Contents

[1.Introduction 2](#_Toc382824946)

[1.1 Purpose 2](#_Toc382824947)

[1.2 Document Conventions 2](#_Toc382824948)

[1.3 Intended Audience and Reading Suggestions 2](#_Toc382824949)

[1.4 Product Scope 3](#_Toc382824950)

[2. Overall Description 4](#_Toc382824951)

[2.1 Product Perspective 4](#_Toc382824952)

[2.2 Product Functions 4](#_Toc382824954)

[2.3 User Classes and Characteristics 5](#_Toc382824955)

[2.4 Operating Environment 6](#_Toc382824956)

[3.External Interface Requirements 7](#_Toc382824957)

[3.1 User Interfaces 7](#_Toc382824958)

[3.2 Software Interfaces 10](#_Toc382824959)

[Both the client and server computer do not need internet browser to work online. 10](#_Toc382824960)

[3.3 Communications Interfaces 10](#_Toc382824961)

[4 System Features 11](#_Toc382824962)

[4.1 Functional Requirements 10](#_Toc382824961)

[4.2 Non Functional Requirements](#_Toc382824961) 11

[4.3 Behavior Requirements](#_Toc382824961) 11

[5 Software Quality Attributes 12](#_Toc382824962)

[5.1 Performance](#_Toc382824961) 12

[5.2 Safety and security](#_Toc382824961) 12

# 1.Introduction

## Purpose

The purpose of this document is to describe the Software Requirement Specifications of a Bus Ticket Management System. It aims to lay down guidelines that have to be followed while developing the bus ticket management system. It also includes the description of software and the IDE’s used.

## Document Conventions

The standard IEEE format for Software Requirement Specification has been followed throughout the document.

Some standards that were followed while writing this document are as follows:

* Headings-Times New Roman, Size-18 pts, Bold.
* Subheadings- Times New Roman, Size-14 pts, Bold.
* Body or Text-Arial, Size-12.
* Italics for Comments.
* Single-Spaced Text.

## Intended Audience and Reading Suggestions

This document intends to cater to all potential stakeholders in the system. These include, but are not necessarily limited to, the following: Users(People looking to book a ticket) and developers(Programmers).

The objective of this document therefore is to formally describe the system’s high level requirements including functional requirements, non-functional requirements and constraints. The detailed structure of this document is organized as follows:

* Section 2 of this document provides an overview of the domain that the proposed Bus Ticket Management System will support. These include a general description of the product, user characteristics, general constraints, and any assumptions for this system. This model demonstrates the development team's understanding of the business domain and serves to maximize the team's ability to build a system that truly does support the business.
* Section 3 of this document describes the various interfaces of the software system including the User Interfaces, Software Interfaces. It includes screenshots of the actual developed Bus Ticket Management System.
* Section 4 organizes the functional requirements for the product by system features, the major services provided by the Bus Ticket Management System.
* Section 5 describes the non functional requirements of the Bus Ticket Management System.

## Product Scope

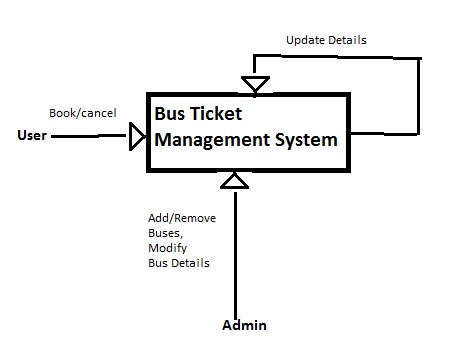
This software resides on a local computer and anyone willing to use this software has to access it on the computer locally, any changes made are reflected immediately and the user who uses the system next will see the latest updated information.

So the product scope is limited to local computers that are present at travel offices, and this software can try and replace the hard copies of all travel details that are maintained by travel agents.

