Bus Ticket Reservation System

Software Requirements Document

V 1.0

Table of Contents

Introduction

1.1 About this document

1.1.1 Purpose & Scope of the document

The purpose of the software requirements document is to systematically capture requirements for the project and the system **Bus Ticket Reservation System (BTR)** application. BTR is a system used for managing the activities of a managing the activities of a travel agency like Booking ticket, Schedule travel etc.

The scope of this document is limited to addressing the requirements from a user, quality, and non-functional perspective. It is recommended that design aspects are not added in this document

1.1.2 Intended Audience

Project Team

1.2 About the Software System

The client would like to develop an independent application **Bus Ticket Reservation System (BTR)**. The core essence for the system is to managing the activities of a travel agency like Booking ticket, Schedule travel etc.

The following section will cover aspects related to Bus Ticket Reservation System.

1.2.1 Purpose

The Bus Ticket Reservation System (BTR) core essence for the system is managing the activities of a travel agency like Booking ticket, Schedule travel etc.

The following are the important modules in the system

a) Customer Registration
b) Book Ticket
c) Cancel Ticket
d) View history and update profile
e) Manage Master Data

1.2.2 Scope of the system

The scope of the system is explained through its modules as follows

- ○ Customer Registration - The system should calculate the priority of the customer, generate the customer id and store the details in the system.

- ○ Book Ticket - By administrators to book the tickets for the customer into the system

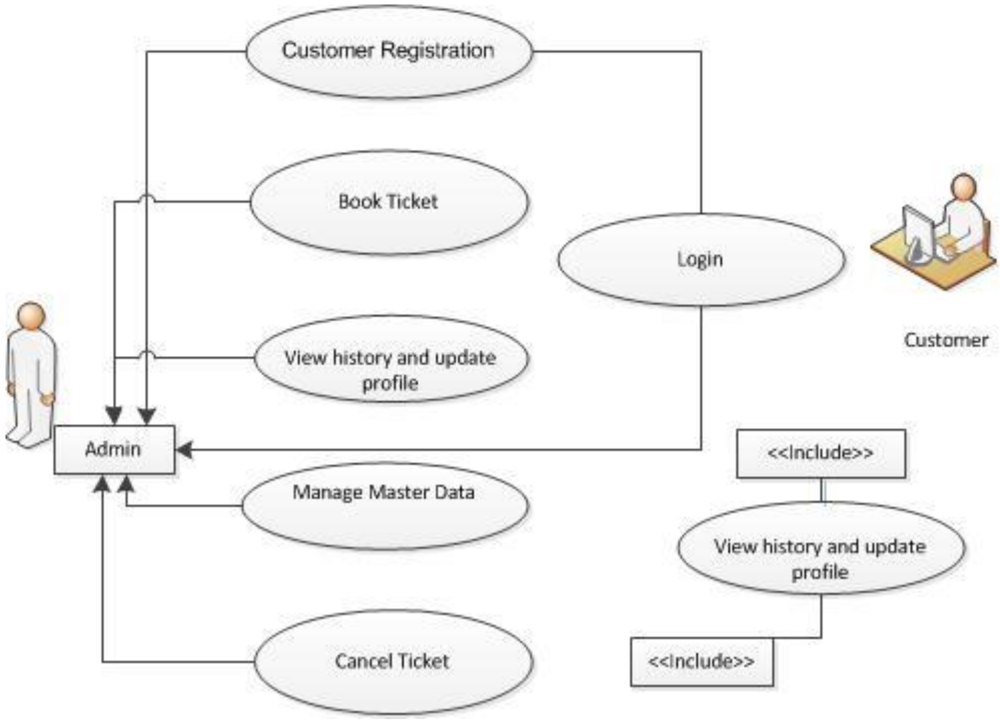
- ○ Cancel Ticket - By administrators to cancel the tickets for the customer into the system

- ○ View history and update profile – By administrators to view history and update the customer information like name, address, contact no etc.

- ○ Manage Master Data - By administrators to manage master data like Bus information, route information, bus schedules etc. The administrator should have the data to be updated into the system.

