

CSN-351

DATABASE MANAGEMENT SYSTEM

Bus Booking Management System Project

Group Members:

- | | |
|-------------------|----------|
| 1. Balne Niteesha | 18114017 |
| 2. Diksha | 18114019 |
| 3. Karan Singh | 18114035 |
| 4. Kavya Barnwal | 18114039 |
| 5. Khushi | 18114040 |

Abstract

Traveling is a large growing business across all countries. Bus reservation system deals with maintenance of records of details of each passenger. It also includes maintenance of information like schedule and details of each bus.

We observed the working of the Bus reservation system and after going through it, we got to know that there are many operations, which they have to do manually. It takes a lot of time and

causes many errors while data entry. Due to this, sometimes a lot of problems occur and they are facing many disputes with customers. To solve the above problem, and further maintain records of passenger details, seat availability, bus availability and other things, we are offering this proposal of a computerized reservation system.

By using this software, we can reserve tickets from any part of the world, through telephone lines, via the internet. Customers can check availability of buses and reserve selective seats. The project provides and checks all sorts of constraints so that the user does give only useful data and thus validation is done in an effective way.

Introduction

The focus of the project is to computerize traveling companies to manage data, details of customers, details of various buses so as to ease companies' tasks and shifting the ticket booking process to an online platform for the easier access of customers. It reduces the possibility of errors or any discrepancy in any kind of data. It replaces all the paperwork.

This bus booking management system has three modules.:

- First module helps the customer to login or register into the system.
- Second module helps to inquire about the availability of seats in a particular bus at a particular date. It helps him to reserve a ticket or cancel a reserved ticket.
- There is a third module which allows the admin of the system to add seats to various buses and to view customers databases.

Features of the system

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features:

- ✓ Availability of seats can be enquired very easily.
- ✓ Passengers can also cancel their tickets easily.
- ✓ Minimum time needed for the various processing
- ✓ Better Service
- ✓ Ensures data accuracy.
- ✓ Records are efficiently maintained by DBMS.
- ✓ DBMS also provides security for the information.
- ✓ Any person across the world, with required setup can access this service.
- ✓ This would help the corporation prepare and organize its schedules more efficiently on the basis of traffic demand.

