς όλους τους υπόλοιπους.</p>
<b>View Debts from Others</b>	Αυτή η περίπτωση χρήσης επιτρέπει στους λειτουργούς να προβάλλουν τις τρέχουσες οφειλές από άλλους λειτουργούς προς τους ίδιους, ανανεωμένες με βάση υποβληθείσες πληρωμές ή νέες διελεύσεις.
<b>View My Debts</b>	Αυτή η περίπτωση χρήσης επιτρέπει στους λειτουργούς να προβάλλουν τις τρέχουσες οφειλές τους προς άλλους λειτουργούς, ανανεωμένες με βάση υποβληθείσες πληρωμές ή νέες διελεύσεις.
<b>View My Payments</b>	Αυτή η περίπτωση χρήσης επιτρέπει στους λειτουργούς να προβάλλουν τις εξωτερικές πληρωμές που έχουν πραγματοποιήσει για τις οποίες έχουν ενημερώσει το σύστημα με υποβολή αποδεικτικού.
<b>View Payments from Others</b>	Αυτή η περίπτωση χρήσης επιτρέπει στους λειτουργούς να προβάλλουν τις εξωτερικές πληρωμές που έχουν πραγματοποιήσει άλλοι λειτουργοί για οφειλές προς τους ίδιους, για τις οποίες έχουν ενημερώσει το σύστημα με υποβολή αποδεικτικού.

<b>View Passes and Costs Analysis information/statistics</b>	<p>Αυτή η περίπτωση χρήσης επιτρέπει στους λειτουργούς να επιλέξουν έναν άλλον λειτουργό και μια συγκεκριμένη χρονική περίοδο, και να προβάλλουν δύο διαφορετικές αναλύσεις. Είτε μια ανάλυση των διελεύσεων με το συνολικό τους κόστος από οχήματα με tags του επισκέπτη λειτουργού στους δικούς τους σταθμούς είτε των διελεύσεων από οχήματα με δικά τους tags σε σταθμούς του επιλεγμένου λειτουργού. Επιπλέον, παρέχει πολύτιμα στατιστικά γραφήματα σχετικά με τις διελεύσεις και το κόστος.</p> <p>Εάν ο χρήστης είναι διαχειριστής ή εκπρόσωπος του Υπουργείου, έχει τη δυνατότητα να επιλέξει τόσο τον τοπικό (home) όσο και τον επισκέπτη (visiting) λειτουργό.</p>
<b>View Toll Station passes</b>	<p>Αυτή η περίπτωση χρήσης επιτρέπει στους λειτουργούς να επιλέξουν έναν από τους σταθμούς τους και μια χρονική περίοδο και να προβάλλουν όλες τις πληροφορίες σχετικά με τις διελεύσεις από αυτόν τον σταθμό κατά τη διάρκεια αυτής. Επιπλέον, παρέχει στατιστικά γραφήματα σχετικά με τις διελεύσεις.</p> <p>Εάν ο χρήστης είναι διαχειριστής ή εκπρόσωπος του Υπουργείου Μεταφορών, έχει τη δυνατότητα να επιλέξει οποιονδήποτε σταθμό από οποιονδήποτε λειτουργό.</p>
<b>Upload passes csv</b>	Ο διαχειριστής μπορεί να ανεβάσει ένα αρχείο CSV που περιέχει τις διελεύσεις που θα προστεθούν στη βάση δεδομένων.
<b>Upload Payment Proof</b>	Οι λειτουργοί μπορούν να επιλέξουν να ανεβάσουν ένα αρχείο PDF με μέγεθος <10MB ως αποδεικτικό πληρωμής επιλεγμένης οφειλής ή μέρους της.



# User Requirements

## User Interfaces:

Οι διεπαφές με τις οποίες αλληλεπιδρά ο χρήστης είναι οι εξής:

- **Command Line Interface (CLI):** Σε αγγλική γλώσσα, είναι προσβάσιμη μέσω γραμμής εντολών
- **Web Application:** Πρόκειται για γραφικό διαδικτυακό περιβάλλον, προσβάσιμο από web browser, το οποίο θα παρέχει πρόσβαση στις λειτουργίες του συστήματος.

