

ONLINE CAB SCHEDULING SYSTEM

CONTENTS	Page No.
<i>Acknowledgement</i>	<i>3</i>
<i>Declaration</i>	<i>4</i>
<i>1. Introduction & Objectives of the Project</i>	
<i>1.1. Introduction</i>	<i>7</i>
<i>1.2. Objectives</i>	<i>8</i>
<i>1.3. Advantage</i>	<i>10</i>
<i>1.4. Project Category</i>	<i>11</i>
<i>1.5. Drawback of Current Manual-System</i>	<i>12</i>
<i>1.6. Proposed System</i>	<i>14</i>
<i>1.7. Need</i>	<i>15</i>
<i>2. Project Selection</i>	
<i>2.1 Tools/Platforms, Hardware & Software Requirements</i>	<i>16</i>
<i>2.2 Introduction to JSP</i>	<i>17</i>
<i>2.3 MS SQL</i>	<i>25</i>
<i>3. System Study</i>	
<i>3.1 Preliminary Investigation</i>	<i>28</i>
<i>3.2 System Development Life Cycle</i>	<i>29</i>
<i>3.3 Feasibility Study</i>	<i>33</i>
<i>3.4 Economic Feasibility</i>	<i>34</i>
<i>3.5 Technical Feasibility</i>	<i>39</i>
<i>3.6 Operational Feasibility</i>	<i>40</i>
<i>3.7 Feasibility Report</i>	<i>40</i>
<i>4. System Analysis</i>	
<i>4.1 Importance of Computerized System</i>	<i>41</i>
<i>4.2 System Analysis</i>	<i>43</i>

<i>4.3 Methods Used For Gathering Information</i>	<i>45</i>
<i>5. System Design</i>	
<i>5.1 Physical Design</i>	<i>50</i>
<i>5.2 Data Flow Diagram</i>	<i>52</i>
<i>5.3 E-R Diagram</i>	<i>57</i>
<i>5.4 Data Modeling</i>	<i>70</i>
<i>5.5 Modules</i>	<i>73</i>
<i>5.6 Scheduling</i>	<i>77</i>
<i>6. Coding</i>	<i>78</i>
<i>7. Code Efficiency</i>	<i>139</i>
<i>8. Optimization of Code</i>	<i>139</i>
<i>9. Testing</i>	<i>140</i>
<i>10. Test Cases</i>	<i>145</i>
<i>11. Verification & Validation</i>	<i>152</i>
<i>12. System Implementation Maintenance and Review</i>	<i>153</i>
<i>12.1 Hardware Evaluation Factors</i>	<i>153</i>
<i>12.2 Software Evaluation Factors</i>	<i>154</i>
<i>12.3 Conversion and Training</i>	<i>155</i>
<i>12.4 Training Needs</i>	<i>156</i>
<i>13. Cost Estimation of the Project</i>	<i>157</i>
<i>14. Security and Validation Checks</i>	<i>161</i>
<i>15. Scope of Future Application</i>	<i>165</i>
<i>16. Conclusion</i>	<i>167</i>
<i>17. Bibliography</i>	<i>168</i>

1. INTRODUCTION AND OBJECTIVES OF THE PROJECT

1.1 Introduction

Online Cab Scheduling System is software which is helpful for Cab travelers, who wants to travel by Cab all over different routes. This system provide different routes for passengers, passengers can also have a look over how many trips are their available for specific route and what is the time table for different traveling. Passengers can book his ticket for his travel by Cab with a specific route and for a specific day.

This system is useful for passenger who wants to know about Cab routes for Travelling between two or many places, how much it will cost and how much time it would take for traveling. This system is also useful for employees like Cab drivers and conductors who are working for this system they can know about his work status information and his working time table with their route information he has to follow over his work.

In this software a project manager works for managing Cab routes, employees and passengers details. He can add a Cab and its details including Cab route details. Project manager allocate work between employees and manage time between multiple routes.

There are mainly four modules in this software:

- ✓ Cab Management
- ✓ Route Management
- ✓ Employee Management
- ✓ Passenger Management

In Cab module a project manager add or delete a Cab details to the database. He can set the details of the route and the timing. Cab details add total number of available

seats, reserved seats and total seating capacity and the route in which the Cab is going to be are added.

Route Management module deals with the route management of the Cab. A project manager can add the route and if he wants he can change it or delete that route. Route modules contain time table for different routes and their related stoppage.

Employee management module is work over maintaining employees details. Employee details of any Cab can add to the database and this is helpful to the Cab operators to provide all work status report and working route and time table report to the employees working with this system.

The passenger module mainly used to give advantage for the passengers. This module is helpful for the passengers to get details of the route and the details of the Cab which have trip to that route. He can also get the information of the number of seats available in a particular day. This system is useful for reserving tickets for specific route and for a specific date. Details of every passenger are stored in the system.

1.2 Objectives of the Project

Online Cab Scheduling System is built to provide facilities to those who are interested in knowing about Cab and their routes. This system provide information regarding all available routes and their time table. If any user wants to know about available seats for specific date this system help him/her for this type of information.

Employees are also facilitated by providing them work status report and their work time table report. In this system all information regarding employees maintained by project manager and work allocation is done by this system manager.

Online Cab Scheduling System is work between passengers and working employees this system is used for benefiting Cab travelers and to provide detailed information regarding Cab routes and availability of seats.

Category of the Project

This project is a "Web Application". It uses the concept of OOPs, RDBMS and Multimedia for Conferencing and Internet technologies.

The Project aims at creating a website using java and MySQL for a web based platform. It uses some features of servlet, applet, JSP and JDBC as well, to speed up a few sections of this project.

I. The User's layer: tools and languages used → J2EE (JSP, Java Server Faces), HTML, DHTML, and JavaScript

Editor: Eclipse

There is a front end which is web-pages which are displayed before the user and user directly interact with the software. Web pages are developed by using various design tools like Flash, Photo Shop etc and documentation languages like HTML and DHTML.

II. The Presentation layer: tools and languages used → JSP, SERVLET, JavaServer Faces

Editor: Eclipse

At this layer all the services related with presentations like format check, mailing interface, parsing, deployment descriptor etc. this layer is developed in java using servlet, JSP.

III. The Integration Level: tool/technology used → JDBC 3.0

JDBC Type 4 Driver is used.

This tier is dedicated for database connectivity and connection management with the database. This tier is developed in java and the components to be used JDBC 2.0.

IV. The Data Resources level: [MySQL Community Server 5.0]

At the back end, there is a data base which maintains data. This is to be developed in MySQL. In this server tables are created and all primary key –foreign key constraints are implemented. The data base follows all the integrity constraints.

Platform used: Microsoft Windows 2000 Advanced Server

Features of Windows 2000 Advanced Server which may contribute to this project

- This is server edition and contains all the features of server.
- Easy to use and configure different components of the system.
- Provide good base to develop enterprise software.
- Provide multi-user facilities to distribute and reintegrate different components of the project easily.

Architecture implemented: Java 2 SDK Enterprise Edition 1.3

1.3 Advantage

I have designed the given proposed system in the JSP, SERVLETS to automate **Online Cab Scheduling System**.

The following steps that give the detailed information of the need of proposed system are:

Performance: During past several decades, the records are supposed to be manually handled for all activities. The manual handling of the record is time consuming and highly prone to error. To improve the performance of the Cab Agencies, the computerized system is to be undertaken. The computerized project is fully computerized and user friendly even that any of the members can see the report and status of the company.

Efficiency: The basic need of this website is efficiency. The website should be efficient so that whenever a new user submit his/her details regarding ticket reservations, cancellations etc. the website is updated automatically. This record will be useful for other users instantly.

Control: The complete control of the project is under the hands of authorized person who has the password to access this project and illegal access is not supposed to deal with. All the control is under the administrator and the other members have the rights to just see the records not to change any transaction or entry.

Security: Security is the main criteria for the proposed system. Since illegal access may corrupt the database. So security has to be given in this project.

1.4 Project Category

I. The User's layer: tools and languages used → JSP, HTML, and DHTML.

Editors: - Eclipse IDE

There is a front end which is web-pages which are displayed before the user and user directly interact with the software. Web pages are developed by using various design tools like Flash, Photo Shop etc and documentation languages like HTML and DHTML.

II. The Presentation layer: tools and languages used → JSP, SERVLET

Editor: Eclipse IDE

At this layer all the services related with presentations like format check, mailing interface, parsing, deployment descriptor etc. this layer is developed in java using servlet, JSP.

III. The Integration Level: tool/technology used → JDBC Type 4

JDBC Type 4 Driver (Pure Java implementation) is used.

This tier is dedicated for database connectivity and connection management with the database. This tier is developed in java and the components to be used JDBC 2.0.

IV. The Data Resources level: [MySQL]

At the back end, there is a data base which maintains data. This is to be developed in MySQL. In this server tables are created and all primary key –foreign key constraints are implemented. The data base follows all the integrity constraints.

Cabiness Intelligence

Gain deeper insight into your Cabiness with integrated, comprehensive analysis and reporting for enhanced decision making.

High Availability

Ensure Cabiness continuity with the highest levels of system availability through technologies that protect your data against costly human errors and minimize disaster recovery downtime.

Performance and Scalability

Deliver an infrastructure that can grow with your Cabiness and has a proven record in handling today's large amounts of data and most critical enterprise workloads.

Security

Provide a secure environment to address privacy and compliance requirements with built-in features that protect your data against unauthorized access.

Manageability

Manage your infrastructure with automated diagnostics, tuning, and configuration to reduce operational costs while reducing maintenance and easily managing very large amounts of data.

Developer Productivity

Build and deploy critical Cabiness-ready applications more quickly by improving developer productivity and reducing project life cycle times.

1.5 Drawbacks of Current Manual- System

- The current manual system has a lot of paper work and it does not deal with exact details.
- To maintain the records of candidates and companies manually, is a Time-consuming job.
- With the increase in database, it will become a massive job to maintain the database.
- Requires large quantities of file cabinets, which are huge and require quite a bit of space in the office, which can be used for storing records of previous claims.
- Lack of security for the records, anyone disarrange the records of your system.

THE NEED OF NEW SYSTEM

- 1. Problem of Reliability:** Current system is not reliable. It seems to vary in quality from one month to the next. Some times it gives good output, but some times the output is worst.
- 2. Problem of Accuracy:** There are too many mistakes in reports.
- 3. Problem of timeliness:** In the current system the reports and output produced is mostly late and in most of the cases it is useless because it is not on time.
- 4. Problem of Validity:** The output and reports mostly contains misleading information. The customer's information is sometimes not valid.
- 5. Problem of Economy:** The current system is very costly. We have to spend lots of money to keep the system up and going, but still not get the desired results.
- 6. Problem of Capacity:** The current system is suffering from problem of capacity also. The staff for organization is very less and the workload is too much. Few peoples cannot handle all the work.

1.6 PROPOSED SYSTEM

1. **Details:** The new proposed system stores and maintains all the online users etc.
2. **Calculations:** The new proposed system updates tables and other information automatically and it is very fast and accurate.
3. **Registers:** There is no need of keeping and maintaining accounts and information manually. It remembers each and every record and we can get any report at any time.
4. **Speed:** The new proposed system is very fast with 100% accuracy and saves time.
5. **Manpower:** The new proposed system needs less manpower. Less people can do the large work.
6. **Efficiency:** The new proposed systems complete the work of many people in less time.
7. **Past details:** The new proposed system contains the details of each enquiry done by visitor.
8. **Reduces redundancy:** The most important benefit of this system is that it reduces the redundancy of data within the data.
9. **Work load:** Reduces the work load of the data store by helping in easy updates of the products and providing them with the necessary details together with financial transactions management.
10. **Easy statements:** Month-end and day-end statement easily taken out without getting headaches on browsing through the day end statements.

1.7 Need

I have designed the given proposed system in the JSP, SERVLETS to automate the process. The following steps that give the detailed information of the need of proposed system are:

- **Performance:** During past several decades, the records are supposed to be manually handled for all activities. The manual handling of the record is time consuming and highly prone to error. To improve the performance of the Company system, the computerized system is to be undertaken. The computerized project is fully computerized and user friendly even that any of the members can see the report and status of the company.
- **Efficiency:** The basic need of this website is efficiency. The website should be efficient so that whenever a new user submits his/her details the website is updated automatically. This record will be useful for other users instantly.
- **Control:** The complete control of the project is under the hands of authorized person who has the password to access this project and illegal access is not supposed to deal with. All the control is under the administrator and the other members have the rights to just see the records not to change any transaction or entry.
- **Security:** Security is the main criteria for the proposed system. Since illegal access may corrupt the database. So security has to be given in this project.

2. PROJECT SELECTION

2.1 Software & Hardware Requirements

The need of today's software development is competence in a GUI based front-end tool, which can connect to Relational Database engines. This gives the programmer the opportunity to develop client server based commercial applications.

FRONT END

JSP programming tools are complete programming environments. It allows programmers to build a GUI program using the various on-screen controls such as buttons, text, menus, boxes etc. These controls are placed on a form and then the processing details related with each control are filled in.

In the Cabiness world, competitive strategies have become the order of the day to improve quality, cut costs and provide a high response customer service base. Most organizations today need to be market driven and do a lot of value addition to their products and services. This naturally calls for rational decision making, which requires information. Information Technology or IT provides that effective channel to support and implement this strategy. Client/Server is the technology that empowers the desktop, thus setting a trend for the way successful organizations will use technology in the next decade.