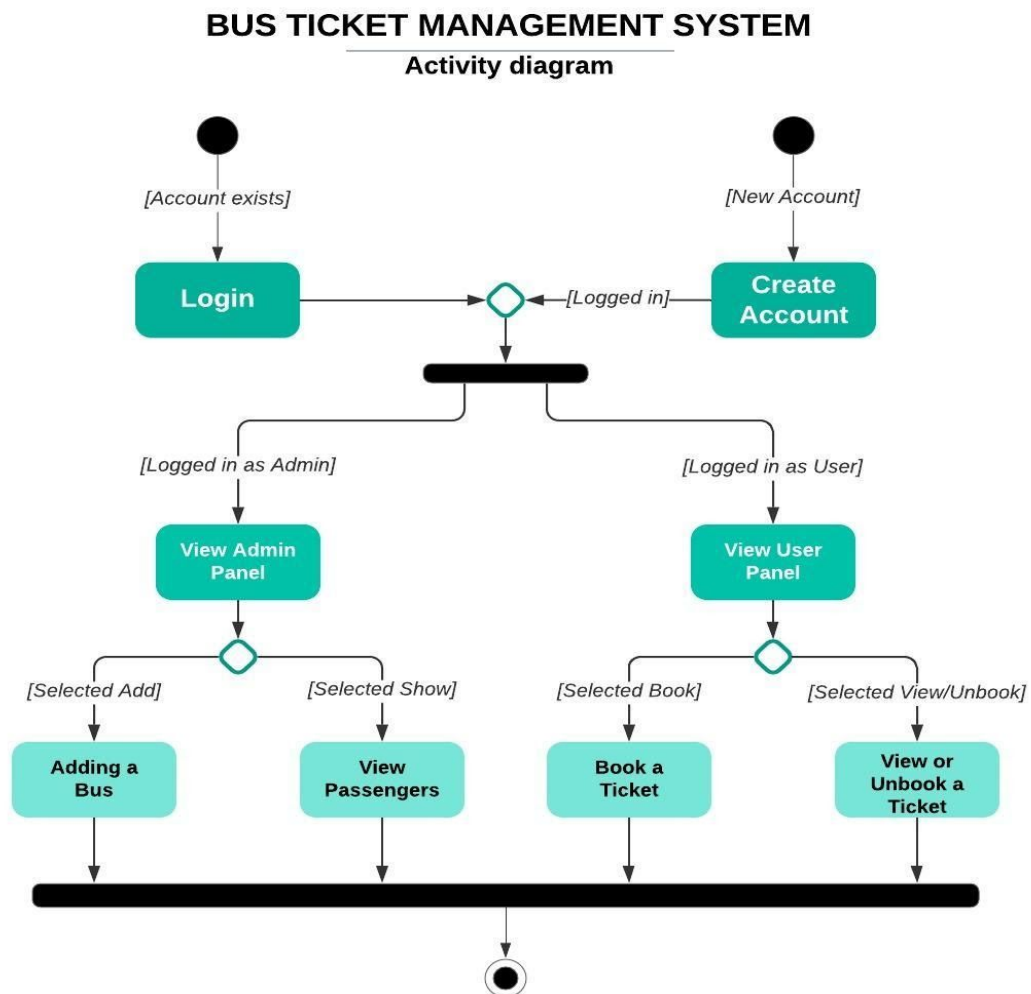
- **Hardware Requirements:**

- PC with Pentium IV processor. (optional)
- 512 MB RAM or above.
- 40 GB Hard Disk or above.
- Java Virtual Machine

