

---

# SOFTWARE REQUIREMENTS SPECIFICATION

for

## BUS PASS SYSTEM

Version 1.0

Prepared by :

1. Harsh Verma(200002038)
2. Siddartha Chennareddy(200002070)
3. Ujjwal Hendwe(200001028)
4. Pradeep Reddy(200001005)
5. Sakshi Mahajan(200001041)

April 14, 2022

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Purpose . . . . .	3
1.2	Intended Audience and Reading Suggestions . . . . .	3
1.3	Project Scope . . . . .	3
1.4	References . . . . .	3
1.5	Overview . . . . .	3
<b>2</b>	<b>Overall Description</b>	<b>5</b>
2.1	Product Perspective . . . . .	5
2.2	User Class and Characteristics . . . . .	6
2.3	Product Features . . . . .	6
2.4	Operating Environment . . . . .	8
2.5	Design and Implementation Constraints . . . . .	8
2.6	Assumptions and Dependencies . . . . .	8
<b>3</b>	<b>System Features</b>	<b>9</b>
3.1	Description and Priority . . . . .	9
3.2	Stimulus/Response Sequences . . . . .	9
3.3	Functional Requirements . . . . .	9
<b>4</b>	<b>External features</b>	<b>12</b>
4.1	User Interface . . . . .	12
4.2	Hardware Interfaces . . . . .	12
4.3	Software Interface . . . . .	12
4.4	Communication Interface . . . . .	12
<b>5</b>	<b>Non-Functional features</b>	<b>13</b>
5.1	Performance Requirements . . . . .	13
5.2	Safety Requirements . . . . .	14
5.3	Security Requirements . . . . .	14
5.4	Software Quality Attributes . . . . .	14

# 1 Introduction

## 1.1 Purpose

The goal of this document is to create an online bus pass system that allows people to purchase city bus passes and intercity bus tickets. Passengers can order their tickets online instead of going to the ticket counter, saving time and effort.

## 1.2 Intended Audience and Reading Suggestions

This project is a prototype for a bus pass system available to the entire audience. This project is beneficial to both the bus management staff and the passengers.

## 1.3 Project Scope

The objective of this system is to make bus administration easier and to offer a user-friendly application for passengers looking to purchase bus tickets and passes. With its bus management and reservation functionalities, the system is built on a relational database. There will be a database server that will handle many major cities in India as well as numerous Bus Service Providers that will provide passengers with various types of buses. Above all, the goal is to provide a pleasant customer experience at a reasonable cost.

## 1.4 References

- <https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database>
- <https://www.geeksforgeeks.org/non-functional-requirements-in-software-engineering/>
- Database System Concepts -6th edition, Abraham Silberschatz, Henry F.Korth, S. Sudarshan
- R. S. Pressman, Software Engineering: A Practitioner's Approach

## 1.5 Overview

The other sections of this paper provide a basic overview, including information about the project's users, the product's hardware, and the product's functional and data needs.

The project's overall description, as well as the assumptions established in designing the Bus Pass System, are detailed in section 2 of this article. The functional requirements are presented in Section 3. It also provides a user's perspective on the product. Section 4 examines the need for external interfaces, including software, hardware, and other factors. Section 5 goes through the non-functional characteristics in detail, including the performance requirements.

## 2 Overall Description

### 2.1 Product Perspective

The Bus Pass System Database System stores the following information:

- **User description:**  
It includes customer ID, first name, last name, gender, date of birth, address, password and phone number. This information can be used to maintain track of a customer's records in case of an emergency or for any other reason.
- **Bus and Bus Service Provider details:**  
This contains information about the bus, such as the bus number, starting point, and terminal point, as well as the stops in between, capacity, bus facilities and features, bus rating, bus driver information, bus fare, and Bus Service Provider information, such as name, address, phone number.
- **Seat Availability:**  
This covers the number of seats reserved and seats available on a certain bus on a given day.
- **Passenger details:**  
It contains information such as the passenger's name, age, gender, and the account from which the ticket was purchased. This information can be used to maintain track of a customer's records in case of an emergency or for any other reason.
- **Reservation (Ticket) description:**  
Customer information, seat number, bus number, date of booking, kind of ticket, date of travel, starting point, finishing point, transaction information, bus information, facilities supplied, and other route information are all included.
- **Driver details:**  
It comprises the driver's information, including the bus he or she is driving, license information, and contact information.
- **Route and Price Details:**  
It includes the route id of all the routes and the price for travelling in that route.
- **Bus Pass Details:**  
It offers information about the application as well as its current state. This table also contains formal documentation, as well as a soft copy of the bus pass if it has been issued.

