# Railway Reservation System

A CS814 Course Project Report

Under the guidance of

Dr. Mahendra Pratap Singh

Submitted By

Sagar Khatri (202CS025)

Piyush Kumar Mishra (202CS018)

Department of Computer Science and Engineering
National Institute of Technology Karnataka
P.O. Srinivasnagar, Surathkal, Mangalore-575025
Karnataka, India
January 2021

# Contents

# 1. Introduction

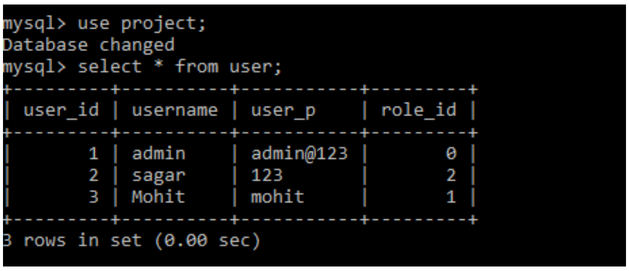
This Railway reservation system can perform the basic function like reservation and cancellation. The users are provided to register on the server for getting access to the database, and query result retrieval upon registration completion user. Each user has an account, which is essentially referred to as the ‘view level’ of customer account contents comprehensive information of the user entered during the user registration, and allow the user to access their past reservations, cancellations enquiry about trains and train schedule, seat availability and make afresh reservations. The user will also be able to update their account details, etc.

The master user of this system is the Railway Administrator who can login using a master password, and once a user is authenticated as an admin, he or she can access and modify information is stored in the database of the system.

Railway Reservation System is an Application which can be used for the reservation of tickets, their cancellation or for viewing the train available for one’s journey and their fares.
The Application is developed on java using the Software “NETBEANS” and as the application is database oriented so for that it uses mysql database in MySQL Server.

There are three categories in which the Users are divided - i. Admin
ii. Employee
iii. Passenger

An Admin can Change role of users to Employee or Passenger or to Admin. But no user has access to modify admin roles.



```
mysql> use project;
Database changed
mysql> select * from user;
+---------+----------+-----------+---------+
| user_id | username | user_p    | role_id |
+---------+----------+-----------+---------+
|       1 | admin    | admin@123 |       0 |
|       2 | sagar    | 123       |       2 |
|       3 | Mohit    | mohit     |       1 |
+---------+----------+-----------+---------+
3 rows in set (0.00 sec)
```

Here role_id 0 has Administrative roles.
role_id 1 has Passenger roles.
role_id 2 has employee roles.

## 1.1 Objective of Project on Railway Reservation System:

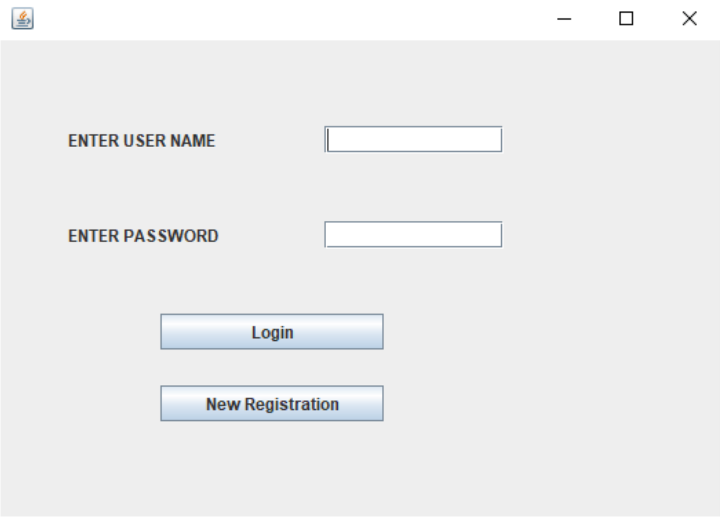
The main objective of the project on railway reservation system is to manage the detail of train, ticket, booking, passenger, train schedule. It manages all the information about train, train route, train schedule. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for the managing the train ticket train route booking. It tracks all the detail about the booking, passengers, train schedule.

