**RAILWAY RESERVATION SYSTEM**

**SUBJECT: PROJECT**



**ACADEMY OF TECHNOLOGY**

 Aedconagar,Adisaptagram,Hooghly,West Bengal,PIN-712121

**Name-RAJIB BANIK**

**Department­-EE**

**Year-2nd**

**RAILWAY RESERVATION SYSTEM**

**A PROJECT REPORT**

***In partial fulfillment for the award of the degree***

***Of***

**CORE JAVA**

***Under the Guidance of***

**MEGHALI DAS**

***Project Carried Out At***

****

**Ardent Computech Pvt. Ltd. (An ISO 9001:2008 Certified)**

**CF-137, Sector - 1, Salt Lake City, Kolkata - 700 064**

***Submitted By:***

***RAJIB BANIK***



**ACADEMY OF TECHNOLOGY**

**In association with**



***(Note:All entries of the proforma of approval should be filled up with appropriate and complete information. Incomplete proforma of approval in any respect will be summarily rejected.)***

1. Title of the Project: **RAILWAY RESERVATION SYSTEM**

2. Name and Address of the Guide: **MISS MEGHALI DAS**

Subject Matter Expert & Assistant Technical Head (.NET Domain)

Ardent Computech Pvt. Ltd. (An ISO 9001:2008 Certified)

CF-137, Sector - 1, Salt Lake City, Kolkata - 700 064

PhD**\***  M.Tech.**\***B.E.\***/**B.Tech.\*MCA**\*** M.Sc.**\***

4. Educational Qualification of the Guide:

5. Working / Training experience of the Guide:**8 Years**

6. Project Version Control History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Primary Authors** | **Description of Version** | **Date Completed** |
| Final | RAJIB BANIK  SUJOY DUTTA  DEBAYAN GHOSH  AKASH SHARMA | Project Report | 14th January, 2019 |

Signature of Team Member Signature of Approval

Date: 14.01.2019 Date: 14.01.2019

**For Office Use Only**

**Miss MEGHALI DAS**

Project Proposal Evaluator

**Approved**

**Not Approved**

**Project Responsibility Form**

**RAILWAY RESERVATION SYSTEM**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **GROUP NO.** | **NAME OF MEMBER** | **RESPONSIBILITY** | | 2 | RAJIB BANIK  SUJOY DUTTA  AKASH SHARMA  DEBAYAN GHOSH | Project Leader & PPT, Coding & Designing, System Analysis, Testing, Implementation and Maintenance |     Each group member must participate in project development and developing the ideas for the required elements. Individual group members will be responsible for completing tasks which help to finalize the project and the performance. All group members must be assigned a task. | |  | | --- | |  | |

Date: 14.01.2019

Name of the Student Signature Of Student

1. SUJOY DUTTA

2. RAJIB BANIK

3DEBAYAN GHOSH

4

**DECLARATION**

I hereby declare that the project work being presented in the project proposal entitled **“RAILWAY RESERVATION SYSTEM”** in partial fulfillment of the requirements for the award of the degree of **JAVA 2ND STANDARD EDITION** at**ARDENT COMPUTECH PVT. LTD, SALTLAKE, KOLKATA, WEST BENGAL**, is an authentic work carried out under the guidance of **MISS MEGHALI DAS**. The matter embodied in this project work has not been submitted elsewhere for the award of any degree of our knowledge and belief.

Date: 14.01.2019

Name of the Student

1. RAJIB BANK
2. SUJOY DUTTA
3. DEBAYAN GHOSH
4. AKASH SHARMA

Signature of the student

a.

b.

c.

d.

****

**Ardent Computech Pvt. Ltd. (An ISO 9001:2008 Certified)**

**CF-137, Sector - 5, Salt Lake City, Kolkata - 700 064**

**CERTIFICATE**

This is to certify that this proposal of minor project entitled **“RAILWAY RESERVATION SYSTEM”** is a record of bonafide work, carried out by **RAJIB BANIK** under my guidance at **Ardent Computech Pvt. Ltd**. In my opinion, the report in its present form is in partial fulfillment of the requirements for the award of the degree of **JAVA 2ND STANDARD EDITION** and as per regulations of the **Ardent*®.*** To the best of my knowledge, the results embodied in this report, are original in nature and worthy of incorporation in the present version of the report.

**Guide / Supervisor**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**MISS MEGHALI DAS**

Subject Matter Expert & Assistant Technical Head (Java Domain)

Ardent Computech Pvt. Ltd. (An ISO 9001:2008 Certified)

CF-137, Sector - 5, Salt Lake City, Kolkata - 700 064

**ACKNOWLEDGEMENT**

Success of any project depends largely on the encouragement and guidelines of many others. I take this sincere opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project work.

I would like to show our greatest appreciation to **Miss Meghali Das**, Project Manager at Ardent, Kolkata. I always felt motivated and encouraged every time by her valuable advice and constant inspiration; without her encouragement and guidance this project would not have materialized.

Words are inadequate in offering my thanks to the other trainees and other members at Ardent Computech Pvt. Ltd. for their encouragement and cooperation in carrying out this project work. The guidance and support received from my university teachers and Ardent facultywas vital for the success of this project.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **Name of the Topic** | **Page No.** |
| 1. | Company Profile | 10 |
|  |  |  |
| 2. | Introduction | 11-14 |
| 3. | Objective | 13 |
| 4. | Scope | 14 |
|  |  |  |
| 5. | System Analysis | 15-22 |
| 6. | Identification of Need | 16 |
| 7. | Feasibility Study | 17 |
| 8. | Work flow | 18-19 |
| 9. | Functional Requirements | 20 |
| 10. | Non-Functional Requirements | 21 |
| 11. | Hardware and Software Requirements | 22 |
|  |  |  |
| 12. | System Design | 23-65 |
| 13. | Data Flow Diagram(DFD) | 24-27 |
| 14. | Entity-Relationship Diagram | 28-30 |
| 15. | Modularization Details | 31-32 |
| 16. | Database Design | 33-42 |
| 17. | User Interface Design | 43-65 |
|  |  |  |
| 18. | Implementation & Testing | 66-78 |
| 19. | Introduction | 67 |
| 20. | Objectives of Testing | 68-69 |
| 21. | Test Cases | 70-73 |
| 22. | White Box Testing | 74 |
| 23. | Black Box Testing | 75 |
| 24. | Output Testing | 76 |
| 25. | User Acceptance Testing | 77 |
| 26. | Integration Testing | 78 |
|  |  |  |
| 27. | Gantt Chart | 79-83 |
|  |  |  |
| 28. | System Security Measures | 80-83 |
| 29. | Database Security Measures | 81 |
| 30. | System Security Measures | 82 |
| 31. | Limitations | 83 |
|  |  |  |
| 32. | Conclusion | 84 |
|  |  |  |
| 33. | Future Scope & Further Enhancements | 85 |
|  |  |  |
| 34. | Bibliography | 86 |

**ARDENT COMPUTECH PVT. LTD.**

Ardent Computech Private Limited is an ISO 9001-2008 certified Software Development Company in India. It has been operating independently since 2003. It was recently merged with ARDENT TECHNOLOGIES.

Ardent Technologies

ARDENT TECHNOLOGIES is a Company successfully providing its services currently in UK, USA, Canada and India. The core line of activity at ARDENT TECHNOLOGIES is to develop customized application software covering the entire responsibility of performing the initial system study, design, development, implementation and training. It also deals with consultancy services and Electronic Security systems. Its primary clientele includes educational institutes, entertainment industries, resorts, theme parks, service industry, telecom operators, media and other business houses working in various capacities.

Ardent Collaborations

ARDENT COLLABORATIONS, the Research Training and Development Department of ARDENT COMPUTECH PVT LTD is a professional training Company offering IT enabled services & industrial trainings for B-Tech, MCA, BCA, MSC and MBA fresher’s and experienced developers/programmers in various platforms. Summer Training / Winter Training / Industrial training will be provided for the students of B.TECH, M.TECH, MBA, MCA and BCA only. Deserving candidates may be awarded stipends, scholarships and other benefits, depending on their performance and recommendations of the mentors.

# Associations

Ardent is an ISO 9001:2008 company.

It is affiliated to National Council of Vocational Training (NCVT), Directorate General of Employment & Training (DGET), Ministry of Labor & Employment, and Government of India.