## 2.2 User Class and Characteristics

Users of the system should be able to retrieve bus information between two given cities with the given date of journey from the database. The system will support two types of user: Customer, and Admin. Customers will have access to customer functions, and the Admin will have access to both customer and bus management functions. The customer should be able to do the following functions:

- Book/cancel bus tickets from one city to another
- Buy monthly city bus passes
- View their itinerary tickets and can download them
- Edit their profile information

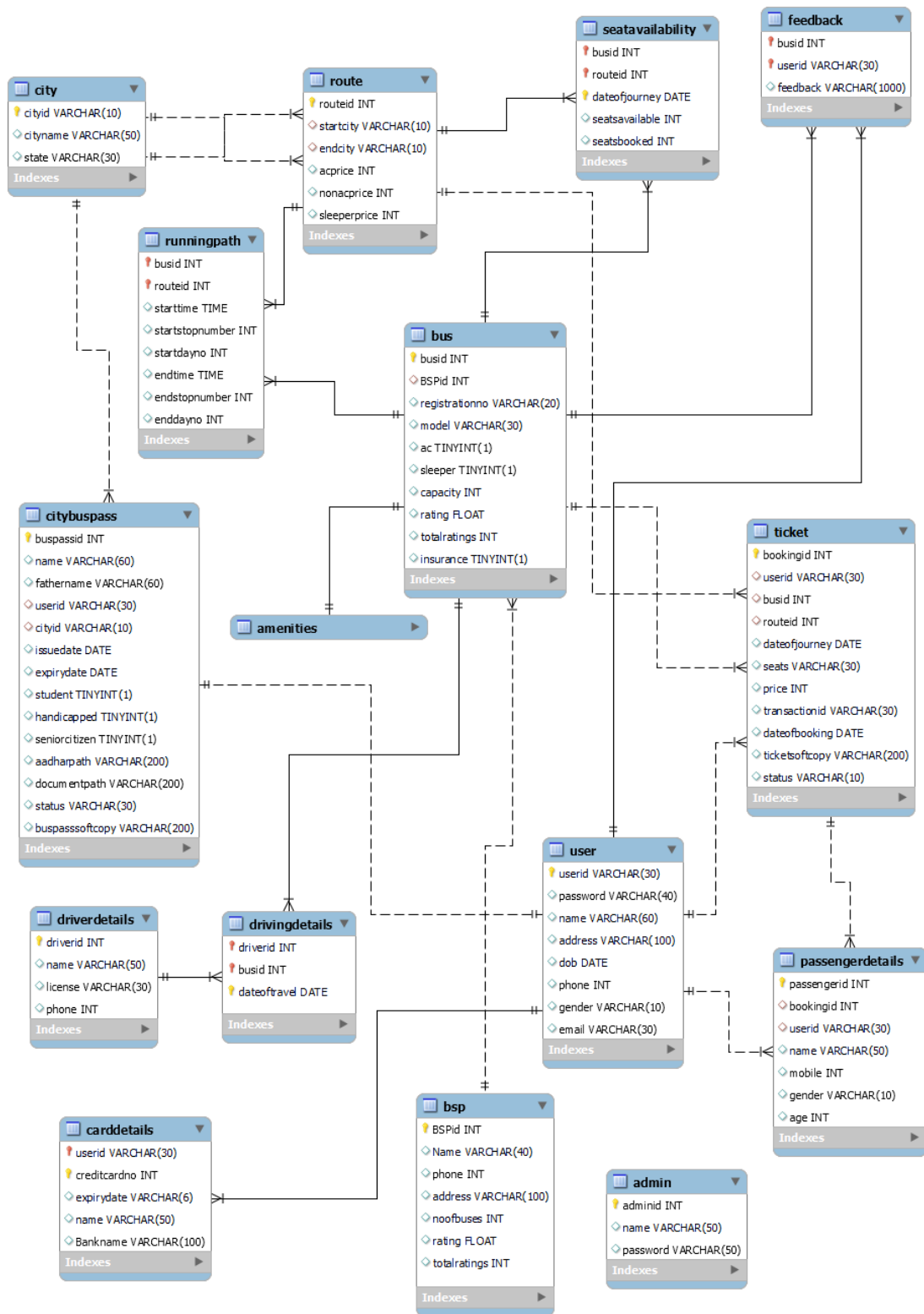
The Admin should have following management functionalities:

- Add new buses
- Update bus terminals, ticket rates, and other bus information
- Govern whether a bus is running on a particular day or not and release tickets accordingly
- Verify the documents uploaded for the bus pass
- Issue the bus pass
- Revoke customer's bus pass if any error is discovered during the verification process

Each bus has a limited number of available seats. There are a number of buses which depart from or arrive at different cities on different dates and times.

## 2.3 Product Features

The major features of bus database system are as shown below in Entity Relation Model (ER model) -



## 2.4 Operating Environment

Operating environment for the bus management system is as listed below:-

- Web Browser: Any web browser like Chrome, Explorer
- Operating system: Windows, MacOS
- Database server: MySQL Database
- Development Tools: Microsoft Visual Studio
- Backend Framework: Django
- Frontend: HTML, CSS, Bootstrap, JavaScript
- Deployment: Heroku, Pythonanywhere

## 2.5 Design and Implementation Constraints

- The information of all users, buses, admins, must be stored in a database that is accessible by the website.
- MySQL server will be used as SQL engine and database.
- Admins and users must use their correct login credentials to access the system.
- The bus pass system must be accessible to the users 24\*7.
- Users may access from any electronic device which has internet browsing capabilities and an internet connection.
- MySQL commands for the applications to retrieve/append data.

## 2.6 Assumptions and Dependencies

- Assuming that the buses run on time, a static current running status model is developed.
- Once the tickets are released for a bus on a particular day, the bus will undoubtedly run on that day.
- Once a seat has been reserved for a segment of the bus route, it will not be available to other passengers during the entire journey.



## 3 System Features

### 3.1 Description and Priority

The bus pass system keeps track of buses, seats, prices, and reservations. It also keeps track of bus passes for trips inside a certain city. Of course, this initiative is a high priority because travelling across cities in buses without previous bookings is quite difficult. It simplifies the procedure of purchasing city bus tickets while also lowering the financial load.

### 3.2 Stimulus/Response Sequences

- Register/Login to the website
- Modify user's profile
- Recharge one's wallet using card or UPI if not recharged
- Apply for a city bus pass and submit the documents
- Search for buses between two cities
- Displays a detailed list of available buses
- Book a ticket on a particular bus.
- Cancel an existing reservation
- Rate your travel and provide any useful feedback

### 3.3 Functional Requirements

**On the user portal:**

- Registration and Login:
  - If the user is not yet registered, he or she must first provide their personal information and register.
  - After registering, the user can use their login credentials to access the website.
- Applying for Bus Pass:

- The user must complete an application form that requests personal information as well as a series of formal papers for verification of the information provided.
- Once the application is validated, payment must be done on the website.
- After the transaction, user will be able to access the soft copy of their bus pass.
- Lost/Misprint Bus Passes:
  - User can register any complaint or query in the website regarding lost/misprinted bus passes.
  - His or her query will be investigated by the administrator, and a new bus pass will be provided, cancelling the previous one.
- Recharging the Wallet:
  - After logging in, user must recharge their wallet using card or UPI for making any reservation/booking on the website.
- Selecting the bus and seats depending on their interest:
  - User needs to enter their start point and destination along with the date of travel for searching buses.
  - On searching, they get a set of buses along with the amenities they provide and the cost of travel.
  - User can also apply for Flexi-Ticket depending on the buses.
  - User needs to select one of the bus and select the seats they want to travel in.
- Transaction and Ticket Generation:
  - They will enter the information of those many passengers based on the quantity of seats selected. They will then proceed to the payment stage.
  - The user needs to pay from their wallet.
  - After successful payment, the ticket will be generated and will be sent to the user's registered mobile and mail.
- Cancellation of ticket:
  - Tickets can be cancelled up to 24 hours before departure using user's booking ID.
  - The cancellation fees will stay the same, at 20% of the ticket price.
  - The money will be returned to the individual's wallet.
- Accessing previous records of tickets:

- The user will be able to access their previous tickets in a folder in the website itself.
- He/she will be able to make any set of changes in the folder like deleting any previous ticket.
- Bus Tracking:
  - Assuming that the buses run on time, a static kind of model is built.
  - The user can access this for knowing the current running status of the bus.
- Rating and Feedback:
  - After the travel, the user can rate (in stars) their travel and can also provide their valuable feedback on the website.