Actor	Description
Admin	Ο Διαχειριστής είναι υπεύθυνος για τη διαχείριση και τη συντήρηση του συστήματος. Έχει δικαιώματα διαχείρισης των δεδομένων των λειτουργών, των σταθμών και των συναλλαγών. Μπορεί να επαναφέρει σταθμούς στο σύστημα, να διαχειρίζεται λογαριασμούς χρηστών, να επιβλέπει την ορθή λειτουργία των υπηρεσιών και να επιλύει τεχνικά ή διαχειριστικά ζητήματα. Ο Διαχειριστής έχει τα υψηλότερα δικαιώματα στο σύστημα και μπορεί να εκτελέσει ενέργειες που δεν είναι διαθέσιμες σε άλλους ρόλους, όπως η επανακίνηση του συστήματος και η εισαγωγή δεδομένων διελεύσεων. Για την πρόσβαση σε αυτές τις λειτουργίες, ο Διαχειριστής πρέπει να συνδεθεί στο σύστημα με τα διαπιστευτήριά του.
Ministry	Το Υπουργείο Μεταφορών (Ministry) είναι ένας εξουσιοδοτημένος φορέας που επιβλέπει τις διελεύσεις και τις οικονομικές υποχρεώσεις μεταξύ των λειτουργών αυτοκινητοδρόμων. Έχει πρόσβαση σε αναφορές και στατιστικά δεδομένα που αφορούν τόσο τους τοπικούς (home) όσο και τους επισκέπτες (visiting) λειτουργούς. Για την πρόσβαση σε αυτές τις δυνατότητες, ο χρήστης του Υπουργείου πρέπει να είναι συνδεδεμένος στο σύστημα με κατάλληλα διαπιστευτήρια.
Operator	Ο Λειτουργός Αυτοκινητοδρόμου είναι υπεύθυνος για τη διαχείριση διοδίων και συναλλαγών που σχετίζονται με οφειλές και πληρωμές μεταξύ λειτουργών. Ο χρήστης αυτός μπορεί να προβάλλει ανάλυση διελεύσεων, να διαχειρίζεται τις οικονομικές του υποχρεώσεις και να υποβάλλει αποδεικτικά πληρωμών. Επιπλέον, έχει πρόσβαση σε στατιστικά στοιχεία που αφορούν τις διελεύσεις και το συνολικό κόστος των συναλλαγών. Για να χρησιμοποιήσει τις συγκεκριμένες λειτουργίες, ο χρήστης πρέπει να έχει συνδεθεί στο σύστημα με τα διαπιστευτήριά του.

# Technical Requirements

Το λογισμικό Payway έχει κρίσιμες απαιτήσεις επίδοσης σε αξιοπιστία, αποδοτικότητα και ταχύτητα.

## Performance

- **Response Time**

Το σύστημα πρέπει να επεξεργάζεται και να αποκρίνεται στα αιτήματα των χρηστών με καθυστέρηση μικρότερη των 5 δευτερολέπτων για τουλάχιστον 95% των αιτημάτων, ακόμα και υπό συνθήκες υψηλού φόρτου.

- **Concurrency**

Η εφαρμογή πρέπει να επιτρέπει την ταυτόχρονη εκτέλεση πολλαπλών ενεργειών από διαφορετικούς χρήστες χωρίς υποβάθμιση της απόδοσης.

- **Data Processing Efficiency**

Το σύστημα πρέπει να είναι σε θέση να διαχειρίζεται μεγάλο όγκο δεδομένων, μέχρι και 30,000 εγγραφών, σε πραγματικό χρόνο, εξασφαλίζοντας ότι οι λειτουργίες του εκτελούνται αποδοτικά, δηλαδή σε λιγότερο του ενός λεπτού.