## 1.2 Functionalities provided by Railway Reservation System are as follows:

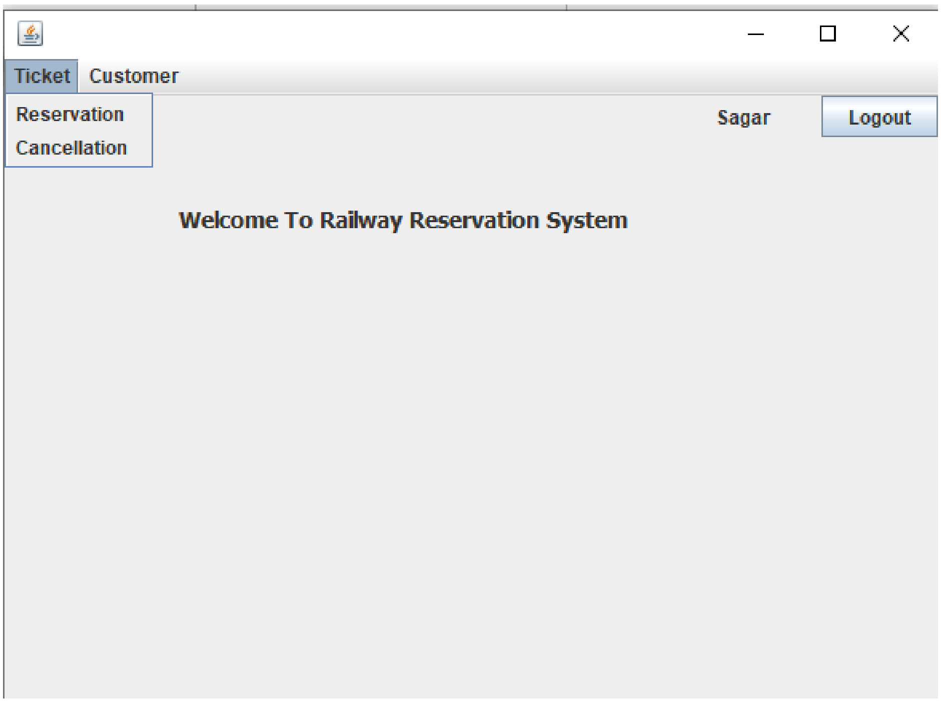
- Provides the searching facility based on various factors such as train, booking, passenger, train schedule.
- Railway reservation system also manage the train route detail online for passenger detail, online for passenger detail, train schedule detail, train.
- It tracks all the information of Ticket, Train Route, Passenger etc.
- Manage the information Ticket.
- Show the information and description of the Train, Ticket.
- To increase efficiency of managing the Train, Ticket.
- It deals with monitoring the information and transaction of passenger.
- Manage the information of train data.
- Editing, adding, and updating of records is improved which results in proper resource management of train data.
- Manage the information of passenger.
- Integration of all records of train schedule.

## 1.3. Project Interface Examples

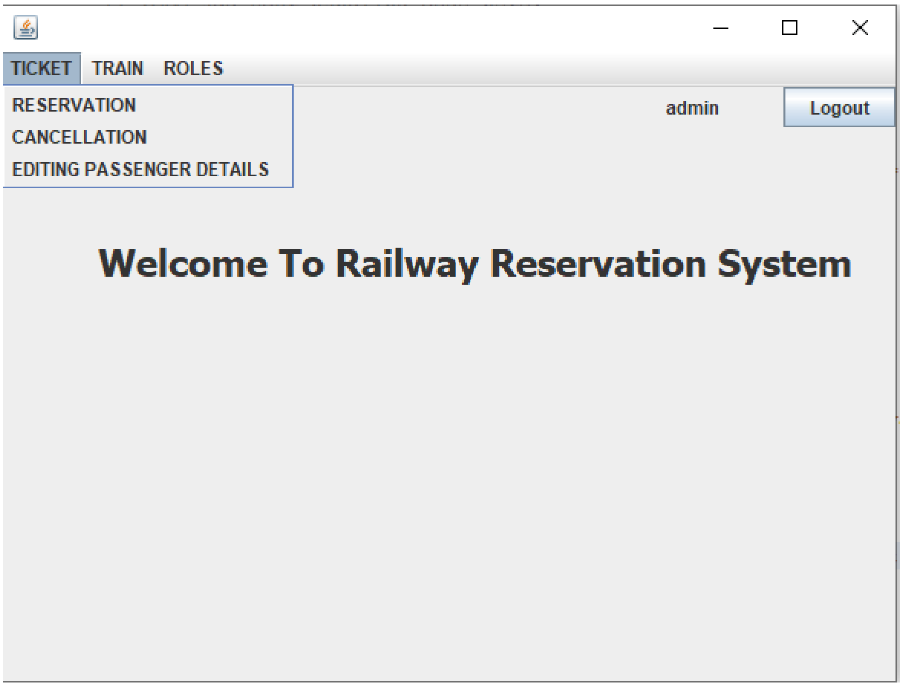
Common Login Interface for all type of users:



**Passenger Interface:**



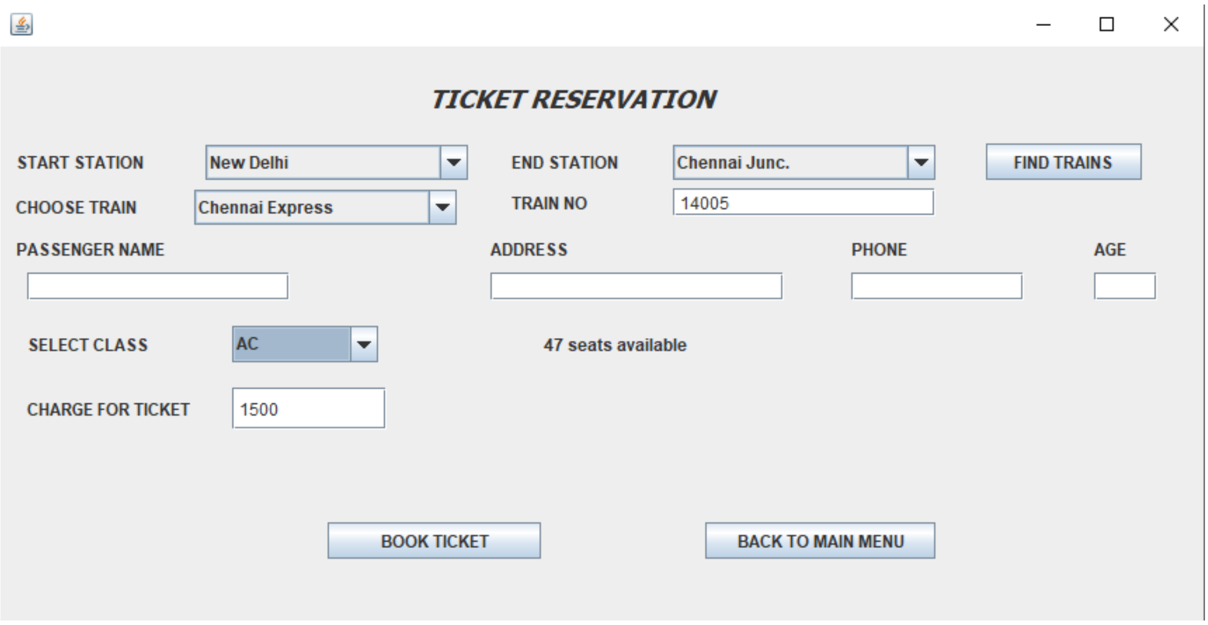
## Admin Interface:



admin

Welcome To Railway Reservation System

## Ticket Reservation:



*TICKET RESERVATION*

START STATION: New Delhi

END STATION: Chennai Junc.