Use Case Diagram

Bus Ticket Reservation System



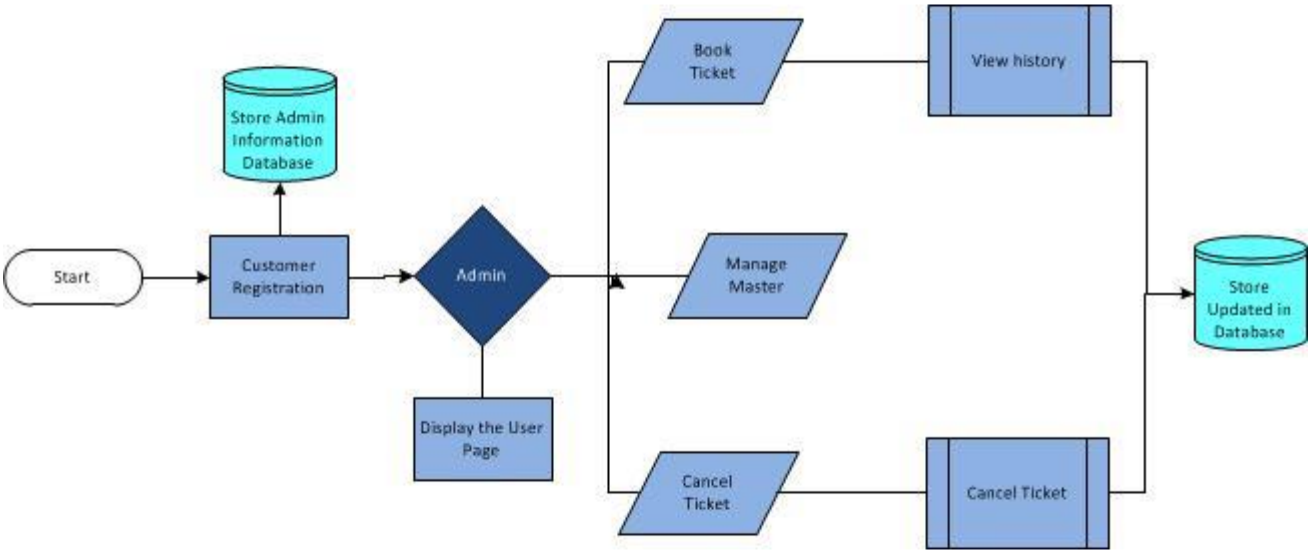
1.2.3 Exclusions

The system will operate only on the modules discussed above and will not include any additional functionality.

1.2.4 System Perspective

Bus Ticket Reservation System is an independent software system developed to manage the activities like BTR Customer Registration, Book Ticket, Cancel Ticket, View history & update profile and Manage Master Data using the Java/ Dotnet architecture.

1.2.5 System diagram



1.2.6 Architecture diagram

Physical Architecture:

A physical architecture is an arrangement of physical elements, (system elements and physical interfaces) that provides the designed solution for a product, service, or enterprise. It is intended to satisfy logical architecture elements and system requirements. Auto Identification Process follows a three layered architecture namely presentation layer, business logic layer and data access layer.

- **Presentation Tier** is the tier in which the users interact with an application. Presentation Tier contents Shared UI code, Code Behind and Designers used to represent information to user.

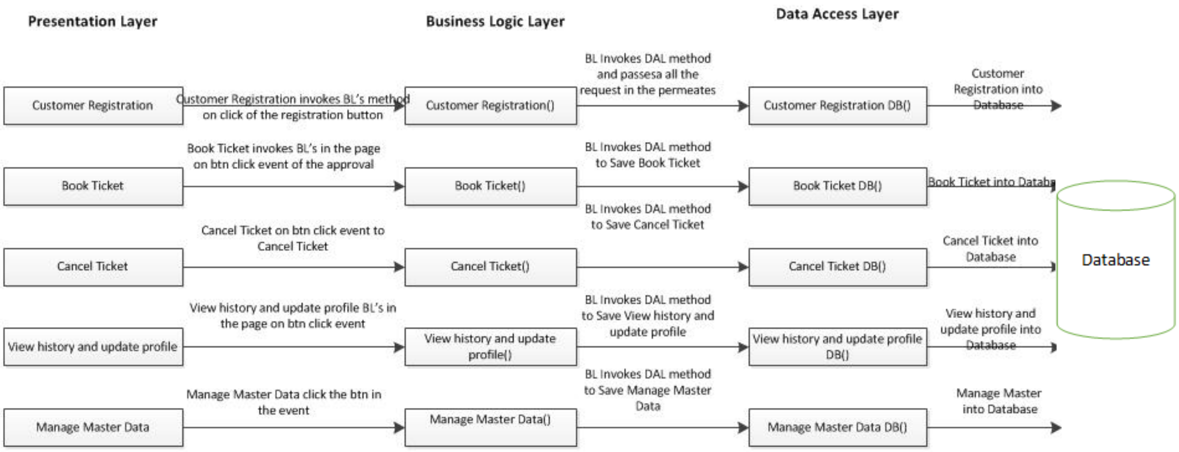
- **Business Tier** is mainly working as the bridge between Data Tier and Presentation Tier. All the Data passes through the Business Tier before passing to the presentation Tier. Business Tier is the sum of Business Logic Layer, Data Access Layer and Value Object and other components used to add business logic.

