ONLINE TICKETING SYSTEM

By

Sunny Sharma – 081412

Akash Chaudhary - 081419

Purnima Choudhry – 081414

Under the Supervision of

Mr. Nitin Rakesh

विद्या तत्व ज्योतिसम:

JAYPEE UNIVERSITY OF
INFORMATION TECHNOLOGY

Submitted in partial fulfilment of the Degree of

Bachelor of Technology

In

Information Technology

DEPARTMENT OF INFORMATION TECHNOLOGY

Jaypee University of Information and Technology

Waknaghat, Solan – 173234, Himachal Pradesh

TABLE OF CONTENTS

II

CERTIFICATE

This is to certify that the project report entitled **"Online Ticketing System",** submitted by **Sunny Sharma, Akash Chaudhary, Purnima Choudhry** in partial fulfillment for the award of degree of Bachelor of Technology in Information Technology to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor

Supervisor's Name : Mr. Nitin Rakesh

Designation: Sr. Lecturer.

Date: 30/05/12.

ACKNOWLEDGEMENT

We express our heartfelt gratitude to all the people who helped us in the successful completion of this project.

First of all, we thank the almighty god who showered his immense blessings on us, which helped us to complete this project successfully.

Mr. Nitin Rakesh , our project supervisor for his constant guidance throughout our project work and for giving us the accessory environment to acquire knowledge and skill. We would also like to thank the Teaching and Non teaching staff of the Computer Science Department for sharing their knowledge with us.

We are also grateful to Dr. S.P Ghrera, Head of Computer Science Department, Jaypee University of Information and Technology for his valuable suggestions and guidance during the execution of this project. We also express our whole hearted thanks to our colleagues for their encouragement to bring this project to a successful completion.

Name of the student

Sunny Sharma– 081412

Akash Chaudhary - 081419

Purnima Choudhry- 081414

Date: 30th May 2012.

IV

SUMMARY

Online Ticketing System offers a web application for the passengers to reserve their train tickets from the comfort of their office or home without having to stand in long queues at railway stations or having to take the trouble to go to a travel agent.

The special benefit of using this application is that it eliminates the need to take ticket printouts and carry them while travelling for verification. A unique code is available with the passenger in his mobile device which is verified and acts as his ticket.

The objective of this project is to help the users enquire for training schedule, fare enquiry, PNR enquiry, train/fare accommodation, online reservation, customer services like SMS services, email confirmations and rules and information regarding the processes. It also offers railway relevant services with mobile application which saves a lot of time and saves money to board at the particular booking stations. Passengers can get to know their PNR status on mobiles. Passengers have everything in their own hands; they don't need any official or known person for travelling. This is an online application which of course is user friendly and it will also make people more familiar and closer to avail railway services which will avoid much traffic problems too.

We have used JavaScript as the client side scripting language, JSP and HTML for designing the web pages.

The database engine has been embedded using an active MySQL server, a browser which acts as a client and an Apache HTTP server. The project may be extended in future as a generic application to book tickets for movies, bus ticket reservations, flight bookings and many more applications.

Project Members

Sunny | AChaudhary

Sunny Sharma | Akash Chaudhary | Purnima Choudhry

Supervisor

Mr. Nitin Rakesh

Date: 30/05/12

V

LIST OF FIGURES

VI

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Abstract

Today computers have become a part of life all over the world. The project "**ONLINE TICKETING SYSTEM**" is being proposed for booking the railways tickets online. This system will provide a user account for every user. The user can login the system and book tickets instantly and cancel the same. He is also able to access his/her account from anywhere just by providing the correct login id and password.

The administrator should have the ability to modify train schedule, modify the ticket costs etc. with the privilege to view the reports of the all the passengers.

The proposed system is an improvement of the existing online railway reservation with additional features like SMS confirmation to the user and timely updations of the waiting status. The passenger does not need to carry tickets while travelling.

1.2 Motivation

This project concerns the comfort ability and flexibility to the user to develop an online reservation system for users to book tickets online. It aims to model the existing railway systems and to provide a comprehensive set of features to enhance the operational limits such as the number of users of such systems.

In future, this application may be extended as a generic e-ticketing application for booking of tickets by users in other departments like booking of movie tickets, bus reservation, airline reservation, and many more such systems. Moreover, modifications are suggested for greater efficiency while ticket booking is performed.

1.3 Organization of Thesis

The remainder of this project report is divided into Seven Chapters. They are as follows: -

Chapter 2 covers the literature review on the existing railway reservation systems and our proposed system features.

Chapter 3 describes the requirement engineering process with the analysis of requirements and feasibility study.

Chapter 4 describes the performance evaluation of the developed system and the hardware and software requirements.

Chapter 5 discusses the design criteria of hardware system, the software layout and the database

Chapter 6 provides the implementation of the website and suggestions for further development to improve the system

Chapter 7 concludes the report and outlines the contributions of the project..

CHAPTER 2

LITERARY SURVEY

2.1 Referential Work

Indian railways are one of the largest and busiest railway systems in the world. Every year it transports about 6 billion people and 7 billion tones of freight. Indian Railways is also one of the earliest government departments who introduced computers in the country and worked towards areas of computerization like Passenger Reservation System (PRS), and Freight Operations Information System (FOIS).

Indian Railways now provides the passengers the facility of online reservation wherein reservation seeking passengers can reserve the tickets online over the internet. Earlier, passengers had to wait at ticket counters to book tickets. Later, travel agents checked the tickets with the ticket counters which turned out to be a lengthy and very tedious process.

We all are aware of Indian Railway Catering and Tourism Corporation (IRCTC) that provides an online booking facility through its web site. Customers can book tickets online with a credit card, debit card or a net banking account. Thus, with the development of www.irctc.co.in many additional services like checking of PNR status and booking and cancellation of tickets instantly proved to be very beneficial and improved the railway services.

This is simply an interface for making reservations, but it moreover employs a huge database underneath holding support for many other services. It is a hugely popularized website that not only offers the facility for online reservation of tickets but also, provides the plethora of information regarding the railways for the passengers.

Some of the salient features provided by this website are:

- Find trains
- Checking the seat availability
- Booking a train ticket online for all classes
- Online payment available with multiple payment options such as Credit Card, Netbanking, EzCash

- Printing online tickets (e-tickets)
- Cancellation of e-tickets (subjected to terms and conditions of the site)
- The person in whose name the online booking has been made needs to prove his identity and may need t produce an identity document during the course of his journey. These identity cards must be issued by a reputed organization in the Central/State Government and include: PAN Card, Voter ID Card and Drivers license
- Passengers travelling on tickets that have been booked online need to take a print out of the e ticket and keep it with themselves during the course of the journey an produce it when required.
- Implementations of the project to be developed are already in industrial use in the form of 'www.irctc.co.in', indianrail.gov.in etc. Hence, these websites have been used for suggesting improvements in the design, performance and to offer greater usability.
- Apart from the industrial application, it is also a research oriented project that involves the task of performance evaluation of different database designs, for efficiency, is in this spirit.

2.2 Proposed System

The proposed system will be available to the users online and can be logged in as a user or as the administrator. The system will be developed in such a way that anyone who wants to check the train timings or the PNR status can do it very easily by logging on to the website from their place or a nearby cyber cafe. Also those who are interested to book their tickets can book online and avail the SMS service to receive alerts of their reservation status from time to time. This will discard the need to access mails regularly.

2.2.1 Features of the proposed system

- Fully automatic mailing System to inform a reservation has been modified.
- Sending emails and SMS alerts for recent updates
- The first conflicting reservation is queued to the waiting list. Whenever the resources become available, the reservation will be automatically accepted and the user will be notified via email and SMS.
- The system will generate a ticket code on each reservation that will be unique and provided to the user.

- The unique code will be sent to the user via SMS on his mobile and via email to his email id
- Waiting list status alerts to user via SMS and email.

2.2.2 Advantages over existing system:

- Saves time and resources
- No need to carry tickets while travelling
- No need to access mails repeatedly
- User's job restricted to booking tickets

CHAPTER 3

REQUIREMENT ENGINEERING PROCESS

3.1 Problem Statement

The main objective of developing this online railway reservation application is to reserve, cancel and review railway tickets online.

It also intends to send a confirmation of the reservation to the user via SMS and email.

3.2 Feasibility Study

We use My SQL as the backend and JSP as the front-end tool to develop this application. We can make the customers train journey as comfort as possible by providing them a unique code and eliminating the need to carry tickets. It is also beneficial to record train details and provide the administrator the benefit to make updations easily. Thus it is worth carrying out the entire requirement engineering process and system development.

3.3 Requirements Definition

The train details consists of train name, train no, destination, source, number of passengers travelling and number of seats available.

The reservation details consist of passengers name, age, sex, source, email id, contact no, PNR, and seat number.

The administrator can:

- Add Train
- View Train List
- Add Class Details
- Get Passenger List
- Add user

- View User List
- Cancel Ticket

The user can:

- Search Train
- View PNR Status
- Cancel Ticket
- View Booked History
- Change Password

3.4 Requirement Analysis

With the development of this project it will be easier for the railway system to maintain the train schedules and make the customer's journey as comfortable as possible. Moreover it would help in reducing the financial crisis of the system. At the client's side, making reservations and cancellations anywhere, anytime with the internet is possible.

The requirement analysis shows that the integration of this software with existing system is more complex in the limited schedule.

CHAPTER 4

PROJECT DESCRIPTION

4.1 Evaluation Parameters

4.1.1 Hardware Requirements:

Processor	:	Pentium4
RAM	:	1 GB
Mouse	:	Standard Mouse
Keyboard	:	Logitech Keyboard
Processor Speed	:	2.4GHZ

4.1.2 Software Requirements:

Operating System.	:	Windows XP
Developing Tool	:	JSP, Javascript
Database	:	My SQL

4.1.3 Display Mode:

Color Quality	:	Highest [32 BIT]
Screen Resolution	:	1024 by 768 Pixels

4.1.4 Development and operating environment:

This project is developed by

- Using WIN 32 platform.
- Coding By JSP
- Slide representation by MS power point.
- Documentation by MS word.

4.1.5 Compatible Browsers

- Internet Explorer v6.0 and above

4.2 Functional Requirements

Functional requirements for this project are :

Function 1: "Train Details"

The administrator must be able to view the details of the train that is the train name, train number, source, destination, etc.

Description:

When needed the administrator must be able to update the train details for the convenience of the customer.

INPUT: Details of train (internal and external).

SOURCE: Trains source

OUTPUT: Number of passengers.

DESTINATION: Area to be reached.

ACTION: To update the train details we require the train name, number, destination, timing, number of seat available.

REQUIRES: The database having the details of the trains.

PRECONDITION: None

POSTCONDITION: None.

SIDE EFFECTS: None

Function 2: "Railway Reservation Details"

The staff of railway system must be able to view the customer details for their reservation(i.e.) customer name, age, sex, seat number, ticket number.