# Overall Description

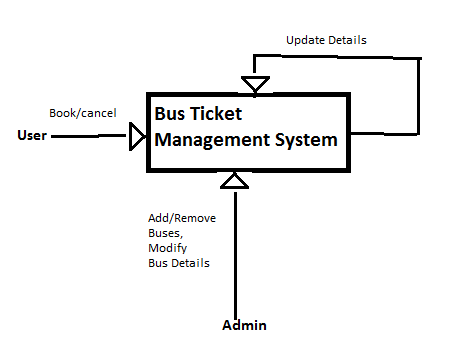
## Product Perspective

The Bus Ticket Management System that is to be developed is a new self contained product that will contain information about a set of buses and passengers travelling on that particular bus. The perspective of the product is to replace the travel books used by travel agents and replace them with a well maintained, reliable soft copy of the travel details.



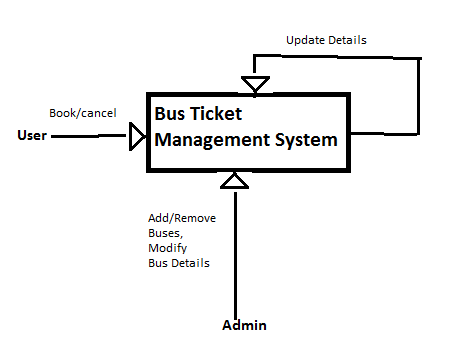
Update

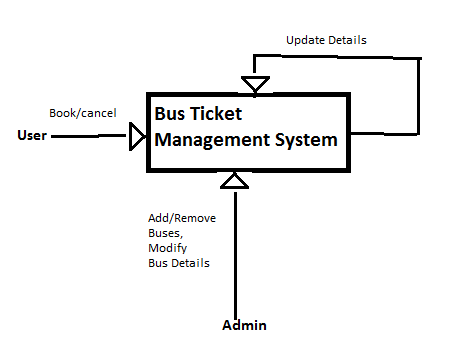
Book/cancel Tickets



Bus Ticket Management System

User





## 

Modify Bus

Details,

Add/Remove

Buses

Admin

**Figure 2.1: Logic Diagram of the Software**

## Product Functions

A software system needs to be developed to maintain a Bus-Ticket Management System.

The software includes functionalities for

1.User

2.Administrator

The user can book/cancel tickets.

1. To Book a ticket He / She selects the SOURCE Station &

DESTINATION Station(for a given Date),  upon which a list of buses plying on the chosen route is displayed.

  The user selects a bus, which results in the displaying of the Seat Matrix of the bus     showing the available seats and booked seats.

  The user selects a seat and enters details like

  1.Name

  2.Address

  3.Contact No.

  4.E-Mail

  after which the a ticket describing all the travel details is generated.

1. To cancel  a ticket He / She enters Bus Number, Seat Number, Date of Journey.

    The ticket is cancelled and the cancelled seat is shown to be available in the seat matrix.

The administrator of the Bus-Ticket-Management system has to verify his/her identity by entering the password, which will give him/her administrative privileges.

The administrator can

1. Modify routes of buses

2. Alter schedules of buses

3. Modify Prices of the tickets

4. Add or remove buses

The changes made by the administrator are updated and user sees the latest updated information.

## User Classes and Characteristics

It identifies the various use classes that will use system. The system will be used in a Travel agency. The administrators, front-desk staff and passengers will be the main users .Given the condition that not all the users are computer-literate. The system is also designed to be user-friendly. It uses a Graphical User Interface (GUI).

Assumptions: All users have general reception and secretarial duties. Every staff has some basic computer training. They are responsible for checking seat availability and reserving it for

appropriate passenger.

They all have post-secondary education relating to general business administration practices. Every administrator has basic computer training. They are responsible for all

of the scheduling and updating day/night employee shifts.

## Operating Environment

The Bus Ticket Management System is developed on the Netbeans IDE, and compiled as a Java Project. The source code files are put together in a src folder and included under one package.