2.2 Introduction to JSP

Adding dynamic content via expressions

As we saw in the previous section, any HTML file can be turned into a JSP file by changing its extension to .jsp. Of course, what makes JSP useful is the ability to embed Java. Put the following text in a file with .jsp extension (let us call it **hello.jsp**), place it in your JSP directory, and view it in a browser.

```
<HTML>
<BODY>
Hello! The time is now <%= new java.util.Date() %>
</BODY>
</HTML>
```

Notice that each time you reload the page in the browser, it comes up with the current time.

The character sequences `<%=` and `%>` enclose Java expressions, which are evaluated at run time.

This is what makes it possible to use JSP to generate dynamic HTML pages that change in response to user actions or vary from user to user.

Exercise: Write a JSP to output the values returned by `System.getProperty` for various system properties such as `java.version`, `java.home`, `os.name`, `user.name`, `user.home`, `user.dir` etc.

Scriptlets

We have already seen how to embed Java expressions in JSP pages by putting them between the `<%=` and `%>` character sequences.

But it is difficult to do much programming just by putting Java expressions inside HTML.

JSP also allows you to write blocks of Java code inside the JSP. You do this by placing your Java code between `<%` and `%>` characters (just like expressions, but without the `=` sign at the start of the sequence.)

This block of code is known as a "scriptlet". By itself, a scriptlet doesn't contribute any HTML (though it can, as we will see down below.) A scriptlet contains Java code that is executed every time the JSP is invoked.

Here is a modified version of our JSP from previous section, adding in a scriptlet.

```
<HTML>
<BODY>
<%
    // This is a scriptlet. Notice that the "date"
    // variable we declare here is available in the
    // embedded expression later on.
    System.out.println( "Evaluating date now" );
    java.util.Date date = new java.util.Date();
%>
```

```
Hello! The time is now <%= date %>
</BODY>
</HTML>
```

If you run the above example, you will notice the output from the "System.out.println" on the server log. This is a convenient way to do simple debugging (some servers also have techniques of debugging the JSP in the IDE. See your server's documentation to see if it offers such a technique.)

By itself a scriptlet does not generate HTML. If a scriptlet wants to generate HTML, it can use a variable called "out". This variable does not need to be declared. It is already predefined for scriptlets, along with some other variables. The following example shows how the scriptlet can generate HTML output.

```
<HTML>
<BODY>
<%
    // This scriptlet declares and initializes "date"
    System.out.println( "Evaluating date now" );
    java.util.Date date = new java.util.Date();
%>
Hello! The time is now
<%
    // This scriptlet generates HTML output
    out.println( String.valueOf( date ));
%>
</BODY>
</HTML>
```

Here, instead of using an expression, we are generating the HTML directly by printing to the "out" variable. The "out" variable is of type [javax.servlet.jsp.JspWriter](#).

Another very useful pre-defined variable is "request". It is of type [javax.servlet.http.HttpServletRequest](#)

A "request" in server-side processing refers to the transaction between a browser and the server. When someone clicks or enters a URL, the browser sends a "request" to the server for that URL, and shows the data returned. As a part of this "request", various data is available, including the file the browser wants from the server, and if the request is coming from pressing a SUBMIT button, the information the user has entered in the form fields.

The JSP "request" variable is used to obtain information from the request as sent by the browser. For instance, you can find out the name of the client's host (if available, otherwise the IP address will be returned.) Let us modify the code as shown:

```
<HTML>
<BODY>
<%
    // This scriptlet declares and initializes "date"
    System.out.println( "Evaluating date now" );
    java.util.Date date = new java.util.Date();
%>
```

Hello! The time is now

```
<%
    out.println( date );
    out.println( "<BR>Your machine's address is " );
    out.println( request.getRemoteHost());
%>
</BODY>
</HTML>
```

A similar variable is "response". This can be used to affect the response being sent to the browser. For instance, you can call `response.sendRedirect(anotherUrl);` to send a response to the browser that it should load a different URL. This response will actually go all the way to the browser. The browser will then send a different request, to "anotherUrl". This is a little different from some other JSP mechanisms we will come across, for including another page or forwarding the browser to another page.

Exercise: Write a JSP to output the entire line, "Hello! The time is now ..." but use a scriptlet for the complete string, including the HTML tags.

Mixing Scriptlets and HTML

We have already seen how to use the "out" variable to generate HTML output from within a scriptlet. For more complicated HTML, using the out variable all the time loses some of the advantages of JSP programming. It is simpler to mix scriptlets and HTML.

Suppose you have to generate a table in HTML. This is a common operation, and you may want to generate a table from a SQL table, or from the lines of a file. But to keep our example simple, we will generate a table containing the numbers from 1 to N. Not very useful, but it will show you the technique.

Here is the JSP fragment to do it:

```
<TABLE BORDER=2>
<%
    for ( int i = 0; i < n; i++ ) {
        %>
        <TR>
        <TD>Number</TD>
        <TD><%= i+1 %></TD>
        </TR>
    }
%>
</TABLE>
```

You would have to supply an int variable "n" before it will work, and then it will output a simple table with "n" rows.

The important things to notice are how the %> and <% characters appear in the middle of the "for" loop, to let you drop back into HTML and then to come back to the scriptlet.

The concepts are simple here -- as you can see, you can drop out of the scriptlets, write normal HTML, and get back into the scriptlet. Any control expressions such as a "while" or a "for" loop or an "if" expression will control the HTML also. If the HTML is inside a loop, it will be emitted once for each iteration of the loop.

Another example of mixing scriptlets and HTML is shown below -- here it is assumed that there is a boolean variable named "hello" available. If you set it to true, you will see one output, if you set it to false, you will see another output.

```
<%
  if ( hello ) {
    %>
    <P>Hello, world
    <%
  } else {
    %>
    <P>Goodbye, world
    <%
  }
%>
```

It is a little difficult to keep track of all open braces and scriptlet start and ends, but with a little practice and some good formatting discipline, you will acquire competence in doing it.

Exercise: Make the above examples work. Write a JSP to output all the values returned by System.getProperties with "
" embedded after each property name and value. Do not output the "
" using the "out" variable.

JSP Directives

We have been fully qualifying the java.util.Date in the examples in the previous sections. Perhaps you wondered why we don't just import java.util.*;

It is possible to use "import" statements in JSPs, but the syntax is a little different from normal Java. Try the following example:

```
<%@ page import="java.util.*" %>
<HTML>
<BODY>
<%
  System.out.println( "Evaluating date now" );
  Date date = new Date();
%>
Hello! The time is now <%= date %>
</BODY>
</HTML>
```

The first line in the above example is called a "directive". A JSP "directive" starts with <%@ characters.

This one is a "page directive". The page directive can contain the list of all imported packages. To import more than one item, separate the package names by commas, e.g.

```
<%@ page import="java.util.*,java.text.*" %>
```

There are a number of JSP directives, besides the page directive. Besides the page directives, the other most useful directives are include and taglib. We will be covering taglib separately.

The include directive is used to physically include the contents of another file. The included file can be HTML or JSP or anything else -- the result is as if the original JSP file actually contained the included text. To see this directive in action, create a new JSP

```
<HTML>
<BODY>
Going to include hello.jsp...<BR>
<%@ include file="hello.jsp" %>
</BODY>
</HTML>
```

View this JSP in your browser, and you will see your original hello.jsp get included in the new JSP.

Exercise: Modify all your earlier exercises to import the java.util packages.

JSP Declarations

The JSP you write turns into a class definition. All the scriptlets you write are placed inside a single method of this class.

You can also add variable and method declarations to this class. You can then use these variables and methods from your scriptlets and expressions.

To add a declaration, you must use the <%! and %> sequences to enclose your declarations, as shown below.

```
<%@ page import="java.util.*" %>
<HTML>
<BODY>
<%!
    Date theDate = new Date();
    Date getDate()
    {
        System.out.println( "In getDate() method" );
        return theDate;
    }
%>
Hello! The time is now <%= getDate() %>
</BODY>
</HTML>
```

The example has been created a little contrived, to show variable and method declarations.

Here we are declaring a Date variable theDate, and the method getDate. Both of these are available now in our scriptlets and expressions.

But this example no longer works! The date will be the same, no matter how often you reload the page. This is because these are declarations, and will only be evaluated once when the page is loaded! (Just as if you were creating a class and had variable initialization declared in it.)

Exercise: Modify the above example to add another function `computeDate` which re-initializes `theDate`. Add a scriptlet that calls `computeDate` each time.

Note: Now that you know how to do this -- it is in general not a good idea to use variables as shown here. The JSP usually will run as multiple *threads* of one single instance. Different threads would interfere with variable access, because it will be the same variable for all of them. If you do have to use variables in JSP, you should use *synchronized* access, but that hurts the performance. In general, any data you need should go either in the *session* object or the *request* object (these are introduced a little later) if passing data between different JSP pages. Variables you declare inside *scriptlets* are fine, e.g. `<% int i = 45; %>` because these are declared inside the local scope and are not shared.

JSP Tags

Another important syntax element of JSP are tags. JSP tags do not use `<%`, but just the `<` character. A JSP tag is somewhat like an HTML tag. JSP tags can have a "start tag", a "tag body" and an "end tag". The start and end tag both use the tag name, enclosed in `<` and `>` characters. The end starts with a `/` character after the `<` character. The tag names have an embedded colon character `:` in them, the part before the colon describes the type of the tag. For instance:

```
<some:tag>
body
</some:tag>
```

If the tag does not require a body, the start and end can be conveniently merged together, as

```
<some:tag/>
```

Here by closing the start tag with a `/>` instead of `>` character, we are ending the tag immediately, and without a body. (This syntax convention is the the same as XML.)

Tags can be of two types: loaded from an external tag library, or predefined tags. Predefined tags start with **jsp:** characters. For instance, `jsp:include` is a predefined tag that is used to include other pages.

We have already seen the `include` directive. `jsp:include` is similar. But instead of loading the text of the included file in the original file, it actually calls the included target at run-time (the way a browser would call the included target. In practice, this is actually a simulated request rather than a full round-trip between the browser and the server). Following is an example of `jsp:include` usage

```
<HTML>
<BODY>
Going to include hello.jsp...<BR>
<jsp:include page="hello.jsp"/>
</BODY>
</HTML>
```

Try it and see what you get. Now change the "jsp:include" to "jsp:forward" and see what is the difference. These two predefined tags are frequently very useful.

Exercise: Write a JSP to do either a forward or an include, depending upon a boolean variable (hint: The concepts of mixing HTML and scriptlets work with JSP tags also!)

JSP Sessions

On a typical web site, a visitor might visit several pages and perform several interactions.

If you are programming the site, it is very helpful to be able to associate some data with each visitor. For this purpose, "session"s can be used in JSP.

A session is an object associated with a visitor. Data can be put in the session and retrieved from it, much like a Hashtable. A different set of data is kept for each visitor to the site.

Here is a set of pages that put a user's name in the session, and display it elsewhere. Try out installing and using these.

First we have a form, let us call it GetName.html

```
<HTML>
<BODY>
<FORM METHOD=POST ACTION="SaveName.jsp">
What's your name? <INPUT TYPE=TEXT NAME=username SIZE=20>
<P><INPUT TYPE=SUBMIT>
</FORM>
</BODY>
</HTML>
```

The target of the form is "SaveName.jsp", which saves the user's name in the session. Note the variable "session". This is another variable that is normally made available in JSPs, just like out and request variables. (In the @page directive, you can indicate that you do not need sessions, in which case the "session" variable will not be made available.)

```
<%
String name = request.getParameter( "username" );
session.setAttribute( "theName", name );
%>
<HTML>
<BODY>
<A HREF="NextPage.jsp">Continue</A>
</BODY>
</HTML>
```

The SaveName.jsp saves the user's name in the session, and puts a link to another page, NextPage.jsp.

NextPage.jsp shows how to retrieve the saved name.

```
<HTML>
<BODY>
```

```
Hello, <%= session.getAttribute( "theName" ) %>  
</BODY>  
</HTML>
```

If you bring up two different browsers (not different windows of the same browser), or run two browsers from two different machines, you can put one name in one browser and another name in another browser, and both names will be kept track of.

The session is kept around until a timeout period. Then it is assumed the user is no longer visiting the site, and the session is discarded.

Projecthelpline.in

2.3 My SQL

- **Introduction**

- MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons:
- MySQL is released under an open-source license. So you have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table.
 - ◀ The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

What is SQL Used for:

1. Scalability and Flexibility

The MySQL database server provides the ultimate in scalability, sporting the capacity to handle deeply embedded applications with a footprint of only 1MB to running massive data warehouses holding terabytes of information. Platform flexibility is a stalwart feature of MySQL with all flavors of Linux, UNIX, and Windows being supported. And, of course, the open source nature of MySQL allows complete customization for those wanting to add unique requirements to the database server.

2. High Performance

A unique storage-engine architecture allows database professionals to configure the MySQL database server specifically for particular applications, with the end result being amazing performance results. Whether the intended application is a high-speed transactional processing system or a high-volume web site that services a billion queries a day, MySQL can meet the most demanding performance expectations of any system. With high-speed load utilities, distinctive memory caches, full text indexes, and other performance-enhancing mechanisms, MySQL offers all the right ammunition for today's critical business systems.