- **Data Tier** is basically the server which stores all the application's data. Data tier contents Database Tables, XML Files and other means of storing Application Data.

Logical Architecture:

The Logical Architecture defines the Processes (the activities and functions) that are required to provide the required User Services. Many different Processes must work together and share information to provide a User Service. The Processes can be implemented via software, hardware, or firmware. The Logical Architecture is independent of technologies and implementations

3-Layered Architecture Logical Diagram


Presentation Layer
Business Logic Layer
Data Access Layer
Customer Registration
Customer Registration invokes BL's method on click of the registration button
Customer Registration()
BL Invokes DAL method and passesa all the request in the permeates
Customer Registration DB()
Customer Registration into Database
Book Ticket
Book Ticket invokes BL's in the page on btn click event of the approval
Book Ticket()
BL Invokes DAL method to Save Book Ticket
Book Ticket DB()
Book Ticket into Databa
Cancel Ticket
Cancel Ticket on btn click event to Cancel Ticket
Cancel Ticket()
BL Invokes DAL method to Save Cancel Ticket
Cancel Ticket DB()
Cancel Ticket into Database
View history and update profile
View history and update profile BL's in the page on btn click event
View history and update profile()
BL Invokes DAL method to Save View history and update profile
View history and update profile DB()
View history and update profile into Database
Manage Master Data
Manage Master Data click the btn in the event
Manage Master Data()
BL Invokes DAL method to Save Manage Master Data
Manage Master Data DB()
Manage Master into Database
Database

1.2.7 System Environment

Bus Ticket Reservation System m application will be operated from the client server with parallel processor support. When a user connects to the Web Server, the Web Server will interact with the Database after processing the business logic to transfer data to and from a database.

- Java/J2EE or .NET Framework application deployment.
- My Sql/Sql server database for data storage

1.2.8 User Characteristics

The application should be user friendly; hence the user should not need any software and hardware knowledge. The user may be new to the client products having never used them before.

The types of users likely to use the system are as follows,

- Client
- Delivery Assurance Group, Process Engineering group
- Developers, testers, other associates in the project

1.2.9 Impact of the System

This is a new product which is developed for internal users. Expected impact of the product is to automate existing manual processes in order to make them more efficient and cost effective.

1.2.10 Assumptions, Dependencies, Risks / Constraints

Assumptions:

The BTR Admin must possess transaction data along with their application Bus Ticket Reservation System (BTR). The core essence for the system is to managing the activities of a travel agency like Booking ticket, Schedule travel etc.

Dependencies:

If there is any similar Identification process required for other applications of the same client, this system should be able to accommodate such a reusability dependency.

Risks:

If there is any failure in identification processes, the system should be able to handle such a financial risk.

1.2.11 Design Constraints

Application should have single login feature. Login functionality should be the welcome feature for the application.

Only administrator can access the application.

None of the application features can be accessed without login.

Frameworks

- **SpringBoot Rest API for connecting to database**
- **Client App using Spring Framework/ Spring Boot**
- **Use RabbitMQ for sending the messages**

System Requirements

1.3 Functional Requirements

1.3.1 Bus Ticket Reservation System - Customer Registration

Bus Ticket Reservation System **Req-2.1.1**	**Customer Registration**
When the users enter the following link **http://localhost:8080/BusTicketReservationSystem** in the browser, Home page appears which will have a short welcome message; a brief introduction to the purpose of the Bus Ticket Reservation System, the user can choose appropriate options. Under home page, we need below interface and Home page Interface Requirements. Enroll Patient form should have basic attributes/fields as mentioned in UI Interface Screen section. At any given point of time, the user can Deselect/Reset all or can go to home page by the clicking home.	
UI Interface Screen	Name - The Customer name should be filled by the user Address - The Address should be filled by the user City - The city should be filled by the user State - The Citizenship should be filled by the user Country - The Country Type should be filled by the user Pin code - The pin code should be filled by user Email Address - The Email Address should be filled by the user Gender - The Gender should be filled by the user Contact no - The Contact no should be filled by the user Date of Birth - Enter the input to represents the customer DOB in DD/MM/YYYY format Customer type - Customer Type Customer pwd - The password should be filled by user

