|  |
| --- |
|  |

|  |
| --- |
| Imam Abdulrahman Bin Faisal University |
| College of Computer Science & Information Technology  Department of Computer Science |

**CS 411 – Software Engineering**

**Term 1 – 2018/2019**

Software Requirements Specification

For

Railway.Manage();



Version 0.1

CS Year 4, G1

Ms.Wadha Almattar

*October 31,2018*

This Software Requirements Specification (SRS) was prepared and provided as a deliverable for Software Engineering, CS 411, Term 1, and it will be used by all developers and stakeholders.

This document is based in part on the IEEE Recommended Practice for SRS Descriptions.

Team Members:

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | ID | Role |
| 1 | Muneera abdullah alhajri | 2160007230 | Leader |
| 2 | Reema Ibrahim Alyousef | 2160003133 | Member |
| 3 | Rahaf Saleh Alzahrani | 2160006662 | Member |
| 4 | Tasneem Hamdy dosoqi | 2160007430 | Member |

**Table of Content**

[Revision History 2](#_Toc436290787)

[1. Introduction 3](#_Toc436290788)

[1.1 Purpose 3](#_Toc436290789)

[1.2 Scope 3](#_Toc436290790)

[1.3 Definitions, Acronyms, and Abbreviations 3](#_Toc436290791)

[1.4 References 3](#_Toc436290792)

[2. Overall description 4](#_Toc436290793)

[2.1 Product perspective 4](#_Toc436290794)

[2.2 Product functions 4](#_Toc436290795)

[3. Specific requirements 5](#_Toc436290796)

[3.1 External interface requirements 5](#_Toc436290797)

[3.1.1 User interfaces 5](#_Toc436290798)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| All members | Oct 30, 2018 | Prepared initial version | 0.1 |

# Table of Tables

Table 1 : List of Definitions 5

Table 2 : List of Acronyms 5

Table 3 describes the login feature provided in the system. 10

# Table of Figures

Figure 1 : System Architecture 8

Figure 2 Use-Case Diagram 9

# Introduction

The following subsection defines all the topics related to the Railway.Manage(); project contained in the Software Requirements Specifications (SRS) document. It provides an overview about the SRS document’s purpose, scope, definitions, acronyms, abbreviations and references.   
The aim of the SRS document is to specify the client requirements in a manner that both the client and the developer can understand.

## Purpose

The purpose of Software Requirements Specifications document is to set up an agreement between the client and the developer, in which all the requirements for the Railway.Manage(); development is precisely detailed and completely satisfied. This can assist the developers greatly when designing the software. Furthermore, this document describes all the functionalities that are important for the Railway.Manage(); to have. Keep in mind, some of the concepts and ideas that are considered and outlined in this document might be discarded as the team proceeds with the project development. This document will also be used as a reference to verify the end software.

The intended audience for the SRS document are the software development team, project designer, project supervisor (Mrs. Wadha Almattar), client (Mr. Albert Peterson). The software development team can use this document as a reference for software implementation and for the traceability of the functions implemented in the software.   
Mrs. Wadha Almattar will review the document and suggest any improvements that would improve the software. Most importantly, this document will help Mr. Albert Peterson to bridge up the communication gap and understand the software clearly.

## Scope

The SRS document will outline all the concepts, ideas and requirement that have been collected form the client. Also, it shows how the software will work by defining all the functions that are important for the Railway.Manage(); to have. The aim of this software is to ease the organization and maintenance of a train station.by enabling the end-users (passengers) to access the time table of the station, book a trip, inquire or update their ticket info at any time. This desktop software will benefit the passengers in which they can access and control their trips through their own PCs. This software will be only for desktop computers, any other devices like smartphones or tablets will be beyond the scope of this software.   
Railway.Manage(); has two main actors:

* **Operators:** They will be responsible about all aspects of the system with no restrictions, since they are permitted to access each component of the system and are also allowed to update or delete entries in time tables.
* **Passengers:** They will be able to access and review timetable of the station, book a trip, or inquire and monitor their ticket.