3. High Availability

Rock-solid reliability and constant availability are hallmarks of MySQL, with customers relying on MySQL to guarantee around-the-clock uptime. MySQL offers a variety of high-availability options from high-speed master/slave replication configurations, to specialized Cluster servers offering instant failover, to third party vendors offering unique high-availability solutions for the MySQL database server.

4. Robust Transactional Support

MySQL offers one of the most powerful transactional database engines on the market. Features include complete ACID (atomic, consistent, isolated, durable) transaction support, unlimited row-level locking, distributed transaction capability, and multi-version transaction support where readers never block writers and vice-versa. Full data integrity is also assured through server-enforced referential integrity, specialized transaction isolation levels, and instant deadlock detection.

5. Web and Data Warehouse Strengths

MySQL is the de-facto standard for high-traffic web sites because of its high-performance query engine, tremendously fast data insert capability, and strong support for specialized web functions like fast full text searches. These same strengths also apply to data warehousing environments where MySQL scales up into the terabyte range for either single servers or scale-out architectures. Other features like main memory tables, B-tree and hash indexes, and compressed archive tables that reduce storage requirements by up to eighty-percent make MySQL a strong standout for both web and business intelligence applications.

6. Strong Data Protection

Because guarding the data assets of corporations is the number one job of database professionals, MySQL offers exceptional security features that ensure absolute data protection. In terms of database authentication, MySQL provides powerful mechanisms for ensuring only authorized users have entry to the database server, with the ability to block users down to the client machine level being possible. SSH and SSL support are also provided to ensure safe and secure connections. A granular object privilege framework is present so that users only see the data they should, and powerful data encryption and decryption functions ensure that sensitive data is protected from unauthorized viewing. Finally, backup and recovery utilities provided through MySQL and third party software vendors allow for complete logical and physical backup as well as full and point-in-time recovery.

PROJ

3. SYSTEM STUDY

3.1 Preliminary Investigation

System development, a process consisting of two major steps of system analysis and design, start when management or sometimes system development personnel feel that a new system or an improvement in the existing system is required. The system development life cycle is classically thought of as the set of activities that analysts, designers and users carry out to develop and implement an information system. The system development life cycle consists of the following activities:

- Preliminary investigation
- Determination of system requirements
- Design of system
- Development of software
- System testing
- Implementation, evaluation, and maintenance

A request to take assistance from information system can be made for many reasons, but in each case someone in the organization initiates the request is made, the first system activity the preliminary investigation begins. This activity has three parts:

- 1) Request clarification
- 2) Feasibility study
- 3) Request approval

Request clarification: Many requests from employees and users in the organizations are not clearly defined, therefore it becomes necessary that project request must be examined and clarified properly before considering systems investigation.

3.2 System Development Life Cycle

Systems are created to solve problems. One can think of the systems approach as an organized way of dealing with a problem. In this dynamic world, the subject System Analysis and Design (SAD), mainly deals with the software development activities.

DEFINING A SYSTEM

A collection of components that work together to realize some objective forms a system. Basically there are three major components in every system, namely input, processing and output.

In a system the different components are connected with each other and they are interdependent. For example, human body represents a complete natural system. We are also bound by many national systems such as political system, economic system, educational system and so forth. The objective of the system demands that some output is produced as a result of processing the suitable inputs.

SYSTEM LIFE CYCLE

System life cycle is an organizational process of developing and maintaining systems. It helps in establishing a system project plan, because it gives overall list of processes and sub-processes required for developing a system.

System development life cycle means combination of various activities. In other words we can say that various activities put together are referred as system development life cycle. In the System Analysis and Design terminology, the system development life cycle means software development life cycle.

Following are the different phases of software development cycle:

- System study
- Feasibility study
- System analysis
- System design
- Coding
- Testing
- Implementation
- Maintenance

Projecthelpline.in

The Different Phases Of Software Development Life Cycle Are Shown Below.

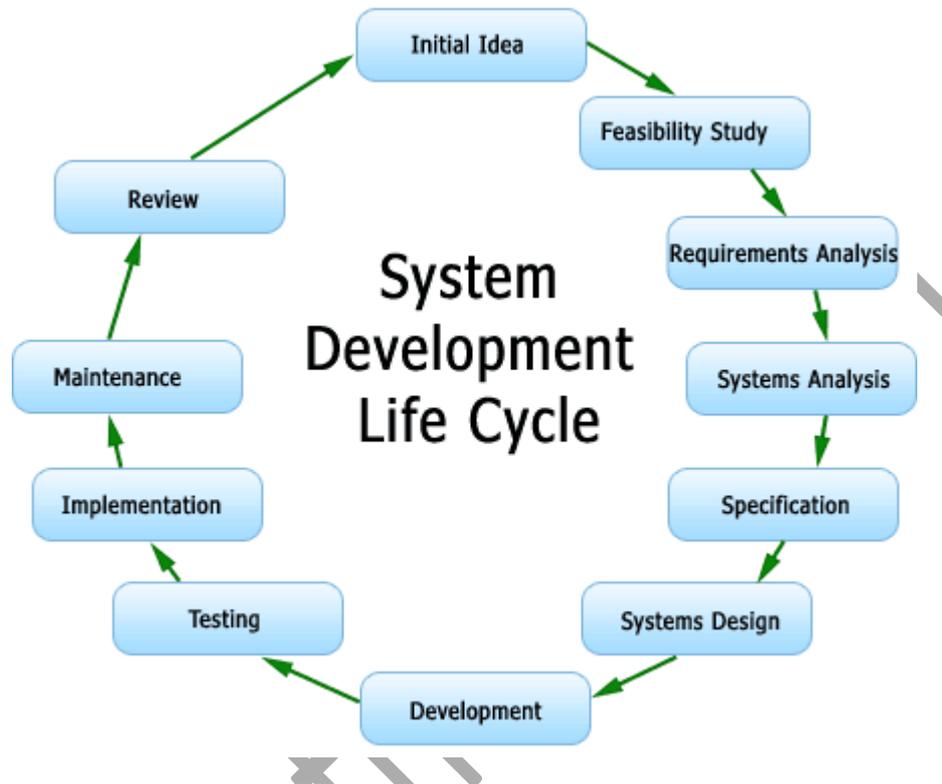


FIG: SHOWING GENERAL LIFE CYCLE PROCESS AND PERCENTAGE OF TIME DEVOTED

A system analysis is a separation of a substance into parts for study and their implementation and detailed examination.

Before designing any system it is important that the nature of the Cabiness and the way it currently operates are clearly understood. The detailed examination provides the specific data required during designing in order to ensure that all the client's requirements are fulfilled. The investigation or the study conducted during the analysis phase is largely based on the feasibility study. Rather it would not be wrong to say that the analysis and feasibility phases overlap. High-level analysis begins during the feasibility study. Though analysis is represented as one phase of the system development life cycle (SDLC), this is not true. Analysis begins with system initialization and continues until its maintenance. Even after successful implementation of the system, analysis may play its role for periodic maintenance and up gradation of the system.

One of the main causes of project failures is inadequate understanding, and one of the main causes of inadequate understanding of the requirements is the poor planning of system analysis.

Analysis requires us to recall the objectives of the project and consider following three questions:

- ✓ What type of information is required?
- ✓ What are the constraints on the investigation?
- ✓ What are the potential problems that may make the task more difficult?

Projecthelpline.in

3.3 Feasibility Study

The basic premise of system analysis is being done here. The primary goal of the system analysis stage is to identify problems and determine how they can be solved with the computer system. In formal SDLC methodologies, the first step in system analysis is feasibility study. A feasibility study is the quick examination of the problems, goals, expected cost of the system. The objective is to determine whether the problem can reasonably be solved with a computer system. In some cases, there may be a better alternative, or perhaps it is simply a short-term annoyance and will gradually disappear. In other cases, the problem may turn out to be more complex than was thought and involves users across the company. Also, some problems may not be solvable with today's technology. It might be better to wait for better technology. In any case, you need to determine the scope of the project to gain a better idea of cost, benefits, and objectives. The feasibility study is typically written so that non-programmers can easily understand it. It is used to "sell" the project to the upper management and as a starting point for the next step. Additionally, it is used as a reference to keep the project on track, and to evaluate the progress of the project team. Is the project cost-effective or is there a cheaper solution? Will the proposed system improve the operation of the bank; will complicating factors prevent it from achieving its goals? Does the technology exist and does the firm have the staff to make the technology work? When the proposal is determined to be feasible, the team leaders are appointed and a plan and schedule are created. The schedule contains a detailed listing of what parts of the project are completed at each time. Of course, it is extremely difficult to estimate the true cost and completion dates. Nonetheless, the schedule is an important tool to evaluate the status of the project and the progress of the team.

Steps in feasibility Analysis are:

1. Identify deficiency by pinpointing, missing functions, unsatisfactory, performance, Excessive cost of operations.
2. Set goals to remove these deficiencies.
3. Goals must be quantified, realizable within the constraints of an organization, broken down into sub goals agreeable to all concerned.
4. Set goals not only to remove deficiencies but also to effectively meet competition. For instance, goals must be based on what competitors do.

3.4 Economic Feasibility

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More commonly known as cost / benefit analysis; in this procedure we determine the benefits and savings that are expected from a proposed system and compare them with costs. We found the benefits outweigh the costs; we take a decision to design and implement the new proposed system.

During the feasibility phase, broad alternative solutions are examined. For each alternate solution the cost and benefits have to be examined before designing one of the alternatives.

Broad solutions will consist of:

1. Specifications of information to be made available by the system.
2. Description of what will be done manually and what the computer will do.
3. Specification of new computing equipment needed or specification of expansion of an existing computer.

➤ COST AND BENEFIT ANALYSIS

Developing an IT application is an investment. Since after developing that application it provided the organization with profits. Profits can be monetary or in the form of an improved working environment. However, it carries risks because in

some cases an estimate can be wrong and the project might not actually turn out to be beneficial.

Cost benefit analysis helps to give management a picture of the cost, benefits and risks. It usually involves comparing alternate investments.

Cost benefit determines the benefits and savings that are expected from the system and compares them with the expected cost.

In performing cost and benefit analysis it is important to identify cost and benefits factors. Cost and benefits can be categorized into the following categories:

- i. Development cost** - Development costs is the cost that are incurred during the development of the system. It is one time investment.
- ii. Operating cost** - Operating cost are the expenses required for the day to-day running of the system. As, operating cost are wages, supplies and overheads.
- iii. Hardware/Software cost** - It includes the cost of purchasing or leasing of computes and it's peripherals. Software costs involves required software cost.
- iv. Personnel cost** - It is the money spent on the people involved in the development of the system.
- v. Facility cost** - Expenses that are incurred during the preparation of the physical site where the system will be operational. These can be wiring, flooring, acoustics, lighting, and air-conditioning.
- vi. Supply cost** - These are variable costs that are very proportionately with the amount of use of paper, ribbons, disks, and others.

➤ **BENEFITS**

We can define benefits as

Profit or Benefit = Income - Cost

Benefits can be accrued by:

Increasing income or

Decreasing costs or Both

My proposed project to Online Cab Scheduling System does everything those 3 - 4 employees or accountants are currently doing on paperwork except on a computer. Due to this factor, if the organization goes ahead with my project, they would not need any personnel, and their costs of sustaining the organization go down radically, the software itself requires minimal memory to run as files are stored in a very defragmented manner and can easily be moved around as well a zipped, to preserve even more space. Hence, funds spent on storage, are almost trifling, will not pose a problem in the future while operating my project.

All of this comes at a very low price of a computer and my software, which, if we consider all the benefits organization will be getting out of it, is completely woo* it.

Let's take the costs required to assemble and run my project Online Cab scheduling System into consideration:

ITEM	COST (Rs.)
Computer	40,000 (depend upon the configuration)
Laser Printer	7,000
Tomcat Server	
My Sql	
Scanner	4,000
Project cost	4,000 (approx.)
Total	55,000

We should know that we need a person who can work with these software's and so his training will cost around Rs 4,000. Here we see that the total price to get one computer up and running with the database management system is close to Rs.50,000 as we don't really need a very high performance computer as this project is entirely text based and requires only minimal processor speeds for computing. If the organization wants to have around 2 computers then we have the total costs being:

$$55,000 * 2 = \text{Rs. } 1,10,000$$

This would mean that the this project would need around 1 lakh to run my system successfully and resourceful companies are planning to reduce the staff, suppose if they remove one employee. Salary of one employee being around Rs.5,000. In one year annual salary of employee is equal to Rs. 60,000 and salary to three employees is Rs. 1,80,000 for one year. From this analysis it can be seen that whatever money the companies will invest on making their system a computerized system will be recovered and half years of time (approx.). This analysis shows that how management is benefited by computerized system.

BENEFITS:

- 1) Fast and easy access to all Online Cab Scheduling System related procedures and functions.
- 2) No need for large storage spaces sized of rooms for storing the cabinets because all the information about the members and other details is saved in the computer's hard disks.
- 3) High level of security and authentication of each and every user.
- 4) Reliability is increased, as backups of files, and records can be made and saved in various different locations and information will be highly secure, unlike in file cabinets where entries can easily be ripped or tampered with by users.
- 6) The reception/front office will look much more neater and cleaner the environment they need, as there Won't be any cupboards or drawers which make the Companies overcrowded.
- 7) There will be no longer the need for all the paper work required to make timely reports lists or other lists as the program generates them at anytime at a very quick pace.