Description:

When the administrator wants to update the reservation details of a passenger travelling in a in a particular train.

INPUT: Passengers details.

SOURCE: Trains source.

OUTPUT: Reserved seats for the passengers.

DESTINATION: Area where the passenger to be reached.

ACTION: To update the reservation details we require name of the passenger, age, sex etc.

REQUIRES: The database having the details of the reservation.

PRECONDITION: The customer's approach a week prior.

POSTCONDITION: None.

SIDE EFFECTS: None.

4.3 Non functional Requirements

- Resource utilization of efficiency : high
- Security : securable
- Mode of use : user friendly.
- Other than the above mentioned requirements, the software should be feasible in order to keep a check if it could be developed within the system boundaries.
- Also the costs and risks associated with the project should be minimal.

CHAPTER 5

DESIGN PHASE

5.1 Architectural Design

Architecture design consists of the diagrammatic representation of the project which acts as a blue print of the software development.

This Architecture design is classified into several types:

- Dataflow diagram.
- Entity relationship diagram.
- Use case diagram.

5.1.1 Data Flow diagram

A **data flow diagram (DFD)** represents the "flow" of data through an information system modelling its *process* aspects in a graphical manner. Often they act as a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used to visualize the process of data processing (structured design).

A DFD shows various kinds of data input and output from the system, where the data will come from and go to, and where the data will be stored. It contains no information about the timing of processes or information about whether processes will operate sequentially or parallels.

Data flow diagram is graphical representation that depicts information flow and the transforms that are applied as data move from input to output.

The basic form of a data flow diagram is also known as a data flow graph or a bubble chart. The data flow diagrams are used to represent a system or software at any level of abstraction (level 0,level 1,etc). The data flow diagram are mainly classified into two module they are,

- Level 0 data flow diagram
- Level 1 data flow diagram
- Level 2 data flow diagram

LEVEL 0:

The level 0 data flow diagram is also called as a fundamental system model or a context model that represents the entire software element as a single bubble with input and output data indicated by incoming and outgoing arrows, respectively. All the additional processes and information flow paths are represented in this form of data flow diagram.

The level 0 data flow diagram of our project consists of module like passengers detail, train detail, display etc.

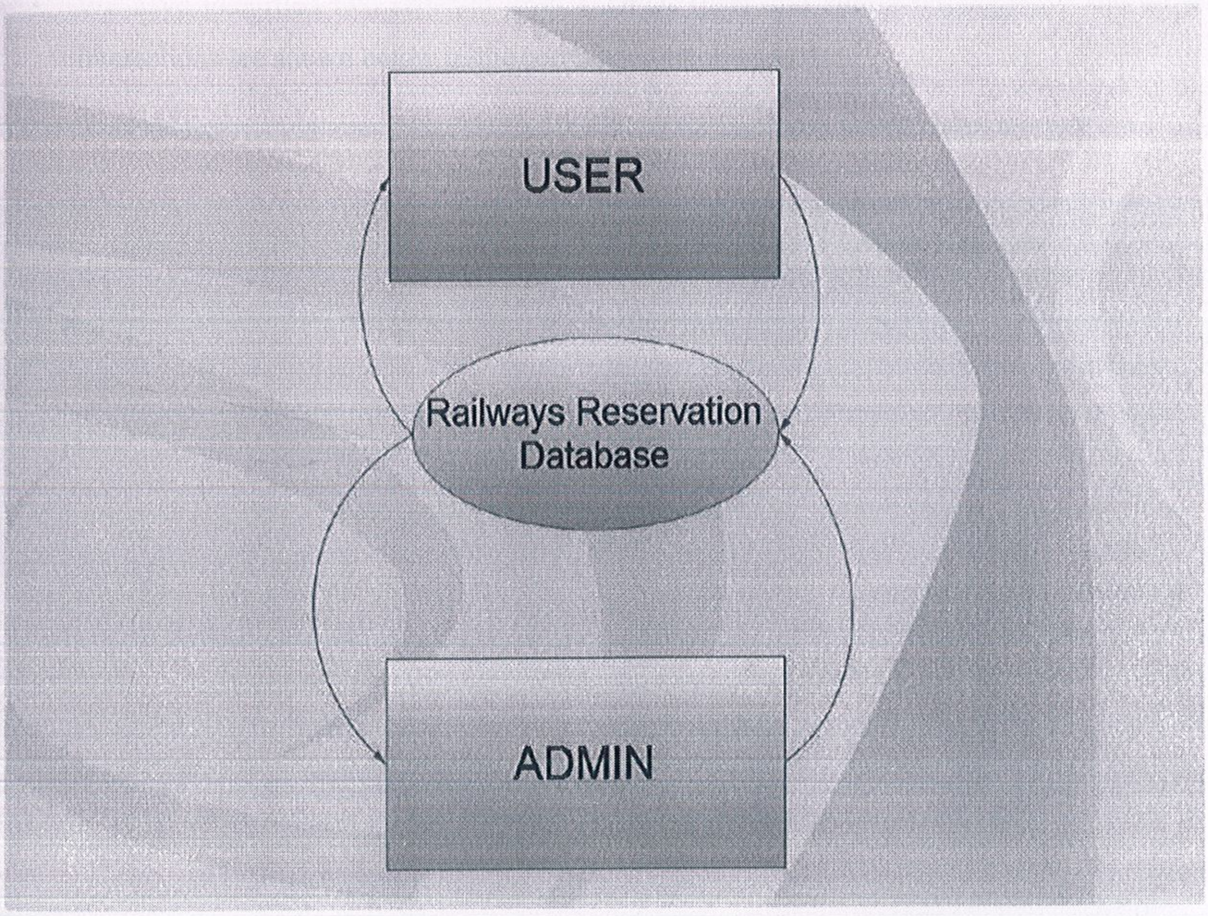


Figure 1: Level 0 DFD

LEVEL 1 DFD:

- The next stage is to create the Level 1 Data Flow Diagram. This level shows the whole system. The purpose of a level 1 DFD is to describe the working of the system by placing identifiable processes on the Level 1 DFD.
- It is more specific than context or level 0 DFD
- Generally contains sub-processes of general system process.
- We have identified two separate level 1 DFDs (user and admin respectively) and all the interactions are shown below in the corresponding diagrams.

