Software Requirements Specification

Version 1.0

<<Annotated Version>>

February 25 2024

“Route ride manager”

Project members :

Bohdan Kharysh

Nikita Karazhyn

Lesnenko Bohdan

Viktor Levitskyi

Byk Illia

Table of contents

* 1.0 Introduction
  + 1.1 *Purpose*
  + 1.2 *Scope of Project*
  + 1.3 *Glossary*
  + 1.4 *References*
  + 1.5 *Overview of Document*
* 2.0 Overall description
  + 2.1 *System Environment*
  + 2.2 *Functional Requirements Specification*
    - *2.2.1 User Use Cases*
      * *Use case: add a new ticket*
      * *Use case: return ticket*
    - *2.2.2 Moderator Use Cases*
      * *Use case: add a new ticket*
      * *Use case: return ticket*
      * *Use case:* check stats
      * *Use case:* append user
      * *Use case:* delete user
    - *2.2.3 Admin Use Cases*
      * *Use case: add a new ticket*
      * *Use case: return ticket*
      * *Use case:* check stats
      * *Use case:* append user
      * *Use case:* delete user
      * *Use case:* change ticket price
      * *Use case:* change the route
  + 2.3 *User Characteristics*
* 3.0 Requirements Specification
  + 3.1 *External Interface Requirements*
  + 3.2 *Functional Requirements*
    - 3.2.1 Add a new ticket
    - 3.2.2 Return ticket
    - 3.2.3 Check stats
    - 3.2.4 Append user
    - 3.2.5 Delete user
    - 3.2.6 Change ticket price
    - 3.2.7 Change the route

**1.0 Introduction**

## *1.1. Purpose*

The purpose of this document is to describe the software product - web application. It will provide necessary info about this system, such as the purpose of creation, functions of the application , interaction with the users, requirements of hardware, some functions and features.

This document is intended for both the stakeholders and the developers of the system.

*1.2 Scope of Project*

This software system will be a web application that allows to control bus infrastructure. It helps to track tickets, prices, luggage delivery, as well as adding/deleting new cashiers(users) or changing ticket prices/routes.

This system will be designed to get all necessary info, which allows to understand proficiency of different branches/routes, the percentage of successful luggage delivery etc, which is crucial for any business.

The software will facilitate managing all services that are provided , and help to directly track something by moderator when needed. It's also possible to add new users into the system, or change the prices/routes.(only allowed for admins)

The system also contains a database where stored info about cities, routes, users of system and tickets.

## *1.3. Glossary*

| **Term** | **Definition** |
| --- | --- |
| User | a person who can append info about ticket orders to the system, or make a ticket return. |
| Moderator | a person who is allowed to track luggage, routes, prices, and get info about previous orders from a database.  also can append/remove cashiers  (people with role “user”) |
| Admin | a person who has all the moderator rights + can change routes/prices. |
| DB | Collection of all the information monitored by this system. |
| Software Requirements Specification (SRS) | A Software Requirements Specification (SRS) document is a detailed description of the software system to be developed. |
| Customer | a person who order ticket |
| Cashier | a person with the role “user”, who sells tickets to the customers. |

## *1.4. References*

- User Roles and Permissions Model

- E-Commerce Platform Model

## *1.5. Overview of Document*

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

# **2.0. Overall Description**

## *2.1 System Environment*

The system environment in the ticket-selling application is carefully designed to cater to the three distinct roles of admin, moderator, and user. The admin role holds the key to all information within the system, including sales data, tickets information, and schedules , as well as having the ability to append/delete cashiers, change prices/routes.

Moderator role has abilities to review all data and to append/delete cashiers, also it's possible to append ticket info to the system, or return ticket.

The user role is more limited in its access to information. Users are primarily focused on appending information regarding ticket purchases.

If needed, it also possible for them to return the ticket,

All of them have access through the web application after logging in.

## *2.2 Functional Requirements Specification*

This section outlines the use cases for each of the active roles separately.

*2.2.1 User Use Cases*

User has the following cases of use:

add a new ticket , return ticket

*Use case: add a new ticket*

Diagram:

Brief Description: At the customer request, User creates a new ticket which includes info about the order, and appends it to the DB.

Initial Step-By-Step Description:

Before this use case can be initiated, the user has to be logged in the system to append information.

1.User selects the option to register a ticket.

2.The Customer provides info about his ticket.(which includes type of ticket, destination, etc.)