3.5 Technical feasibility

Today, very little is technically impossible. Consequently, technical feasibility looks at what is practical and reasonable. Technical feasibility addresses three major issues:

1. Is the proposed technology or solution practical?
2. Do we currently possess the necessary technology?
3. Do we possess the necessary technical expertise, and is the schedule reasonable?

Is the Proposed Technology or Solution Practical?

The technology for any defined solution is normally available. The question whether that technology is mature enough to be easily applied to our problems. Some firms like to use state-of-the-art technology, but most firms prefer to use mature and proven technology. A mature technology has a larger customer base for obtaining advice concerning problems and improvements.

Do We Currently Possess the Necessary Technology?

Assuming the solution's required technology is practical, we must next ask ourselves, is the technology available in our information systems shop? If the technology is available, we must ask if we have the capacity. For instance, will our current printer be able to handle the new reports and forms required of a new system?

If the answer to any of these questions is no, then we must ask ourselves, Can we get this technology? The technology may be practical and available, and, yes, we need it. But we simply may not be able to afford it at this time. Although this argument borders on economic feasibility, it is truly technical feasibility. If we can't afford the technology, then the alternative that requires the technology is not practical and is technically infeasible!

3.6 Operational Feasibility

It is mainly related to human organizational and political aspects. The points to be considered are:

- o What changes will be brought with the system?
- o What organizational structures are disturbed?
- o What new skills will be required? Do the existing staff members have these skills?
- o If not, can they be trained in due course of time?

Generally project will not be rejected simply because of operational infeasibility but such considerations are likely to critically affect the nature and scope of the eventual recommendations.

For operational feasibility study we appointed a small group of people who are familiar with information system techniques, who understand the parts of the Cabiness that are relevant to the project and are skilled in system analysis and design process.

3.7 FEASIBILITY REPORT

After studying the feasibility of the project we came to the following points, these results may change according to further analysis and design.

PROJECT NAME: ONLINE CAB SCHEDULING SYSTEM

DEFINITION OF PROBLEM OR OPPORTUNITY: We have to make a computerized system (software) to make the working of companies easy and efficient so that software will replace the manual work with automated computerized process.

EXPECTED BENEFITS:

- Reduce the number of employee.
- Save money.
- Increase the efficiency of workers.
- Reduce the response time.
- Improve the service quality.
- Reduce the bulk of paper work.
- Reduce the chance of error by human.
- Increase the accuracy in result.

4. SYSTEM DESIGN

4.1 Importance of Computerized

ONLINE CAB SCHEDULING SYSTEM

There are several attributes in which the computer based information works. Broadly the working of computer system is divided into two main groups:

- ◆ Transaction System
- ◆ Decision Support System

Transaction System:

A transaction is a record of some well-defined single and usually small occurrence in a system. Transactions are input into the computer to update the database files. It checks the entering data for its accuracy. This means that numeric data appears in numeric field and character data in character field. Once all the checks are made, transaction is used to update the database. Transaction can be inputted in on-line mode or batch mode. In on-line mode, transactions are entered and updated into the database almost instantaneously. In batch mode, transactions are collected into batches, which may be held for a while and inputted later.

Decision Support System:

It assists the user to make analytical decision. It shows the various data in organized way called analysis. This analysis can be made to syrdy preferences and help in making decisions.

Computer system works out best with record maintenance. It will tell you which customer would get how much pending/reports statements. It will also help to search the information about a particular person by simply entering his telephone number. User can store information as per requirement, which can be used for comparison with other reports.

PRINCIPLES OF SYSTEM ANALYSIS

Principles:

1. Understand the problem before you begin to create the analysis model.
2. Develop prototypes that enable a user to understand how human machine interaction will occur.
3. Record the origin of and the reason for every requirement.
4. Use multiple views of requirements like building data, function and behavioral models.
5. Work to eliminate ambiguity.

A Complete Structure:

The limited time and resources have restricted us to incorporate, in this project, only the main activities that are performed in news sites, but utmost care has been taken to make the system efficient and user friendly.

For the optimum use of practical time it is necessary that every session is planned.

Planning of this project will include the following things:

- Topic Understanding.
- Modular Break – Up of the Syst
- Processor Logic for Each Module.
- Database Requirements.

Topic Understanding:

It is vital that the field of application as introduced in the project may be totally a new field. So as soon as the project was allocated to me, I carefully went through the project to identify the requirements of the project.

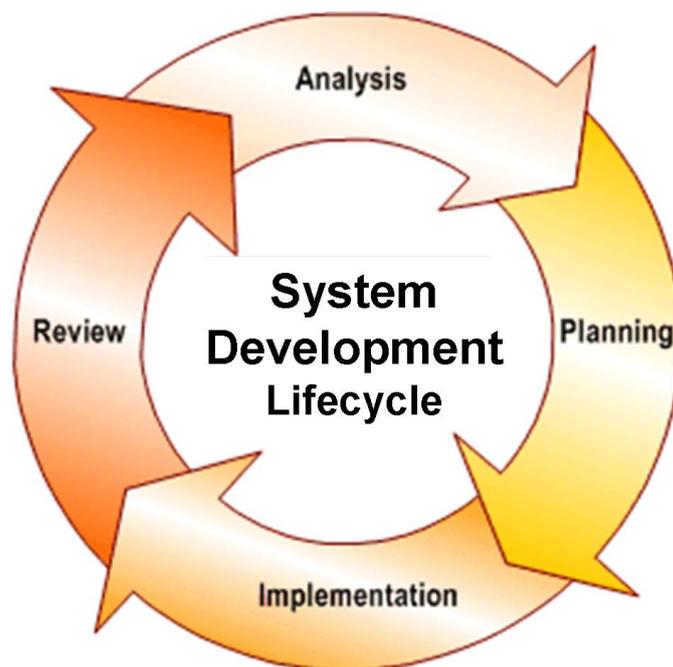
Modular Break –Up of the System:

- Identify The Various Modules In The System.
- List Them In The Right Hierarchy.
- Identify Their Priority Of Development
- Description Of The Modules:

4.2 System Analysis

A system analysis is a separation of a substance into parts for study and their implementation and detailed examination.

Before designing any system it is important that the nature of the Cabiness and the way it currently operates are clearly understood. The detailed examination provides the specific data required during designing in order to ensure that all the client's requirements are fulfilled. The investigation or the study conducted during the analysis phase is largely based on the feasibility study. Rather it would not be wrong to say that the analysis and feasibility phases overlap. High-level analysis begins during the feasibility study. Though analysis is represented as one phase of



the system development life cycle (SDLC), this is not true. Analysis begins with system initialization and continues until its maintenance. Even after successful implementation of the system, analysis may play its role for periodic maintenance and up gradation of the system.

One of the main causes of project failures is inadequate understanding, and one of the main causes of inadequate understanding of the requirements is the poor planning of system analysis.

Analysis requires us to recall the objectives of the project and consider following three questions:

FIG: SHOWING GENERAL LIFE CYCLE PROCESS AND PERCENTAGE OF TIME DEVOTED

- What are the constraints on the investigation?
- What are the potential problems that may make the task more difficult?

Keeping the above questions in mind and considering the survey conducted to determine the need of the system, the total system was designed and can be described as under:

The three major parts of the system are:

➤ **Providing Information:**

The system is effectively used to provide large variety of information to the interested customer. The major purpose of the site is to easily provide booking details, stock, sales with quick update to latest modifications in the records. This thing is not at all possible in printed material, which are updated only once a few weeks. It also gives information about the general usage of the system for first time visitors. The system itself works as a information provider for Company.

Alert when available: Through the survey it was clearly that there is a need to device an alternative way for providing alert facility to the user. Sometimes the product which customer demand is not available at that moment, user can register demand of customer and when its available, system gives an alert to the user that customer had registered a customer request with the same match.

Constraints: After the objectives were clear during the analysis phase, it was essential to understand the constraints in order to plan and avoid problems arising during detailed analysis.

Technology - the customer may be committed to a particular hardware or software solution. The software required in this case is: compete Java developer kit, Microsoft windows environment for MS - access.

Budget - if budget is a real constraint, the budget of the new system proposed would be constantly compared with that of the existing system or any Alternatives solution. In this case during the economic feasibility study it has been clearly proved that the new system is definitely more feasible than the alternative solution possible. Organization must implement a system which saves the effort, also its provide an easy method for customer who investigate each detail itself.

Scope - what is the area under investigation in this project? What are the boundaries of the system? What is the extent of possible usage of the new system? More and more people are now having access to organization and watch independently Details of new upcoming stock. Hence the scope is constantly increasing. However its usage can be increased many folds with a little investment from the organization side by implanting touch screen computer kiosks at various convenient positions at the service station.

Environmental Analysis:

The external entities for an organization are its Supplier's customers or any individual.

4.3 Methods Used For Gathering Information

The methods used for gathering information about the existing information system are as followed.

- (a) Review of records.
- (b) Observation of the functioning system.
- (c) Interviews.
- (d) Questionnaires.

In order to create an informative and practical system, a system analyst would have to have some kind of way to view the current system. Receiving feed back on what can be done to improve the current system, and how much the current system is acceptable to the users.

1) Requirement analysis: -

The main part of problem is to obtain a clear understanding of the needs of user and what exactly are desired from the software. It is used for specifying the requirement.

Fact finding tools:

After obtaining the background knowledge, I began to collect data on the existing system's output, input, and costs. The tools used in data collection / information gathering are:

- Review of the written Documents
- On-site Observation
- Interviews
- Questionnaires

Review of the written documents:

In this phase we analyzed all the documents like the day books, supply report, order generating forms, supply forms, account etc. All these things describe the format and functions of the current system included in most manuals are system requirement that help determine how III various objectives are met.

The form is one of the most important source through which I draw some conclusion like:

1. Who use the form(s)? How important are they to the user?
2. Do the forms include all the necessary information? What item should be added or deleted?
3. How readable and easy to follow is the form?
4. How does the information in the form help other user make better decision?
5. What other uses does the form offer the user area? By analyzing all the details we draw a conclusion that what are the merit and De-merit of the current phase. Will the company contain all the back up of all the important document of not sales person contains all the information about the available vehicles or not. But above all there are some problems with the on site observations that one analyst must face during analysis like:

1. Take long time and get inefficient result
2. Attitude and motivation of subject cannot be readily observed
3. Observation are subject to error

4. In a complex situation it can be very time consuming. So for this we switched towards the other fact finding tools like interviews and questionnaires.

5. SYSTEM DESIGN

The design document that we will develop during this phase is the blueprint of the software. It describes how the solution to the customer problem is to be built. Since solution to complex problems isn't usually found in the first try, iterations are most likely required. This is true for software design as well. For this reason, any design strategy, design method, or design language must be flexible and must easily accommodate changes due to iterations in the design. Any technique or design needs to support and guide the partitioning process in such a way that the resulting sub-problems are as independent as possible from each other and can be combined easily for the solution to the overall problem. Sub-problem independence and easy combination of their solutions reduces the complexity of the problem. This is the objective of the partitioning process. Partitioning or decomposition during design involves three types of decisions: -

Define the boundaries along which to break;

Determine into how many pieces to break; and

Identify the proper level of detail when design should stop and implementation should start.

Basic design principles that enable the software engineer to navigate the design process suggest a set of principles for software design, which have been adapted and extended in the following list:

Free from the suffer from "tunnel vision." A good designer should consider alternative approaches, judging each based on the requirements of the problem, the resources available to do the job.

The design should be traceable to the analysis model. Because a single element of the design model often traces to multiple requirements, it is necessary to have a means for tracking how requirements have been satisfied by the design model.

The design should not repeat the same thing. Systems are constructed using a set of design patterns, many of which have likely been encountered before. These patterns should always be chosen as an alternative to reinvention. Time is short and

resources are limited! Design time should be invested in representing truly new ideas and integrating those patterns that already exist.

The design should "minimize the intellectual distance" between the software and the problem as it exists in the real world. That is, the structure of the software design should (whenever possible) mimic the structure of the problem domain.

The design should exhibit uniformity and integration. A design is uniform if it appears that one person developed the entire thing. Rules of style and format should be defined for a design team before design work begins. A design is integrated if care is taken in defining interfaces between design components.

The design activity begins when the requirements document for the software to be developed is available. This may be the SRS for the complete system, as is the case if the waterfall model is being followed or the requirements for the next "iteration" if the iterative enhancement is being followed or the requirements for the prototype if the prototyping is being followed. While the requirements specification activity is entirely in the problem domain, design is the first step in moving from the problem domain toward the solution domain. Design is essentially the bridge between requirements specification and the final solution for satisfying the requirements.

The design of a system is essentially a blueprint or a plan for a solution for the system. We consider a system to be a set of components with clearly defined behavior that interacts with each other in a fixed defined manner to produce some behavior or services for its environment. A component of a system can be considered a system, with its own components. In a software system, a component is a software module.

The design process for software systems, often, has two levels. At the first level, the focus is on deciding which modules are needed for the system, the specifications of these modules, and how the modules should be interconnected. This is what is called the system design or top-level design. In the second level, the internal design of the modules, or how the specifications of the module can be satisfied, is decided. This design level is often called detailed design or logic design. Detailed design essentially expands the system design to contain a more detailed description of the processing logic and data structures so that the design is sufficiently complete for coding.