	Confirm pwd - The Confirm password should be filled by user
Trigger	None
Pre-Conditions	The User need to enter all the details of the customer to be registered into the system.
Post Conditions	The system should find weightage of the customer, generate the customer id and stored along with the customer details.
Success End Condition	On success a message should be displayed "Customer Registered successfully"
Failed End Condition	The system should throw an error message if one of the validations is wrong. The error message and error code needs to be identified during design. The validation and business rules are mentioned in the business rules section.
Steps & Actions	1. System should display all the necessary details. 2. The countries list all values should be fetched from database on page. 3. Selecting the country should automatically get the names of the respective states only. 4. Get all the necessary information. 5. On register or Update the system needs to calculate the weightage based on the customer type 6. Identify the priority based on the customer type details as follows: The priority of the customer should be based on designation. I) The government officer should be given VIP status . II) The Customer type can include government officer, government employee, private employee, student and others. 7. Identify the Weightage of the customer based on the priority 8. Generate the customer ID as follows. **Customer ID= current year (4 digit) +current month (2 digit) +unique number (4 digit).** **Example: 2012055001.** 9. Store the details into the system

Business Rules & Validations	- The name should contain only alphabets and space. - Pin code should be 6 digits. - Contact no should be 10 digits. - Email id should be in a valid format (***abc@xyz.com***). - Age should not be less than 18. - Age should not be above 80 years. - All details are mandatory. - Address should not contain special characters other than white space. - At the time of registration, the following task needs to be done: ○ Calculate the weightage based on customer type. ○ Government officer should be assigned as VIP and other should be consider as Normal Users. ○ Generate the customer number. - Update the above details automatically into the database along with the customer details.

1.3.2 Book Ticket

Bus Ticket Reservation System **Req-2.1.2**	**Book Ticket**
By administrators to book the tickets for the customer into the system. The administrator should have the details from the customer to be entered into the system. The system should check the availability of the seats in the Bus & it should book and generate the ticket id and store the details in the system.	
UI Interface Screen	**Check Ticket** Customer ID - Customer Id should be automatically generated From Location - The from Location need to be Inserted To Location - The To Location need to be inserted

	Journey Date - The Journey Date should be filled by the user Number of Tickets - The User need to enter the no of tickets **Book Ticket** Bus Name - Represents the Bus name Seats Available - Represent the Available Seats Time - Represents the Bus Time Ticket Charge - The Ticket Amount will calculated and displayed _**Note: a) Once all the details are updated for the registration then only it will be processed for registration. The Customer can also flush/clear the values entered for re updating._ ***b) Generate the Ticket ID (e.g.* Ticket ID= Bus id + current date (2 digit) +current month (2 digit) +current year (last 2 digit) + unique number (4 digit)*).***
Trigger	Administrator triggers the functionality once he/she receives the details of the travel from the customer.
Pre-Conditions	The admin should have the details of the travel to be booked into the system.
Post Conditions	The admin should have the details of the travel to be booked into the system.
Success End Condition	On success a message should be displayed “Ticket Booked successfully”.
Failed End Condition	The system should throw an error message if one of the validations is wrong. The error message and error code needs to be identified during design. The validation and business rules are mentioned in the business rules section.
Steps & Actions	1. System should display all the necessary details. 2. Get all the necessary information like Customer ID, from location, to location, Date of Journey, Number of tickets. 3. The From Location and to location must be typed/select. 4. On the check availability it should check the customer id, check the availability and display all the available Bus based on the specified date and place. The Bus schedule details must be displayed on the same details. 5. Select the required Bus and go for the book Ticket. 6. Generate the unique Ticket id for each and every ticket and display the ticket information.