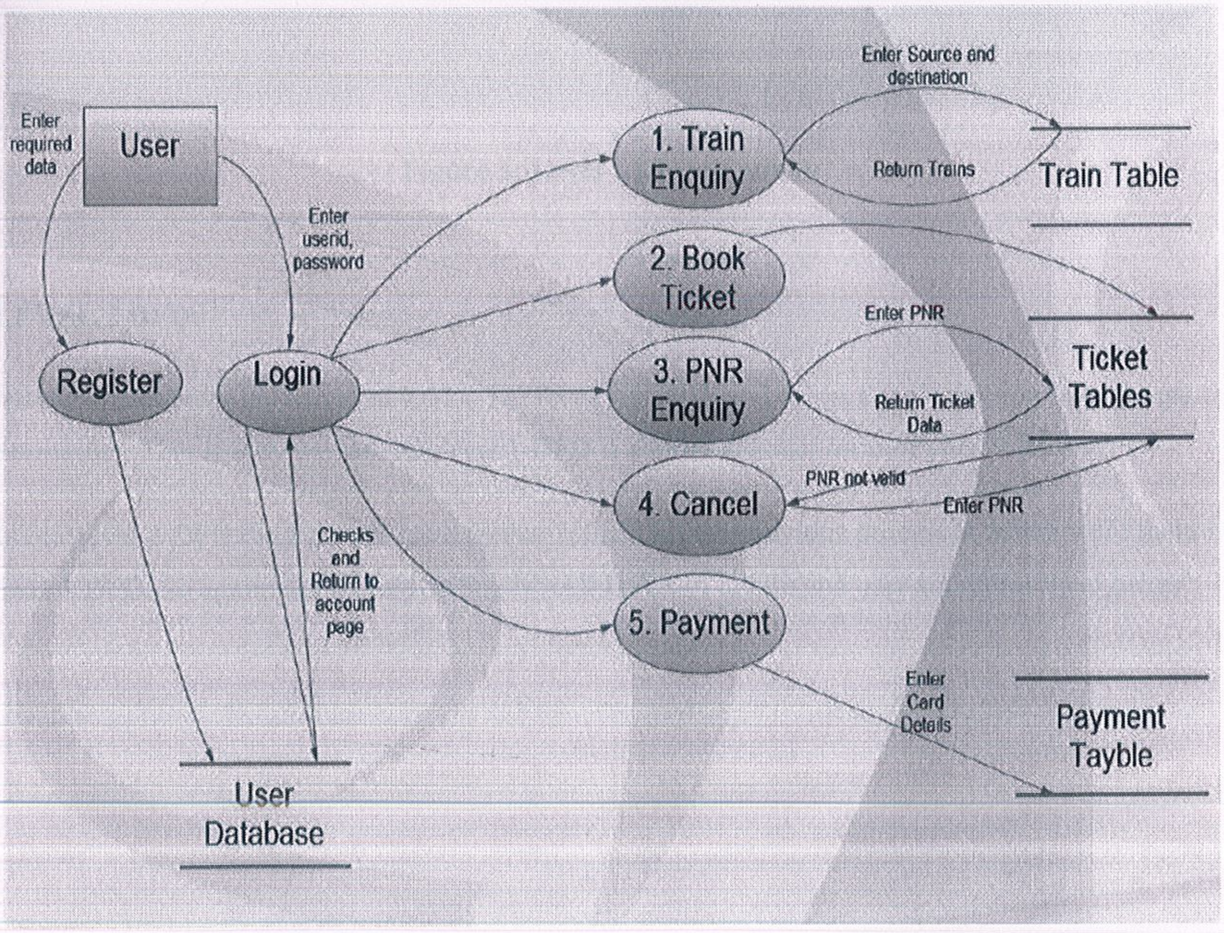


Figure 2: Level 1 DFD of user

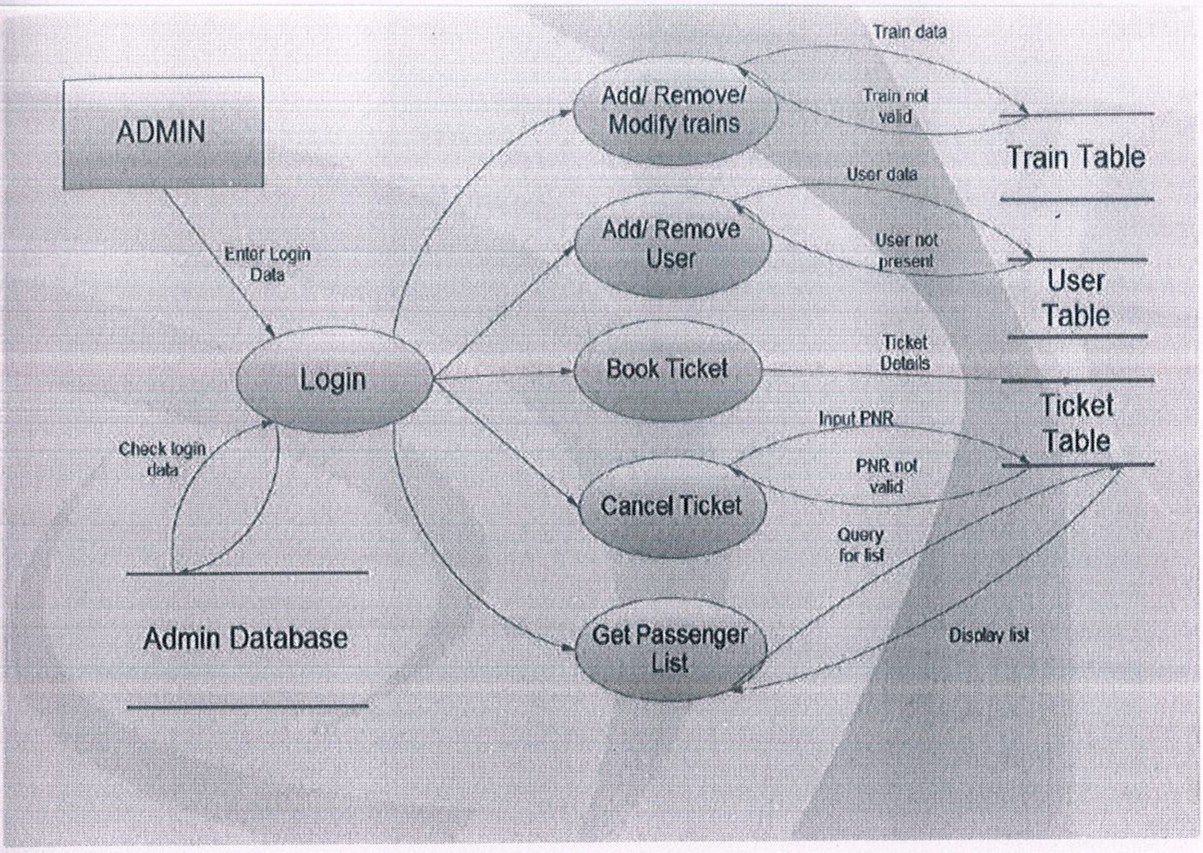


Figure 3: Level 1 DFD of admin

LEVEL 2 DFD:

As we move down the series of DFDs i.e. level 0,level 1 and so on the abstraction level keeps on decreasing. That is why we say that level 1 DFD is a more abstract form of level 2 DFD.

In our level 2 DFD diagram we have further refined all the previous processes which were there in level 1 DFD. So again, the corresponding level 2 DFDs of admin and user are represented below

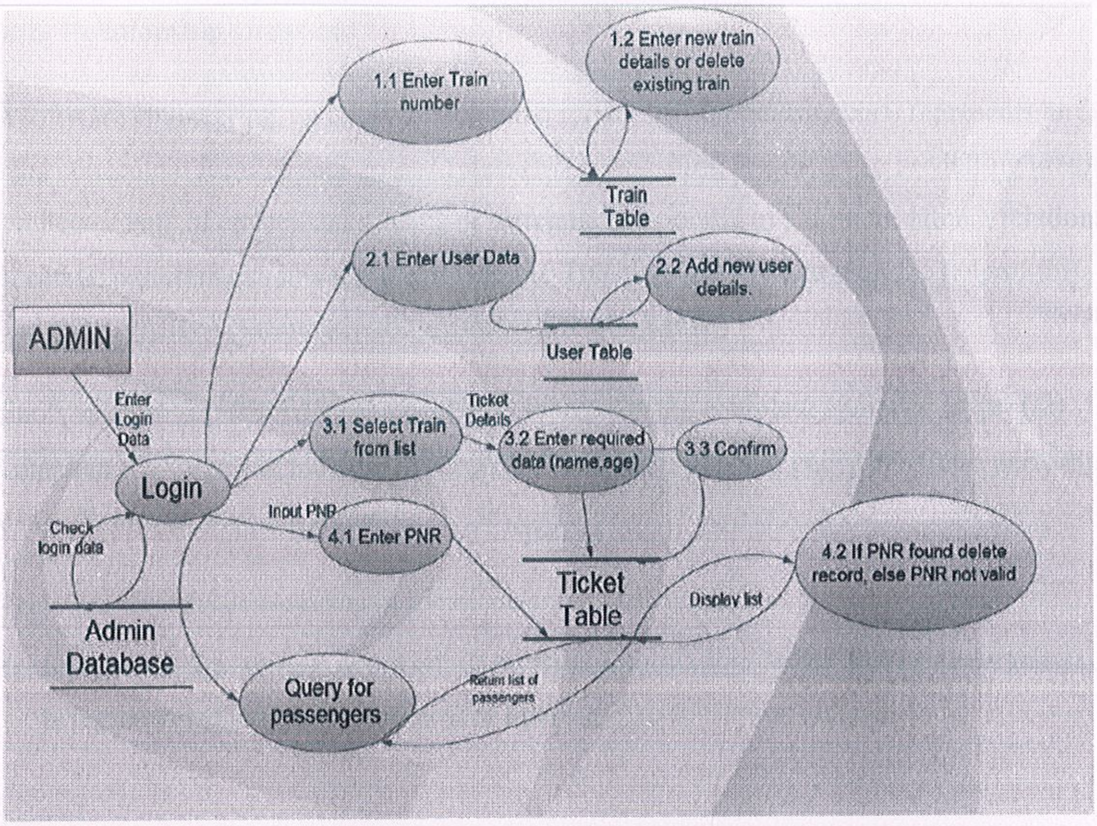

1.1 Enter Train number
1.2 Enter new train details or delete existing train
Train Table
2.1 Enter User Data
2.2 Add new user details.
User Table
ADMIN
Enter Login Data
Ticket Details
3.1 Select Train from list
3.2 Enter required data (name,age)
3.3 Confirm
Login
Input PNR
4.1 Enter PNR
Check login data
4.2 If PNR found delete record, else PNR not valid
Ticket Table
Display list
Admin Database
Query for passengers
Return list of passengers

Figure 4: Level 2 DFD of admin

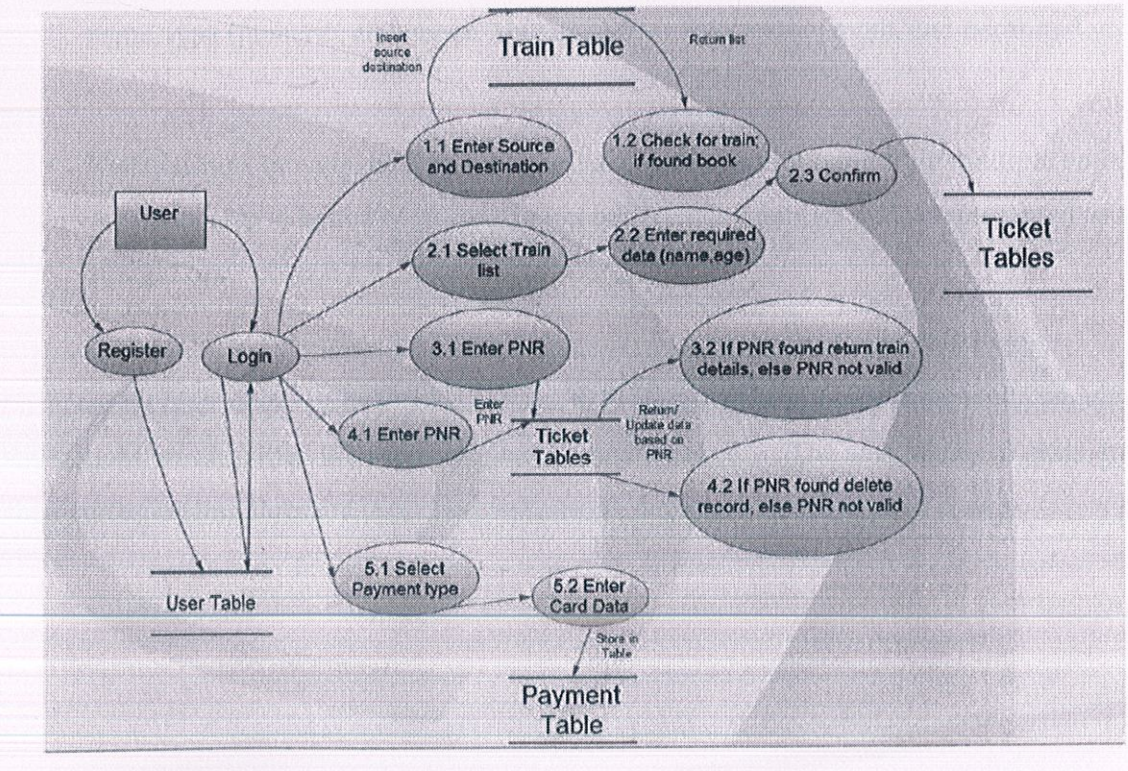

Insert source destination
Train Table
Return list
1.1 Enter Source and Destination
1.2 Check for train; if found book
2.3 Confirm
User
Ticket Tables
2.1 Select Train list
2.2 Enter required data (name,age)
Register
Login
3.1 Enter PNR
3.2 If PNR found return train details, else PNR not valid
Enter PNR
4.1 Enter PNR
Ticket Tables
Return/ Update data based on PNR
4.2 If PNR found delete record, else PNR not valid
5.1 Select Payment type
5.2 Enter Card Data
User Table
Store in Table
Payment Table

Figure 5: Level 2 DFD of user

5.5.2 Entity Relationship Diagram:

In software engineering, an entity-relationship model (ER model for short) represents an abstract and conceptual form of data. Entity-relationship modelling is a database modelling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion. The diagrams created with the help of this process are called **entity-relationship diagrams** or **ER diagrams**.

Entity relationship (ER) diagram generally represents the complete operation of project. This ER diagram is a complete blue print of a project in which each and every operation performed at a time is explained briefly and fluently.

There are three basic elements in ER models:

- Entities are the "things" about which we seek information. In the area of the organization being modeled, entities are objects of interest.

- Attributes are the data which we gather about the entities. Attributes gives us a detailed information about an entity type. Attributes have an internal structure, for example: name, type (numeric, alphabetic, alphanumeric), maximum length, and permitted values.

- Relationships provide the structure needed to gather information from multiple entities. A relationship is basically an association between two entities and is represented as a straight line between two entities connecting them.

In online reservation system this ER diagram is classified in to many aspects. In which the railway stations are classified in to stations junction and cities etc… The availability of seats, trains and their corresponding fares are explained and arrival, departure is represented clearly.