## Scalability

- **System Growth Support**

Το λογισμικό πρέπει να υποστηρίζει την εύκολη προσθήκη νέων λειτουργιών στο σύστημα, χωρίς να επηρεάζεται η συνολική απόδοση.

- **Service Expansion**

Πρέπει να επιτρέπει την εύκολη επέκταση των υπηρεσιών της, προσφέροντας νέες δυνατότητες στους χρήστες χωρίς να απαιτείται εκτεταμένη αναδιοργάνωση του υπάρχοντος συστήματος.

## Security

- **Authentication and Authorization**

Το σύστημα πρέπει να διαθέτει ισχυρούς μηχανισμούς ελέγχου ταυτότητας και εξουσιοδότησης, ώστε να αποτρέπει μη εξουσιοδοτημένη πρόσβαση.

- **Data Integrity**

Για την προστασία της ακεραιότητας των δεδομένων, πρέπει όλες οι ενέργειες επεξεργασίας τους στο σύστημα να είναι ατομικές, ώστε να αποφεύγονται ασυνέπειες και μερικές ενημερώσεις.

## Maintainability

- **Code Documentation**

Ο πηγαίος κώδικας της εφαρμογής πρέπει να είναι καλά τεκμηριωμένος και δομημένος, ώστε να επιτρέπει την εύκολη αναβάθμιση και συντήρηση.

- **System Monitoring**

Το σύστημα πρέπει να διαθέτει μηχανισμούς παρακολούθησης και διάγνωσης σφαλμάτων, ώστε να εντοπίζονται και να επιλύονται προβλήματα γρήγορα.

## Usability

- **User-Friendly Interface**

Η εφαρμογή πρέπει να διαθέτει εύχρηστο και όσο το δυνατόν πιο απλοποιημένο περιβάλλον χρήστη, ώστε οι λειτουργοί των αυτοκινητοδρόμων να μπορούν να εκτελούν τις απαραίτητες ενέργειες χωρίς περιττή πολυπλοκότητα.

- **Automated Operations**

Το σύστημα πρέπει να υποστηρίζει αυτοματοποιημένες λειτουργίες, μειώνοντας την ανάγκη για χειροκίνητες διαδικασίες από τους χρήστες.

## Availability

- **High Availability**

Η εφαρμογή πρέπει να διαθέτει μηχανισμούς αποκατάστασης σφαλμάτων και διαχείρισης αστοχιών, διασφαλίζοντας διαθεσιμότητα υπηρεσίας 99.9% (uptime).

## Hardware Requirements

### Network

- **Bandwidth and Throughput:**

Ο server που φιλοξενεί το λογισμικό PayWay θα πρέπει να έχει ικανοποιητικής ταχύτητας σύνδεση στο διαδίκτυο, ώστε να διασφαλίζεται η ομαλή ροή δεδομένων και η γρήγορη απόκριση του REST API.

### Server

- **Server Processing Power:**

Ο server που φιλοξενεί το PayWay πρέπει να διαθέτει τουλάχιστον έναν τετραπύρινο επεξεργαστή (Intel, AMD) ή κάποιο αντίστοιχο cloud instance.

- **Memory:**

Ο server πρέπει να διαθέτει τουλάχιστον 30GB αποθηκευτικού χώρου για αποθήκευση των στοιχείων του backend, καθώς και μνήμη RAM μεγέθους κατ' ελάχιστον 8GB, ώστε να διασφαλίζεται η ταχεία λειτουργία.

### Client

- **Operating System Compatibility:**

Το frontend της εφαρμογής είναι web-based και έχει ελεγχθεί και μπορεί να λειτουργεί σε Windows (10 και άνω), macOS (10.15 και άνω) και δημοφιλείς διανομές Linux (Ubuntu 20.04+).