### **On the Admin portal:**

- Login:
  - The admin needs to login using the login credentials. Only then, he/she can make any changes in the database and the website.
- Release tickets for the buses:
  - The admin needs to confirm whether a bus is running on a particular day or not.
  - The administrator must then release the tickets for that day's bus.
- Appoint new Admins:
  - The Admin has the right to appoint new admins by creating a new set of login credentials.
- Add new buses or remove existing ones:
  - Admin can add new buses to the routes.
  - He/she can remove any existing buses from the database as well.
- Validate the bus passes:
  - Verify the documents uploaded by the applicant.
  - Issue the soft copy of the verified bus pass.
- Lost/Misprint bus passes:
  - The administrator must validate the user's complaint and then issue a replacement bus pass as necessary.

## 4 External features

There are many types of interfaces as such supported by the bus pass system namely; User Interface, Software Interface, Hardware Interface and Communication Interface.

### 4.1 User Interface

The user interface for the software shall be compatible to any browser such as Internet Explorer, Mozilla or Google Chrome by which user can access to the system.

- Front end - HTML, CSS, JavaScript, Bootstrap
- Back end - Django Framework
- Visual Studio Code

### 4.2 Hardware Interfaces

- An electronic device that can connect to the internet. Because the software must run over the internet, all of the hardware required to connect to the internet will serve as the system's hardware interface.

### 4.3 Software Interface

- Operating System - Windows and MacOS are chosen for their support and user-friendly nature.
- Database - Using MySQL Database for saving records.

### 4.4 Communication Interface

This project is compatible with all web browsers. For reservation forms, ticket purchasing, and so on, we use basic electronic forms.

## 5 Non-Functional features

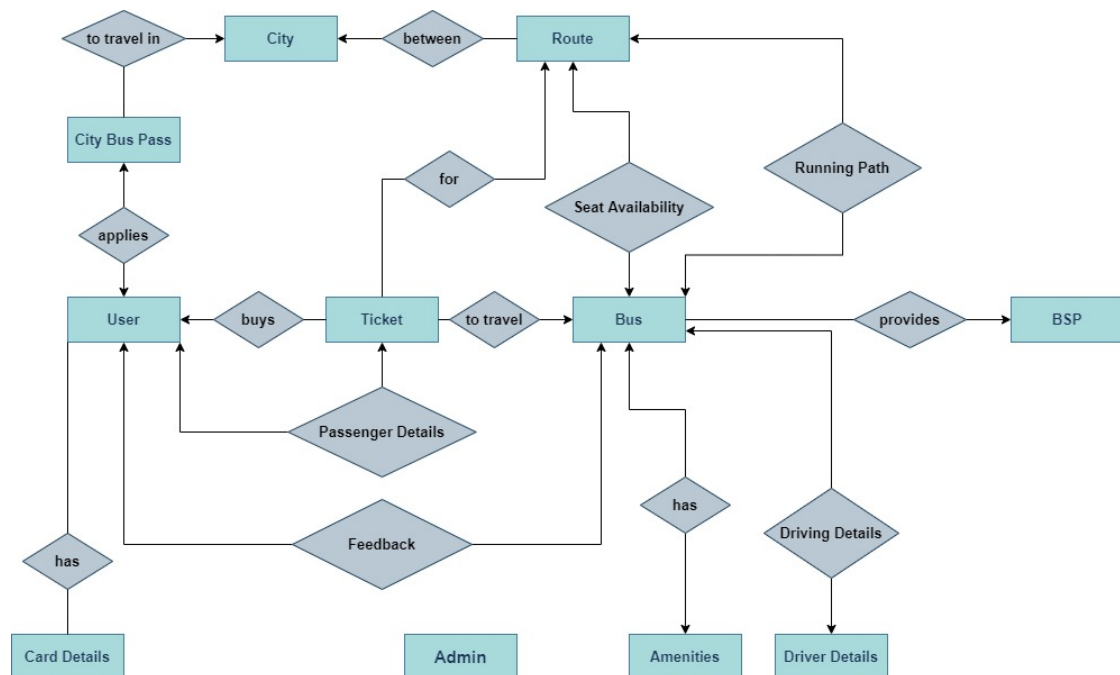
### 5.1 Performance Requirements

The steps involved to perform the implementation of the bus-pass-system database are as listed below.