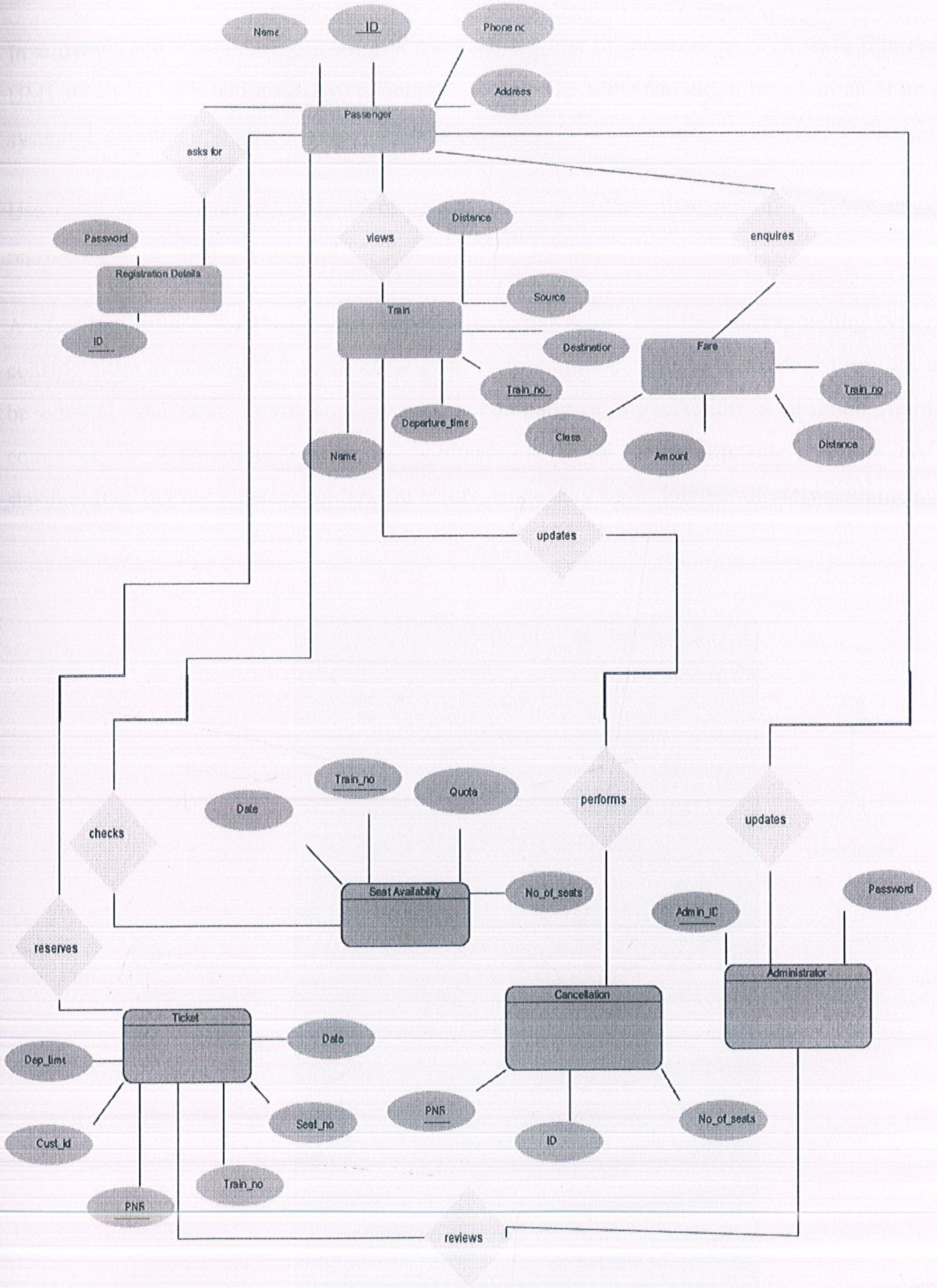


Figure 6: ER Diagram

5.1.3 Use Case Diagram:

In software and systems engineering, a use case is a list of interactions between a role (known in UML as an "actor") and a system, to achieve a goal. The actor can either be a human or an external system depending upon the problem statement.

Use cases are used at a higher level in systems engineering than within software engineering, representing missions or stakeholder goals.

A use case defines the interactions between external actors and the corresponding system under consideration to accomplish a particular goal. Actors must be able to make decisions, but need not be human: "An actor might be a person, a company or organization, a program running in a computer, or a computer system — hardware, software, or middleware". Actors are always stakeholders, but not all stakeholders are actors, since they never interact directly with the system.

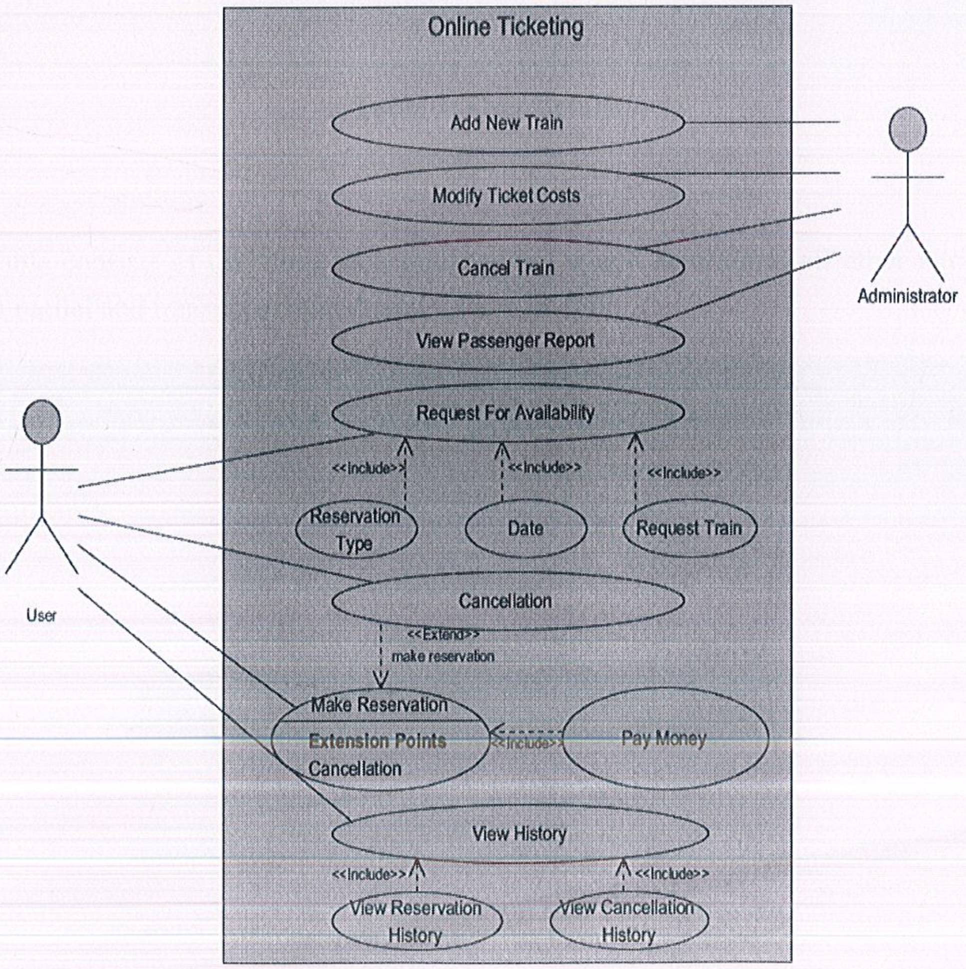


Figure 7: Use Case Diagram

5.2 Database Design

username	password	first_name	last_name	age	email	Registration-date

Table 1: Admin details

The above table consists of username as a primary key which determines all other attributes of it. There is no partial and transitive dependency, so it is in 3NF.

username	password	first_name	last_name	age	email	Registration-date

Table 2: User details

The user table consists of username as a primary key which determines all other attributes of it. There is no partial and transitive dependency, so it is in 3NF.

train_no	train_name	source	destination	distance	arrival_time	dep_time

Name	Departure_ time	date	seat_no

Table 3: Train details

Train number is the primary key of train table. It consists of atomic values of attributes and there is no partial and transitive dependency so it is in 3NF.

1NF – 3NF conversion of train table

train_no	train_name	source	destination	distance	arrival_time	dep_time

Table 4: Train details modified

train_no	class	fare

Table 5: Fare details

PNR	train_no	name	date	seat_no	class	fare

Table 6: Ticket details

train_no	General	Sleeper	Ac I	Ac II	Ac III

Table 7: Class details

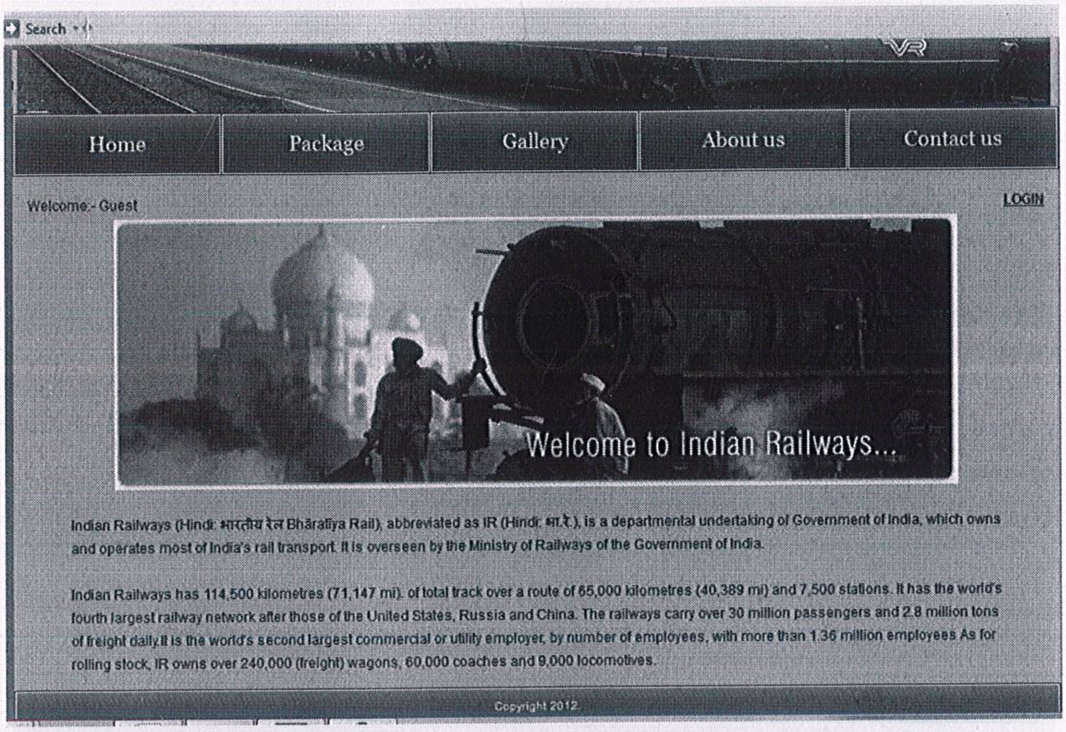
receipt_no	username	card_no	pin_no	bank	type