	7. Identify the Bus information for book ticket 8. Identify the Route Information 9. Identify the Schedule Information 10. Identify the Seat Availability 11. Generate the ticket id as follows. **Ticket ID= Bus id + current date (2 digit) +current month (2 digit) +current year (last 2 digit) + unique number (4 digit).** **Example: B1011005130011** 12. The ticket details into the system and update the number of available tickets in the selected schedule.
Business Rules & Validations	• The customer should be already registered in the system. • From and to should not be the same. • Date should not be less than current date. • All the fields are Mandatory. • Booking can be done only for the next two days. • A person can book maximum 4 tickets at a time. • At the time of Booking ○ Check the available Buses based on the Date, Type, No. of Seats and Place. ○ Get the preferred Bus and Generate the Ticket id for the individual tickets. ○ Store the Information of Ticket in System. ○ Update the number of seats available in the selected schedule. The number of seats allotted for the customer needs to be deducted from the total seats available in the system.

1.3.3 Cancel Ticket

Bus Ticket Reservation System **Req-2.1.3**	**Cancel Ticket**

Administrators to cancel the tickets for the customer into the system. The administrator should have the details from the customer to be entered into the system. The system should check for the ticket information for cancellation calculates the amount for the refund and store the details in the system	
UI Interface Screen	• Customer ID - Customer ID • Ticket ID - Ticket ID • Date of Journey - Enter the input to represents DD/MM/YYYY format *****Note:*** ***I)At the time of Cancellation, the system needs to calculate the cancellation details, store the cancellation details into the system and update the number of seats available.*** ***ii) Once all the details are updated for the registration then only it will be processed for (i.e. Check Ticket or Cancellation) registration. The Customer can also or Reset / clear the values entered for re updating.***
Trigger	Administrator triggers the functionality once he/she receives the details of the travel from the customer.
Pre-Conditions / Assumptions	The admin should have the details of the ticket to be cancelled into the system.
Post Conditions	The system should on successful cancellation the increment the number of available tickets; the cancelled ticket information should be recorded in database.
Success End Condition	On success a message should be displayed "**Ticket Cancelled successfully. Please pay the remaining amount Rs.__ (Balance amount) to the customer".**
Failed End Condition	The system should throw an error message if one of the validations is wrong. The error message and error code needs to be identified during design. The validation and business rules are mentioned in the business rules section.
Steps & Actions	1. System should display all the necessary details. 2. Get Customer ID, Ticket ID to be cancelled. 3. Get the Date of Journey also as a security measure to avoid unintended tickets getting cancelled. 4. On select of ***cancel ticket*** the system must calculate the

amount to be refunded to the customer based on the number of hours before the actual journey time available when the ticket is getting cancelled.

5. On the same screen/window a details should be provided to see the cancellation policy.
6. On successful cancellation it should redirects to success message. The following message should be displayed.

   Ticket Cancelled Successfully

   Please pay the remaining amount Rs.__ (Balance amount) to the customer.
7. Calculate the number of hours between the time of cancellation and scheduled journey time as Number of Hours (H).
8. Identify the Cancellation Rate as follows.

Number of Hours(H)	Cancellation Rate (%)
H=>48	10
48>H>=24	25
24>H>=12	50
H<12	100

   If a customer belonging to VIP customer type is cancelling the ticket before 48 no amount will be deducted.
9. Calculate the Deduction Amount

   Deduction amount per Ticket (D) = Cost per Ticket * (Cancellation Rate/100)
10. Calculate the refund amount which is equal to the **total deduction amount (TD)** deducted from the total amount paid by the customer.

    Refund Amount = Amount Paid by the Customer – deduction amount
11. Store the cancellation details into the system

    Update the number of seats available in the schedule in which the customer cancelled his tickets.

Business Rules & Validations

- Customer id should be existing one.
- Ticket id should be existing one.
- The customer should have booked the ticket with

	the entered ticket id for travelling on the entered journey date. • All the fields are Mandatory. • At the time of Cancellation, the system needs to calculate the cancellation details, store the cancellation details into the system and update the number of seats available.