## 3.The User applies information.(append it to the system and click “Apply” button)

## 4.The system adds the new ticket to the DB.

*Use case: return ticket*

Diagram:

Brief Description: At the customer request, User returns a ticket which includes info about the order, and removes it from the DB

Initial Step-By-Step Description:

Before this use case can be initiated, the user has to be logged in the system to append information.

1.User selects the option to return a ticket.

2.The Customer provides info about his ticket.(which includes type of ticket, destination, etc.)

## 3.The User appends information.

4.The User clicks the “return ticket” button.

## 5.The system removes the ticket from the DB.

*2.2.2 Moderator Use Cases*

Moderator has the following cases of use:

add a new ticket , return ticket , check stats , append user , delete user.

*Use case: add a new ticket*

*(Same as use case for a User , check 2.2.1 Use case: add a new ticket)*

*Use case:* return ticket

*(Same as use case for a User , check 2.2.1 Use case:* return ticket*)*

*Use case:* check stats

Diagram:

Brief Description: Moderator gets all stats from the BD.

Initial Step-By-Step Description:

Before this use case can be initiated, the Moderator has to be logged in the system to get the information.

1.Moderator selects the option to get stats.

## *2.System gets the info from DB , and displays it on the Moderator Screen.*

*Use case:* append user

Diagram:

Brief Description: Moderator appends user to the System.

Initial Step-By-Step Description:

Before this use case can be initiated, the Moderator has to be logged in the system to apply changes.

1.Moderator selects the option to append user

*2.Moderator appends info about the User*

## *3.System* appends user.

*Use case:* delete user

Diagram:

Brief Description: Moderator deletes user from the System.

Initial Step-By-Step Description:

Before this use case can be initiated, the Moderator has to be logged in the system to apply changes.

1.Moderator selects the option to delete user

2.System display list of Users*.*

## 3.Moderator click on the User and apply removing.

4.System removes User.

*2.2.3 Admin Use Cases*

Admin has the following cases of use:

add a new ticket, return ticket, check stats, append user, delete user, change ticket price for the route, change the route.

*Use case: add a new ticket*

*(Same as use case for a User , check 2.2.1 Use case: add a new ticket)*

*Use case:* return ticket

*(Same as use case for a User , check 2.2.1 Use case:* return ticket*)*

*Use case:* check stats

*(Same as use case for a Moderator , check 2.2.2 Use case:* check stats*)*

*Use case:* append user

*(Same as use case for a Moderator , check 2.2.2 Use case:* append user*)*

*Use case: delete user*

*(Same as use case for a Moderator , check 2.2.2 Use case: delete user)*

*Use case:* change ticket price

Diagram:

Brief Description: Admin changes ticket price for the route

Initial Step-By-Step Description:

Before this use case can be initiated, the Admin has to be logged in the system to apply changes.

1.Admin selects the option to change ticket price for the route

## 2.System displays all the routes and the prices.

3.Admin select the price that he wants to change from the list and enter new value

4.Admin applies changes.

5.System applies changes to the DB.

*Use case:* change the route

Diagram:

Brief Description: Admin changes the route

Initial Step-By-Step Description:

Before this use case can be initiated, the Admin has to be logged in the system to apply changes.

1.Admin selects the option to change a route

## 2.System displays all the routes(as a start city and finish city)

3.Admin choose the cities from the dropdown list

4.Admin applies changes.

5.System applies changes to the DB

## *2.3 User Characteristics*

Users are expected to be Internet literate and be able to use a search engine.

Also he needs a keyboard to log in into the System and to append Customer info to the System.

He has to be able to use buttons.

Moderator is expected to be Internet literate and be able to use a search engine.

Also he needs a keyboard to log in into the System and to append Customer info to the System.

He has to be able to use buttons.

Admin is expected to be Internet literate and be able to use a search engine.

Also he needs a keyboard to log in into the System and to append Customer info to the System.

He has to be able to use buttons and dropdown lists

**3.0 Requirements Specification**

*3.1 External Interface Requirements*

The only link to an external system is the link to the Route ride manager (RRM) Database to verify the membership of a User.

The RRM Database necessary fields for adding a ticket are personal info, route, type of ticket, date of the ride.

Other info that is placed inside DB is used to perform some additional functions connected to Manager or Admin work(such as stats, or adding new users, etc.)