Table 8: Payment details

CHAPTER 6

RESULTS

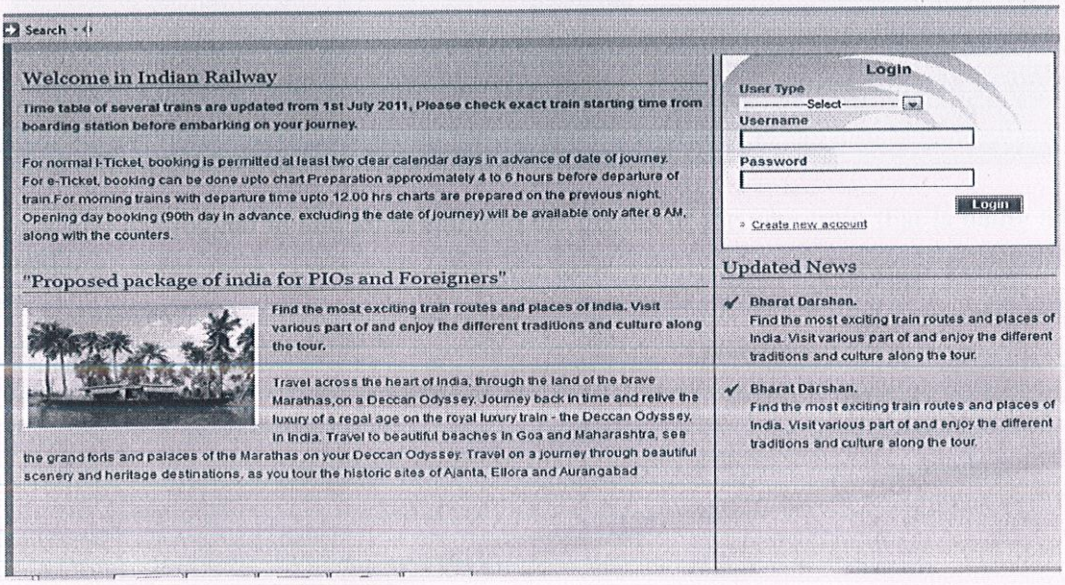
6.1 Implementation



Indian Railways (Hindi: भारतीय रेल Bhāratiya Rail), abbreviated as IR (Hindi: भा.रे.), is a departmental undertaking of Government of India, which owns and operates most of India's rail transport. It is overseen by the Ministry of Railways of the Government of India.

Indian Railways has 114,500 kilometres (71,147 mi). of total track over a route of 65,000 kilometres (40,389 mi) and 7,500 stations. It has the world's fourth largest railway network after those of the United States, Russia and China. The railways carry over 30 million passengers and 2.8 million tons of freight daily.It is the world's second largest commercial or utility employer, by number of employees, with more than 1.36 million employees As for rolling stock, IR owns over 240,000 (freight) wagons, 60,000 coaches and 9,000 locomotives.

Welcome to Indian Railways website



Welcome in Indian Railway

Time table of several trains are updated from 1st July 2011, Please check exact train starting time from boarding station before embarking on your journey.

For normal I-Ticket, booking is permitted at least two clear calendar days in advance of date of journey. For e-Ticket, booking can be done upto chart Preparation approximately 4 to 6 hours before departure of train.For morning trains with departure time upto 12.00 hrs charts are prepared on the previous night. Opening day booking (90th day in advance, excluding the date of journey) will be available only after 8 AM, along with the counters.

"Proposed package of india for PIOs and Foreigners"

Find the most exciting train routes and places of India. Visit various part of and enjoy the different traditions and culture along the tour.

Travel across the heart of India, through the land of the brave Marathas,on a Deccan Odyssey. Journey back in time and relive the luxury of a regal age on the royal luxury train - the Deccan Odyssey, in India. Travel to beautiful beaches in Goa and Maharashtra, see the grand forts and palaces of the Marathas on your Deccan Odyssey. Travel on a journey through beautiful scenery and heritage destinations, as you tour the historic sites of Ajanta, Ellora and Aurangabad

Login

User Type: ------------Select------------

Username:

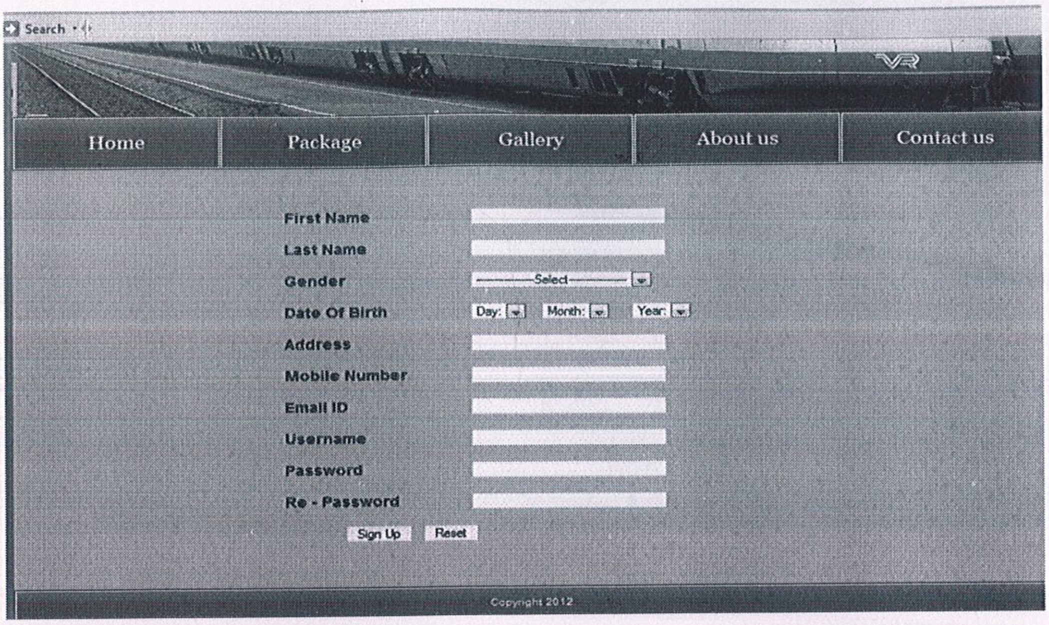
Password:

Updated News

✓ **Bharat Darshan.**
Find the most exciting train routes and places of India. Visit various part of and enjoy the different traditions and culture along the tour.

✓ **Bharat Darshan.**
Find the most exciting train routes and places of India. Visit various part of and enjoy the different traditions and culture along the tour.

This is the login page. Here the user can login as admin or passenger by entering the required fields.



First Name

Last Name

Gender ————Select————

Date Of Birth Day: Month: Year:

Address

Mobile Number

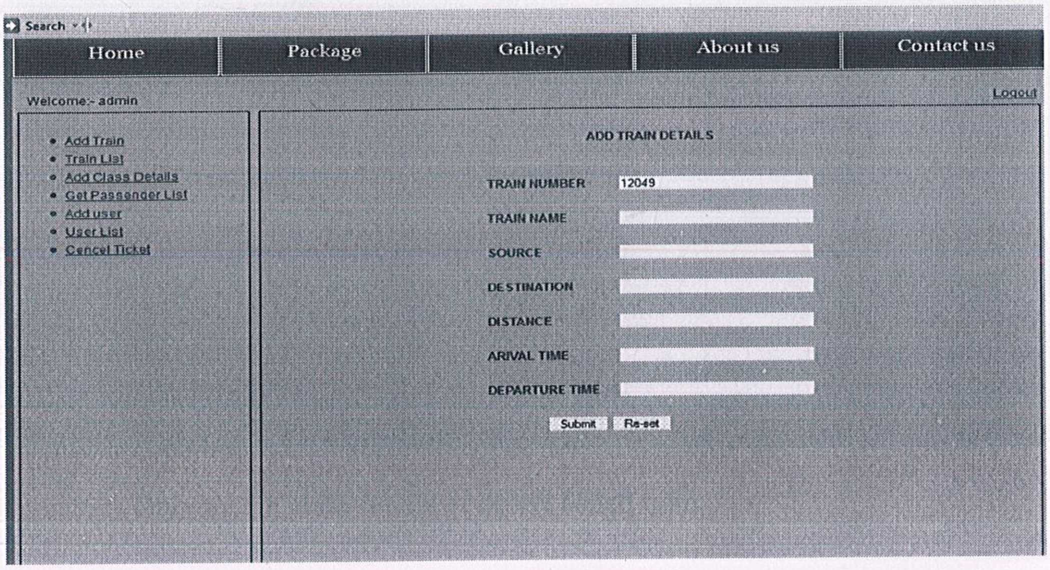
Email ID

Username

Password

Re - Password

If the passenger is new to the website then first he/she has to create an account in order to use any functionalities that are being offered.