CHOOSE TRAIN: Chennai Express

TRAIN NO: 14005

PASSENGER NAME

ADDRESS

PHONE

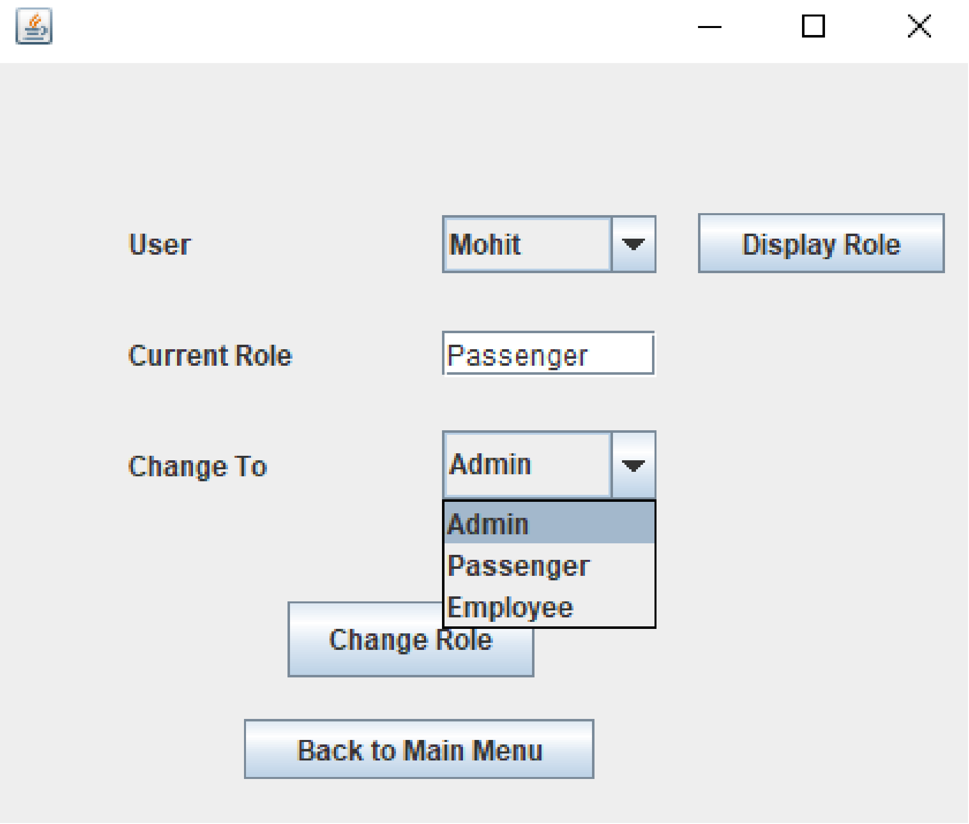
AGE

SELECT CLASS: AC

47 seats available

CHARGE FOR TICKET: 1500

**Change Roles:**



User Mohit

Current Role Passenger

Change To Admin

Roles for various users are implemented through database.

# 2. Authorization

**RBAC**: Role-based access control (RBAC) systems assign access and actions according to a person’s role within the system. Everyone who holds that role has the same set of rights. Those who hold different roles have different rights.

Role-based access control (RBAC)is a policy-neutral access-control mechanism defined around roles and privileges. The components of RBAC such as role-permissions, user-role and role-role relationships make it simple to perform user assignments. A study by NIST has demonstrated that RBAC addresses many needs of commercial and government organizations. RBAC can be used to facilitate administration of security in large organizations with hundreds of users and thousands of permissions. Although RBAC is different from MAC and DAC access control frameworks, it can enforce these policies without any complication.

## 2.1. Need of RBAC based authorization

Every company has sensitive documents, programs, and records. Protect them too strictly, and company’s work grinds to a halt. Leave them open, and catastrophic security issues can arise.

Using RBAC we can grant access to those who need it while blocking those who don’t need access. Make changes based on a person’s role rather than individual attributes. We can make these changes quickly by altering access by role.

### 2.1.1. What is RBAC

All role-based control systems share core elements, such as:

- **Administrators**: They identify roles, grant permissions, and otherwise maintain security systems.
- **Roles**: Workers are grouped together based on the tasks they perform.
- **Permissions**: Access and actions attach to each role, and they outline what people can and cannot do.

RBAC systems do not require:

- **Differentiation of individual freedoms**: Access is defined by a person’s role, not that person’s preferences or wishes. This makes it easy to manage permissions.
- **Intensive maintenance**: Permissions follow roles. A new job function becomes a new role applied to dozens (or hundreds or thousands) of employees with only a small amount of work for the administrator. Promotions involve changing roles, not editing permissions as line items.

### 2.1.2. Roles within RBAC

Roles dictate authorization within an RBAC system. It’s critical to define them properly. Otherwise, large groups of people withing company can’t do their work.

Roles can be defined by:

- **Authority**: Senior management needs access to files interns should never see.
- **Responsibility**: A board member and a CEO might hold similar authority within a company, but they are each responsible for different core functions.
- **Competence**: A skilled worker can be trusted to work within sensitive documents without errors, while a novice could make catastrophic mistakes. It’s important to tailor access accordingly.

### 2.1.3. Role-Based Access Control Permissions

Permissions specify what people can access and what they can do in the system. Think of permissions as the rules people follow per the roles have outlined. Permissions should involve:

- **Access**: Who can open a specific drive, program, file, or record? Who shouldn’t even know these things exist? Access will limit what people can see.
- **Reading**: Who can scan through these documents, even if they can’t change anything inside of them? Some roles may have the ability to reference materials but not make changes to them.
- **Writing**: Who can change documents? Does someone else have to approve the changes, or are they permanent? This will define permissions.
- **Sharing**: Who can download a document or send it as an email attachment? As with the other permissions, some users will not be able to share materials even if they can reference them.
- **Finances**: Who can charge money? Who can offer refunds? Permissions could involve the ability to deal with charges and refunds, set up credit accounts, or cancel payments.

### 2.1.4. Role-Based Access Control Benefits

Security options abound, and it’s not always easy to make the right choice for company. RBAC comes with plenty of tried-and-true benefits that set it apart from the competition. An RBAC system can:

- **Reduce complexity**: New employees gain access based on their roles, not on long lists of server and document requirements. This simplifies creating, maintaining, and auditing policies.
- **Allow global administration**: Change access for many employees all at once by altering permissions associated with a role.
- **Ease on boarding**: As people join, move withing, or are promoted within organizations, and we don’t have to worry about the individual’s permissions, just that they’re in the right place. The roles take care of the rest.
- **Reduce Blunders**: Traditional security administration is error-prone. Adding permissions for individuals gives us plenty of options to make a mistake. Change a role’s access, and you’re less likely to give someone too much (or too little) power.
- **Lower overall costs**: When admin duties shrink, companies save on security administration. This saves our organization time and money.