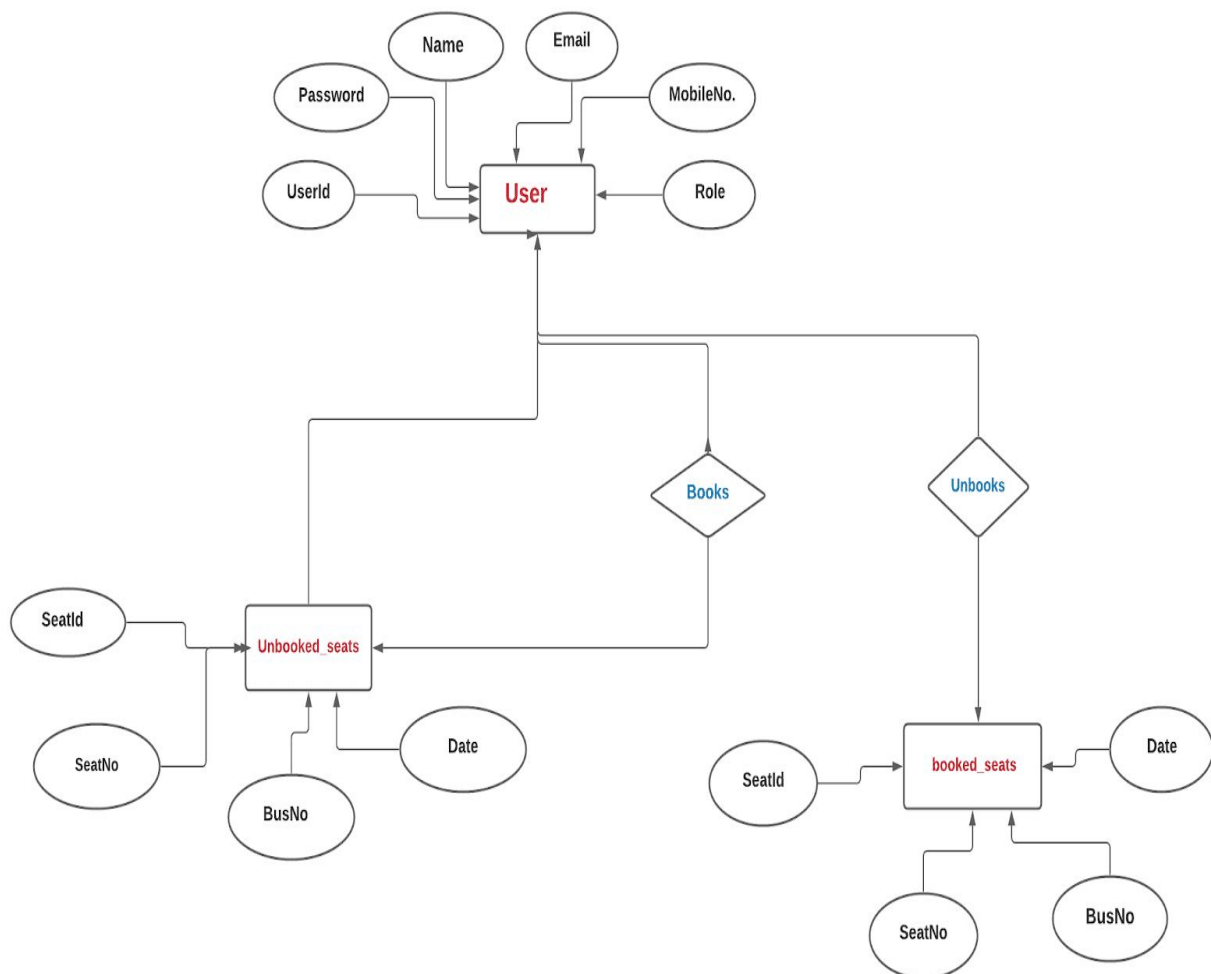
- **Software Requirements:**

- Operating system : Windows (optional)
- Front end : Java Runtime
- Integrated Development Environment(IDE) : Netbeans
- Server to host the database on machine : XAMPP
- JDBC driver : MySQL Connector

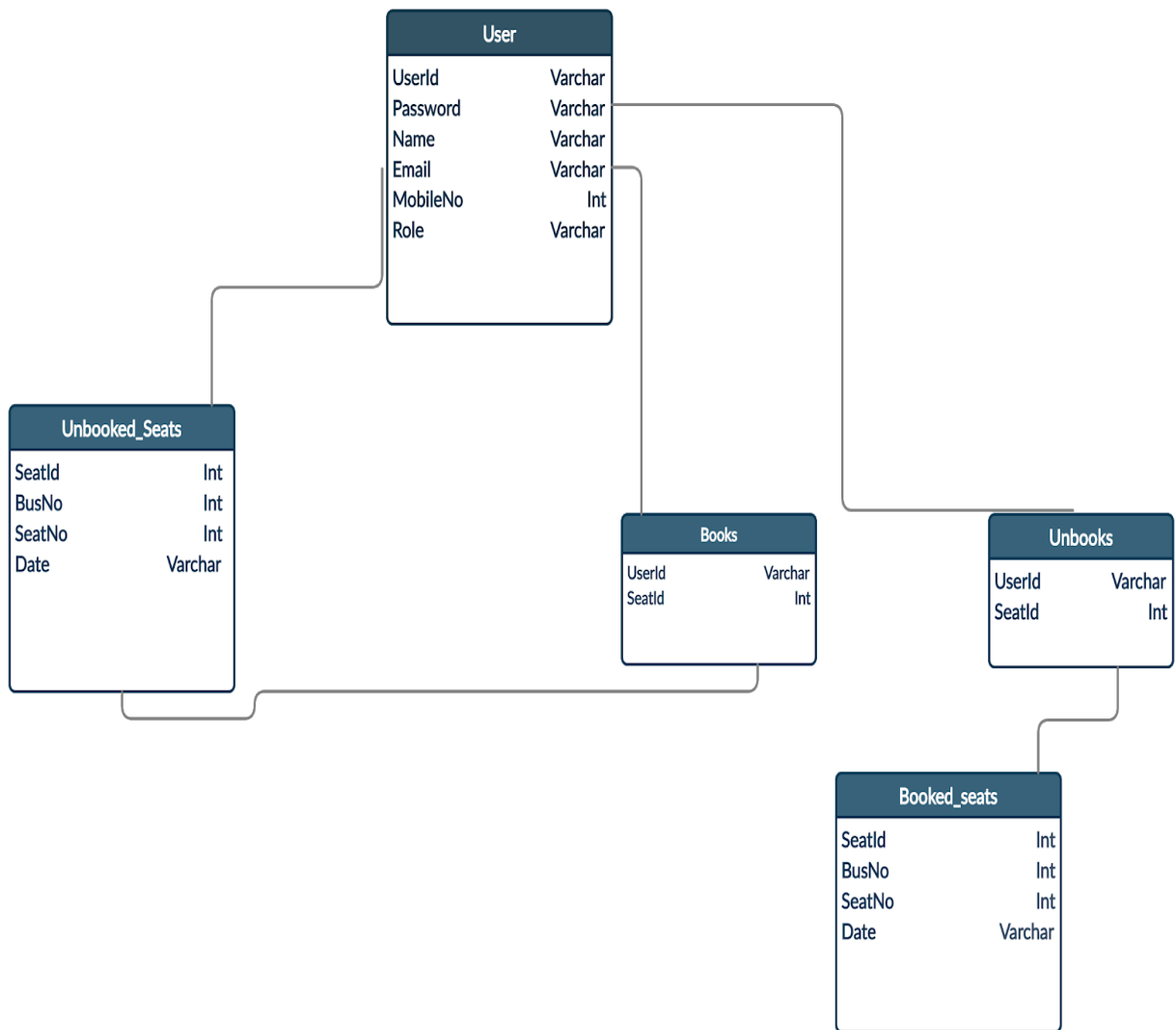
Activity Diagram:



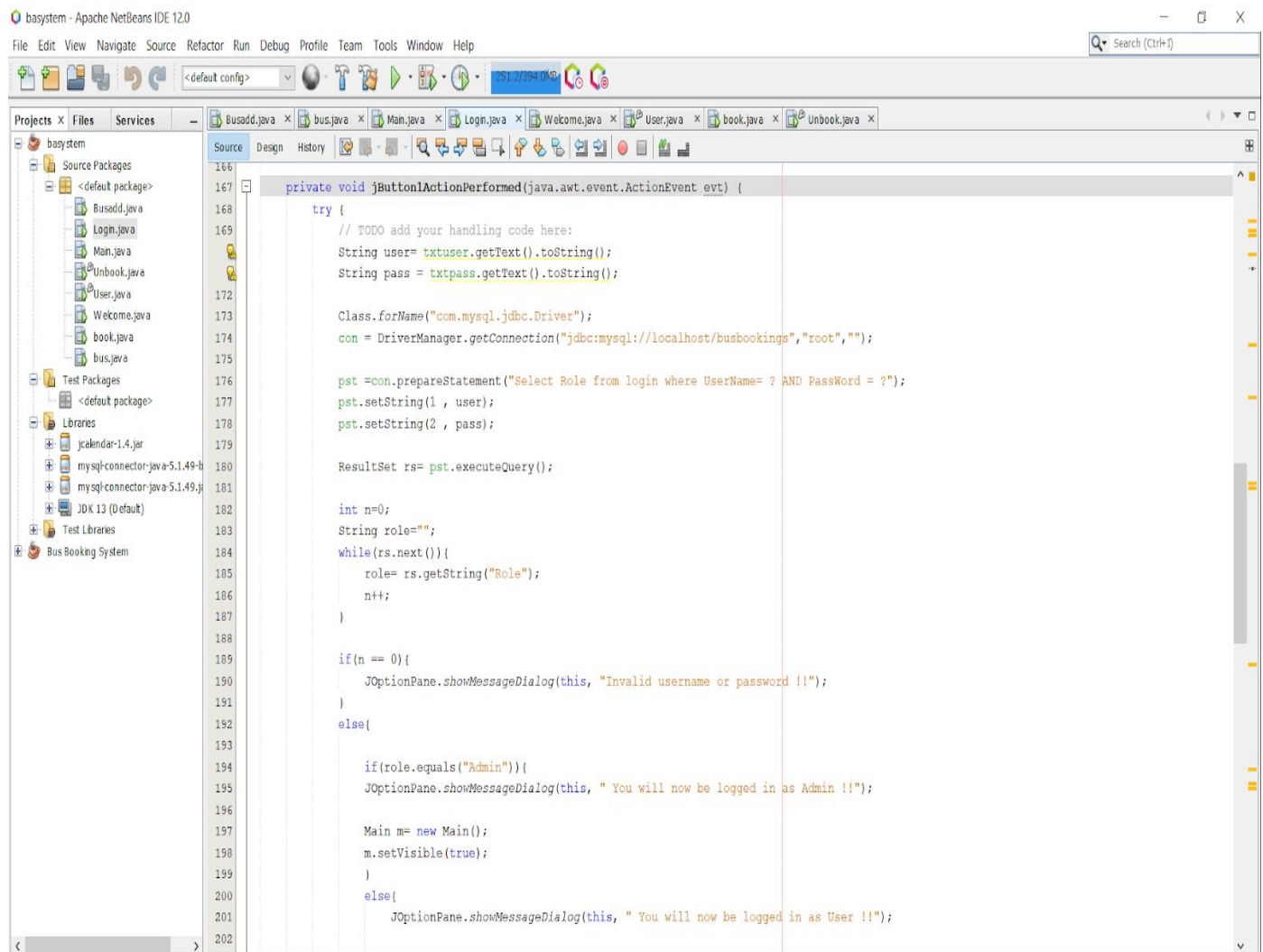
ER Diagram:



Relational Schema:



Code Snippets:



The screenshot displays the Apache NetBeans IDE 12.0 interface. The title bar reads "basytem - Apache NetBeans IDE 12.0". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. A search bar on the right says "Search (Ctrl+F)". The toolbar contains icons for file operations, running, and debugging. The "Projects" tab on the left shows a project named "basytem" with a "Source Packages" folder containing files like Busadd.java, Login.java, Main.java, Unbook.java, User.java, Welcome.java, book.java, and bus.java. Below it are "Test Packages" and "Libraries" including jakarta-1.4.jar, mysql-connector-java-5.1.49-b, and JDK 13 (Default). The "Source" tab is active, showing the code for "private void jButtonActionPerformed(java.awt.event.ActionEvent evt) {" in a file named "Login.java". The code includes comments, variable declarations, database connection setup, query execution, and conditional logic for user authentication.

```
166
167 private void jButtonActionPerformed(java.awt.event.ActionEvent evt) {
168     try {
169         // TODO add your handling code here:
170         String user= txtuser.getText().toString();
171         String pass = txtpass.getText().toString();
172
173         Class.forName("com.mysql.jdbc.Driver");
174         con = DriverManager.getConnection("jdbc:mysql://localhost/busbookings","root","");
175
176         pst =con.prepareStatement("Select Role from login where UserName= ? AND PassWord = ?");
177         pst.setString(1 , user);
178         pst.setString(2 , pass);
179
180         ResultSet rs= pst.executeQuery();
181
182         int n=0;
183         String role="";
184         while(rs.next()){
185             role= rs.getString("Role");
186             n++;
187         }
188
189         if(n == 0){
190             JOptionPane.showMessageDialog(this, "Invalid username or password !!");
191         }
192         else{
193
194             if(role.equals("Admin")){
195                 JOptionPane.showMessageDialog(this, " You will now be logged in as Admin !!");
196
197                 Main m= new Main();
198                 m.setVisible(true);
199             }
200             else{
201                 JOptionPane.showMessageDialog(this, " You will now be logged in as User !!");
202             }
203         }
204     } catch (Exception ex) {
205         JOptionPane.showMessageDialog(this, ex.getMessage());
206     }
207 }
```

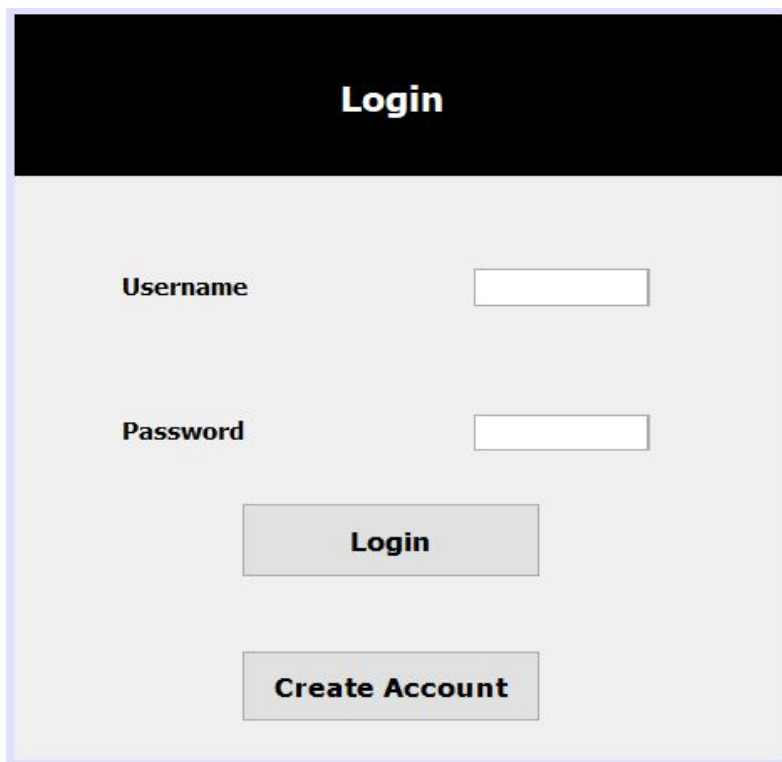
System:

Login and create account windows:

You can either **login** by giving your credentials or **create a new account** . If the username or password is wrong or such an entry does not exist in the database , a window pops up showing the message

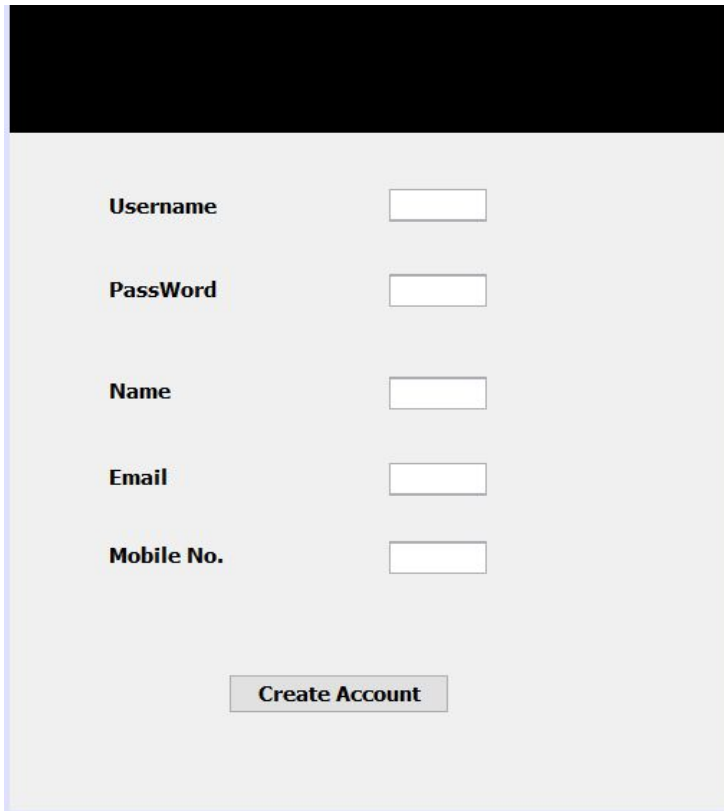
“Invalid username or password!” . Also , if a username is already taken while creating a new account , a window pops up with the message:

“Username already taken!” .



The image shows a login and account creation form. It has a black header with the word "Login" in white. Below the header, there are two input fields: "Username" and "Password". Below the "Password" field, there are two buttons: "Login" and "Create Account".

Login	
Username	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="Login"/>	
<input type="button" value="Create Account"/>	



Admin View:

An admin manages the whole system , so he can add buses available on a particular date and the status of their seats in the database , with the help of the “Add” button , after which the **Add Seat** window opens .

The admin can also view all the bookings of any bus on any particular day , with the help of the “Show” button , after which , the **Bus Records** window opens up.