Welcome:- admin

ADD TRAIN DETAILS

TRAIN NUMBER 12049

TRAIN NAME

SOURCE

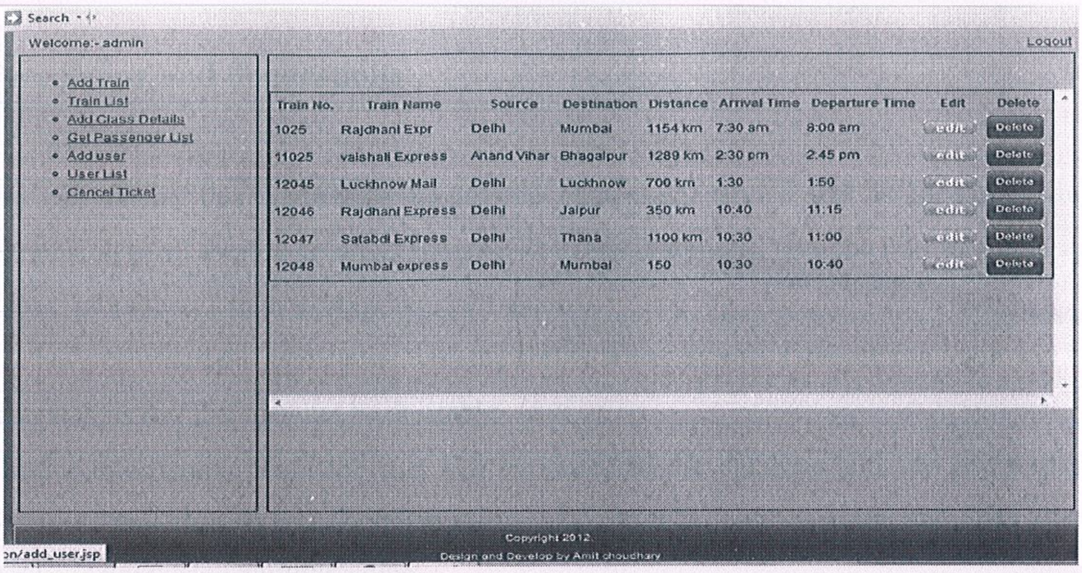
DESTINATION

DISTANCE

ARIVAL TIME

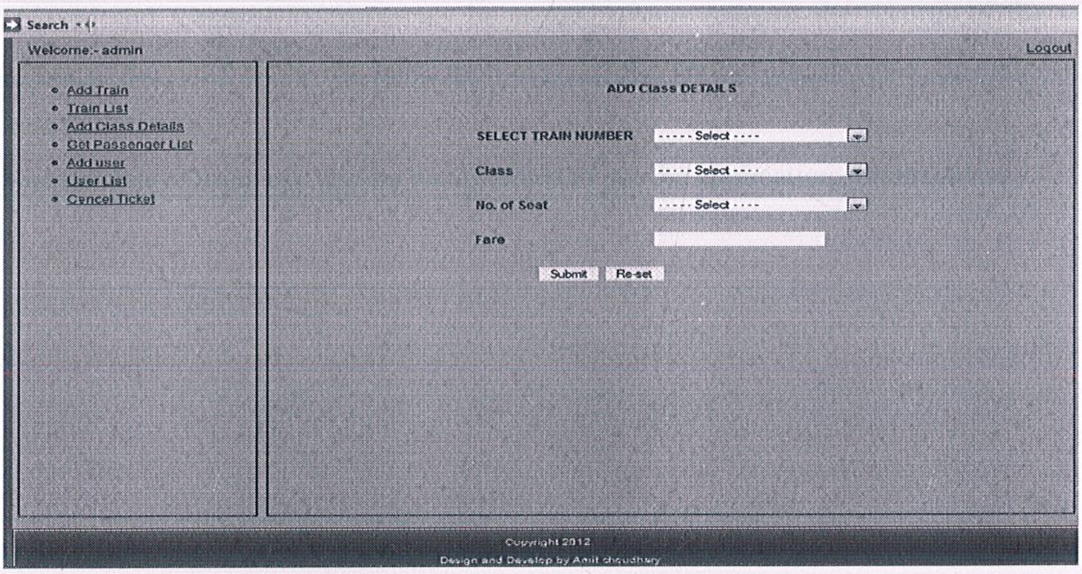
DEPARTURE TIME

Here, the admin has logged in and he is entering the details of the new train that is being added by admin itself.



Train No.	Train Name	Source	Destination	Distance	Arrival Time	Departure Time	Edit	Delete
1025	Rajdhani Expr	Delhi	Mumbai	1154 km	7:30 am	8:00 am	edit	Delete
11025	vaishali Express	Anand Vihar	Bhagalpur	1289 km	2:30 pm	2:45 pm	edit	Delete
12045	Luckhnow Mail	Delhi	Luckhnow	700 km	1:30	1:50	edit	Delete
12046	Rajdhani Express	Delhi	Jaipur	350 km	10:40	11:15	edit	Delete
12047	Satabdi Express	Delhi	Thana	1100 km	10:30	11:00	edit	Delete
12048	Mumbai express	Delhi	Mumbai	150	10:30	10:40	edit	Delete

This page shows the list of all trains that are available on the website.



ADD Class DETAILS

SELECT TRAIN NUMBER ----- Select ----

Class ----- Select ----

No. of Seat ----- Select ----

Fare

Here the admin is entering the class details of newly added train.

6.2 Future Scope and Suggestions

- We can extend this application to provide multi-train service that is for connecting 2 stations, if at all there is no direct train between any 2 stations, then the user may also make use of more than 1 train to book the reservations.
- We have provided the application with the ability where at max 2 people can login and book reservations for, this may not be desirable
- Greater information about the user can be stored in the database and two users may be updated about changes in reservation status upon cancellations, via email and SMS alerts at the same time using multithreading.
- Distributed system architecture may also be implemented where the website may be deployed and a shared database may be created which contains the records of the users. Access to the database at the same time may invoke certain triggers and changes can be made accordingly.
- More number of users can also be incorporated to use the application simultaneously.

CHAPTER 7

CONCLUSION

This project on **ONLINE TICKETING SYSTEM involves** providing a web application to users to reserve railway tickets online after logging in to the system. The details such as the Train number, Train ticket are used to know about the different trains. The programs are coded in easier and structured manner so that any possible modification can be done easily. The passenger details and train details are provided so that the passenger's reports are generated easily.

This application may be extended as a generic e-ticketing application for booking of tickets by users in other departments like booking of movie tickets, bus reservation, airline reservation, and many more such systems.

The project can be viewed as a comprehensive one for those involved with Web Application Development. It engrosses most of the aspects of session management and database connectivity both theoretical and system oriented.

It involves managing large databases and designing appropriate web pages that are user friendly and interactive. Also, the concept of scripting using Java comes into play with the performance considerations of the solution provided at both the administrative and the user end.

REFERENCES

Books

- Professional JSP 2nd Edition by Larry Kim, Published by Wrox Press; 2 (April 2001)
- Java: The Complete Reference - Herbert Schildt -, 5th edition , Published by McGraw-Hill Osborne Media; 5th edition(August 13, 2002)

Web Pages

www.irctc.com

www.railaustralia.com

www.indianrail.gov in.com

www.wikipedia.com

www.askme.com

APPENDIX A

Team member workload

Purnima Choudhry

- Design interface of user login
- Design interface of cancel ticket
- Programming of cancel ticket
- Programming of user login
- Connectivity with database system

Sunny Sharma

- Design interface of admin login
- Design interface of home page
- Design of database queries
- Programming of home page
- Programming of admin login
- Programming of email validation

Akash Chaudhary

- Design interface of modify ticket
- Design interface of book ticket
- Design of database queries
- Programming of modify ticket
- Programming of book ticket
- Programming of email validation

APPENDIX B

Project plan

PROJECT PLAN

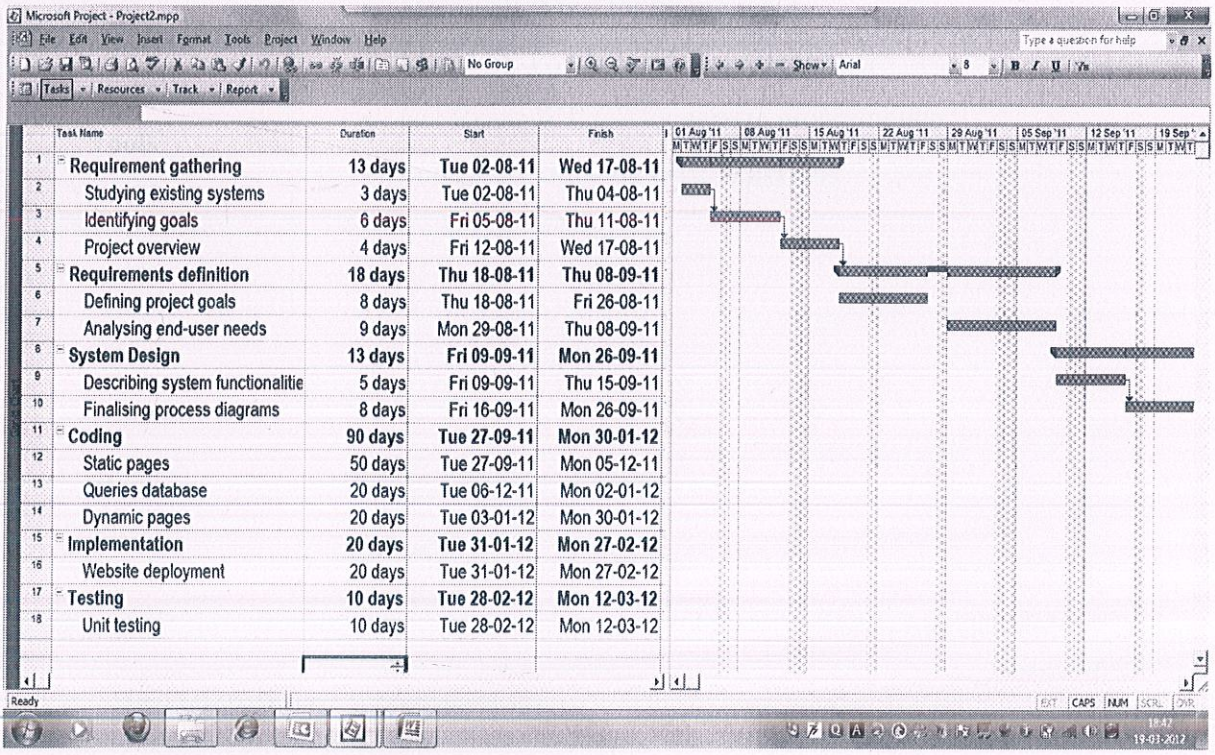
Student Name : Sunny Sharma, Akash Chaudhary, Purnima Choudhry

Project Title : Online Ticketing System

Project Supervisor : Mr. Nitin Rakesh

Date : May 2012

Project Objective(s) : - To build an efficient web application to reserve, cancel and review railway tickets online.



	Task Name	Duration	Start	Finish
1	**Requirement gathering**	**13 days**	**Tue 02-08-11**	**Wed 17-08-11**
2	Studying existing systems	3 days	Tue 02-08-11	Thu 04-08-11
3	Identifying goals	6 days	Fri 05-08-11	Thu 11-08-11
4	Project overview	4 days	Fri 12-08-11	Wed 17-08-11
5	**Requirements definition**	**18 days**	**Thu 18-08-11**	**Thu 08-09-11**
6	Defining project goals	8 days	Thu 18-08-11	Fri 26-08-11
7	Analysing end-user needs	9 days	Mon 29-08-11	Thu 08-09-11
8	**System Design**	**13 days**	**Fri 09-09-11**	**Mon 26-09-11**
9	Describing system functionalitie	5 days	Fri 09-09-11	Thu 15-09-11
10	Finalising process diagrams	8 days	Fri 16-09-11	Mon 26-09-11
11	**Coding**	**90 days**	**Tue 27-09-11**	**Mon 30-01-12**
12	Static pages	50 days	Tue 27-09-11	Mon 05-12-11
13	Queries database	20 days	Tue 06-12-11	Mon 02-01-12
14	Dynamic pages	20 days	Tue 03-01-12	Mon 30-01-12
15	**Implementation**	**20 days**	**Tue 31-01-12**	**Mon 27-02-12**
16	Website deployment	20 days	Tue 31-01-12	Mon 27-02-12
17	**Testing**	**10 days**	**Tue 28-02-12**	**Mon 12-03-12**
18	Unit testing	10 days	Tue 28-02-12	Mon 12-03-12

Project Resources Plan

Equipment	**Computer with development Environment** • An active MySQL server • A browser which acts as a client • Client side scripting language : HTML, Javascript • Server side Scripting language : JSP • Server : Apache Tomcat 6.0.20
Tools	• Microsoft Word for typing report • Microsoft PowerPoint for presentation • Net Beans IDE 6.8 for system programming • Eclipse for drawing UML diagrams