1.3.4 Update Profile

Bus Ticket Reservation System **Req-2.1.4**	**Update Profile**
Administrators to view history and update the customer information like name, address, contact no etc. The admin can also remove the customer information from the system. The administrator should have the details from the customer to be updated into the system.	
UI Interface Screen	**<u>Get Customer Details</u>** Customer ID - Customer Id Get Details - The user should enter the details **<u>Update Profile</u>** Name - The Customer name should be filled by the user Address - The Address should be filled by the user City - The city should be filled by the user State - The Citizenship should be filled by the user Country - The Country Type should be filled by the user Pincode - The pin code should be filled by user and it should contain 6 digits only. Email Address - The Email Address should be filled by the user Date of Birth - Enter the input to represents the customer DOB in DD/MM/YYYY format

<table>
<tr><td></td><td>Gender - The Gender should be filled by the user

<b><i>**Note:</i></b>

<b><i>i) Once all the details are updated for the registration then only it will be processed for Updation.</i></b>

<b><i>ii) At the time of registration, the following task needs to be done, calculate the weightage based on customer type.</i></b>
<table>
<tr><th>Customer Type</th><th>Priority</th><th>Weightage</th></tr>
<tr><td>Normal</td><td>Normal</td><td>5</td></tr>
<tr><td>VIP</td><td>High</td><td>10</td></tr>
</table>
</td></tr>
<tr><td><b>Trigger</b></td><td>Administrator triggers the functionality once he/she receives the details from the customer.</td></tr>
<tr><td><b>Pre-Conditions / Assumptions</b></td><td>The admin should have the details of the customer to be updated into the system.</td></tr>
<tr><td><b>Post Conditions</b></td><td>The system should view history, update the user information and remove the customer details from the system</td></tr>
<tr><td><b>Success End Condition</b></td><td>On success a message should be displayed <b>“Information updated successfully”.</b></td></tr>
<tr><td><b>Failed End Condition</b></td><td>The system should throw an error message if one of the validations is wrong. The error message and error code needs to be identified during design. The validation and business rules are mentioned in the business rules section.</td></tr>
<tr><td><b>Steps &amp; Actions</b></td><td>1. Initially the system should accept the details for accepting the customer id and which fetches the customer information from the d/b system.
2. Customer details needs to be shown with a set of editable fields for entering the modifiable customer data. The other details must be shown as non-editable fields.
3. Customer Name, address, city, country, state, pin code, email id, contact number and customer type can be modified. The other fields like Date of Birth and gender needs to be displayed as non-editable fields.
4. The country details needs to be populated from the database.
5. The state details also must be populated from the database based on the country selected.</td></tr>
</table>

6. Identify the priority based on the customer type details as follows:

Customer Type	Priority
Normal	Normal
VIP	High

7. Identify the Weightage of the customer based on the priority as follows:

Priority	Weightage
Normal	5
High	10

Business Rules & Validations

Get Customer Details

- Customer id should be existing one.

Update details

- Customer id should be existing
- Email id should be in a valid format(abc@xyz.com)
- Email id should not be already existing in the system
- Contact number should be a 10 digit number.
- Address should not contain special characters other than white space.
- The name should contain only alphabets and space.
- Pin code should be 6 digits.
- At the time of registration, the following task needs to be done:
    - Calculate the weightage based on customer type.
    - Update the details for the entered customer number.

1.3.5 Manage Master Data

Bus Ticket Reservation System **Req-2.1.5**	**Manage Master Data**
Administrators to manage master data like Bus information, route information, bus schedules etc.	

The administrator should have the data to be updated into the system.	
UI Interface Screen	**Main Details** • Add new Bus - Add new Bus • Add new route -Add new Bus • Schedule Bus -Schedule Bus **Add new bus** • Bus name -To Add Bus name • Bus Type - To Add Bus Type • No of seats - No of Seats **Add new route** • From - From • To -To • Cost - cost **Schedule bus** • From - From • Bus Name - Bus Name • Time - Time • schedule -schedule
Trigger	Administrator triggers the functionality once he/she has the details
Pre-Conditions / Assumptions	The admin should have the data to be updated into the system.
Post Conditions	The system should update the bus information, route information and schedules from the system
Success End Condition	On success a message should be displayed **"Information updated successfully".**
Failed End Condition	The system should throw an error message if one of the validations is wrong. The error message and error code needs to be identified during design. The validation and business rules are mentioned in the business rules section.

Steps & Actions

1. System should display all the necessary information for to Add new Bus, Add new Route and Schedule bus.
2. In the Add new bus , necessary details like Bus name, Bus type, No of Seats should be displayed.
3. On adding the information , the system should generate a unique bus ID and check if the no of seats provided matches with maximum number of seats specified for that bus type and store the data into the database.
4. In the Add new route , necessary details like from, to, Cost should be displayed.
5. On adding the information , the system should generate a unique route ID and Store the information in the database.
6. In schedule bus , necessary details for from, to, Bus Name, Time should be provided.
7. On entering the information, the system should find the route id and bus id and store in the database along with time.
8. The system should also ensure that no bus is scheduled on the same time.
9. Identify the Bus information as follows.