## Definitions, Acronyms, and Abbreviations

Table : List of Definitions

|  |  |
| --- | --- |
| Terminology | Definition |
| SPMP | Software Project Management Plan is a complete plan for software development that includes standards, methods and tools needed for development. As well as stating the budget and duration of the project. |
| SRS | A document that specifies all the requirements that needs to be completed before the completion of the project |
| Program | A set of instructions for the computer to follow. |

Table 2 below specifies the acronyms used in this document:

Table : List of Acronyms

|  |  |
| --- | --- |
| Acronym | Definition |
| SRS | Software Requirements Specification |
| E-Mail | Electronic mail |
| PC | Personal Computer |
| IEEE | Institute of Electrical and Electronics Engineers |

## References

[1] IEEE Standard for Software Project. [Online] 1998. IEEE Std 1058-1998.

# Overall description

This section of the document contains an overall description of Railway.Manage ();. These two sections include explanation about the application’s perspective and its basic functionalities.

## Product perspective

Railway.Manage (); is a one-stop stand-alone software that delivers different functionalities to both the passengers (users) and the operators of the train station to meet their needs. It is a desktop software where the passengers be able to access and review the timetable of the station, book a trip, inquire and monitor their ticket or check the trip schedule during the booking process. Operators are responsible about all aspects of the system. there are no restrictions for them because they can access any component of the system. They can view the timetable for each train and edit them freely.

Railway.Manage (); system has a single database connected to it. Moreover, the system architecture allows the passengers (end users) to access the database indirectly via application. Also allows the operators who have authorization to reach the database directly and manage the application. Figure 2.1 shows the overall architecture of the system.

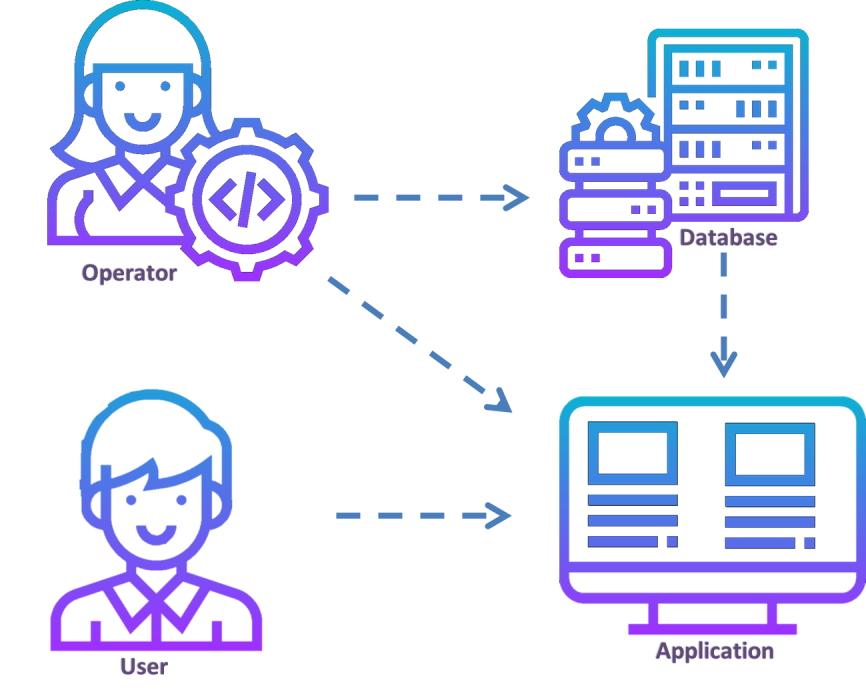


Figure : System Architecture

## Product functions

Railway.Manage ( ); software system is intended to offer major functions and services that serve the end users need. The description about each major function are listed below.   
Figure 2 illustrates the functions provided in Railway.Manage();