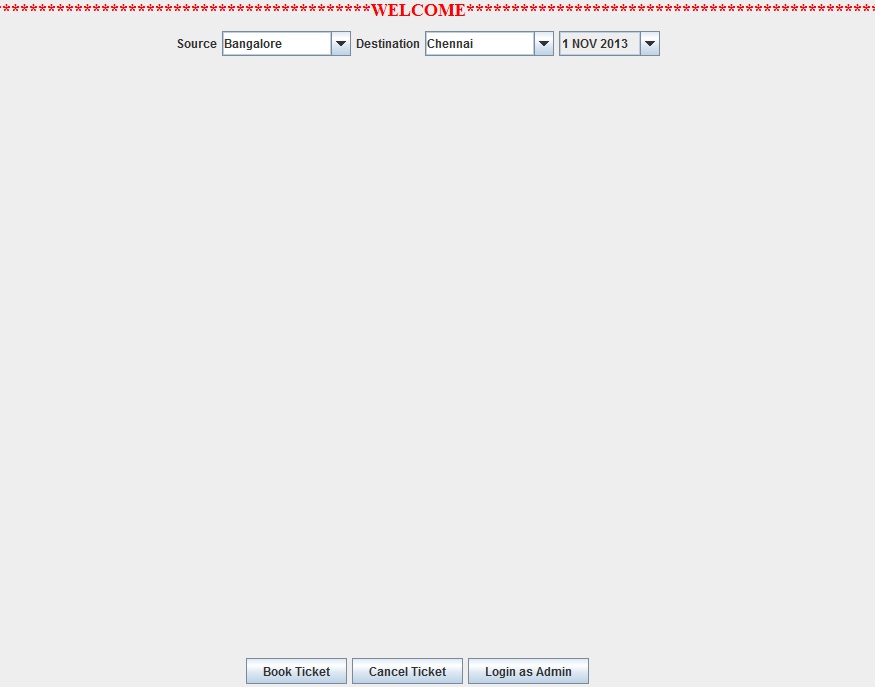
This can be run on Operating Systems which have the Java Development Kit along with Java Run Time environment installed on it.

# 

# 3.External Interface Requirements

## User Interfaces

The Bus Ticket Management System will not be web based. Format of main screen shall be standard and flexible. The system shall be user friendly designed. Pages shall be connecting each other in a consistent way. Operations which can be done with the system shall be repeatable. The design of the pages should allow the users to do this.

****

**Main Booking Page**

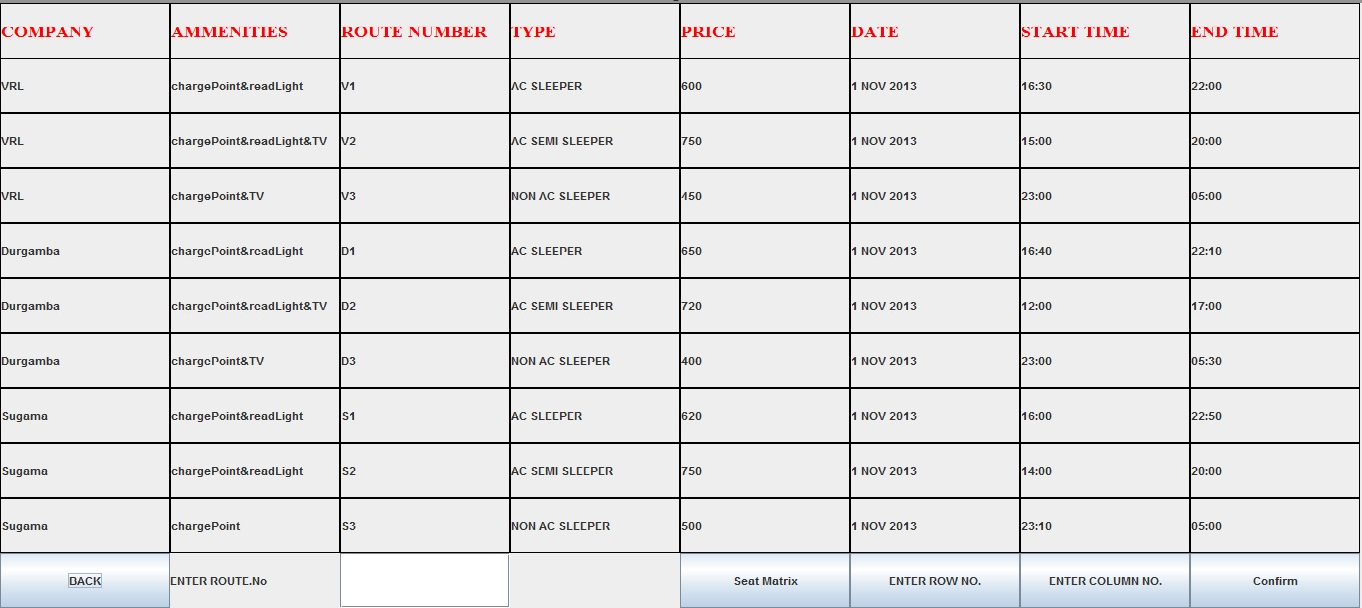
As it can be seen above, the main interface includes a Book ticket button which will resume only after the source and destination of the user is inserted with the date. It also includes a button for Cancel ticket if the user changes their mind. The third button is for the Admin to login, once pressed it will ask for their username and password and the system retrieves the admin menu.