Because the detailed design is an extension of system design, the system design controls the major structural characteristics of the system. The system design has a major impact on the testability and modifiability of a system, and it impacts its efficiency. Much of the design effort for designing software is spent creating the system design.

The input to the design phase is the specifications for the system to be designed. Hence, reasonable entry criteria can be that the specifications are stable and have been approved, hoping that the approval mechanism will ensure that the specifications are complete, consistent, unambiguous, etc. The output of the top-level design phase is the architectural design or the system design for the software system to be built. This can be produced with or without using a design methodology. Reasonable exit criteria for the phase could be that the design has been verified against the input specifications and has been evaluated and approved for quality.

A design can be object-oriented or function-oriented. In function-oriented design, the design consists of module definitions, with each module supporting a functional abstraction. In object-oriented design, the modules in the design represent data abstraction (these abstractions are discussed in more detail later). In the function-oriented methods for design and describe one particular methodology the structured design methodology in some detail. In a function-oriented design approach, a system is viewed as a transformation function, transforming the inputs to the desired outputs. The purpose of the design phase is to specify the components for this transformation function, so that each component is also a transformation function. Hence, the basic output of the system design phase, when a function oriented design approach is being followed, is the definition of all the major data structures in the system, all the major modules of the system, and how the modules interact with each other.

Once the designer is satisfied with the design he has produced, the design is to be precisely specified in the form of a document. To specify the design, specification languages are used. Producing the design specification is the ultimate objective of the design phase. The purpose of this design document is quite different from that of the design notation. Whereas a design represented using the design notation is largely to be used by the designer, a design specification has to be so precise and complete that it can be used as a basis of further development by other

programmers. Generally, design specification uses textual structures, with design notation helping in understanding.

5.1 Physical Design

The design phase focuses on the detailed implementation of the system recommended in the feasibility. Emphasis is on translating performance specifications into design specifications. The design phase is a transition from user-oriented document to a programmer-oriented document.

a. Design Methodology:

Design Methodology is a way to transform the "art" of system analysis and design into an "engineering - type" discipline. It explains the relationship amongst various modules and programs within the system. It standardizes the approach to analysis and design, simplifies design by segmentation, improves documentation and subsequent maintenance and enhancements.

The following structured diagram can appropriately represent the relationship between various modules.

b. Design Overview:

In analyzing the present system a great deal of information was collected during the investigation and feasibility phases through list of problems and requirements, interview reports, questionnaires, onsite observations, manuals and determining potential solutions.

It is important to record this information in an unambiguous, concise manner which will be clear and accessible to others, and which can be used by other analysts and designers involved in developing the system. Structured techniques help us to record the information in this way, using diagrams and minimum amount of the text.

Structured analysis is a set of techniques and graphical tools that allow the analyst to develop a new kind of system specification that are easily understandable to the user. The traditional approach of organizing data through flowcharts support future developments and simplify communication with the user but focus on the cost/benefit and feasibility analysis, project management,

hardware and software selection, and personal considerations. In contrast, structured analysis considers new goals and structured tools for analysis, which provide the basis for design and implementation.

c. Process Modeling:

System design goes through two phases of development: logical and physical. Logical implementation represented by Data Flow Diagram shows the logical flow of a system and defines the boundaries of the system it describes the input (source), outputs (destinations), data bases (data stores), and procedures (data flows) - all in the format that meets the user's requirements. The logical implementation of the whole project can be represented as under through Data Flow Diagrams (DFD).

Projecthelpline.in

5.2 Data Flow Diagram

Data flow diagrams are the most commonly used way of documenting the processing of the candidate system. As their name suggests they are a pictorial way of representing the flow of data into, around, and out of the system. They are easily understandable and are less prone to misinterpretation than textual description. A complete set of DFDs provides a compact top - down representation of the system, which makes it easier for the user and the analyst to envisage the system as a whole.

DFDs are constructed using four major components:

- **External entities** - represents the sources of the data that enter the system or the recipients of the system that leave the system.

for example - passenger is the usual receiver of information and supplier of data during form filling.

- **Data stores** - represent the stores of the data within the system example: computer files, databases or in the manual system files, etc. data stores can not be linked directly by data flows either to each other or to external entities without an intervening process to transform them.

- **Processes** - represent activities in which data is manipulated by being stored or retrieved or transformed in some way.

Process names are generally unambiguous and convey as much meaning as possible without being too long. Example: verify data, acquired time schedule etc.

Data flows - represents the movement of data between other components.

Data flow diagrams are used to describe how the system transforms information.

They define how information is processed and stored and identify how the information flows through the processes.

- i. Bubbles:** - A circle or bubble represents that transform data from once form to another by performing some tasks with the data.
- ii. Data store:-**A data store is place where data is held temporarily from one transaction to next or is stored permanently.
- iii. External Entity:-**Which defines a source or destination of system data also called an external entity Based on the working process of the proposed system Data Flow Diagram (DFD) is a model, Which gives the insight into the

information domain and functional domain at the same time can be drawn using OMT symbols, DFD is refined into different levels. The more refined DFD is details of the system are incorporated. In the process of creating a DFD, we decompose the system into different functional subsystems. The DFD refinement results in a corresponding refinement of data.

0 level Data Flow Diagram:

The data flow diagram is pictorial representation of the flow of data in a system. It identifies data flow data stores and functions and sources.

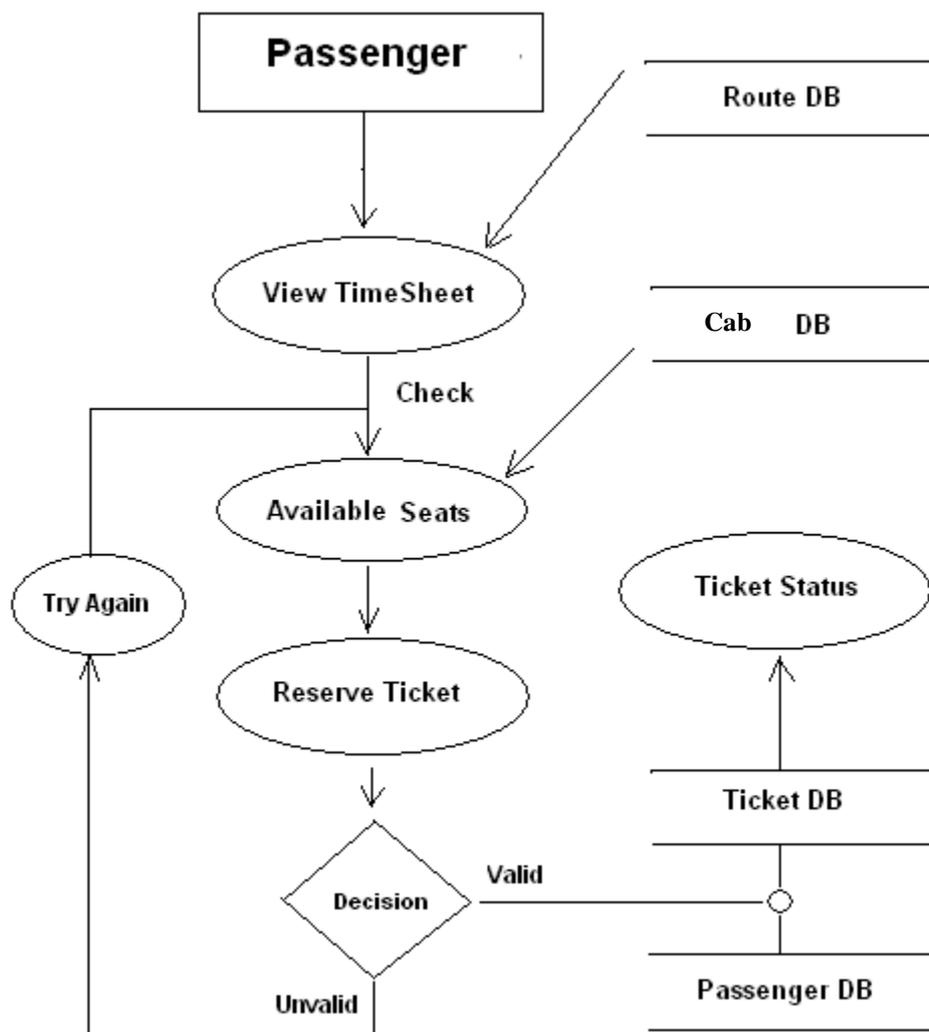
Dataflow diagram show The flow of data and flow of logics involved.

DFD show the passage of data through the system. It focuses on the processes that transform incoming data flows into outgoing data flows

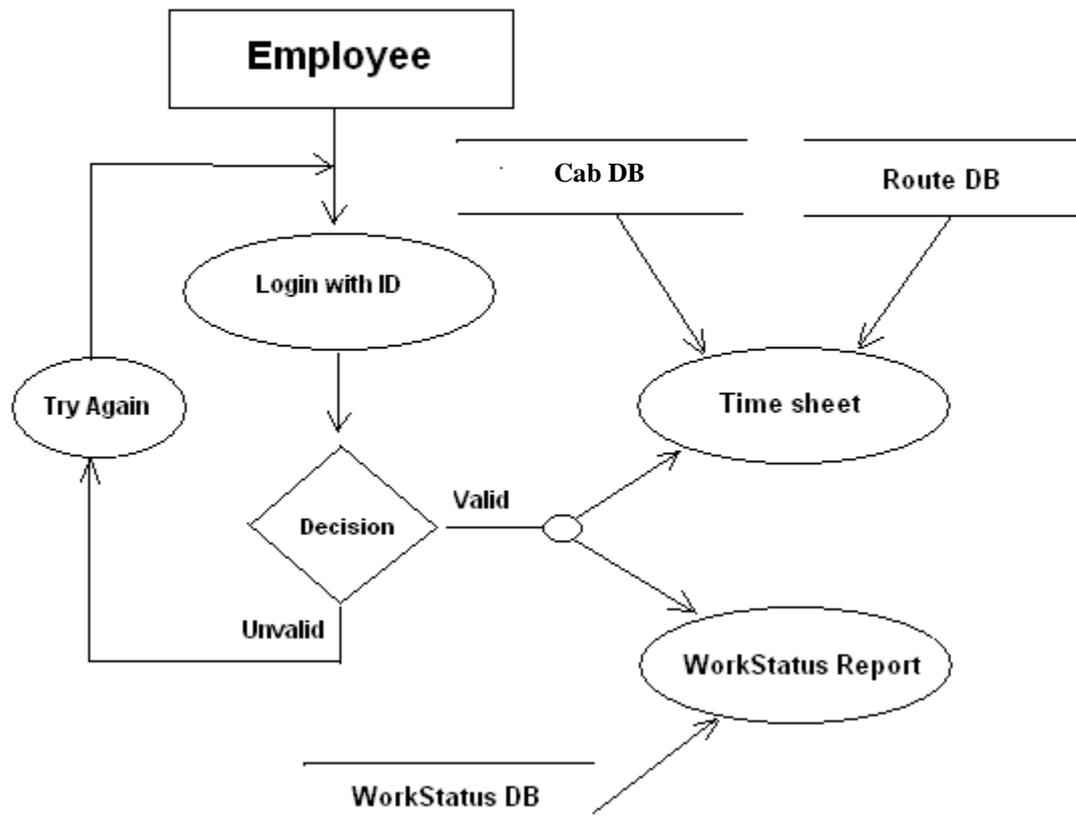
Projecthelpline.in

Passenger Data Flow Diagram:

Below data flow diagram is used to show how user of this system interact with this system. Firstly passenger take a look for available route time table, after checking time table passenger go for checking availability of seats for required route, after this passenger go for reservation for seats on a specific route. If passenger request is right than seat will be reserved and passenger details would send to passenger data base and than to Ticket data base after that ticket status report will be generated, if request is not valid than passenger will be send back previous page.



Employees Data Flow Diagram:



This above diagram is used to show data flow from employees to their output results. In this firstly employees has to login to the system by using his unique ID number if this number is valid than employee can access his time sheet and his work status report which information is provided by the stored data base of the system.

These above diagram shows flow of input data and than process data by system and produce output by processing input data. Above two diagrams show student and faculty member activity between the system.