Figure Use-Case Diagram

* Account   
  A log in feature is provided in the system, this feature is critical to increase the security and reliability of the system. However, operators are the ones who must log in before accessing their functions, while passengers are only required to enter their information.
  + All the operators in the system shall have their own personal account.
  + Each operator shall be able to choose and change his/her password.
  + Each operator shall be able to log in and log out from his/her personal account.
  + Each operator shall be able to choose his/her password.
  + If the operator forgets his/her own password account, an email will be sent to his/her personal email after receiving a message with special code.
* Booking
  + Each passenger can choose a seat on the train and choose their desired classes (silver , gold, economic) with each having unique characteristics.
  + During booking, the passenger must enter correct personal information for the required information. Each ticket will include: name, age, date of trip, time departure and arrival station.
  + The passenger is required to choose if the ticket is a one way or a round trip.
  + If the booking is done successfully, a number will be generated. This number is used to enable the passenger to view the ticket when desired.
* Modifying: The Operator is responsible for editing and maintaining the train schedule, which is shown in the various monitors throughout the train station.  
  Through editing, the operator is capable of achieving the following:
* Operator can add a new trip to the train schedule whenever a new trip is scheduled.
* It’s possible to modify the trips in the schedule, whether it might be to change the state of a trip, or to announce any urgent changes.
* Trips whose state is “Arrived” will be deleted shortly after by the operator.
* Passenger Information: Before booking a ticket, the passenger is required to enter their personal information, this information includes:
* The passenger’s contact information, represented as their E-mail and phone number. This data is used to reach them when any updates regarding their trip occur.
* Personal info such as first and last name, alongside the national number. Which will be essential for identifying the passenger inside the database.

All of these are stored in the database in their corresponding tables.

* Train Schedule: Throughout the train station itself, many monitors will be displaying the train schedule to inform the visitors of the current state of the trips going in and out of the station.  
  Information displayed in the schedule are: Departure and arrival time, alongside the name of the station the train departed from.
* Ticket Viewing: The passenger will be asked to enter the previously mentioned generated number if they want to view their ticket. (This number is provided to the user right after booking successfully).

# Specific requirements

This chapter underlines some essential requirements for the development process, such as the external interface requirements, in addition to the functional requirements for each intended user.  
These requirements were built based on the client’s specifications during the conducted meetings and were reviewed and checked by all the team members.

## External interface requirements

Inputs and outputs of the system are explained in this section, alongside a description of all the user interfaces provided in the system.

### User interfaces

Since interfaces are considered the main connection between the program and the user, the team made it a priority to develop an easily accessible and clear interface for the Railway.Manage(); system. To achieve this goal, the team employed the following features in the application:

* Visual aid: The most important functions to the user are made to be the main focus of their vision, either by placing them in the most significant areas to the user’s eyes or highlighting their importance by using different fonts and tools.
* Labels: Are relevant to their meaning, besides using simple and clear language.
* Language: English is the language used in the program.
* Font: The font is appropriately sized to suit the average user.  
  + - 1. **Home page:**

The home page is the portal for both the passengers and the operators, it allows the user to choose the action they wish to make. Whether they want to book, view a ticket or to view the train schedule, besides providing a button to access a separate interface for the exclusive operator functions.

* + - 1. **Quick Booking interface:**This interface is accessible from the home page. It is designed to allow the user to book a ticket through multiple stages, each represented in separate pages and are sequentially ordered to appear to the user right after each other.

* + - * 1. **Passenger information:**When a user chooses to book, they will be asked to enter their personal and contact information first. After finishing this process, the user must press “Next” to advance to next the stage for booking.
        2. **Booking:**Here the user is required to specify various matters regarding their trip, for example they will be asked to select the departure and arrival date, alongside selecting the class, type of trip (round or one way) and the number of adults and children boarding the train.
        3. **Seat selection:**

This interface comes right after the booking interface when the user proceeds to push the “Next” button. In this page the user is asked to select their desired seat in the train.

* + - * 1. **Successful Booking:**

The aim of this interface is to inform the passenger that the booking has been done successfully, and to give them a generated number which they can use to view their ticket at any time. This interface will only be visible after all the previously mentioned stages are done.

* + - 1. **View Ticket:**

Can only be accessed from the home page, this interface allows the user to view their ticket at any time, the ticket information will be visible after entering the generated number that the passenger acquired after booking successfully.

* + - 1. **Train Schedule:**

Here the user can view information about all the trips arriving to the station, this information includes the departing station and the departure and arrival time of the trip.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Format | Level | I/O | Comment |
| Username | Text | Mandatory | Input | Unique, represented as the operator’s name record in the database. |
| Password | Encrypted text | Mandatory | Input | Must correspond to the username in the database. |

* + - 1. **Operator login:**To access the operator portal, the operator must firstly login successfully, which will be done by asking the operator to enter a valid username and its corresponding password.

**Table 3 describes the login feature provided in the system.**

* + - 1. **Operator Interface:**After logging in, this interface will be shown to the operator, where it is possible for the operator to manage the train schedule, either by adding, deleting or editing a trip value from the database.