Bus ID	Bus Name	Type	Capacity
B101	KPN Travels	A/C Semi Sleeper	30
B102	ABT Travels	A/C Semi Sleeper	30
B103	PNK SRT Travels	A/C Sleeper	22
B104	ABT Travels	Non A/C seater	40
B105	KPN Travels	Non A/C seater	40
B106	PNK SRT Travels	Non A/C seater	40

10. Identify the Maximum seats by type information for buses as follows

Type	Maximum Seats

A/C Semi Sleeper	35
A/C Sleeper	25
Non A/C seater	45

11. Identify the Route Information as follows:

Route ID	From	To	Cost
R1	Coimbatore	Chennai	600
R2	Coimbatore	Salem	300
R3	Coimbatore	Bangalore	500
R4	Coimbatore	Cochin	350
R5	Chennai	Coimbatore	600
R6	Salem	Coimbatore	300
R7	Bangalore	Coimbatore	500

12. Identify the Schedule Information as follows:

SID	Route ID	Bus ID	Time
S1	R1	B101	10.00 P.M
S2	R2	B102	9.00 P.M
S3	R3	B103	8.45 P.M
S4	R4	B104	7.30 P.M

13. Identify the Seat Availability as follows:

SID	Date of Journey	Available Seats
S1	10/5/2012	30
S2	10/5/2012	30
S1	11/5/2012	30
S3	12/5/2012	22
S4	10/5/2012	40

Business Rules & Validations

a) In add new bus

- Bus type should be existing one.
- Unique bus ID should be automatically generated.
- No of seats provided must be less than or equal to the no of seats specified for that bus type.
- All the details are Mandatory

b) In add new route

	• Route should not be the existing one. • Unique route Id should be generated automatically. • All the details are Mandatory **c) In Schedule** • The route should be an existing one. • Bus should be existing one. • There should not be any bus scheduled in same time. • All details are mandatory.

1.4 Nonfunctional Requirements

1.4.1 UI Requirements: Inn

1. The front end should be user-friendly and pleasant
2. Any error message or exception displayed to the user should be user-readable (and not technical)
3. All entered values should be validated
4. The UI screens are designed with any theme color along with the logo of the company.
5. Exceptions should not be shown on console.
6. No debug message should come on console.
7. All the errors and exceptions should be logged to log file.
8. Log file path should be configurable.
9. Database connections should be configurable.
10. It should not be hard coded.
11. Requirements involving UI features should be taken care in console accordingly

1.4.2 Performance Requirements: Pnn

1. All front-end pages should be served up in less than 5 seconds post click when up to 1000 users are on the application concurrently.
2. Since it’s a financial operations application, transactions should be safe and fast.

1.4.3 Installation, Deployment & Operational Requirements: Onn

1. Application Java and J2EE deployment with My SQL database or SQL Server database for persistence

   Or

   Application .NET Framework deployment with SQL Server database for persistence
2. This system must be up 24X7

1.4.4 Maintainability & Portability Requirements: MPnn

1. If the allocated memory space for the existing application needs to be increased then it should be possible without any impact to performance

1.4.5 Security Requirements Snn:

1. No unauthorized users should be able to log on to the system
2. User Id/Login Id should have combination of letter and numeric values, none of the special characters are allowed (#,@,$,%,&).
3. Validate the user id/login id and password, accordingly to the user it will redirect to the appropriate page.

2.0 Requirements / Use Case Summary Table

Business Requirement ID	Software Requirements ID	Short Description	Requirement Provider (Originator)	Priority	Complexity	Requirement type	Remarks
BR001	SR001	Customer Registration	Client	High	Medium	UI screen & backend processing	-

BR002	SR002	Book Ticket	Client	Medium	Medium	UI screen & backend processing	-
BR003	SR003	Cancel Ticket	Client	High	Medium	UI screen & backend processing	-
BR004	SR004	View history and update profile	Client	Medium	Medium	UI screen & backend processing	-
BR005	SR005	Manage Master Data	Client	High	High	UI screen & backend processing	-