- **Browser Support:**

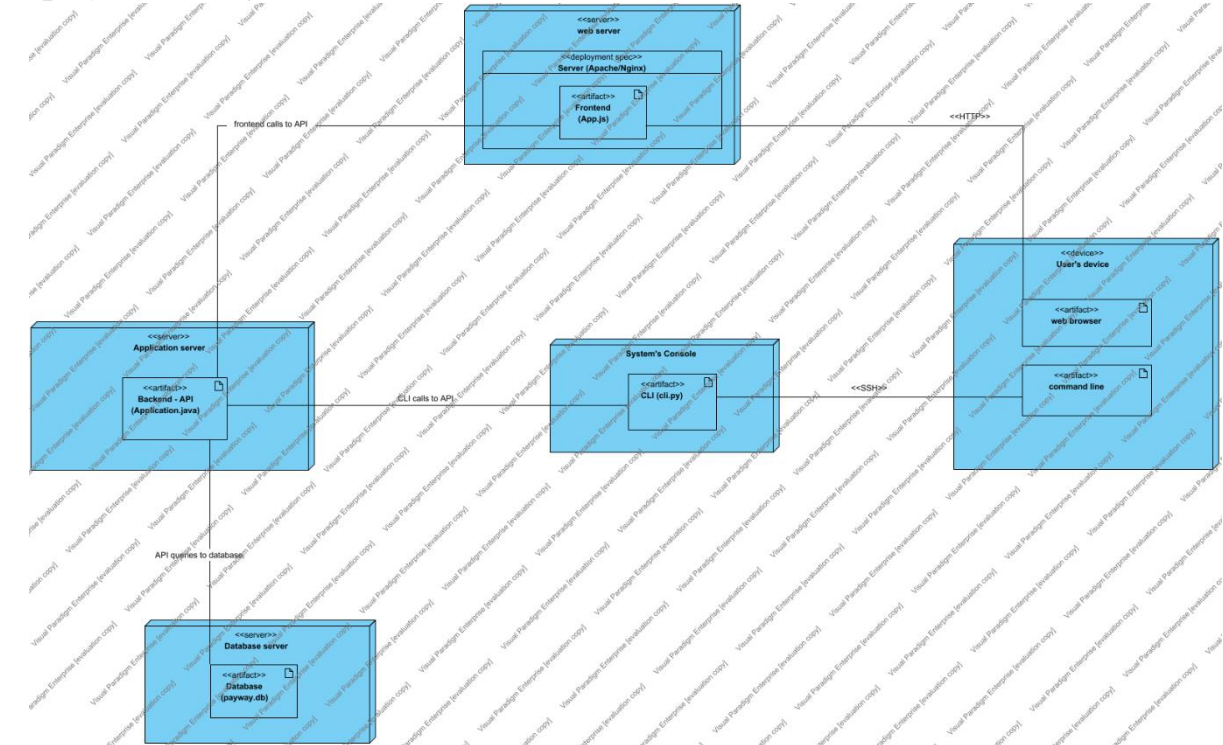
Το λογισμικό είναι προσβάσιμο από τις τελευταίες εκδόσεις μοντέρνων browsers (Google Chrome, Mozilla Firefox, Safari, Microsoft Edge).

- **Processor and Memory:**

Οι υπολογιστές των χρηστών πρέπει να διαθέτουν σύγχρονο επεξεργαστή (οποιαδήποτε αρχιτεκτονικής και γενιάς από το 2018 και έπειτα) για ομαλή αλληλεπίδραση με την εφαρμογή, ενώ συνιστάται τουλάχιστον 4GB RAM για την ομαλή λειτουργία του frontend και της CLI διεπαφής τοπικά.