******

***Railway reservation system***

**INTRODUCTION**

* **The RAILWAY RESERVATION SYSTEM** in java is a great project. This project is designed for successful completion of project on Railway ReservationSystem. The Indian Railways (IR) carries about 5.5 lakhs passengers in reserved accommodation every day. The Computerized Passenger Reservation System(PRS) facilates the booking and cancellation of tickets from any of the 4000 terminals(i.e. PRS booking window all over the countries). These tickets can be booked or cancelled for journeys commence in in any part of India and ending in any other part, with travel time as long as 72hours and distance upto several thousand kilometers. The project of PRS was launched on 15th November 1985, over Northern Railway with the installation of Integrated Multiple Train Passenger Reservation System(IMPRESS), an online transaction processing system developed by Indian Railways in association with Computer Maintenance Corporation Ltd., at New Delhi. The objective was to provide reserved accommodations on any train from any counter,preparation of train charts and accounting of the money collected. The application was implemented in 1987 Mumbai,Chennai,Kolkata,Secunderabad subsequently.
* Online Booking
* Counter Booking
* **ANALYSIS**

**OBJECTIVE**

Railway Reservation System in Java is planned to with the help of this people can book their tickets online through internet, sitting in their home by a single click of mouse. Using their credit cards people can easily get their tickets done within minutes. There are certai2n charges for online booking as well.. For all those kinds of need Railway Reservation System project in Java contains modules which includes details of

|  |
| --- |
| Ticket Online |
| Internet |
| Availability |
| Price |
| Many more…. . |

**SCOPE**

The purpose of the Railway Reservation System is to simplify and automatic the process of searching for Ticket in case of emergency and maintain the records of Tatkal ticket. Online Railway Reservation System for booking the tickets online the intended user has to enter details like username and password

The system is used for maintaining all the processes Railway Reservation System.

As a whole, the system is focused to work with Railway Reservation system and on additional modification it canalso be used as management system of Railway Reservation System.

**Confirm Reservation for confirm Seat**.

**Reservation against Cancellation.**

**Waiting list Reservation.**

**Online Reservation.**

**Tatkal Reservation.**

***System Analysis***

**IDENTIFICATION OF NEED**

In Online Railway Reservation System for booking the tickets online the intended user has to enter details like username and password. Once the username and passwords are verified then he is allowed to enter the main system wherein he can select the destination, date and no of tickets. For the purchase of the tickets he is asked for payment details (like credit card details etc.). The user is also provided with a provision of cancelling the reservation where in the entire amount is credited back to his account. The analysis part also comprises of the UML diagrams that boost the program.

**FEASIBILITY STUDY**

Feasibility study is made to see if the project on completion will serve the purpose the organization for the amount of work.

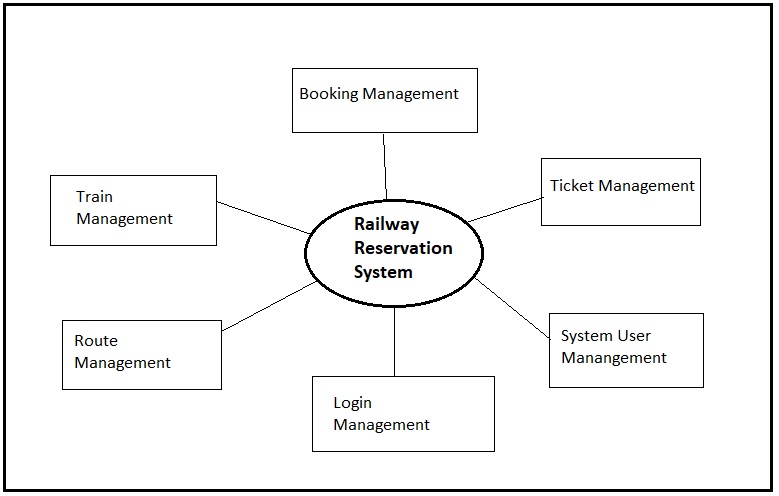
Effortand the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provide the feasibility of the project that is being designed and lists various area that were considered very carefully during the feasibility study of this project such as Technical, Economic and operational feasibilities.

**WORK FLOW**

The job application may ask for information on your most recent jobs, typically two to five positions. Or, the employer may ask for a number of years of experience, typically five to ten years of experience.

Employers generally want information on the company you worked for, your job title, and the dates you were employed there. However, sometimes the employer will ask for a more detailed employment history and more information on the jobs you have held as [part of the hiring process](https://www.thebalance.com/recruitment-and-hiring-process-2062875). For example, he or she might ask for the name and contact information for your previous supervisors.



The developer is responsible for:

* Developing the system, which meets the SRS and solving all the requirements of the system?
* Demonstrating the system and installing the system at client's location after the acceptance testing is successful.
* Submitting the required user manual describing the system interfaces to work on it and also the documents of the system.
* Conducting any user training that might be needed for using the system.
* Maintaining the system for a period of one year after installation.

1. **PERFORMANCE REQUIREMENTS**

It is available during all 24 hours.

 Offered through Mail express, super-fast, Rajdhani & Shatabdi Trains. About 1520 Trainsruns daily.

 Variety of compartments based on comfort:

• AC first class.

• AC sleeper.

• First class.

• AC three tier.

• AC chair car.

• Sleeper class

• Ordinary chair car.

Types of concerns & complexities:

• 44 types of quotas.• 8 types of trains.• 9 types of classes.• 162 types of concessions.• 127 types of bogies

**NON-FUNCTIONAL REQUIREMENTS**

* **Usability Requirement**: The system shall allow the users to access the system from any browsers, no special training is required. The system user friendly and the system is written in simple English.
* **Availability Requirement**: The system is available 100% for the user and is used by 24 hours a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.
* **Accuracy**: The system should accurately provide real time information taking into consideration various issues. The system shall provide 100% access reliability.
* **Performance Requirement**: The information is refreshed at regular intervals depending upon whether some updates have occurred or not. The system shall respond the member in less than 2 seconds.
* **Security Requirement**: System will use a secured database and the system will have different users and each user has different types of constraints.
* **Reliability Requirement**:The system has to be 100% reliable due to the importance of data and the damages that can be caused by incorrect data. The system will run 7 days a week and 24 hours a day.

## **HARDWARE and SOFTWARE REQUIREMENTS**

**HARDWARE REQUIREMENTS**

* Computer that has a 1.6GHz or faster processor
  + 1 GB (32 Bit) or 2 GB (64 Bit) RAM (Add 512 MB if running in a virtual machine)
  + HDD 20 GB Hard Disk Space and Above Hardware Requirements5400 RPM hard disk drive
  + DirectX 9 capable video card running at 1024 x 768 or higher-resolution display

**SOFTWARE REQUIREMENTS**

 WINDOWS OS (XP/2000/200 Server/2003 Server/Vista or7/8/10)

 Eclipse oxygen

 Internet Information Server 8.0 (IIS)

 JDK 8.1

 Oracle 11G Express Edition

***SYSTEM DESIGN***

**DATA FLOW DIAGRAM**

**LEVEL 1**

Railway Reservation System

Confirmation or rejection details

**Registers**

Railway Reservation System

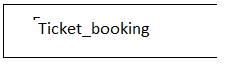
Sending User details

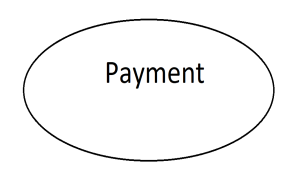
Form

****

Confirmation

Login details

****

****

Confirmation

**ENTITY RELATIONSHIP DIAGRAM**

User\_login

1

BOOK

M

TICKET\_BOOKING

has

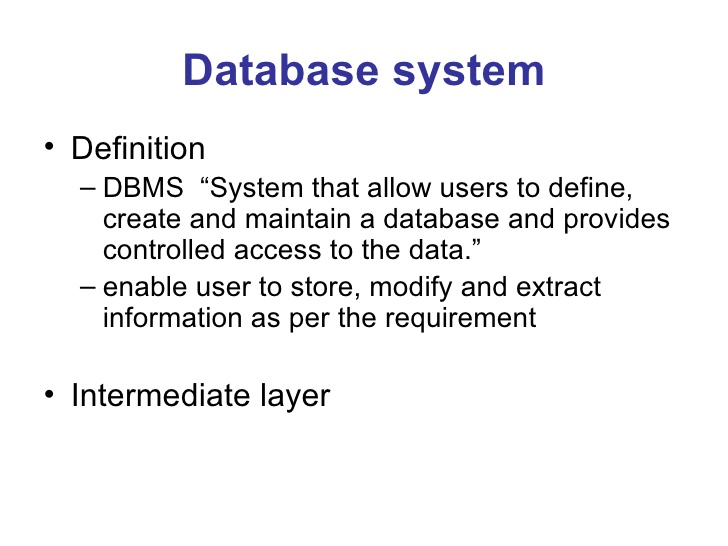
Train\_name

**DATA INTEGRITY AND CONSTRAINTS**

**Data integrity is normally enforced in a database system by a series of integrity constraints or rules. Three types of integrity constraints are an inherent part of the relational data model: entity integrity, referential integrity and domain integrity:  Entity integrity concerns the concept of a primary key. Entity integrity is an integrity rule which states that every table must have a primary key and that the column or columns chosen to be the primary key should be unique and not null.**