**The Admin Menu**

The Admin Menu Page contains 4 fields which are labeled as ‘Add a bus’, ‘Modify a bus’, ‘Book a Ticket’ and ‘Remove a bus’. The admin can enter values to these required fields by clicking on the respective button.

There shall be other pages which have functionality related with the customer operations and admin tool operations. The example figures are added.

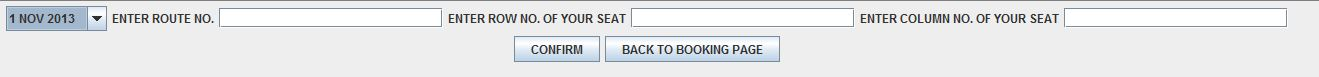
****

**List Of Buses Plying on a route.**

The bus list is a table containing the Company name along with the details of the bus:

* Amenities: gives all extra features present in that particular bus
* Route number: to differentiate between buses of the same company
* Type: tells the user whether the bus is a sleeper/non sleeper and if It has A/c.
* Price: Amount to be paid
* Date: date of departure
* Start and End time: The time of departure and arrival

At the bottom the user has to enter the route number and the choose the seat position from the seat matrix given. Once finished choosing the user can use the confirm button.

****

**Cancel Page**

The cancel page is also present to remove a reservation placed on a seat in any specific bus.

Enter the details of the bus: the date, the route number and your seat number.

## 3.2 Software Interfaces

## Both the client and server computer do not need internet browser to work online.

The Bus Ticket Management System is developed on the Netbeans IDE, and compiled as a Java Project. The source code files are put together in a src folder and included under one package.

This can be run on Operating Systems which have the Java Development Kit along with Java Run Time environment installed on it.

The System can be run on a Java virtual machine.

A Java virtual machine (JVM) interprets compiled Java binary code (called byte code) for a computer's processor (or "hardware A **Java Virtual Machine** (**JVM**) is a process virtual machine that can execute Java byte code. It is the code execution component of the Java platform. Sun Microsystems has stated that there are over 5.5 billion JVM-enabled devices.platform") so that it can perform a Java program's instructions.

Java was designed to allow application programs to be built that could be run on any platform without having to be rewritten or recompiled by the programmer for each separate platform. A Java virtual machine makes this possible because it is aware of the specific instruction lengths and other particularities of the platform.

## Communications Interfaces

This software can only be used on a **local computer** and thus communication functions like e-mail, web browsers and network servers are not included. Any changes made are reflected immediately and the user who uses the system next will see the latest updated information.

# System Features

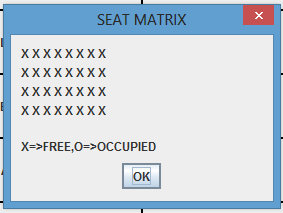
**4.1 Functional Requirements**

**Book Ticket:**

* Check Availability: The passenger must be allowed to see all available options for a journey and see if a particular seat is available or not. He should be able to view all the buses plying on a given route on a specific day and be able to check seat availability on all the buses.
* Book Ticket: Then if the ticket is available then the seat should be booked, by entering the passenger details like Name, Contact No., E-mail id, Age.