ER-Diagram

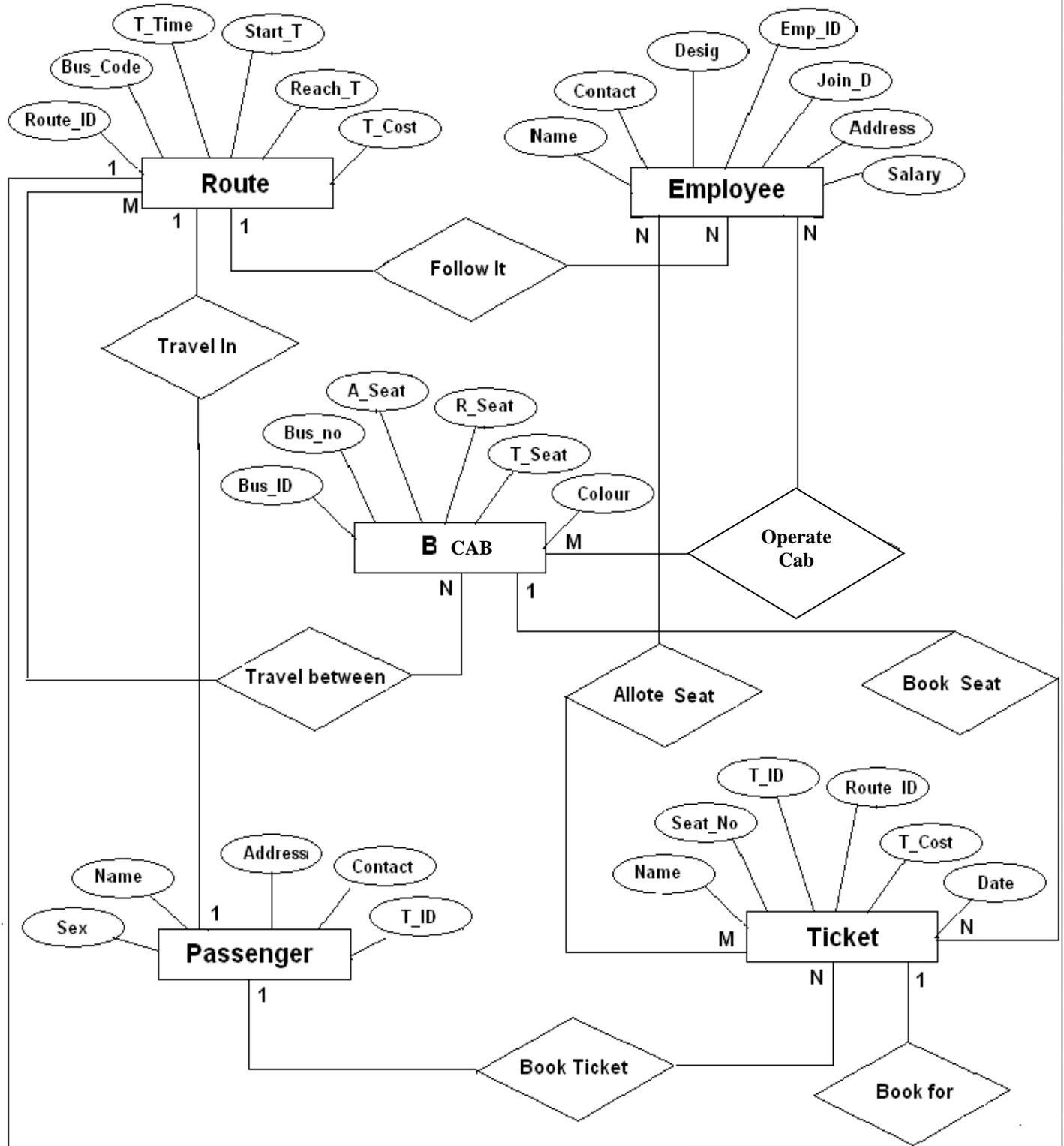
Entity relationship diagram is a high-level conceptual model. This diagram describes data as a collection of entities, relationships and attributes.

ER-Diagrams are composed of:

- ✓ Rectangles representing entity sets.
- ✓ Ellipses representing attributes.
- ✓ Diamonds representing relationship sets.

Projecthelpline.in

5.3 E-R Diagram for Online Cab Scheduling System:



In above diagram we represent Route, Employees, Cab, Ticket and Passenger as entities and their relationship by using diamond shape, numbers associated with these entities represent their relationship status if they are in many to many, one to many or one to one relationship with each other. In ER-Diagram we represent attributes which are in ellipse form, these are used to represent character sticks hold by entities. Above diagram we have Cab_ID, Cab_no, A_seat, R_seat, T_seat and Colour in ellipse form these are attributes of Cab entity and use for specifying character sticks of Cab. All other attributes associated with entities are used for representing specification of that entity.

In our ER-Diagram we have follow it relationship between Route and Employees, Employees and Cab has Operate relationship this relationship contain many to many association, Passenger and Ticket have Book Ticket relationship and contain one to many association. Route and Cab have many to many relationship with 'follow it' relation name. Such kind of other relationships are there and this system follow all the such relationships while creating data base over this system.

This all attributes, entities and their relationships are formed as ER-Diagram.

Feasibility Study

Whatever we think need not be feasible .It is wise to think about the feasibility of any problem we undertake. Feasibility is the study of impact, which happens in the organization by the development of a system. The impact can be either positive or negative. When the positives nominate the negatives, then the system is considered feasible. Here the feasibility study can be performed in two ways such as technical feasibility and Economical Feasibility.

Technical Feasibility:

We can strongly says that it is technically feasible, since there will not be much difficulty in getting required resources for the development and maintaining the system as well. All the resources needed for the development of the software as well as the maintenance of the same is available in the organization here we are utilizing the resources which are available already.

Economical Feasibility

Development of this application is highly economically feasible .The organization needed not spend much more for the development of the system already available. The only thing is to be done is making an environment for the development with an effective supervision. If we are doing so, we can attain the maximum usability of the corresponding resources .Even after the development, the organization will not be in a condition to invest more in the organization .There fore, the system is economically feasible.

INPUT DESIGN

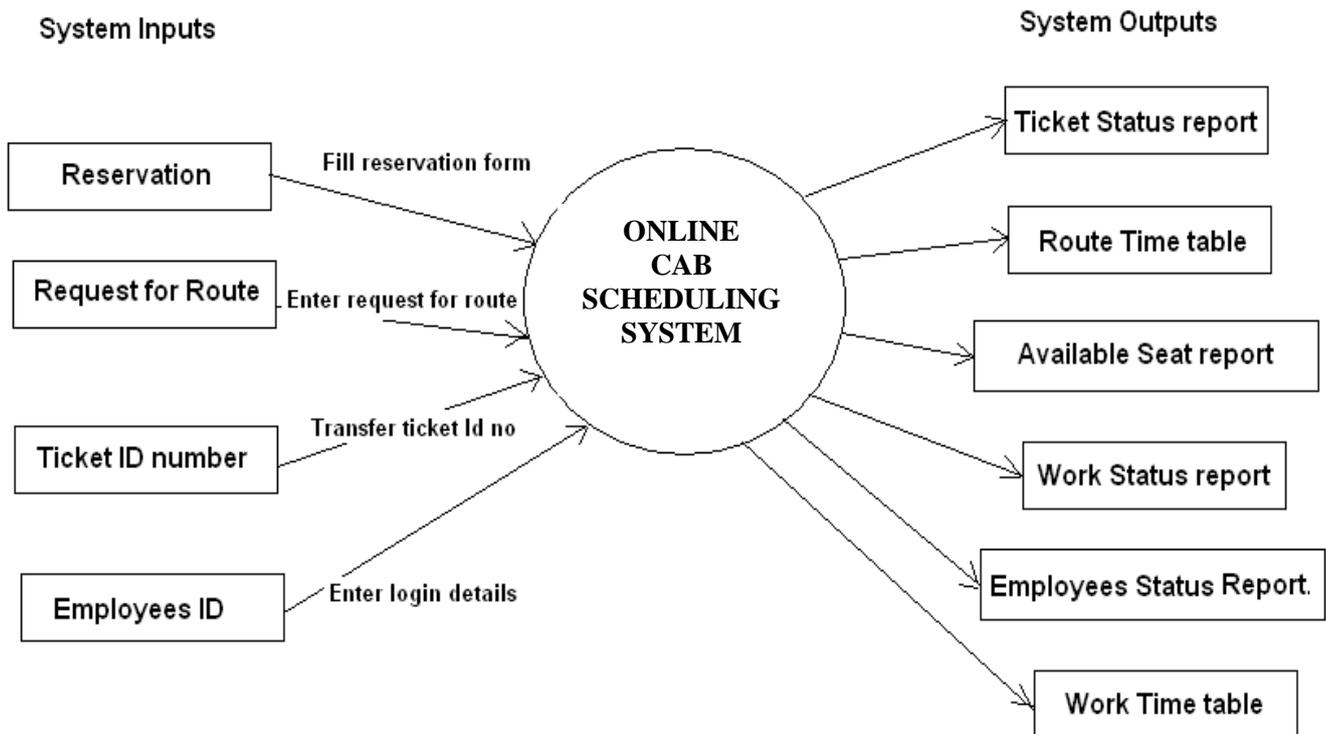
Input design is the process of converting user-oriented input to a computer based format. Input design is a part of overall system design, which requires very careful attention.

In this system main input needed is as following:

1. Passengers Request for Reservation.
2. Passengers Request for Route Time table.
3. Ticket ID number for knowing Ticket Status Report.
4. Employees ID for accessing Employees Status Report.

The goal of designing input data is to make enter easy, logical and free from errors as possible. The format in which the data fields are entered should be given in the input form .Here data entry is online it makes use of processor that accepts commands and data from the operator through a key board. The input required is analyzed by the processor. It is then accepted or rejected.

Input and Output Design Diagram:



Output Design

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of these result for latter consultation .Computer output is the most important and direct source of information to the users. Designing computer output should proceed in an organized well throughout the manner. The right output must be available for the people who find the system easy to use.

Main Outputs for this system are as following:

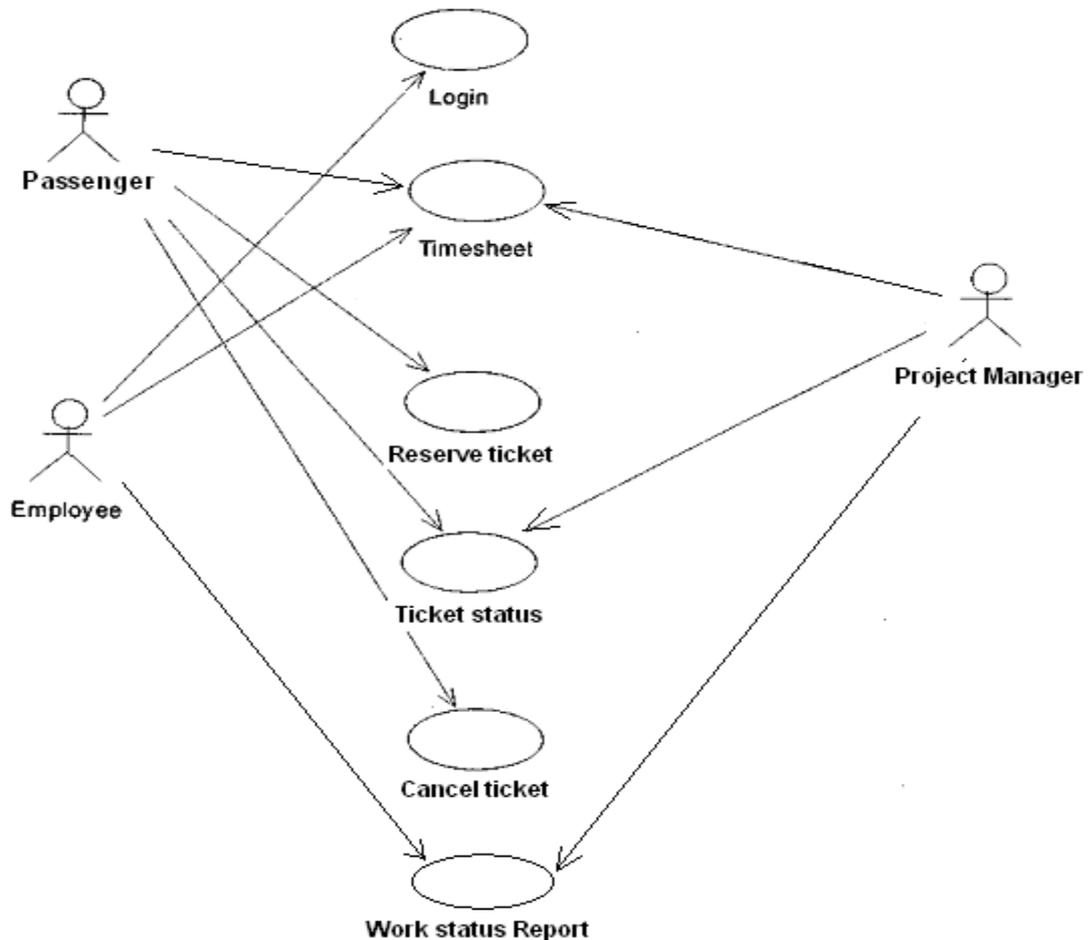
1. Ticket Status report.
2. Employee Work Status report
3. Employee Work Time table
4. Route Time table
5. Available Seat report

The outputs have been defined during the logical design stage.

Use Case Diagrams

Use case diagram describes the interaction between the system and users. This shows which actor performs which action so that it is a graphical representation of actor and their associative actions.