** Concerns the concept of a foreign key. The referential integrity rule states that any foreign-key value can only be in one of two states. The usual state of affairs is that the foreign-key value refers to a primary key value of some table in the database. Occasionally, and this will depend on the rules of the data owner, a foreign-key value can be null. In this case we are explicitly saying that either there is no relationship between the objects represented in the database or that this relationship is unknown.**

** Domain integrity specifies that all columns in a relational database must be declared upon a defined domain. The primary unit of data in the relational data model is the data item. Such data items are said to be non-decomposable or atomic. A domain is a set of values of the same type.**

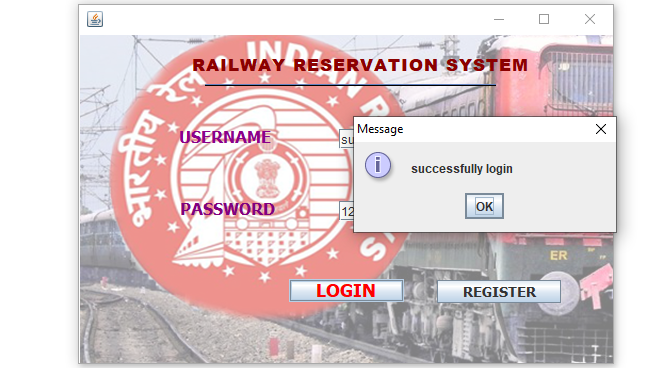


**USER INTERFACE DESIGN**

User interface design (UID) or user interface engineering is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing the user experience. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals (usercentred design). Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to it. Graphic design and typography are utilized to support its usability, influencing how the user performs certain interactions and improving the aesthetic appeal of the design; design aesthetics may enhance or detract from the ability of users to use the functions of the interface. The design process must balance technical functionality and visual elements (e.g., mental model) to create a system that is not only operational but also usable and adaptable to changing user needs. Interface design is involved in a wide range of projects from computer systems, to cars, to commercial planes; all of these projects involve much of the same basic human interactions yet also require some unique skills and knowledge. As a result, designers tend to specialize in certain types of projects and have skills centredon their expertise, whether that be software design, user research, web design, or industrial design.

**Login page**

***snapshot***

****

**Coding**

**public** login() {

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 550, 367);

contentPane = **new** JPanel();

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel lblUsername = **new** JLabel("USERNAME");

lblUsername.setForeground(**new** Color(139, 0, 139));

lblUsername.setFont(**new** Font("Tahoma", Font.***BOLD***, 16));

lblUsername.setBounds(100, 94, 124, 17);

contentPane.add(lblUsername);

textField = **new** JTextField();

textField.setBounds(260, 94, 86, 20);

contentPane.add(textField);

textField.setColumns(10);

JLabel lblPassword = **new** JLabel("PASSWORD");

lblPassword.setForeground(**new** Color(128, 0, 128));

lblPassword.setFont(**new** Font("Tahoma", Font.***BOLD***, 16));

lblPassword.setBounds(101, 166, 105, 17);

contentPane.add(lblPassword);

textField\_1 = **new** JTextField();

textField\_1.setBounds(260, 166, 86, 20);

contentPane.add(textField\_1);

textField\_1.setColumns(10);

JButton btnLogin = **new** JButton("LOGIN");

btnLogin.setForeground(Color.***RED***);

btnLogin.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

String user=textField.getText().toString();

String password=textField\_1.getText().toString();

**try**{

**int** i=0;

Class.*forName*("com.mysql.jdbc.Driver");Connection

con=DriverManager.*getConnection*("jdbc:mysql://localhost:3306/railway","root","");

//String sql="SELECT \* FROM user\_login WHERE username=? and password=?";

PreparedStatement ps= con.prepareStatement("select \* from user\_login where username=? and password=?");

ps.setString(1, user);

ps.setString(2,password);

ResultSet rs=ps.executeQuery();

**if**(rs.next()){

JOptionPane.*showMessageDialog*(**null** , "successfully login");

PassengerInformation pi = **new** PassengerInformation();

pi.setVisible(**true**);

}

**else**{

JOptionPane.*showMessageDialog*(**null**, "username and password invalid or register first ");

}

}

**catch**(Exception e1)

{

System.***out***.println(e1);

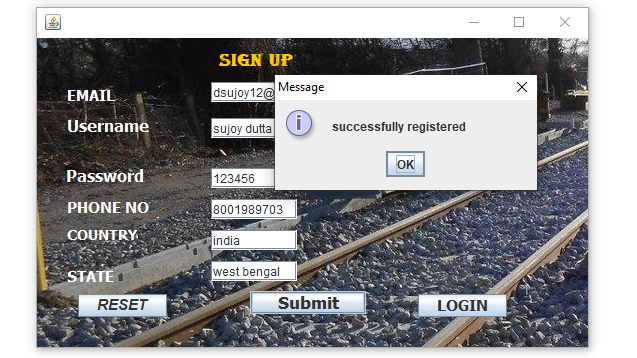
}

}

});

**Signup page**

**snapshot**

****

**CODING**

**import** java.awt.BorderLayout;

**import** java.awt.EventQueue;

**import** javax.swing.JFrame;

**import** javax.swing.JPanel;

**import** javax.swing.border.EmptyBorder;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JTextField;

**import** java.awt.Font;

**import** javax.swing.JButton;

**import** java.awt.event.ActionListener;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.awt.event.ActionEvent;

**import** javax.swing.ImageIcon;

**import** java.awt.Color;

**public** **class** Registration **extends** JFrame {

**private** JPanel contentPane;

**private** JTextField textField;

**private** JTextField textField\_1;

**private** JTextField textField\_2;

**private** JTextField textField\_3;

**private** JTextField textField\_4;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

Registration frame = **new** Registration();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** Registration() {

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 567, 348);

contentPane = **new** JPanel();

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel lblRegistration = **new** JLabel("SIGN UP");

lblRegistration.setForeground(Color.***ORANGE***);

lblRegistration.setFont(**new** Font("Algerian",Font.***BOLD***, 18));

lblRegistration.setBounds(182, 11, 211, 22);

contentPane.add(lblRegistration);

JLabel lblName = **new** JLabel("Username");

lblName.setForeground(Color.***WHITE***);

lblName.setFont(**new** Font("Tahoma", Font.***BOLD***, 16));

lblName.setBounds(30, 80, 100, 17);

contentPane.add(lblName);

textField = **new** JTextField();

textField.setBounds(174, 80, 138, 20);

contentPane.add(textField);

textField.setColumns(10);

JLabel lblPassword = **new** JLabel("Password");

lblPassword.setForeground(Color.***WHITE***);

lblPassword.setFont(**new** Font("Tahoma", Font.***BOLD***, 16));

lblPassword.setBounds(29, 131, 88, 14);

contentPane.add(lblPassword);

textField\_1 = **new** JTextField();

textField\_1.setBounds(174, 130, 138, 20);

contentPane.add(textField\_1);

textField\_1.setColumns(10);

JButton btnSubmit = **new** JButton("Submit");

btnSubmit.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent arg0) {

String user=textField.getText().toString();

String password=textField\_1.getText().toString();

String email=textField\_2.getText().toString();

String phone\_no=textField\_3.getText().toString();

String country=textField\_4.getText().toString();

**try**

{

Connection con=**null**;

**int** i=0;

Class.*forName*("com.mysql.jdbc.Driver");

con=DriverManager.*getConnection*("jdbc:mysql://localhost:3306/railway","root","");

PreparedStatement ps=con.prepareStatement("insert into user\_login values(?,?,?,?,?)");

ps.setString(1, user);

ps.setString(2, password);

ps.setString(3, email);

ps.setString(4, phone\_no);

ps.setString(5,country);

i=ps.executeUpdate();

**if**(i>0)

{

JOptionPane.*showMessageDialog*(**null** , "successfully registered");

}

**else**

{

JOptionPane.*showMessageDialog*(**null** , "plz enter all information");

}

}

**catch**(Exception e)

{

System.***out***.println(e);