*3.2 Functional Requirements*

3.2.1 Add a new ticket

| **Use Case Name** | Add a new ticket |
| --- | --- |
| **XRef** | Section 2.2.1, add a new ticket. |
| **Trigger** | User selects the option to add a new ticket. |
| **Precondition** | User`s screen with “add a new ticket” and “return ticket” buttons |
| **Basic Path** | 1.The user presses the button “add a new ticket”  2.The system displays empty fields for necessary information  3.The User enters the information and clicks the "Confirm" button.  4.The system records ticket info into DataBase. |
| **Alternative Paths** | None |
| **Postcondition** | Main screen is shown |
| **Exception Paths** | User can quit at any time. |
| **Other** | None |

3.2.2 Return ticket

| **Use Case Name** | Return ticket |
| --- | --- |
| **XRef** | Section 2.2.1, Return ticket |
| **Trigger** | User selects the option to Return ticket |
| **Precondition** | User`s screen with “add a new ticket” and “return ticket” buttons |
| **Basic Path** | 1.The user presses the button “Return ticket”  2.The system displays empty fields for necessary information  3.The User provides info about ticket.  4.The User clicks the “return ticket” button.  5.The system applies changes into the DataBase. |
| **Alternative Paths** | None |
| **Postcondition** | Main screen is shown |
| **Exception Paths** | User can quit at any time. |
| **Other** | None |

3.2.3 Check stats

| **Use Case Name** | Check stats |
| --- | --- |
| **XRef** | Section 2.2.2, Check stats |
| **Trigger** | Moderator selects the option to Check stats |
| **Precondition** | Moderator screen with “Check stats” and other buttons |
| **Basic Path** | 1.Moderator presses the button “Check stats”  2.The system displays all the necessary information  3.Moderator press “finish” button. |
| **Alternative Paths** | None |
| **Postcondition** | Main screen is shown |
| **Exception Paths** | Moderator can quit at any time. |
| **Other** | None |

3.2.4 Append user

| **Use Case Name** | Append user |
| --- | --- |
| **XRef** | Section 2.2.2, Append user |
| **Trigger** | Moderator selects the option “Append user” |
| **Precondition** | Moderator screen with “Append user” and other buttons |
| **Basic Path** | 1.Moderator presses the button “Append user”  2.The system displays empty fields  3.Moderator append necessary info.  4.Moderator press “Apply” button  5.System add new user. |
| **Alternative Paths** | None |
| **Postcondition** | Main screen is shown |
| **Exception Paths** | Moderator can quit at any time. |
| **Other** | None |

3.2.5 Delete user

| **Use Case Name** | Delete user |
| --- | --- |
| **XRef** | Section 2.2.2, Delete user |
| **Trigger** | Moderator selects the option “Delete user” |
| **Precondition** | Moderator screen with “Delete user” and other buttons |
| **Basic Path** | 1.Moderator presses the button “Delete user”  2.The system displays an empty list of Users.  3.Moderator choose necessary User  4.Moderator press “Apply” button  5.System delete a user. |
| **Alternative Paths** | None |
| **Postcondition** | Main screen is shown |
| **Exception Paths** | Moderator can quit at any time. |
| **Other** | None |

3.2.6 Change ticket price

| **Use Case Name** | Change ticket price |
| --- | --- |
| **XRef** | Section 2.2.3, Change ticket price |
| **Trigger** | Admin selects the option “Change ticket price” |
| **Precondition** | Admin screen with “Change ticket price” and other buttons |
| **Basic Path** | 1.Admin presses the button “Change ticket price”  2.The system displays a list of routes with prices.  3.Admin choose the necessary field and change the value.  4.Admin press “Apply” button  5.System changes the price. |
| **Alternative Paths** | None |
| **Postcondition** | Main screen is shown |
| **Exception Paths** | Admin can quit at any time. |
| **Other** | None |

3.2.7 Change the route

| **Use Case Name** | Change the route |
| --- | --- |
| **XRef** | Section 2.2.3, Change the route |
| **Trigger** | Admin selects the option “Change the route” |
| **Precondition** | Admin screen with “Change the route” and other buttons |
| **Basic Path** | 1.Admin presses the button “Change the route”  2.The system displays an list of start and finish cities.  3.Admin choose the necessary field and change the value from the dropdown list.  4.Admin press “Apply” button  5.System changes the price. |
| **Alternative Paths** | None |
| **Postcondition** | Main screen is shown |
| **Exception Paths** | Admin can quit at any time. |
| **Other** | None |