**Report Generation:**

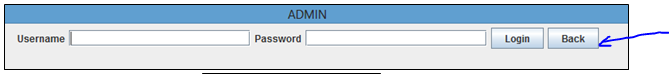
* Bus List: The Bus Ticket Management System should contain files that contain the list of buses travelling on a particular route.
* Passenger List: The Bus Ticket Management System should contain files that contain the list of passengers travelling on a particular bus.
* **Seat Availability:** The Bus Reservation System should generate reports on seat availability about the following information: bus number, seat number, occupied/unoccupied. It does this by displaying the seat matrix of the bus.

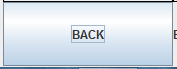


* **Passenger related Information:** Each passenger should have the following mandatory information: first name, last name, phone number, Bus details .
* **Bus related Information:** Each bus should have following information: bus number, no of seats, bus type: normal , AC, Sleeper, Source, Destination.
* **Update Bus Information:** The Bus Reservation System shall allow the Administrator to update any of the bus information like type, source , destination, ticket price.

**4.2 Non Functional Requirements**

* There is a back button on all pages, which lets the user go back to the immediate previous page.

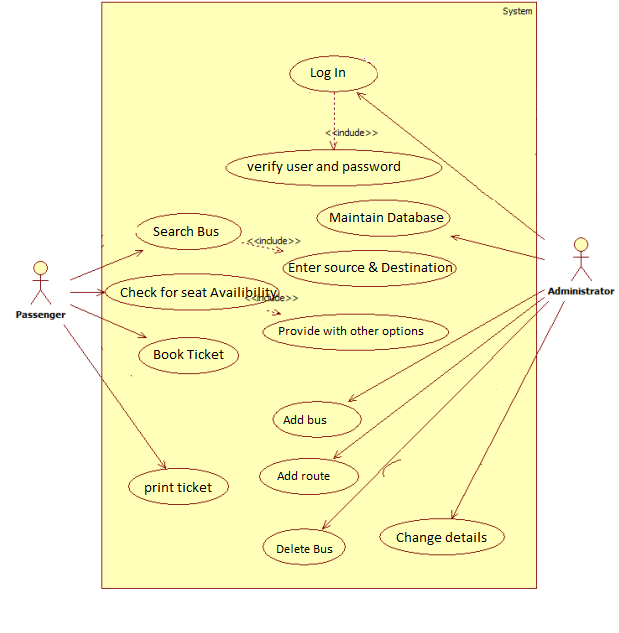




**4.3 Behavior Requirements**

**Use Case View**

A use case defines a goal-oriented set of interactions between external actors and the system under consideration. Since sometimes we will not be able to specify completely the behavior of the system by just State Diagrams, hence we use use-cases to complete it.



**Use Case Diagram for the Bus Ticket Management System.**

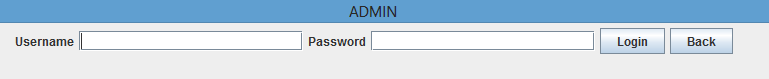
**5 Software Quality Attributes**

**5.1 Performance:**

* **Back Up :** The system shall be able to provide the capability to store the Data. The software should write to files, and data is read from these files when the user checks queries for data.
* **Response Time**: The system shall give responses in 1 second after checking the user’s information. This is possible because all the data resides on a personal computer locally, and there is no delay as files need not be passed from one computer to another.

**5.2 Safety and Security Requirements**

* **Login ID** : Admin who uses the system shall have a Login ID and Password.



* **Modification** : Any modification (insert, delete, update) to the Database shall be synchronized and done only by the administrator, who can do so only after verifying his credentials by typing username and password.
* **Administrators' Rights** : Administrators shall be able to view and modify all information in Bus Reservation System.