}

}

});

btnSubmit.setFont(**new** Font("Tahoma", Font.***BOLD***, 17));

btnSubmit.setBounds(213, 253, 116, 23);

contentPane.add(btnSubmit);

JLabel lblEmail = **new** JLabel("EMAIL");

lblEmail.setForeground(Color.***WHITE***);

lblEmail.setFont(**new** Font("Tahoma", Font.***BOLD***, 15));

lblEmail.setBounds(30, 45, 82, 24);

contentPane.add(lblEmail);

textField\_2 = **new** JTextField();

textField\_2.setBounds(174, 44, 88, 20);

contentPane.add(textField\_2);

textField\_2.setColumns(10);

JLabel lblPhoneNo = **new** JLabel("PHONE NO");

lblPhoneNo.setForeground(Color.***WHITE***);

lblPhoneNo.setFont(**new** Font("Tahoma", Font.***BOLD***, 15));

lblPhoneNo.setBounds(30, 171, 87, 17);

contentPane.add(lblPhoneNo);

textField\_3 = **new** JTextField();

textField\_3.setBounds(174, 171, 86, 20);

contentPane.add(textField\_3);

textField\_3.setColumns(10);

JLabel lblCountry = **new** JLabel("COUNTRY");

lblCountry.setForeground(Color.***WHITE***);

lblCountry.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

lblCountry.setBounds(30, 212, 76, 14);

contentPane.add(lblCountry);

textField\_4 = **new** JTextField();

textField\_4.setBounds(174, 211, 86, 20);

contentPane.add(textField\_4);

textField\_4.setColumns(10);

JButton btnExit = **new** JButton("LOGIN");

btnExit.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent arg0) {

login lg = **new** login();

lg.setVisible(**true**);

}

});

btnExit.setFont(**new** Font("Tahoma", Font.***BOLD***, 15));

btnExit.setBounds(381, 256, 89, 23);

contentPane.add(btnExit);

JButton btnReset = **new** JButton("RESET");

btnReset.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent arg0) {

textField\_2.setText(**null**);

textField.setText(**null**);

textField\_1.setText(**null**);

textField\_3.setText(**null**);

textField\_4.setText(**null**);

}

});

btnReset.setFont(**new** Font("Arial", Font.***BOLD*** | Font.***ITALIC***, 15));

btnReset.setBounds(41, 256, 89, 23);

contentPane.add(btnReset);

JLabel lblNewLabel = **new** JLabel("New label");

lblNewLabel.setIcon(**new** ImageIcon("C:\\Users\\Sujoy PC\\Desktop\\C77FS2EWsAAcEMt.jpg"));

lblNewLabel.setBounds(0, 0, 551, 309);

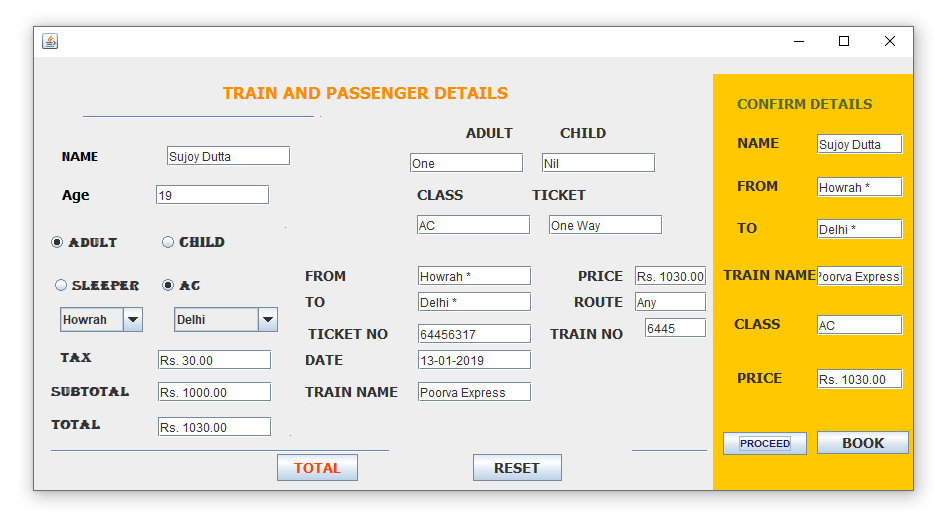
contentPane.add(lblNewLabel);

}

}

**TICKET BOOKING PAGE**

**SNAPSHOTS:**

****

**CODING**

**public** **class** PassengerInformation **extends** JFrame {

**private** JPanel contentPane;

**private** JTextField tf15;

**private** JTextField tf22;

**private** JTextField tf1;

**private** JTextField tf3;

**private** JTextField tf6;

**private** JTextField tf7;

**private** JTextField tf;

**private** JTextField tf10;

**private** JTextField tf11;

**private** JTextField tf4;

**private** JTextField tf5;

**private** JTextField tf2;

**private** JTextField tf8;

**private** JTextField tf9;

**private** JTextField tf13;

**private** JTextField tf14;

**private** JTextField tf12;

**static** **int** *count*=1000;

**private** JTextField textField;

**private** JTextField tf16;

**private** JTextField textField\_2;

**private** JTextField tf17;

**private** JTextField tf18;

**private** JTextField tf19;

**private** JTextField tf20;

**static** String *name*;

**static** String *price1*;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

PassengerInformation frame = **new** PassengerInformation();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** PassengerInformation() {

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 895, 472);

contentPane = **new** JPanel();

contentPane.setForeground(**new** Color(0, 0, 0));

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

JLabel lblName = **new** JLabel("NAME");

lblName.setFont(**new** Font("Tahoma", Font.***BOLD***, 13));

lblName.setForeground(**new** Color(0, 0, 0));

tf15 = **new** JTextField();

tf15.setColumns(10);

JLabel lblAge = **new** JLabel("Age\r\n");

lblAge.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

lblAge.setForeground(**new** Color(0, 0, 0));

tf22 = **new** JTextField();

tf22.setColumns(10);

JRadioButton rb6 = **new** JRadioButton("ADULT\r\n");

rb6.setFont(**new** Font("Algerian", Font.***BOLD***, 14));

JRadioButton rb7 = **new** JRadioButton("CHILD");

rb7.setFont(**new** Font("Algerian", Font.***BOLD***, 15));

JComboBox cb1 = **new** JComboBox();

cb1.setModel(**new** DefaultComboBoxModel(**new** String[] {"SOURCE", "Howrah", "Kolkata"}));

cb1.setBounds(43, 11, 74, 20);

contentPane.add(cb1);

JComboBox cb2 = **new** JComboBox();

cb2.setModel(**new** DefaultComboBoxModel(**new** String[] {"DESTINATION", "Delhi", "Patna"}));

contentPane.add(cb2);

JRadioButton rb2 = **new** JRadioButton("SLEEPER");

rb2.setFont(**new** Font("Algerian", Font.***BOLD***, 14));

JRadioButton rb1 = **new** JRadioButton("AC");

rb1.setFont(**new** Font("Algerian", Font.***BOLD***, 14));

JButton btnReset = **new** JButton("RESET");

btnReset.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

JLabel lblTax = **new** JLabel("TAX");

lblTax.setFont(**new** Font("Algerian", Font.***BOLD***, 14));

tf1 = **new** JTextField();

tf1.setColumns(10);

JLabel lblTotal = **new** JLabel("TOTAL");

lblTotal.setFont(**new** Font("Algerian", Font.***BOLD***, 14));

tf3 = **new** JTextField();

tf3.setColumns(10);

JButton btnTotal = **new** JButton("TOTAL");

btnTotal.setForeground(**new** Color(255, 69, 0));

btnTotal.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

btnTotal.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

**double** tax = 2.0;

**double** fare = 1500;

**double** fare2 = 200;

**double** totalTax, eco = 3.85, fclass = 5.60;

**if**((rb2.isSelected()) && (rb7.isSelected()) && cb1.getSelectedItem().equals("Howrah") && cb2.getSelectedItem().equals("Delhi"))

{

totalTax = (tax \* fare)/100;

String tTax = String.*format*("Rs. %.2f", totalTax);

tf1.setText(tTax);

String Subtotal = String.*format*("Rs. %.2f", fare);

tf2.setText(Subtotal);

String Total = String.*format*("Rs. %.2f",fare + totalTax);

tf3.setText(Total);

tf13.setText(Total);

// ==============net side==================//

tf4.setText("SL");

tf5.setText("One Way");

tf7.setText("One");

tf6.setText("Nil");

tf11.setText("Poorva Express");

}

**else** **if**((rb2.isSelected()) && (rb6.isSelected()) && cb1.getSelectedItem().equals("Howrah") && cb2.getSelectedItem().equals("Delhi"))