## 2.2. Component of RBAC in our Application:

Railway Reservation system has following component:

**Admin user** is provided with functionalities:

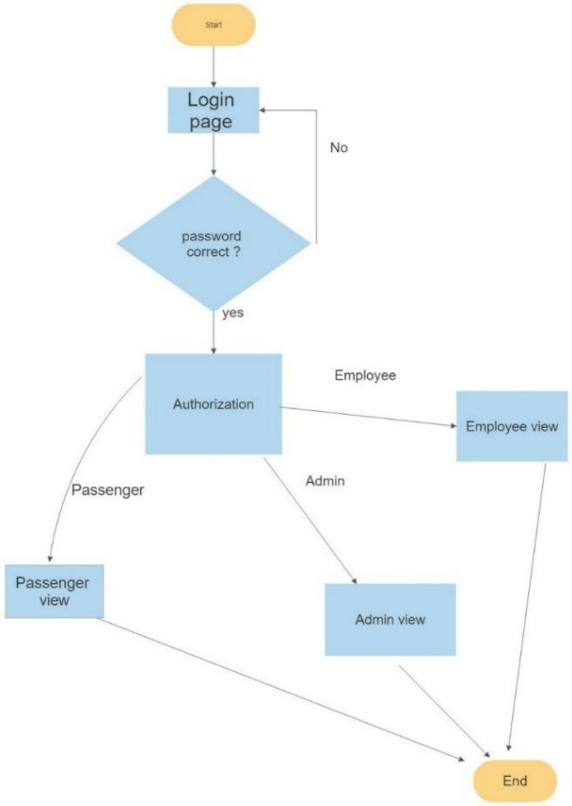
- Ticket Reservation for a train.
- Cancellation for the Reserved ticket.
- Editing of Customer Records.
- Adding a new Train.
- Searching a Train Details.
- Editing Train Details.
- Changing Roles of User.

**Employee** is provided with functionalities:

- Ticket Reservation for a train for all customer.
- Cancellation for the Reserved ticket for all customers.
- Editing of all Customer Records for the Reserved Ticket.

**A Passenger** is provided with functionalities:

- Ticket Reservation for a train for itself.
- Cancellation for the Reserved ticket for itself.
- Editing of its own Customer Records for the Reserved Ticket.



## 2.3. Components of administrative model

To ensure better functioning and smooth running of an organization, these are following components in our project:

- **Planning**: In this project the admin has to create a database such that it will store all the information about user, trains and passengers in the System and will have to add trains to the database with Train no. used to identify the train uniquely.

- **Organization**: In the Railway Reservation System the Administration decides what kind of service to be provided to the Employee and to passengers. Also, it has to maintain the database correctly and make the necessary changes whenever necessary.

- **Directing**: Administrator should be able to identify the requirements and need of organisation and should be able to provide the required roles to the respective users.

- **Coordinating**: Here whenever a new user is added, it is given a passenger role by default and will have a restricted functionality. So, in order to give the appropriate role to the newly joined employee, administrator, employee and organisation has to coordinate with each other.

- **Reporting:** Administrator keep track of the changes made related to the Employees or the trains and role assignment to user and report them to the organisation to maintain the authenticity and for improvement of the system.

# 3. Conclusion:

In our project reservation system, we have stored all the information about that train schedule, and the user booking ticket and even the status of trains, seat, etc. This database is helpful for the application which facilitate passengers to book that train ticket and check the detail of the terrain and their status from their place itself. So, it avoids inconvenience of going to railway station for each and every query. Here we have three categories of users for different functionalities and we assign the roles to user using RBAC. Our system has predefined set of permissions (for example- we are having 9 set of permissions which implies the maximum operation an admin can do with the system). We had considered the most important requirements only many more feature in detail can be added to our project in order to obtain even more user-friendly application these applications are already in progress and in future, they can be upgraded and may become part of an aging technology.

# 4. References

[1] S ANDHU , R. 1997. Roles versus groups. In Proceedings of the 2nd ACM Workshop on Role-Based Access Control (Fairfax, VA, Nov. 6-7). ACM Press, New York, NY.

[2] https://www.okta.com/identity-101/what-is-role-based-access-controlrbac/