# SOFTWARE REQUIREMENT SPECIFICATIONS

**Project Name** : Ticketing System with Multiple Stops

**Document Title** : SRS

**Project Type** : File Handling in C

**Project Timeline** : 05.12.2022 to 12.12.2022

# TABLE OF CONTENTS

# INTRODUCTION

Ticketing System with Multiple Stops is a system to provide route information of the bus with all stops along with the fares. It allows us to view the booked and vacant seats. It facilitates booking tickets in an optimal way by checking if the seat requested by the user is available or not, thereby generating the ticket if the seat is available.

## 1.1 Overview:

The Ticketing System with Multiple Stops facilitates an over-the-counter bus ticket booking functionality. The input to the system will be the passenger's boarding point and drop-off point along with their desired seat number. The system checks for the availability of the requested seat. If it's available along the passenger's route, it collects the passenger's information such as name, phone number, age, etc. Finally, the ticket will be generated.

## 1.2 Project Scope:

The objective of the project is to provide a ticketing system that ensures seat allocation and ticket booking in an optimal manner with ease. To summarize, it covers the following aspects -

- Efficient booking management
- Seat allocation in an optimal way
- Ticket generation and storing in a file as backup

# OVERALL DESCRIPTION

## 2.1 Project Feature

- The user interface is attractive and user-friendly. The entire process is menu driven.
- The system displays the routes and fares for user understandability.
- Seats booked and seats vacant are available to analyze the availability of the seats
- Once there is a vacant seat, a ticket will be generated for the user with all the relevant information.
- The ticket records are stored in the file as a backup.

## 2.2 User Needs

1. User Characteristics: The user should be familiar with menu-driven Applications.
2. General Constraints: A full internet connection is required for Linux (Operating System).
3. Intended audience:
    - Passengers
    - Clerks

## 2.3 Operating Environment

The operating environment for the application is listed below

- Operating system: Any Linux-based OS

# SYSTEM FEATURES

## 3.1 Functional Requirements

**TS_FR01:**

Storing route information: Using file handling in C, storing the route information with all the stops and distance from the start stop in a file and loading the same into the main memory.

**TS_FR02:**

Displaying the details of Route: Route information with all stops is displayed to the passenger or user using a file and reading from it.

**TS_FR03:**

Displaying price chart: The price chart is displayed to the user as a matrix by storing it in a file and reading from it.

**TS_FR04** :

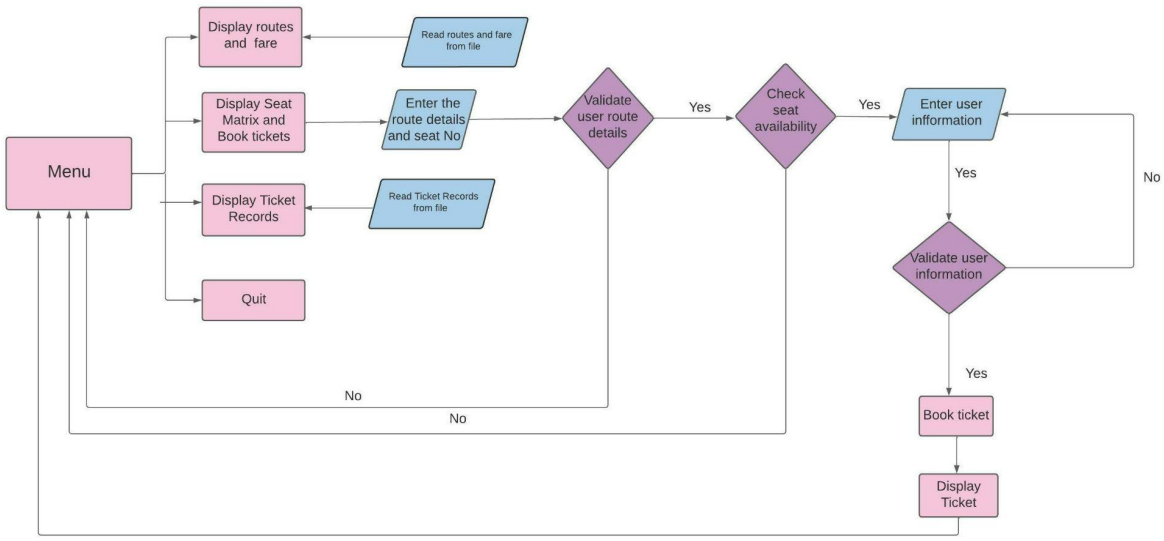
Reading Start stop and Stop stop: Taking the input from the user about the user's boarding point, drop off point and then fetching the fare for the user's journey.

## TS_FR05:

Displaying booked seats and vacant seats: A seat matrix with the available seats and booked seats will be displayed to the user.

## TS_FR06:

Ticket Booking in an optimal way: Ticket booking is done in an improved and optimized way at intermediate stops as well the ticket will be stored in a file and can be viewed later.



## 3.2 System Requirements

System Requirements are types of functional requirements. These are features that are required in order for a system to function.

Software Interface:

Operating System: Windows XP (32/62 bit) and Linux which supports networking.

Hardware Interface:

Hardware requirements are:

- Processor: i3 or above
- ROM: 1TB (SSD/HDD)
- RAM: 8 GB or above

# NON-FUNCTIONAL REQUIREMENTS

**1. Maintainability**: Software must be capable of being maintained cost-effectively throughout its lifetime and can be modified with additional requirements.

**2. Performance:** Software must be quick to respond to users' actions. Commands must not take much time to run.

**3. Compatibility**: Software must be compatible with all Linux environments.

**4. Scalability:** Performance must be as expected even if the workload is high.

**5. Availability**: Users must access the system whenever they want.

# EXTERNAL INTERFACE REQUIREMENTS

I. User Interface:

a) GUI: There is no GUI involved or created for the project/application

b) CLI: The application is based on CLI, and the commands are given through it.

II. Hardware Interface:

The Application uses/accesses the hard disk for storing the data and to access the files. Access to the hardware requirements is managed by the operating system and the application.

a) LINUX-based operating system.

b) Terminal to run.

# FUTURE ENHANCEMENTS

Future enhancements involve enhancing the system to support multiple routes. This would allow the system to manage multiple buses/trains with different routes.

# APPENDIX

- https://www.javatpoint.com/file-handling-in-c
- https://www.geeksforgeeks.org/menu-driven-program-to-implement-travel-agency/
- https://study.com/academy/lesson/validating-input-data-in-c-programming.html