{

totalTax = (tax \* fare)/100;

String tTax = String.*format*("Rs. %.2f", totalTax);

tf1.setText (tTax);

String Subtotal = String.*format*("Rs. %.2f", fare/2);

tf2.setText(Subtotal);

String Total = String.*format*("Rs. %.2f",fare/2 + totalTax);

tf3.setText(Total);

tf13.setText(Total);

// ==============net side==================//

tf4.setText("SL");

tf5.setText("One Way");

tf7.setText("Nil");

tf6.setText("One");

tf11.setText("Poorva Express");

}

**else** **if**((rb1.isSelected()) && (rb7.isSelected()) && cb1.getSelectedItem().equals("Howrah") && cb2.getSelectedItem().equals("Delhi"))

{

totalTax = (tax \* fare)/100;

String tTax = String.*format*("Rs. %.2f", totalTax);

tf1.setText(tTax);

String Subtotal = String.*format*("Rs. %.2f", fare\*1.5);

tf2.setText(Subtotal);

String Total = String.*format*("Rs. %.2f",fare\*1.5 + totalTax);

tf3.setText(Total);

tf13.setText(Total);

// ==============net side==================//

tf4.setText("AC");

tf5.setText("One Way");

tf7.setText("One");

tf6.setText("Nil");

tf11.setText("Poorva Express");

}

**else** **if**((rb1.isSelected()) && (rb6.isSelected()) && cb1.getSelectedItem().equals("Howrah") && cb2.getSelectedItem().equals("Delhi"))

{

totalTax = (tax \* fare)/100;

String tTax = String.*format*("Rs. %.2f", totalTax);

tf1.setText(tTax);

String Subtotal = String.*format*("Rs. %.2f", fare/1.5);

tf2.setText(Subtotal);

String Total = String.*format*("Rs. %.2f",fare/1.5 + totalTax);

tf3.setText(Total);

tf13.setText(Total);

// ==============net side==================//

tf4.setText("AC");

tf5.setText("One Way");

tf7.setText("Nil");

tf6.setText("One");

tf11.setText("Poorva Express");

}

**if**((rb2.isSelected()) && (rb7.isSelected()) && cb1.getSelectedItem().equals("Howrah") && cb2.getSelectedItem().equals("Patna"))

{

totalTax = (tax \* fare2)/100;

String tTax = String.*format*("Rs. %.2f", totalTax);

tf1.setText(tTax);

String Subtotal = String.*format*("Rs. %.2f", fare2);

tf2.setText(Subtotal);

String Total = String.*format*("Rs. %.2f",fare2 + totalTax);

tf3.setText(Total);

tf13.setText(Total);

// ==============net side==================//

tf4.setText("SL");

tf5.setText("One Way");

tf7.setText("One");

tf6.setText("Nil");

tf11.setText("Jan Shatabdi");

}

**else** **if**((rb2.isSelected()) && (rb6.isSelected()) && cb1.getSelectedItem().equals("Howrah") && cb2.getSelectedItem().equals("Patna"))

{

totalTax = (tax \* fare2)/100;

String tTax = String.*format*("Rs. %.2f", totalTax);

tf1.setText(tTax);

String Subtotal = String.*format*("Rs. %.2f", fare2/2);

tf2.setText(Subtotal);

String Total = String.*format*("Rs. %.2f",fare2/2 + totalTax);

tf3.setText(Total);

tf13.setText(Total);

// ==============net side==================//

tf4.setText("SL");

tf5.setText("One Way");

tf7.setText("Nil");

tf6.setText("One");

tf11.setText("Jan Shatabdi");

}

**else** **if**((rb1.isSelected()) && (rb7.isSelected()) && cb1.getSelectedItem().equals("Howrah") && cb2.getSelectedItem().equals("Patna"))

{

totalTax = (tax \* fare2)/100;

String tTax = String.*format*("Rs. %.2f", totalTax);

tf1.setText(tTax);

String Subtotal = String.*format*("Rs. %.2f", fare2\*1.5);

tf2.setText(Subtotal);

String Total = String.*format*("Rs. %.2f",fare2\*1.5 + totalTax);

tf3.setText(Total);

tf13.setText(Total);

// ==============net side==================//

tf4.setText("AC");

tf5.setText("One Way");

tf7.setText("One");

tf6.setText("Nil");

tf11.setText("Jan Shatabdi");

}

**else** **if**((rb1.isSelected()) && (rb6.isSelected()) && cb1.getSelectedItem().equals("Howrah") && cb2.getSelectedItem().equals("Patna"))

{

totalTax = (tax \* fare2)/100;

String tTax = String.*format*("Rs. %.2f", totalTax);

tf1.setText(tTax);

String Subtotal = String.*format*("Rs. %.2f", fare2/1.5);

tf2.setText(Subtotal);

String Total = String.*format*("Rs. %.2f",fare2/1.5 + totalTax);

tf3.setText(Total);

tf13.setText(Total);

// ==============net side==================//

tf4.setText("AC");

tf5.setText("One Way");

tf7.setText("Nil");

tf6.setText("One");

tf11.setText("Jan Shatabdi");

}

**if**((rb2.isSelected()) && (rb7.isSelected()) && cb1.getSelectedItem().equals("Kolkata") && cb2.getSelectedItem().equals("Delhi"))

{

totalTax = (tax \* fare)/100;

String tTax = String.*format*("Rs. %.2f", totalTax);

tf1.setText(tTax);

String Subtotal = String.*format*("Rs. %.2f", fare);

tf2.setText(Subtotal);

String Total = String.*format*("Rs. %.2f",fare + totalTax);

tf3.setText(Total);

tf13.setText(Total);

// ==============net side==================//

tf4.setText("SL");

tf5.setText("One Way");

tf7.setText("One");

tf6.setText("Nil");

tf11.setText("ANVT Express");

}

**else** **if**((rb2.isSelected()) && (rb6.isSelected()) && cb1.getSelectedItem().equals("Kolkata") && cb2.getSelectedItem().equals("Delhi"))

{

totalTax = (tax \* fare)/100;

String tTax = String.*format*("Rs. %.2f", totalTax);

tf1.setText(tTax);

String Subtotal = String.*format*("Rs. %.2f", fare/2);

tf2.setText(Subtotal);

String Total = String.*format*("Rs. %.2f",fare/2 + totalTax);

tf3.setText(Total);

tf13.setText(Total);

// ==============net side==================//

tf4.setText("SL");

tf5.setText("One Way");

tf7.setText("Nil");

tf6.setText("One");

tf11.setText("ANVT Express");

}

**else** **if**((rb1.isSelected()) && (rb7.isSelected()) && cb1.getSelectedItem().equals("Kolkata") && cb2.getSelectedItem().equals("Delhi"))

{

totalTax = (tax \* fare)/100;

String tTax = String.*format*("Rs. %.2f", totalTax);

tf1.setText(tTax);

String Subtotal = String.*format*("Rs. %.2f", fare\*1.5);

tf2.setText(Subtotal);

String Total = String.*format*("Rs. %.2f",fare\*1.5 + totalTax);

tf3.setText(Total);

tf13.setText(Total);

// ==============net side==================//

tf4.setText("AC");

tf5.setText("One Way");

tf7.setText("One");

tf6.setText("Nil");

tf11.setText("ANVT Express");

}

**else** **if**((rb1.isSelected()) && (rb6.isSelected()) && cb1.getSelectedItem().equals("Kolkata") && cb2.getSelectedItem().equals("Delhi"))

{

totalTax = (tax \* fare)/100;

String tTax = String.*format*("Rs. %.2f", totalTax);

tf1.setText(tTax);

String Subtotal = String.*format*("Rs. %.2f", fare/1.5);

tf2.setText(Subtotal);

String Total = String.*format*("Rs. %.2f",fare/1.5 + totalTax);

tf3.setText(Total);

tf13.setText(Total);

// ==============net side==================//

tf4.setText("AC");

tf5.setText("One Way");

tf7.setText("Nil");

tf6.setText("One");

tf11.setText("ANVT Express");

}

**if**((rb2.isSelected()) && (rb7.isSelected()) && cb1.getSelectedItem().equals("Kolkata") && cb2.getSelectedItem().equals("Patna"))

{

totalTax = (tax \* fare2)/100;

String tTax = String.*format*("Rs. %.2f", totalTax);

tf1.setText(tTax);

String Subtotal = String.*format*("Rs. %.2f", fare2);

tf2.setText(Subtotal);