Bus Booking System

Add

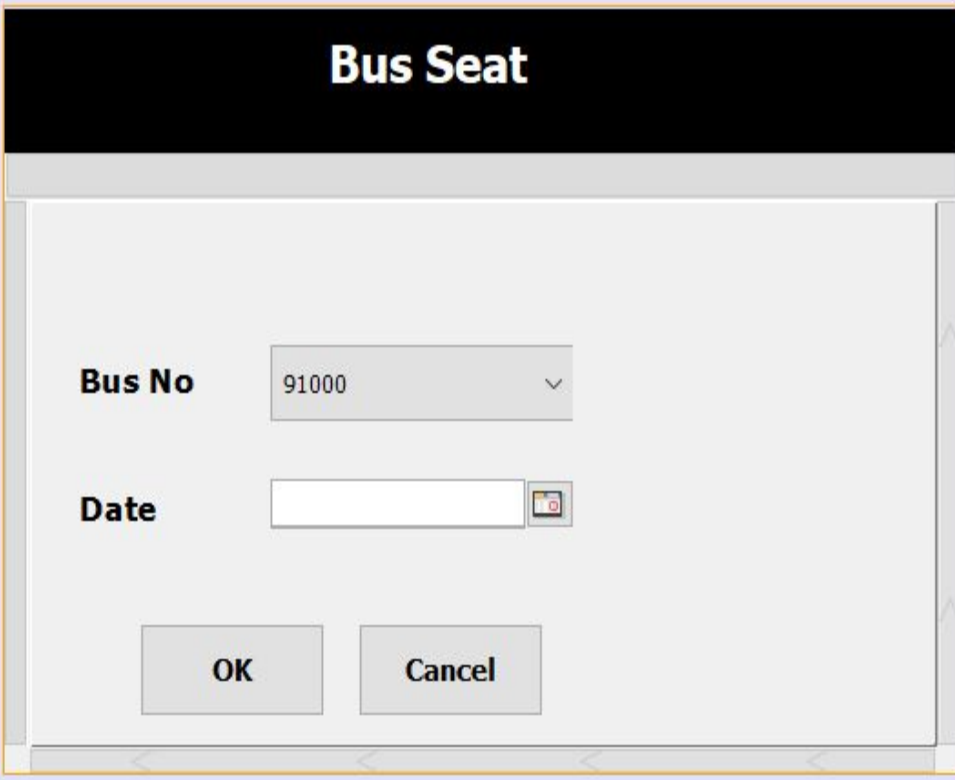
Show

Bus Records

Date

show

Busno	Seats	Status	CustomerName	Mobile	Date
-------	-------	--------	--------------	--------	------

A screenshot of a 'Bus Seat' selection window. The window has a black title bar with the text 'Bus Seat' in white. Below the title bar is a light gray header bar. The main content area is white and contains two input fields: 'Bus No' with a dropdown menu showing '91000' and a 'Date' field with a calendar icon. At the bottom are two buttons: 'OK' and 'Cancel'. The window has a standard Windows-style border with scrollbars on the right and bottom.

Bus Seat

Bus No 91000

Date

OK Cancel

User View:


A user can view all buses and seats available on the particular day , and also book them for himself or any relative (name and mobile No. required) , with the help of **Ticket Booking** window .

He/She can also unbook any of his/her booking for a particular date , booked under any name (which needs to be provided as input) , in the **View Your Bookings** window .

Ticket Booking

Name

Date



Show


Mobile

BusNo.	Seat	Date	Status
--------	------	------	--------

View Your Bookings

Name

Date



Show

Mobile No

Bus No.	Seat	Date
---------	------	------

Queries Used:

1) **Select Queries :**

a) SELECT Role

FROM login

WHERE UserName= ? AND PassWord = ?

b) SELECT seat.Busno, seat.Seats, seat.Status, busbooked.UserName ,
busbooked.MobileNo, seat.Date

FROM seat

LEFT JOIN busbooked ON seat.BusNo=busbooked.BusNo AND seat.Seats =
busbooked.Seat AND busbooked.Date = seat.Date

WHERE seat.Date = date

c) SELECT seat.Busno, seat.Seats,seat.Status,seat.Date

FROM seat

WHERE seat.status = status AND seat.Date = date

d) SELECT busbooked.BusNo , busbooked.Seat , busbooked.Date

FROM busbooked

WHERE Date= date AND UserName = uname AND MobileNo = mno

2) **Insert Queries :**

a) INSERT INTO seat(BusNo , Seats , Date, Status)

VALUES(bno , sno , date , status)

b) INSERT INTO login(UserName , PassWord , Name , Email , MobileNo ,
Role)

VALUES(uname , pass , name , email , mno , role)

3) **Update Queries:**

UPDATE seat

SET Status = status

WHERE BusNo = bno AND Seats = seats AND Date= date

4) **Delete Queries :**

DELETE FROM busbooked

WHERE BusNo = bno AND Seat = sno AND UserName= uname AND MobileNo =
mno AND Date= date

Conclusion:

This project includes a user-friendly interface for Bus ticket management, with optimised queries and normalized databases. They make the system more efficient to store and organize data than spreadsheets; also allow a centralized facility such that the data can easily be accessed, modified and quickly shared among multiple users.