APPENDIX C

Programming code

The Home Page

```
<%--
  Document   : Home
  Created on : Jan 2, 2012, 1:13:49 PM
  Author     : a
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <link href="styles.css" rel="stylesheet" type="text/css" />
    <%
            response.setHeader("Cache-Control", "no-cache"); //HTTP 1.1
            response.setHeader("Pragma", "no-cache"); //HTTP 1.0
            response.setDateHeader("Expires", 0); //prevents caching at the proxy server
            response.setHeader("Cache-Control", "no-store"); //HTTP 1.1
            String uid, adminid = null;
            adminid = (String) session.getAttribute("Admin");
            uid = (String) session.getAttribute("SS");
            if (uid != null) {

    %>
    <title>JSP Page</title>
  </head>
  <body>
    <%@include  file="header.html" %>
    <div id ="main">
      <table width="100%" border="1px"><tr><td style="float: left; margin-left: 10px">Welcome:- <%=uid%></td><td style="float: right; margin-right: 10px"><a href="logout">Logout</a></td></tr></table>
      <div>
        <div style="float:left">
          <%@include file = "menu.html" %>
        </div>
        <div style="float:right;height: 500px; width: 680px; border: 1px solid black; margin-right: 10px">
          <center><img src="images/rail.png" alt="rail image" height="250px" width="600px" style=" margin-top: 15px"/></center>
          <p style=" margin-top: 20px; margin-left: 50px; margin-right: 50px">Indian Railways (Hindi: भारतीय रेल Bhāratīya Rail), abbreviated as IR (Hindi: भा.रे.), is a departmental undertaking of Government of India, which owns and operates most of India's rail transport. It is overseen by the Ministry of Railways of the Government of India.</p>
          <p style=" margin-top: 20px; margin-left: 50px; margin-right: 50px">Indian Railways has 114,500 kilometres (71,147 mi). of total track over a route of 65,000 kilometres (40,389 mi) and 7,500 stations. It has the world's fourth largest railway network after those of the United States, Russia and China. The railways carry over 30 million passengers and 2.8 million tons of freight daily.It is the world's second largest commercial or utility employer, by number of employees, with more than 1.36 million employees As for rolling stock, IR owns over 240,000 (freight) wagons, 60,000 coaches and 9,000 locomotives.</p>
        </div>
      </div>
    </div>
    <%@include file = "footer.html" %>
  </body>
</html>
<%      } else if(adminid != null) {%>


  <title>JSP Page</title>
  </head>
  <body>
    <%@include  file="header.html" %>
    <div id ="main">
      <table width="100%" border="1px"><tr><td style="float: left; margin-left: 10px">Welcome:- <%=adminid%></td><td style="float: right; margin-right: 10px"><a href="logout">Logout</a></td></tr></table>
      <div>
        <div style="float:left">
          <%@include file = "adminmenu.html" %>
        </div>
        <div style="float:right;height: 500px; width: 680px; border: 1px solid black; margin-right: 10px">
          <center><img src="images/rail.png" alt="rail image" height="250px" width="600px" style=" margin-top: 15px"/></center>
          <p style=" margin-top: 20px; margin-left: 50px; margin-right: 50px">Indian Railways (Hindi: भारतीय रेल Bhāratīya Rail), abbreviated as IR (Hindi: भा.रे.), is a departmental undertaking of Government of India, which owns and operates most of India's rail transport. It is overseen by the Ministry of Railways of the Government of India.</p>
          <p style=" margin-top: 20px; margin-left: 50px; margin-right: 50px">Indian Railways has 114,500 kilometres (71,147 mi). of total track over a route of 65,000 kilometres (40,389 mi) and 7,500 stations. It has the world's fourth largest railway network after those of the United States, Russia and China. The railways carry over 30 million passengers and 2.8 million tons of freight daily.It is the world's second largest commercial or utility employer, by number of employees, with more than 1.36 million employees As for rolling stock, IR owns over 240,000 (freight) wagons, 60,000 coaches and 9,000 locomotives.</p>
        </div>
      </div>
    </div>
    <%@include file = "footer.html" %>
  </body>
</html>


<%} else {%>
```

```
<title>JSP Page</title>
</head>
<body>
  <%@include  file="header.html" %>
  <div id ="main">
    <table  width="100%"><tr><td  style="float:  left;  margin-left:  10px">Welcome:-
Guest</td><td  style="float:  right;  margin-right:  10px"><a
href="index.jsp"><span>LOGIN</span></a></td></tr></table>
    <center><img src="images/rail.png" alt="rail image" /></center>
    <p style=" margin-top: 20px; margin-left: 50px; margin-right: 50px">Indian Railways (Hindi:
भारतीय रेल Bhāratīya Rail), abbreviated as IR (Hindi: भा.रे.), is a departmental undertaking of
Government of India, which owns and operates most of India's rail transport. It is overseen by the
Ministry of Railways of the Government of India.</p>
            <p style=" margin-top: 20px; margin-left: 50px; margin-right: 50px">Indian Railways
has 114,500 kilometres (71,147 mi). of total track over a route of 65,000 kilometres (40,389 mi) and
7,500 stations. It has the world's fourth largest railway network after those of the United States,
Russia and China. The railways carry over 30 million passengers and 2.8 million tons of freight
daily.it is the world's second largest commercial or utility employer, by number of employees, with
more than 1.36 million employees As for rolling stock, IR owns over 240,000 (freight) wagons,
60,000 coaches and 9,000 locomotives.</p>
  </div>
  <%@include file = "footer.html" %>
</body>
</html>

<%          }
%>
```