String Total = String.*format*("Rs. %.2f",fare2 + totalTax);

tf3.setText(Total);

tf13.setText(Total);

// ==============net side==================//

tf4.setText("SL");

tf5.setText("One Way");

tf7.setText("One");

tf6.setText("Nil");

tf11.setText("Patna-Kolkata Express");

}

**else** **if**((rb2.isSelected()) && (rb6.isSelected()) && cb1.getSelectedItem().equals("Kolkata") && cb2.getSelectedItem().equals("Patna"))

{

totalTax = (tax \* fare2)/100;

String tTax = String.*format*("Rs. %.2f", totalTax);

tf1.setText(tTax);

String Subtotal = String.*format*("Rs. %.2f", fare2/2);

tf2.setText(Subtotal);

String Total = String.*format*("Rs. %.2f",fare2/2 + totalTax);

tf3.setText(Total);

tf13.setText(Total);

// ==============net side==================//

tf4.setText("SL");

tf5.setText("One Way");

tf7.setText("Nil");

tf6.setText("One");

tf11.setText("Patna-Kolkata Express");

}

**else** **if**((rb1.isSelected()) && (rb7.isSelected()) && cb1.getSelectedItem().equals("Kolkata") && cb2.getSelectedItem().equals("Patna"))

{

totalTax = (tax \* fare2)/100;

String tTax = String.*format*("Rs. %.2f", totalTax);

tf1.setText(tTax);

String Subtotal = String.*format*("Rs. %.2f", fare2\*1.5);

tf2.setText(Subtotal);

String Total = String.*format*("Rs. %.2f",fare2\*1.5 + totalTax);

tf3.setText(Total);

tf13.setText(Total);

// ==============net side==================//

tf4.setText("AC");

tf5.setText("One Way");

tf7.setText("One");

tf6.setText("Nil");

tf11.setText("Patna-Kolkata Express");

}

**else** **if**((rb1.isSelected()) && (rb6.isSelected()) && cb1.getSelectedItem().equals("Kolkata") && cb2.getSelectedItem().equals("Patna"))

{

totalTax = (tax \* fare2)/100;

String tTax = String.*format*("Rs. %.2f", totalTax);

tf1.setText(tTax);

String Subtotal = String.*format*("Rs. %.2f", fare2/1.5);

tf2.setText(Subtotal);

String Total = String.*format*("Rs. %.2f",fare2/1.5 + totalTax);

tf3.setText(Total);

tf13.setText(Total);

// ==============net side==================//

tf4.setText("AC");

tf5.setText("One Way");

tf7.setText("Nil");

tf6.setText("One");

tf11.setText("Patna-Kolkata Express");

}

}

});

btnTotal.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent arg0) {

//time//

Calendar timer = Calendar.*getInstance*();

timer.getTime();

//date//

SimpleDateFormat Tdate = **new** SimpleDateFormat("dd-MM-YYYY");

tf10.setText(Tdate.format(timer.getTime()));

//tf8.setText("Howrah \*");

tf8.setText((String)cb1.getSelectedItem()+ " \*");

tf9.setText((String)cb2.getSelectedItem()+ " \*");

//random no. generator//

**int** num1;

String q1 ="";

num1 = 1325 + (**int**) (Math.*random*()\*4238);

q1 += num1 + 1325;

tf.setText(q1);

**int** num2;

String q2="";

num1=1325+(**int**)(Math.*random*()\*4238);

q1 +=num1+1325;

tf12.setText(q1);

//=================================//

tf14.setText("Any");

}

});

btnReset.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

tf1.setText(**null**);

tf2.setText(**null**);

tf3.setText(**null**);

rb2.setSelected(**false**);

rb1.setSelected(**false**);

rb7.setSelected(**false**);

rb6.setSelected(**false**);

cb1.setSelectedItem("Source");

cb2.setSelectedItem("Destination");

tf4.setText(**null**);

tf5.setText(**null**);

tf7.setText(**null**);

tf6.setText(**null**);

tf8.setText(**null**);

tf14.setText(**null**);

tf9.setText(**null**);

tf11.setText(**null**);

tf.setText(**null**);

tf13.setText(**null**);

tf12.setText(**null**);

tf10.setText(**null**);

}

});

JLabel lblAdult = **new** JLabel("ADULT");

lblAdult.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

tf6 = **new** JTextField();

tf6.setColumns(10);

JLabel lblChild = **new** JLabel("CHILD");

lblChild.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

tf7 = **new** JTextField();

tf7.setColumns(10);

JLabel lblTrainNo = **new** JLabel("TRAIN NO");

lblTrainNo.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

tf = **new** JTextField();

tf.setColumns(10);

JLabel lblDepartureTime = **new** JLabel("DATE");

lblDepartureTime.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

tf10 = **new** JTextField();

tf10.setColumns(10);

JLabel lblTrainName = **new** JLabel("TRAIN NAME");

lblTrainName.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

tf11 = **new** JTextField();

tf11.setColumns(10);

JPanel panel = **new** JPanel();

panel.setBackground(Color.***ORANGE***);

JLabel lblSleeper = **new** JLabel("CLASS");

lblSleeper.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

tf4 = **new** JTextField();

tf4.setColumns(10);

JLabel lblTicket = **new** JLabel("TICKET");

lblTicket.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

tf5 = **new** JTextField();

tf5.setColumns(10);

JLabel lblSubtotal = **new** JLabel("SUBTOTAL");

lblSubtotal.setFont(**new** Font("Algerian", Font.***BOLD***, 14));

tf2 = **new** JTextField();

tf2.setColumns(10);

JLabel lblFrom\_1 = **new** JLabel("FROM");

lblFrom\_1.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

tf8 = **new** JTextField();

tf8.setColumns(10);

JLabel lblTo\_1 = **new** JLabel("TO");

lblTo\_1.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

tf9 = **new** JTextField();

tf9.setColumns(10);

JLabel lblPrice\_1 = **new** JLabel("PRICE");

lblPrice\_1.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

tf13 = **new** JTextField();

tf13.setColumns(10);

JLabel lblRoute = **new** JLabel("ROUTE");

lblRoute.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

tf14 = **new** JTextField();

tf14.setColumns(10);

JLabel lblTicketNo = **new** JLabel("TICKET NO");

lblTicketNo.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

tf12 = **new** JTextField();

tf12.setColumns(10);

JSeparator separator = **new** JSeparator();

JSeparator separator\_1 = **new** JSeparator();

JList list = **new** JList();

JSeparator separator\_2 = **new** JSeparator();

JSeparator separator\_3 = **new** JSeparator();

JSeparator separator\_4 = **new** JSeparator();

JLabel lblTrainAndPassenger = **new** JLabel("TRAIN AND PASSENGER DETAILS");

lblTrainAndPassenger.setForeground(**new** Color(255, 140, 0));

lblTrainAndPassenger.setFont(**new** Font("Tahoma", Font.***BOLD***, 17));

JSeparator separator\_5 = **new** JSeparator();

JSeparator separator\_6 = **new** JSeparator();

JLabel lblName\_1 = **new** JLabel("NAME");

lblName\_1.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

lblName\_1.setBounds(24, 60, 60, 17);

panel.add(lblName\_1);

JButton btnBook = **new** JButton("BOOK");

btnBook.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

btnBook.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

String cname=String.*valueOf*(textField.getText().toString());

String from=String.*valueOf*(tf16.getText().toString());

String to=String.*valueOf*(tf17.getText().toString());

String train\_name=String.*valueOf*(tf18.getText().toString());

String train\_class=String.*valueOf*(tf19.getText().toString());

String price=String.*valueOf*(tf20.getText().toString());

**try**

{

**int** i=0;

**int** c=1;

Class.*forName*("com.mysql.jdbc.Driver");

Connection con=DriverManager.*getConnection*("jdbc:mysql://localhost:3306/railway","root","");

PreparedStatement ps = con.prepareStatement("insert into confirm values(?,?,?,?,?,?)");

ps.setString(1, cname);

ps.setString(2, from);

ps.setString(3, to);

ps.setString(4, train\_name);

ps.setString(5, train\_class);

ps.setString(6,price);

i = ps.executeUpdate();

c++;

**if** (i > 0)

{

*name*=cname;

*price1*=price;

JOptionPane.*showMessageDialog*(btnBook, "Booking Successful, Your seat no. is : "+*count* );

payment ob=**new** payment(cname,price);

ob.setVisible(**true**);

dispose();

*count*--;

}

}

**catch** (Exception ex)

{

//System.out.println(ex);

ex.printStackTrace();