- ER Diagram-

The Entity Relationship Diagram is a method for visually expressing the logical structure of a database. The results of this analysis are then utilised to organise data as a relation, normalise the relation, and create a relation database.

- Entities: In an application, these define different real-world items.
- Properties/Attributes: Which define an entity's attributes and relationships.
- Relationships: Which connect entities and represent meaningful dependencies between them.



- Normalization-

The primary goal of normalisation is to decrease redundancy, which means that data should only be kept once. Several times storing information wastes storage space and increases the total size of the data stored.

Modification anomalies can occur if a database is not correctly designed. When data is added to, altered, or deleted from a database table, modification anomalies occur. Data redundancy can also be a problem with traditional databases and poorly built relational databases. A database can be normalised to eliminate these issues.

The process of breaking down a table into smaller tables is known as normalisation. As a result, each table focuses on a single theme. There are three types of anomaly modifications, and the first, second, and third normal forms (3NF) are deemed sufficient for the majority of practical needs. It should only be considered after a comprehensive examination and comprehension of its implications.

## 5.2 Safety Requirements

If a catastrophic failure, such as a disc crash, causes extensive damage to a large portion of the database, the recovery method restores a previous copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing committed transaction operations from the backed up log, up to the time of failure.

## 5.3 Security Requirements

Security systems, like many other applications, require database storage. All sensitive user information, such as passwords and credit card numbers, will be encrypted and saved in the database. The website will be hosted by a reputable 3rd-party hosting platform, eliminating the possibility of server hacking. No personal information of the customers is shared out of our databases. Information such as information about users' booked tickets will be kept private. Any database modifications, such as insert, delete, update, and so on, can be synchronised rapidly and only the authenticated administrators can execute them.

## 5.4 Software Quality Attributes

- Scalability -

Django has the capacity to scale the website swiftly and flexibly to meet even the highest traffic demands. As the number of users or the demand on the server grows, performance will not be affected. The project's database will be extremely efficient, allowing for quick retrieval of data anytime it is needed.

- Usability-  
The software is user friendly, simple to use and effective. Without any external assistance, anyone may simply comprehend how the website works and learn to utilise it. Manuals will also be provided to the user so that he/she may learn how to utilise the software.
- Performance -  
Since, the software is based on web and has to be run from a web server, it shall take initial load time depending on internet connection strength which also depends on the media from which the software is run. Usually, the time it takes for user interface screens to load will be no more than two to three seconds. The performance will also depend upon hardware components of the user. Within 5 seconds, the log in information will be validated, and the queries will be answered.
- Serviceability -  
The software will be very easy to maintain and repair. Repairs can be completed in a short amount of time.
- Maintainability -  
The website can be easily maintained without changing much code because of Django's ability to create reusable components. Furthermore, because the website is totally dynamic, any data entered by the administrator will be automatically included into the website without the need for any additional adjustments from the developer. The site is examined and maintained on a regular basis, with the goal of resolving any issues that consumers may have throughout the booking process.
- Flexibility -  
The bus management system can be flexibly used for inter-intra state/city bus systems and also possibly used in International bus management systems. Also this system can be further moulded to be used for other reservation systems such as flight booking systems.
- Adaptability -  
The website is flexible enough to adapt to modern technologies in order to meet the changing needs of the users with minimal effort.
- Data Integrity –  
Because the administrator adds the data to the database, the information about the buses will be totally accurate. In the case of user information, it will be validated, and the documentation will be checked by the administrator. The data is thorough, accurate, and trustworthy.
- Reliability –  
The system will be available 24 hours a day, 7 days a week (Availability). For bus information and reservations, the website will be totally trustworthy. The information will be dynamic and updated on a regular basis, and the data of buses

available will be automatically erased from the user-side once the date and time has passed, making the site extremely dependable.