# Deployment Requirements

## Deployment Diagram



## Nodes

Name	Description
Application server	Στον application server τρέχει το backend της εφαρμογής. Λαμβάνει κλήσεις από το frontend και το CLI, τις επεξεργάζεται και δίνει αποτελέσματα. Στα πλαίσια των κλήσεων μπορεί να επικοινωνεί και με την βάση δεδομένων.
Database server	Στον server της βάσης δεδομένων βρίσκεται η βάση δεδομένων. Μπορεί να επικοινωνεί με το backend, απαντώντας στα queries του.
System's Console	Η εφαρμογή μας μέσω του system console παρέχει έναν τρόπο επικοινωνίας με αυτήν πέραν του frontend. Δέχεται εντολές από το command line του user και δημιουργεί τα αντίστοιχα requests στο API. Συνδέεται με ssh με το τελικό μηχάνημα του χρήστη.
User's device	Η συσκευή του χρήστη είναι ο τρόπος με τον οποίο μπορεί να αλληλεπιδρά με την εφαρμογή. Προσφέρονται δύο τρόποι για να πραγματοποιηθεί αυτό, μέσω web browser που δίνει πρόσβαση στο frontend της εφαρμογής και μέσω command line που συνδέεται με ssh στο CLI.
Web server	Στον web server τρέχει το frontend της εφαρμογής. Το frontend μπορεί να επικοινωνεί με το API της εφαρμογής όπου κάνει κλήσεις και λαμβάνει τα αντίστοιχα αποτελέσματα και με τον web browser του χρήστη όπου τα παρουσιάζει διαδραστικά.

## Components

Database Tables
debt
operator
pass
payment
tag
tollstation
user

Backend Components	
Backend Controllers	Backend Services
AdminController.java	ChargeService.java
AnalysisController.java	HealthCheckService.java
Authentication Controller.java	jwtBlackListService.java
ChargesByController.java	PassService.java
HomeController.java	PaymentService.java
PassesCostController.java	TollStationService.java
PaymentController.java	
TollStationPassesController.java	

Frontend Components		
App.js		
Components	Styles	Pages
Charts.js	AdminStyle.css	AdminPage.js
Footer.js	App.css	ChargesByPage.js
Header.js	PassAnalysisStyle.css	Dashboard.css
Loader.js	TotalDebtsPage.css	Dashboard.js
Sidebar.js		DebtsFromOthers.js
Tables.js		DebtsPage.js
		Login.js
		MyPayments.js
		MyStations.js
		PassAnalysisAdmin.js
		PassAnalysisPage.js
		PassCostPage.js
		Passes.js
		PayDebts.js

		<b>PaymentsFromOthers.js</b>
		<b>Stations.js</b>

<b>CLI components</b>
<b>CLI.py</b>

## Artifacts

Name	Description
<b>Backend - API (Application.java)</b>	Στον server του backend τρέχει το Application.java, το οποίο ενεργοποιεί όλες τις λειτουργίες του backend. Παρέχεται το API στο frontend και το CLI για να δέχεται κλήση, υπηρεσίες χειρισμού των κλήσεων για παραγωγή των απαιτούμενων αποτελεσμάτων και διεπαφή με την βάση δεδομένων για εξαγωγή των δεδομένων για χειρισμό των κλήσεων.
<b>CLI (cli.py)</b>	Στο system console τρέχει το αρχείο cli.py, το οποίο μπορεί να δέχεται εντολές από την συσκευή του χρήστη και να τις μετατρέπει σε αιτήματα για το API, επιστρέφοντας τα αντίστοιχα δεδομένα.
<b>command line</b>	Ο χρήστης μπορεί, εάν επιθυμεί να επικοινωνήσει με την εφαρμογή με εντολές που θα δίνει μέσα από terminal στην συσκευή του που θα επικοινωνεί με το CLI.
<b>Database (payway.db)</b>	Η βάση δεδομένων έχει αποθηκευμένα τα απαραίτητα δεδομένα της εφαρμογής. Ανάλογα με τα queries του API συνδυάζονται τα δεδομένα αυτά για να παραχθεί κάθε φορά το ζητούμενο αποτέλεσμα.
<b>Frontend (App.js)</b>	Στον host τρέχει το αρχείο App.js, το οποίο ενεργοποιεί όλες τις λειτουργίες του frontend. Παρέχεται τρόπος επικοινωνίας με το API, επεξεργασία που αφορά το frontend και παρουσίαση στον web browser του χρήστη.
<b>web browser</b>	Ένας web browser προσφέρει στον χρήστη την δυνατότητα να αλληλεπιδρά με την εφαρμογή με διαδραστικό τρόπο.