}

}

});

btnBook.setBounds(104, 357, 92, 23);

panel.add(btnBook);

textField = **new** JTextField();

textField.setBounds(104, 60, 86, 20);

panel.add(textField);

textField.setColumns(10);

JLabel lblNewLabel = **new** JLabel("FROM");

lblNewLabel.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

lblNewLabel.setBounds(24, 104, 60, 14);

panel.add(lblNewLabel);

tf16 = **new** JTextField();

tf16.setBounds(104, 103, 86, 20);

panel.add(tf16);

tf16.setColumns(10);

JLabel lblTo = **new** JLabel("TO");

lblTo.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

lblTo.setBounds(24, 146, 46, 14);

panel.add(lblTo);

textField\_2 = **new** JTextField();

textField\_2.setBounds(128, 81, -58, -1);

panel.add(textField\_2);

textField\_2.setColumns(10);

tf17 = **new** JTextField();

tf17.setBounds(104, 145, 86, 20);

panel.add(tf17);

tf17.setColumns(10);

JLabel lblTrainName\_1 = **new** JLabel("TRAIN NAME\r\n");

lblTrainName\_1.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

lblTrainName\_1.setBounds(10, 193, 108, 14);

panel.add(lblTrainName\_1);

tf18 = **new** JTextField();

tf18.setBounds(104, 192, 86, 20);

panel.add(tf18);

tf18.setColumns(10);

JLabel lblClass = **new** JLabel("CLASS");

lblClass.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

lblClass.setBounds(21, 242, 74, 14);

panel.add(lblClass);

tf19 = **new** JTextField();

tf19.setBounds(104, 241, 86, 20);

panel.add(tf19);

tf19.setColumns(10);

JLabel lblPrice = **new** JLabel("PRICE");

lblPrice.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

lblPrice.setBounds(24, 296, 46, 14);

panel.add(lblPrice);

tf20 = **new** JTextField();

tf20.setBounds(104, 295, 86, 20);

panel.add(tf20);

tf20.setColumns(10);

JButton btnProceed = **new** JButton("PROCEED");

btnProceed.setForeground(**new** Color(25, 25, 112));

btnProceed.setBounds(10, 358, 84, 23);

panel.add(btnProceed);

btnProceed.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

textField.setText(tf15.getText().toString());

tf16.setText(tf8.getText().toString());

tf17.setText(tf9.getText().toString());

tf18.setText(tf11.getText().toString());

tf19.setText(tf4.getText().toString());

tf20.setText(tf13.getText().toString());

String name=String.*valueOf*(textField.getText().toString());

String from=String.*valueOf*(tf16.getText().toString());

String to=String.*valueOf*(tf17.getText().toString());

String train\_name=String.*valueOf*(tf18.getText().toString());

String train\_class=String.*valueOf*(tf19.getText().toString());

String price=String.*valueOf*(tf20.getText().toString());

}

});

btnProceed.setFont(**new** Font("Arial", Font.***BOLD***, 10));

JLabel lblConfirmDetails = **new** JLabel("CONFIRM DETAILS");

lblConfirmDetails.setForeground(**new** Color(85, 107, 47));

lblConfirmDetails.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

lblConfirmDetails.setBounds(24, 22, 190, 14);

panel.add(lblConfirmDetails);

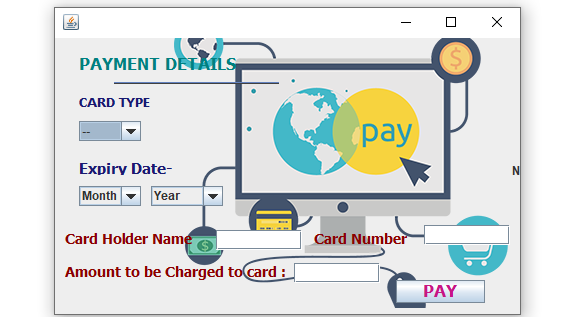
contentPane.setLayout(gl\_contentPane);

}

}

**PAYMENT DETAILS**

**SNAPSHOT**

****

**Code**

**public** **class** payment **extends** JFrame {

**private** JPanel contentPane;

**private** JTextField textField;

**private** JTextField textField\_1;

**private** JTextField textField\_2;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

payment frame = **new** payment(PassengerInformation.*name*,PassengerInformation.*price1*);

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** payment(String name,String price) {

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 481, 315);

contentPane = **new** JPanel();

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel lblName = **new** JLabel("Card Holder Name");

lblName.setForeground(**new** Color(139, 0, 0));

lblName.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

lblName.setBounds(10, 193, 141, 14);

contentPane.add(lblName);

textField = **new** JTextField();

textField.setBounds(161, 192, 86, 20);

contentPane.add(textField);

textField.setColumns(10);

textField.setText(PassengerInformation.*name*);

JLabel lblPrice = **new** JLabel("Amount to be Charged to card :");

lblPrice.setForeground(**new** Color(139, 0, 0));

lblPrice.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

lblPrice.setBounds(10, 218, 231, 31);

contentPane.add(lblPrice);

textField\_1 = **new** JTextField();

textField\_1.setBounds(239, 225, 86, 20);

contentPane.add(textField\_1);

textField\_1.setColumns(10);

textField\_1.setText(PassengerInformation.*price1*);

JButton btnPay = **new** JButton("PAY");

btnPay.setForeground(**new** Color(199, 21, 133));

btnPay.setFont(**new** Font("Tahoma", Font.***BOLD***, 16));

btnPay.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent arg0) {

JOptionPane.*showMessageDialog*(**null**, "payment Sucessful" );

login lo=**new** login();

lo.setVisible(**true**);

}

});

btnPay.setBounds(341, 242, 89, 23);

contentPane.add(btnPay);

JLabel lblCardType = **new** JLabel("CARD TYPE");

lblCardType.setForeground(**new** Color(25, 25, 112));

lblCardType.setFont(**new** Font("Tahoma", Font.***BOLD***, 13));

lblCardType.setBounds(24, 57, 86, 14);

contentPane.add(lblCardType);

JComboBox comboBox = **new** JComboBox();

comboBox.setModel(**new** DefaultComboBoxModel(**new** String[] {"--", "DEBIT", "CREDIT"}));

comboBox.setBounds(24, 83, 62, 20);

contentPane.add(comboBox);

JLabel lblPaymentDetails = **new** JLabel("PAYMENT DETAILS");

lblPaymentDetails.setForeground(**new** Color(0, 128, 128));

lblPaymentDetails.setFont(**new** Font("Tahoma", Font.***BOLD***, 16));

lblPaymentDetails.setBounds(24, 11, 165, 31);

contentPane.add(lblPaymentDetails);

JLabel lblExpiary = **new** JLabel("Expiry Date-");

lblExpiary.setForeground(**new** Color(25, 25, 112));

lblExpiary.setFont(**new** Font("Tahoma", Font.***BOLD***, 15));

lblExpiary.setBounds(24, 123, 96, 14);

contentPane.add(lblExpiary);

JComboBox comboBox\_1 = **new** JComboBox();

comboBox\_1.setModel(**new** DefaultComboBoxModel(**new** String[] {"Month", "01", "02", "03", "04", "05", "06", "07", "08", "09", "10"}));

comboBox\_1.setBounds(24, 148, 62, 20);

contentPane.add(comboBox\_1);

JComboBox comboBox\_2 = **new** JComboBox();

comboBox\_2.setModel(**new** DefaultComboBoxModel(**new** String[] {"Year", "2020", "2021", "2022", "2023", "2024", "2025"}));

comboBox\_2.setBounds(96, 148, 72, 20);

contentPane.add(comboBox\_2);

JLabel lblCardNumber = **new** JLabel("Card Number");

lblCardNumber.setForeground(**new** Color(139, 0, 0));

lblCardNumber.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

lblCardNumber.setBounds(259, 193, 123, 14);

contentPane.add(lblCardNumber);

textField\_2 = **new** JTextField();

textField\_2.setBounds(369, 187, 86, 20);

contentPane.add(textField\_2);

textField\_2.setColumns(10);

JSeparator separator = **new** JSeparator();

separator.setBackground(Color.***DARK\_GRAY***);

separator.setBounds(59, 44, 165, 2);

contentPane.add(separator);

JLabel lblNewLabel = **new** JLabel("New label");

lblNewLabel.setIcon(**new** ImageIcon("C:\\Users\\Sujoy PC\\Desktop\\100.png"));

lblNewLabel.setBounds(119, -6, 465, 276);

contentPane.add(lblNewLabel);

}

}

**IMPLEMENTATION AND TESTING**

TractIn this emerging world of computers all most all manual systemautomated and computerized but maximum of them are so complex and acommon user is unable to operate that software system. We are developing

the “Online Railway Reservation System” to model the present system and

to remove the drawbacks.This project explores how computer technology can be used to solvethe problem of user. We specifically props way to computerize theadministration and engineering offices of Indian Railway Online. A highspeed communication network needs to be developed for interconnecting theoffices of railways. This same communication network can be used toprovide high speed internet connection to the public and leas out network capacity to internet service providers.