Register page

```
<%--
    Document   : Register
    Created on : Dec 1, 2011, 6:15:54 PM
    Author     : a
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <link href="styles.css" rel="stylesheet" type="text/css" />
        <script type="text/javascript">

            function validate(form1){
                var fname = document.form1.txtFirstName
                var Lname = document.form1.txtLastName
                var Gender = document.form1.Gender
                 var address = document.form1.txtaddress
                var MobileNumber = document.form1.txtMobileNumber
                var email = document.form1.txtEmailID
                var username = document.form1.txtUserName
                var password = document.form1.txtPassword
                var RePassword = document.form1.txtRePassword
                var mm = document.form1.birthday_month
                var dd = document.form1.birthday_day
                var yy = document.form1.birthday_year


                if(fname.value == ""){
```

```
    alert('Please enter your first name')
    fname.focus();
    return false
}

if(Lname.value == ""){
    alert('Please enter your last name')
    Lname.focus();
    return false
}

if(Gender.options[0].selected == true){
    alert('Please select your gender')
    Gender.focus();
    return false
}

if(dd.options[0].selected == true){
    alert('Please select your birth date')
    dd.focus();
    return false
}
if(mm.options[0].selected == true){
    aiert('Please select your birth month')
    mm.focus();
    return false
}
if(yy.options[0].selected == true){
    alert('Please select your birth year')
    yy.focus();
    return false
}
 if(address.value == ""){
    alert('Please enter your address')
    address.focus();
```

```
        return false
    }
    if(MobileNumber.value == ""){
        alert('Please enter your contact number')
        MobileNumber.focus();
        return false
    }
    if(email.value == ""){
        alert('Please enter your email ID')
        email.focus();
        return false
    }
    if(username.value == ""){
        alert('Please enter your username')
        username.focus();
        return false
    }
    if(password.value == ""){
        alert('Please enter your password')
        password.focus();
        return false
    }
    if(RePassword.value == ""){
        alert('Please Re-enter  password')
        RePassword.focus();
        return false
    }
    if(RePassword.value != password.value){
        alert('Please enter same password')
        RePassword.focus();
        return false
    }
}
```

```
      </script>
      <title>JSP Page</title>
   </head>
   <body>
      <%@include file = "header.html" %>
      <div id ="main">
         <form  name="form1" action="signUp" method="POST" onsubmit="return validate()">
            <center>

               <fieldset style=" width: 450PX; height: 400px; width:">
                  <h1>&nbsp;</h1>


                  <table cellpadding="4" cellspacing="4" align="center" width="100%">   <center>
                        <tr><td><h3>First Name</h3></td><td style="bgcolor:white"><input
type="text" name="txtFirstName" value="" size="30"/>&nbsp;</td></tr>
                        <tr><td><h3>Last Name</h3></td><td><input type="text"
name="txtLastName" value="" size="30" /></td></tr>
                        <tr><td><h3>Gender</h3></td><td><select name="Gender">
                              <option>-----------------Select-----------------</option>
                              <option>Male</option>
                              <option>Female</option>
                           </select></td></tr>
                        <tr><td><h3>Date Of Birth</h3></td><td><div>


                              <select name="birthday_day" id="birthday_day">
                                 <option value="-1">Day:</option>
                                 <option value="1">1</option>
                                 <option value="2">2</option>
                                 <option value="3">3</option>
                                 <option value="4">4</option>
                                 <option value="5">5</option>
```

```
    <option value="6">6</option>
    <option value="7">7</option>
    <option value="8">8</option>
    <option value="9">9</option>
    <option value="10">10</option>
    <option value="11">11</option>
    <option value="12">12</option>
    <option value="13">13</option>
    <option value="14">14</option>
    <option value="15">15</option>
    <option value="16">16</option>
    <option value="17">17</option>
    <option value="18">18</option>
    <option value="19">19</option>
    <option value="20">20</option>
    <option value="21">21</option>
    <option value="22">22</option>
    <option value="23">23</option>
    <option value="24">24</option>
    <option value="25">25</option>
    <option value="26">26</option>
    <option value="27">27</option>
    <option value="28">28</option>
    <option value="29">29</option>
    <option value="30">30</option>
    <option value="31">31</option>
</select>&nbsp; &nbsp;&nbsp;

<select name="birthday_month">
    <option value="0">Month:</option>
    <option value="1">Jan</option>
    <option value="2">Feb</option>
    <option value="3">Mar</option>
    <option value="4">Apr</option>
    <option value="5">May</option>
    <option value="6">Jun</option>
    <option value="7">Jul</option>
    <option value="8">Aug</option>
    <option value="9">Sep</option>
    <option value="10">Oct</option>
    <option value="11">Nov</option>
    <option value="12">Dec</option>
</select> &nbsp; &nbsp; &nbsp;
<select name="birthday_year" id="birthday_year">
    <option value="-1">Year:</option>
    <option value="2010">2010</option>
    <option value="2009">2009</option>
    <option value="2008">2008</option>
    <option value="2007">2007</option>
    <option value="2006">2006</option>
    <option value="2005">2005</option>
    <option value="2004">2004</option>
    <option value="2003">2003</option>
    <option value="2002">2002</option>
    <option value="2001">2001</option>
    <option value="2000">2000</option>
    <option value="1999">1999</option>
    <option value="1998">1998</option>
    <option value="1997">1997</option>
    <option value="1996">1996</option>
    <option value="1995">1995</option>
    <option value="1994">1994</option>
    <option value="1993">1993</option>
    <option value="1992">1992</option>
    <option value="1991">1991</option>
    <option value="1990">1990</option>
    <option value="1989">1989</option>
    <option value="1988">1988</option>
    <option value="1987">1987</option>
    <option value="1986">1986</option>
```

```
<option value="1985">1985</option>
<option value="1984">1984</option>
<option value="1983">1983</option>
<option value="1982">1982</option>
<option value="1981">1981</option>
<option value="1980">1980</option>
<option value="1979">1979</option>
<option value="1978">1978</option>
<option value="1977">1977</option>
<option value="1976">1976</option>
<option value="1975">1975</option>
<option value="1974">1974</option>
<option value="1973">1973</option>
<option value="1972">1972</option>
<option value="1971">1971</option>
<option value="1970">1970</option>
<option value="1969">1969</option>
<option value="1968">1968</option>
<option value="1967">1967</option>
<option value="1966">1966</option>
<option value="1965">1965</option>
<option value="1964">1964</option>
<option value="1963">1963</option>
<option value="1962">1962</option>
<option value="1961">1961</option>
<option value="1960">1960</option>
<option value="1959">1959</option>
<option value="1958">1958</option>
<option value="1957">1957</option>
<option value="1956">1956</option>
<option value="1955">1955</option>
<option value="1954">1954</option>
<option value="1953">1953</option>
<option value="1952">1952</option>
<option value="1951">1951</option>
```

```
<option value="1950">1950</option>
<option value="1949">1949</option>
<option value="1948">1948</option>
<option value="1947">1947</option>
<option value="1946">1946</option>
<option value="1945">1945</option>
<option value="1944">1944</option>
<option value="1943">1943</option>
<option value="1942">1942</option>
<option value="1941">1941</option>
<option value="1940">1940</option>
<option value="1939">1939</option>
<option value="1938">1938</option>
<option value="1937">1937</option>
<option value="1936">1936</option>
<option value="1935">1935</option>
<option value="1934">1934</option>
<option value="1933">1933</option>
<option value="1932">1932</option>
<option value="1931">1931</option>
<option value="1930">1930</option>
<option value="1929">1929</option>
<option value="1928">1928</option>
<option value="1927">1927</option>
<option value="1926">1926</option>
<option value="1925">1925</option>
<option value="1924">1924</option>
<option value="1923">1923</option>
<option value="1922">1922</option>
<option value="1921">1921</option>
<option value="1920">1920</option>
<option value="1919">1919</option>
<option value="1918">1918</option>
<option value="1917">1917</option>
<option value="1916">1916</option>
```

```
                    <option value="1915">1915</option>
                    <option value="1914">1914</option>
                    <option value="1913">1913</option>
                    <option value="1912">1912</option>
                    <option value="1911">1911</option>
                    <option value="1910">1910</option>
                    <option value="1909">1909</option>
                    <option value="1908">1908</option>
                    <option value="1907">1907</option>
                    <option value="1906">1906</option>
                    <option value="1905">1905</option>
                    <option value="1904">1904</option>
                    <option value="1903">1903</option>
                    <option value="1902">1902</option>
                    <option value="1901">1901</option>
                    <option value="1900">1900</option>
                  </select>
                </div>
              </td></tr>
            <tr><td><h3>Address</h3></td><td><input type="text" name="txtaddress" value="" size="30" /></td></tr>
            <tr><td><h3>Mobile Number</h3></td><td><input type="text" name="txtMobileNumber" value="" size="30" /></td></tr>
            <tr><td><h3>Email ID</h3></td><td><input type="text" name="txtEmailID" value="" size="30" onblur="onlyvhar()" /><span id="errorMesUsername"></span></td></tr>
            <tr><td><h3>Username</h3></td><td><input type="text" name="txtUserName" value="" size="30" /></td></tr>
            <tr><td><h3>Password</h3></td><td><input type="password" name="txtPassword" value="" size="30" /></td></tr>
            <tr><td><h3>Re - Password</h3></td><td><input type="password" name="txtRePassword" value="" size="30" /></td></tr>
            <tr><td colspan="2">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input class=button type="submit" name="Submit"
```

```
value="Sign Up"> &nbsp;&nbsp;&nbsp;<input class=button type="reset" name="Reset"
value="Reset"></td></tr>
              </center> </table>



          </fieldset>
        </center>
      </form>
    </div>
    <%@include file = "footer.html" %>
  </body>
</html>

<%--
  Document   : Home
  Created on : Jan 2, 2012, 1:13:49 PM
  Author     : a
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <link href="styles.css" rel="stylesheet" type="text/css" />

    <title>JSP Page</title>
  </head>
  <body>
    <%@include  file="header.html" %>
    <div id ="main">
      <h1>&nbsp;</h1>
      <div>
```

```
        <center>
            <%
                        String errormsg = null;
                        errormsg = (String) session.getAttribute("loginerror");
                        if (errormsg != null) {
                    %>
                    <span style=" font-size: 40; color:red"><%=errormsg%></span>
                    <%
                            session.removeAttribute("loginerror");
                        } else {
                            session.setAttribute("loginerror", "");
                        }%>
            <div style="height: 500px; width: 680px; border: 1px solid black; margin-right:
10px">
                <center><img src="images/rail1.jpg" alt="rail image" height="250px"
width="500px" styie=" margin-top: 15px"/></center>

                <h1>&nbsp;</h1>
                <h1>&nbsp;</h1>
                <form action="adminLogin" method="POST">
                    <center>
                        <table cellpadding="4" celispacing="4" style=" border: 1px solid white; height:
100px; width: 300px">
                            <tr><td><span>Username</span></td><td><input type="text"
name="txtusername" value="" size="30" /></td></tr>
                            <tr><td><span>Password</span></td><td><input type="password"
name="txtpassword" value="" size="30" /></td></tr>
                            <tr><td colspan="2">&nbsp;&nbsp;<input type="submit" value="Login"
/>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="reset" value="Re-set" /></td></tr>
                        </table>
                        <h1>&nbsp;</h1>

                    </center>
                </form>
                </div>
            </center>
        </div>
    </div>
    <%@include file = "footer.html" %>
  </body>
</html>
```

Add train module

```
/**
 *
 * @author a
 */
public class addtrain extends HttpServlet {
//
//    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
//            throws ServletException, IOException {
//        response.setContentType("text/html;charset=UTF-8");
//        PrintWriter out = response.getWriter();
//        HttpSession session = request.getSession();
//        PreparedStatement ps = null;
//        Connection con = null;
//            String train_no, train_name, source, destination, distance, arrival_time, departure_time =
null;
//        try {
//            train_no = request.getParameter("txtTrainNumber");
//            train_name = request.getParameter("txtTrainName");
//            source = request.getParameter("txtSource");
//            destination = request.getParameter("txtDestination");
//            distance = request.getParameter("txtDistance");
//            arrival_time = request.getParameter("txtArivalTime");
//            departure_time = request.getParameter("txtDepartureTime");
//            System.out.println(train_no);
//            System.out.println(train_name);
//            System.out.println(source);
//          System.out.println(destination);
//          System.out.println(distance);
//          System.out.println(arrival_time);
//          System.out.println(departure_time);
//          con = connection.dbConnection.makeCon();
//          ps = con.prepareStatement("insert into train values(?,?,?,?,?,?,?)");
//          ps.setString(1, train_no);
//          ps.setString(2, train_name);
//          ps.setString(3, source);
//          ps.setString(4, destination);
//          ps.setString(5, distance);
//          ps.setString(6, arrival_time);
//          ps.setString(7, departure_time);
//          int i = ps.executeUpdate();
//          if (i > 0) {
//              session.setAttribute("msg", "Train add successfully");
//              response.sendRedirect("addTrain.jsp");
//          } else {
//              session.setAttribute("msg", "Train not add successfully");
//              response.sendRedirect("addTrain.jsp");
//          }
//      } catch (SQLException e) {
//          e.printStackTrace();
//      } finally {
//          out.close();
//      }
//  }


    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        PrintWriter out = response.getWriter();
        HttpSession session = request.getSession();
        PreparedStatement ps = null;
```

```
Connection con = null;
String train_no, train_name, source, destination, distance, arrival_time, departure_time = null;
try {
    train_no = request.getParameter("txtTrainNumber");
    train_name = request.getParameter("txtTrainName");
    source = request.getParameter("txtSource");
    destination = request.getParameter("txtDestination");
    distance = request.getParameter("txtDistance");
    arrival_time = request.getParameter("txtArivalTime");
    departure_time = request.getParameter("txtDepartureTime");
    System.out.println(train_no);
    System.out.println(train_name);
    System.out.println(source);
    System.out.println(destination);
    System.out.println(distance);
    System.out.println(arrival_time);
    System.out.println(departure_time);
    con = connection.dbConnection.makeCon();
    ps = con.prepareStatement("insert into train values(?,?,?,?,?,?,?)");
    ps.setString(1, train_no);
    ps.setString(2, train_name);
    ps.setString(3, source);
    ps.setString(4, destination);
    ps.setString(5, distance);
    ps.setString(6, arrival_time);
    ps.setString(7, departure_time);
    int i = ps.executeUpdate();
    if (i > 0) {
        session.setAttribute("msg", "Train details has been saved successfully");
        response.sendRedirect("addTrain.jsp");
    } else {
        session.setAttribute("msg", "Train details not saved");
        response.sendRedirect("addTrain.jsp");
    }
} catch (SQLException e) {
```

```
            out.println(e.toString());
        } finally {
            out.close();
        }
    }


}
```

Sign In Module

```
/**
 *
 * @author a
 */
public class signIn extends HttpServlet {

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        HttpSession session = request.getSession(),
        Connection con = null;
        Statement ss = null;
        ResultSet rs = null;
        String username, password, sql, usertype;
        usertype = request.getParameter("usertype");
        username = request.getParameter("text1");
        password = request.getParameter("text2");
        if (usertype.equals("User")) {
            sql = "select * from user_ where username = '" + username + "' and password_ = '" +
password + "'";
            try {
                con = connection.dbConnection.makeCon();
                ss = con.createStatement();
                rs = ss.executeQuery(sql);
                if (rs.next()) {
                    String email_id = rs.getString("email_id");
                    session.setAttribute("mail", email_id);
                    session.setAttribute("SS", username);
                    response.sendRedirect("Home.jsp");
                } else {
                    session.setAttribute("error", "Your userID or passwod is wrong!");
                    response.sendRedirect("index.jsp");
                }
            } catch (Exception e) {
            }
        } else if (usertype.equals("Admin")) {
            if (username.equals("admin") && password.equals("admin")) {
                session.setAttribute("Admin", "admin");
                response.sendRedirect("Home.jsp");
            } else {
                session.setAttribute("error", "Your userID or passwod is wrong!");
                response.sendRedirect("index.jsp");
            }

        }
    }
}
```