**OBJECTIVES OF TESTING**

The objective our test plan is to find and report as many bugs as possible to improve the integrity of our program. Although exhaustive testing is not possible, we will exercise a broad range of tests to achieve our goal. Our user interface to utilize these functions is designed to be user-friendly and provide easy manipulation of the tree. The application will only be used as a demonstration tool, but we would like to ensure that it could be run from a variety of platforms with little impact on performance or usability.

Process Overview

The following represents the overall flow of the testing process:

 Identify the requirements to be tested. All test cases shall be derived using the current Program Specification.

 Identify which particular test(s) will be used to test each module.

 Review the test data and test cases to ensure that the unit has been thoroughly verified and that the test data and test cases are adequate to verify proper operation of the unit.

 Identify the expected results for each test.

 Document the test case configuration, test data, and expected results.

 Perform the test(s).

 Document the test data, test cases, and test configuration used during the testing process. This information shall be submitted via the Unit/System Test Report (STR).

 Successful unit testing is required before the unit is eligible for component integration/system testing.

 Unsuccessful testing requires a Bug Report Form to be generated. This document shall describe the test case, the problem encountered, its possible cause, and the sequence of events that led to the problem. It shall be used as a basis for later technical analysis.

 Test documents and reports shall be submitted. Any specifications to be reviewed, revised, or updated shall be handled immediately.

**TESTING STEPS**

**A test case is a document that describes an input, action, or event and expected response, to determine if a feature of an application is working correctly. A test case should contain particular such as test case identifier, test condition, input data Requirement expected results. The process of developing test cases can help find problems in the requirement or design of an application, since it requires completely thinking through the operation of the application.**

**Unit Testing**

**Unit testing** is a level of software testing and software development process which test source code of individual unit or module of software product. **Unit testing** test as smallest testable part of an application.

Primary goal of unit testing is to take the smallest isolated piece of software application and check whether they behave exactly what user want.

After isolation of software application stub, driver and mock object is used to test by unit testing.

It is used to validate each unit or component of the software perform as designed.

All modules are tested individually and find out the errors before integrating all modules to improve quality software application.

It reduces errors and enhances test coverage by testing each unit of software application.

It tests **“Is the error due to defect in units.”**

It is first level of testing and done prior integration testing.

It follows white box testing.

It is tested by developer itself or sometimes independent testers.

Unit testing done by automated or manually.

It is also called as module testing or component testing.

**Integration Testing**

1. GUI testing  
2. Navigation testing  
3. Functional testing  
4. Non - functional testing (Performance Security)  
5. Data validation testing  
6. Negative testing  
7. Data Base Testing

Test cases for Railway Registration:  
1. Check whether there is a train for your destination.  
2. Check the availability of seats on the required date.  
3. Check whether the available seats for required coach that is A/c or Sleeper coaches.

Test Scenario: Train search with all route stations  
   
1. Enter source and destination codes or source and  destination corresponding trains are displaying or not  
2. Enter train number and date source and destination with all stations are displaying or not   
    
Scenario: check availabilityof seats for required date  
1. Search seats for Ac ,Sleeper ,seat  
2. Check upper middle lower births   
3. Ticket fare for corresponding criteria  
  Booking :  
  Book the tickets with corresponding criteria  
  by cash or card  
  any concetion for senior citizen fi applicable

Scenario:  
Print the ticket   
same for cancelation of ticket

1. Check whether you have money.  
2. Check the train is available & you have to go to the same destination.  
3. Check working hrs of booking office

Step1: Verify there is a train available to the destination.  
Step2: Verify the seats are available on the particular date.  
Step3: Verify the seats available of the coaches like sitting or berth and check the fare of the particular system

Validation

At the culmination of the integration testing, Software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test begin in validation testing. Validation testing can be defined in many ways, but a simple definition is that the validation succeeds when the software functions in a manner that is expected by the customer. After validation test has been conducted, one of the three possible conditions exists.

 The function or performance characteristics confirm to specification and are accepted.  A deviation from specification is uncovered and a deficiency lists is created.  Proposed system under consideration has been tested by using validation test and found to be working satisfactory.

**User Acceptance Testing**

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes whenever required.

This is done in regard to the following point:

* Input Screen Design
* Output Screen Design
* Format of reports and other outputs.

**FUNCTIONAL TESTING**

These are the points concerned during the stress test:

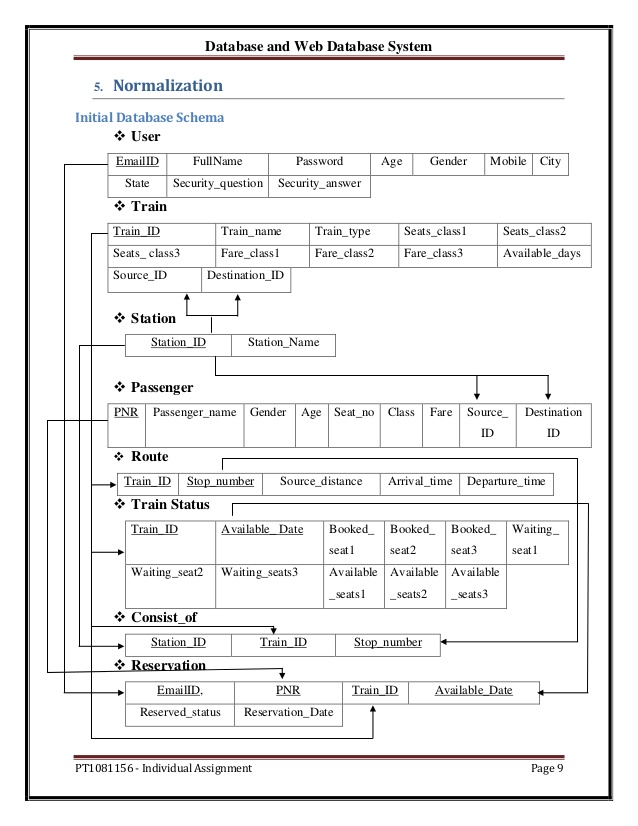
* Nominal input: character is in putted in the place of digits and the system has to flash the message "Data error"
* Boundary value analysis: exhaustive test cases have designed to create an output report that produces the maximum (and minimum) allowable number of table entries.

***SOFTWARE SYSTEM aTTRIBUTES***

**DATABASE SECURITY**

Data and applications reside everywhere, on the network, endpoints, and in the constantly evolving cloud ecosystem. As a result, organizations are now faced with securing a multitude of applications, users, devices, and infrastructures.   
Get a head start in protecting your data from common threats to SaaS apps, such as:

* Accidental data exposure.
* Insider threats like disgruntled employees.
* Credential theft.



**SYSTEM SECURITY**

Watch a detailed demo of the new SaaS security capabilities in PAN-OS® 8.0 and Aperture, including:

* Enhanced SaaS visibility and reporting
* Creation of SaaS security policies for unsanctioned and sanctioned applications
* Prevention of malware with WildFire integration
* Automated risk remediation

**LIMITATIONS**

.Data Redundancy

.Difficulty in accessing the data

**CONCLUSION**

IT USES ADVANCED FEATURES OF MICRO CONTROLLER WITH IR SENSORS AND THE METAL DETECTORS,PROVES TO BE EFFECTIVE IN ACHIEVING THE OBJECTIVES

IT IS APPLICABLEAT EVERY ASPECT OF THE RAILWAYS FOR UNINTERRUPTIBLE SERVICE

SAVING HUMAN LIFE PROTECTION AGAINST ACCIDENTS AND THE COMMUNICABLE MECHANICAL SYSTEMS ARE THE SALIENT FEATURES AND THE ADDED ADVANTAGE OF THIS PROJECT.

**FUTURE SCOPE AND FURTHER ENHANCEMENTS**

### In future, I would like to keep working on this project and make new additions to provide users with more advanced features and more detailed information and to make the system more flexible. I have set our sights on the following additions in future –

###  Railway Reservation Information Management

###  Managing Reports

###  Feedbacks Management

**BIBLIOGRAPHY**

* <https://www.google.com>
* <https://www.slideshare.com>
* <https://www.tutorialspoint.com>
* <https://www.youtube.com>

***THANK YOU***