Projecthelp

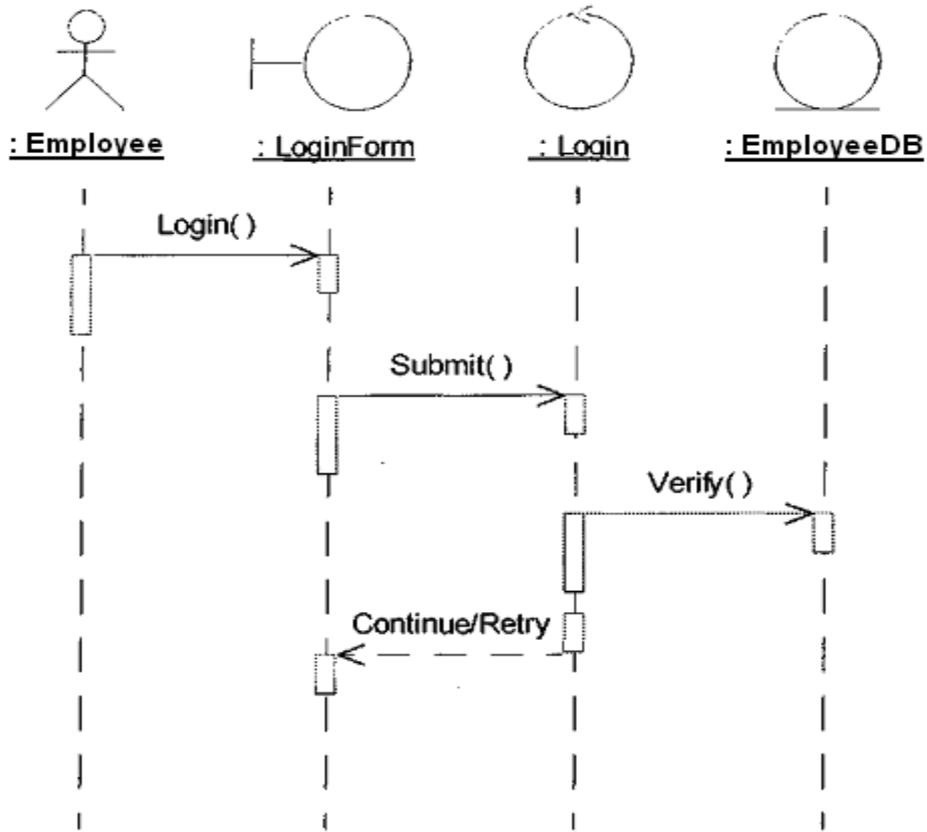


Above use case diagram specifies who is going to do which work for the system, so that it specifies actor and action diagram.

Sequence Diagrams

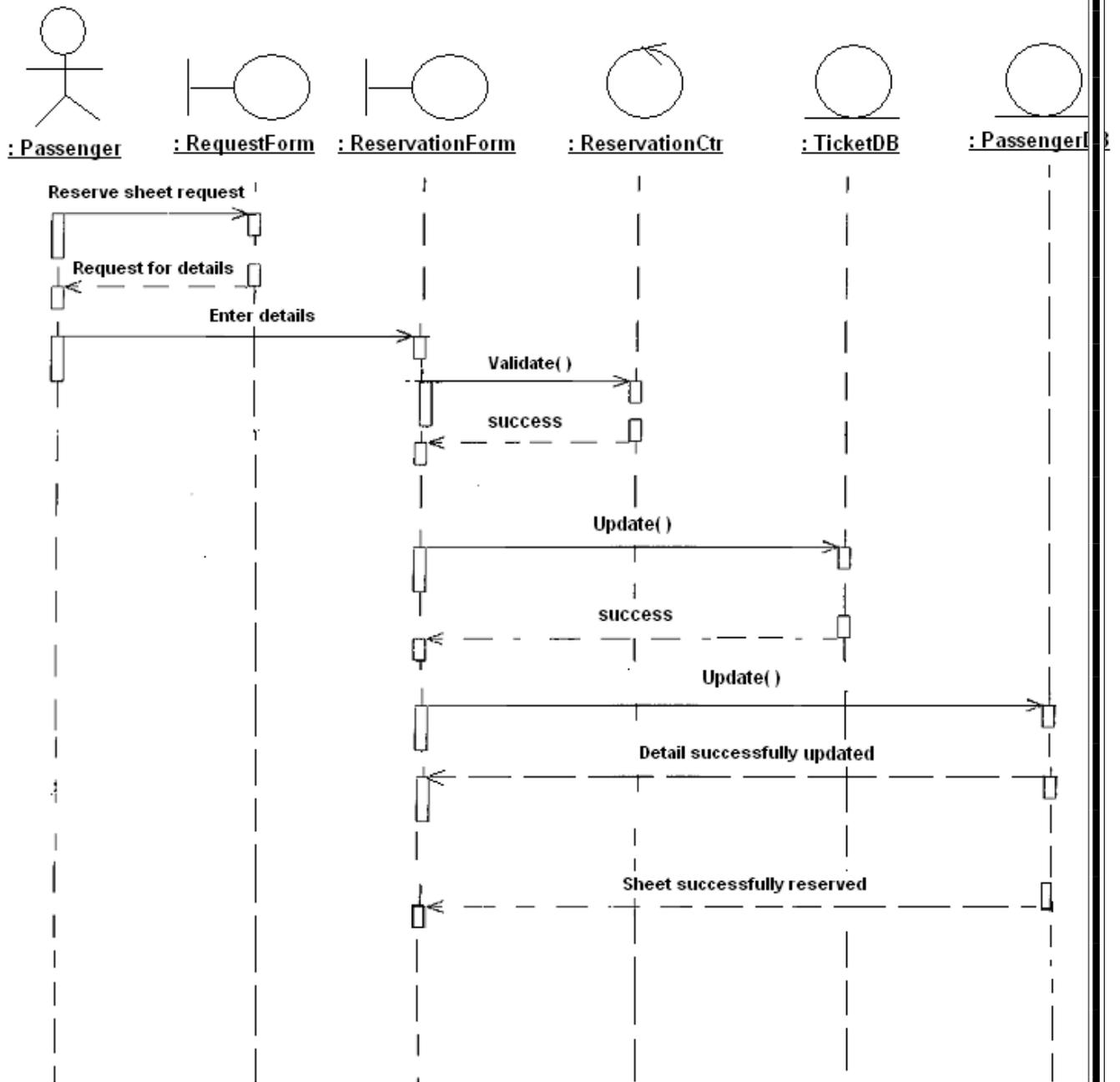
Sequence Diagram is used to show object interaction in a time-based sequence of what happened first, what happened next and so on. Below diagram show how a user interact with system and one by one what happened in the system.

Employees Login :

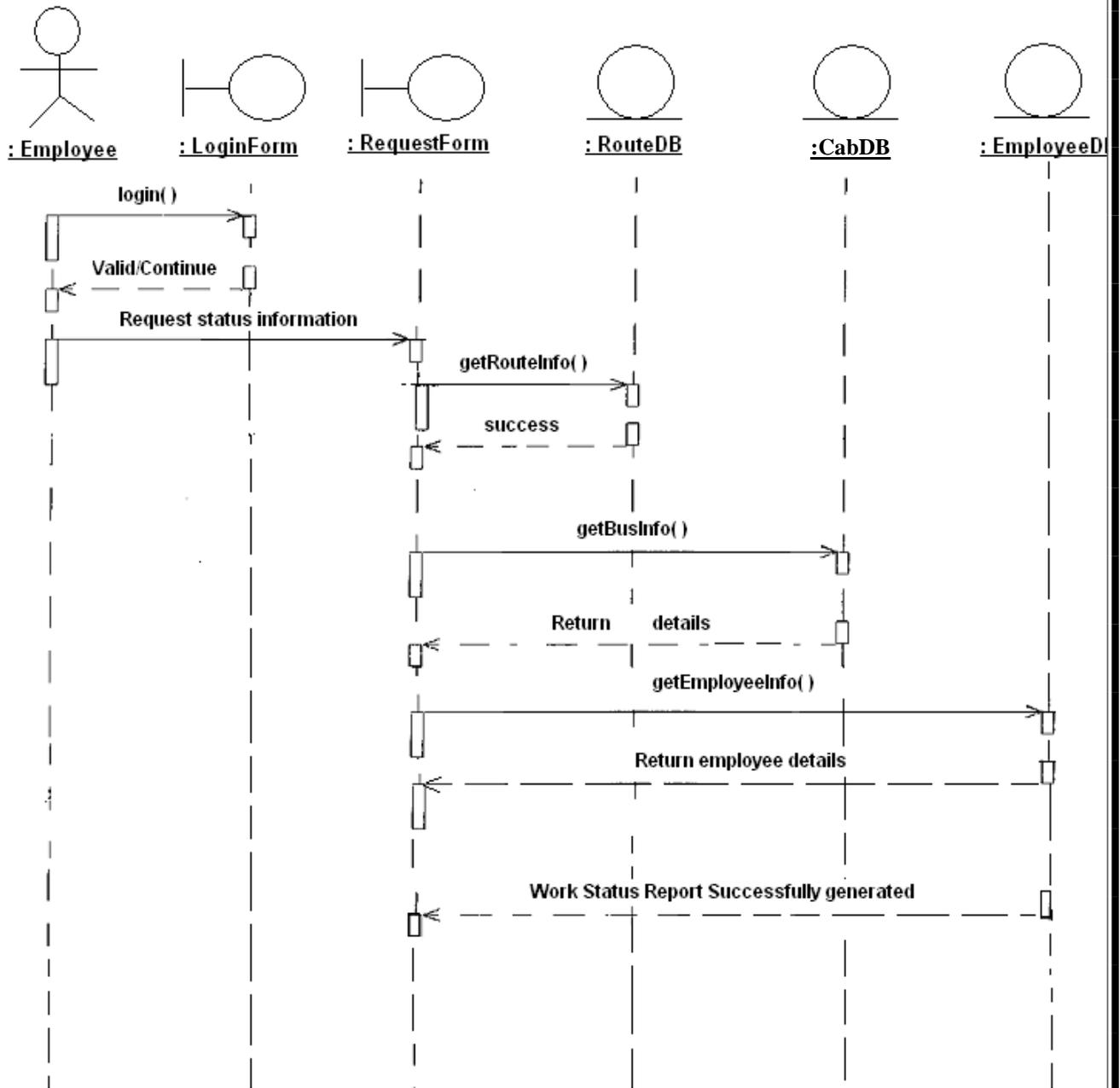


Project

Reserve Ticket:



Work Status Report for Employees:

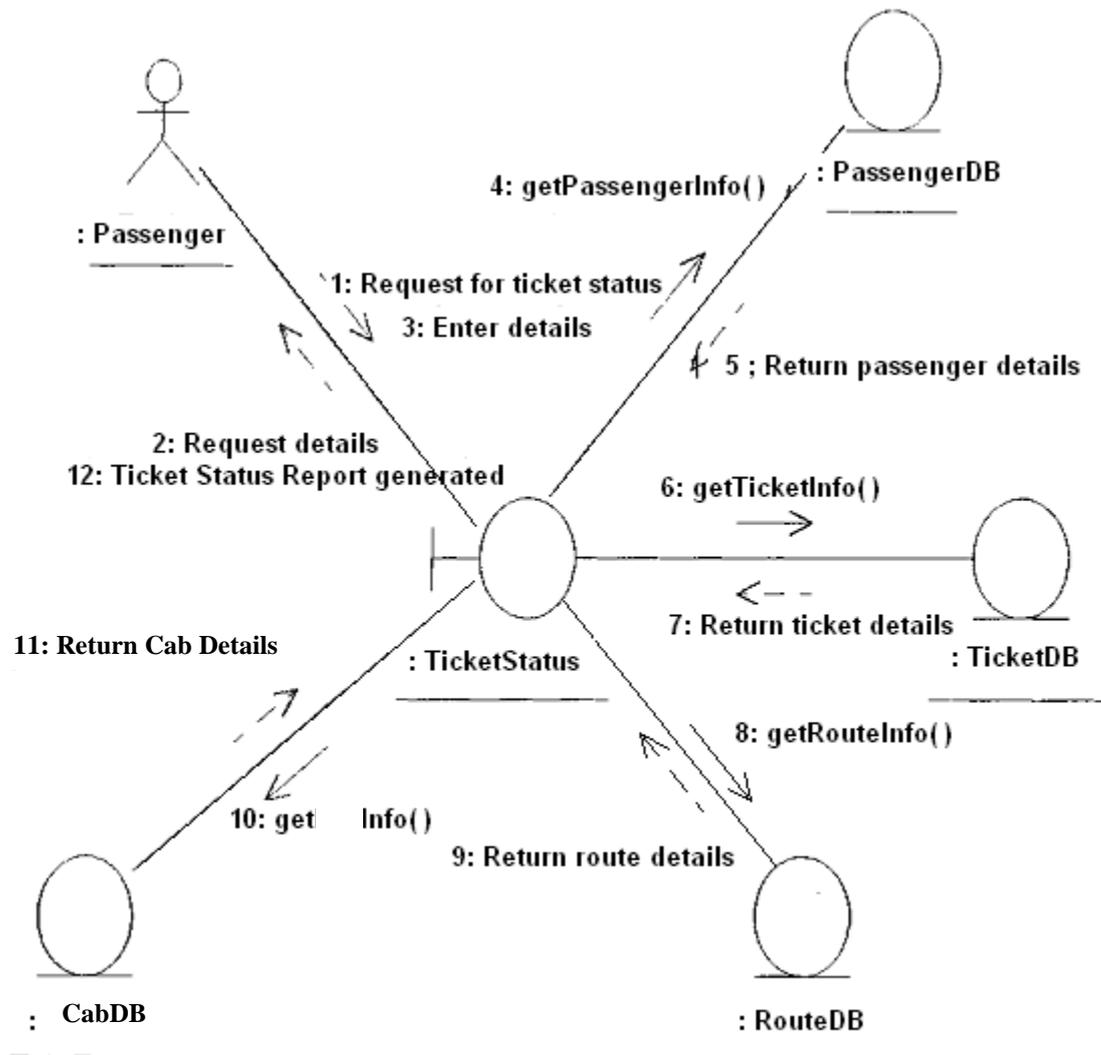


Collaboration Diagrams

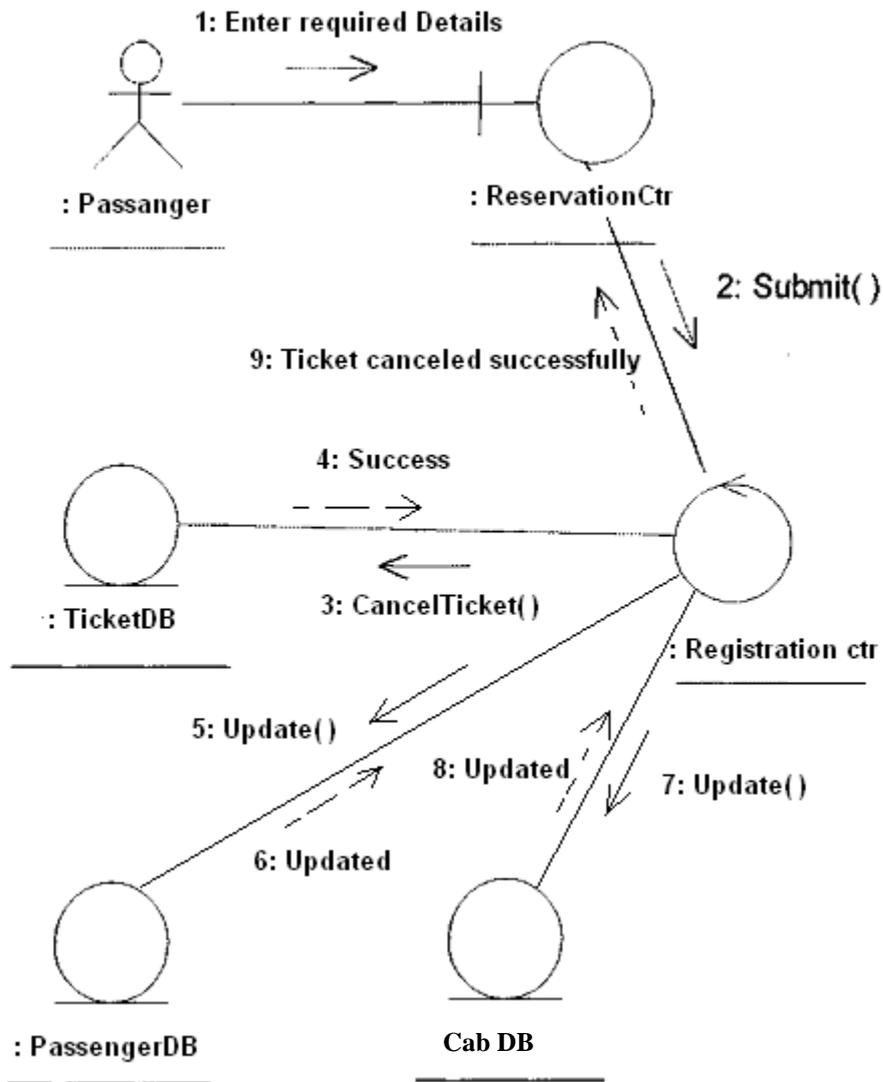
A Collaboration diagram is an interaction diagram that shows the order of messages that implement an operation or a transaction. These below diagrams show objects, their links, and their messages. Sequence diagrams and collaboration diagrams are alternate representations of an interaction.

Projecthelpline.in

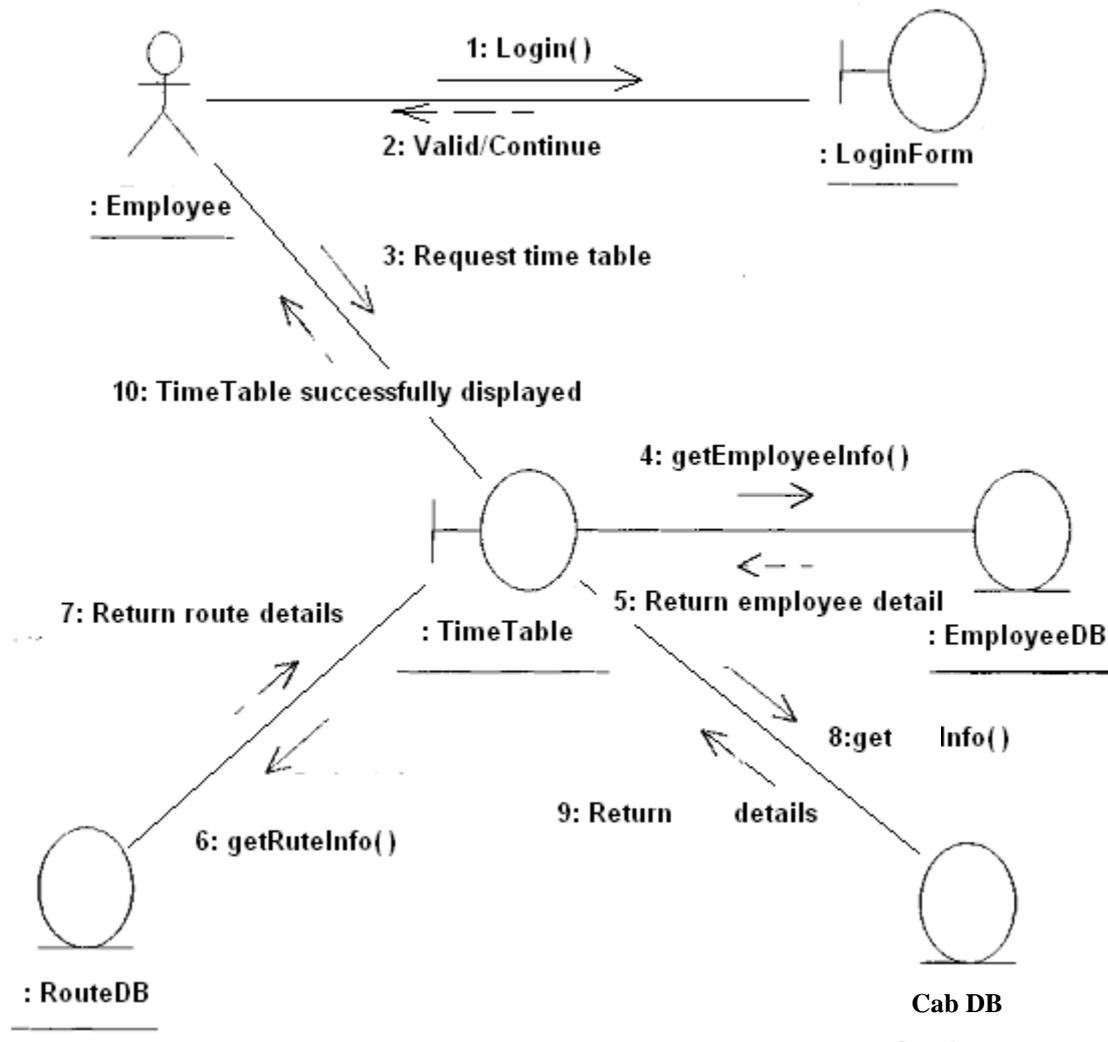
Ticket Status Report:



Cancellation of Ticket:



Time Sheet for Employees:



Data Dictionary

Data dictionary is maintained for holding required information regarding specific object, topic or any kind of real world entity. A Data Dictionary a repository of the elements in a system. As the name suggests, these elements center around data and the way they are structured to meet user requirements and organization needs. In a data dictionary we will find a list of all elements composing the data flowing through a system.

5.4 Data Modeling

The general theme behind a database is to handle information as an integrated whole. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and effectively. After designing input and output, the analyst must concentrate on database design or how data should be organized around user requirements. The general objective is to make information access, easy quick, inexpensive and flexible for other users.

During database design the following objectives are concerned:-

- ✓ Controlled Redundancy
- ✓ Data independence
- ✓ Accurate and integrating
- ✓ More information at low cost
- ✓ Recovery from failure
- ✓ Privacy and security

Table Name: Employees

Column Name	Description	Data Type(Size)	Constraint
Name	Employee name	Varchar2(15)	Not null
Address	Employee address	Varchar2(15)	Not null
Contact	Employee contact no	Number	Not null
Design	Designation of employee	Varchar2(15)	Not null
Emp_ID	Employee ID number	Varchar2(15)	Primary key
Join_D	Employee work join date	Varchar2(15)	Not null
Salary	Employee salary	Number	Not null

Table Name: Route

Column Name	Description	Data Type(Size)	Constraint
Route_ID	Route ID number	Varchar2(15)	Primary key
Cab_Code	Cab Code for route	Varchar2(15)	Not null
T_time	Total time taken by this route	Varchar2(15)	Not null
Start_T	Route start time	Varchar2(15)	Not null
Reach_T	Reach time for route	Varchar2(15)	Not null
T_cost	Total cost for this route	Number	Not null

Table Name: Cab

Column Name	Description	Data Type(Size)	Constraint
Cab_ID	Cab Id number	Varchar2(15)	Primary key
Cab_no	Cab number	Varchar2(15)	Not null
A_seat	Available seats	Number	Not null
R_seat	Reserve seats	Number	Not null
T_seat	Total no of seats	Number	Not null
Colour	Cab color	Varchar2(15)	Not null

Table Name: Ticket

Column Name	Description	Data Type(Size)	Constraint
Name	Passenger name	Varchar2(15)	Not null
No_seat	Book Seat number	Varchar2(15)	Not null
T_ID	Ticket ID no	Number	Primary key
Route_ID	Route Id no	Varchar2(15)	Reference(route)
T_cost	Ticket total cost	Number	Not null
Date	Ticket for date	Varchar2(15)	Not null

Table Name: Passenger

Column Name	Description	Data Type(Size)	Constraint
Name	Passenger name	Varchar2(15)	Not null
Address	Passenger address	Varchar2(15)	Not null
Contact	Passenger contact no	Number	Not null
T_ID	Reserved ticket ID	Varchar2(15)	Not null
Sex	Passenger Sex	Varchar2(15)	Not null

Table Name: Work Status

Column Name	Description	Data Type(Size)	Constraint
Emp_ID	Employee ID no	Varchar2(15)	Reference(Employee)
S_date	Start date from	Varchar2(15)	Not null
E_date	Schedule end date	Varchar2(15)	Not null
Cab_ID	Cab ID no	Varchar2(15)	Reference(Cab)
Start_T	Route start time	Varchar2(15)	Not null
Reach_T	Route reach time	Varchar2(15)	Not null
Route_ID	Route id no	Varchar2(15)	Reference(Route)

Above table shows how data base maintained in our Online Cab Scheduling system. All above tables are interrelated with each other so that we can retrieve associative data from tables. These tables are maintained in back end of system so that only authorized persons work on this data base.

5.5 Modules

“A modularization consists of well-defined manageable units with well defined interfaces among the units”

Desirable property of modular system include

- i. Each module is a well defined sub-system.
- ii. Single, well – defined purpose of each module.
- iii. Modules can be separately compiled and stored in a library.
- iv. Modules can use other module.
- v. Modules should be easier to use than to build.
- vi. Modules should be simpler from outside then from inside.

Design constraints to be observed during modularization are:-

Coupling: “Coupling is a measure of the degree of interdependence between software modules”.

Two modules with high coupling are strongly inter connected and thus dependent on each other.

Two modules with low coupling are not dependent on one another.

Cohesion: “Cohesion is a measure of the degree to which the elements of a module are functionally related”.

A strongly cohesive module implements functions, which is related to one feature of the solution and requires little interaction with other module.

No. Of Modules of Project:

A modularization consists of well-defined manageable units with well defined interfaces among the units.

Module description

There are Four Modules in this project, these MODULES are:-

1. LOGIN MASTER
2. TICKET RESERVATION MODULE

3. TICKET CANCELLATION
4. PASSENGERS REQUEST FOR RESERVATION.
5. PASSENGERS REQUEST FOR ROUTE TIME TABLE.
6. TICKET ID NUMBER FOR KNOWING TICKET STATUS REPORT.
7. EMPLOYEES ID FOR ACCESSING EMPLOYEES STATUS REPORT.
8. REGISTRATION MASTER
9. ROUTES AVAILABLE

1. LOGIN MASTER

Important sub module

- a. User-id and password update master
- b. User-id checker
- c. Login status manager

2. TICKET RESERVATION MODULE

Important sub module

- a. View Ticket Booking Details
- b. Timing
- c. Ticket cancellation facility

3. REGISTRATION MASTER

Important sub module

- a. User first register himself
- b. Then he can see user menu form
- c. User registration form is available

4. ROUTES MASTER

Important sub module

- a. View Route details
- b. Add Route details
- c. Cabs available at a particular route.

5. EMPLOYEE DETAILS

Important sub module

- a. View Drivers and Conductor's Timings details
- b. Add Employee details
- c. Drivers available at a particular route for a particular No..

5.8 Process Logic of each module

Project Plan, Design & Approach

Phase 1: Requirement Gathering and Analysis

Gather system requirements and prepare a System Requirement Specification document.

After collect the information it is analyzed that the available resources can fulfill all the requirements. And it also be examined that what resource will be used.

Phase 2: System Design

Make a detailed analysis of the system and prepare a System Design Document on the basis of SRS.

Phase 3: Prepare UTC & STC for testing the software

Prepare Unit Test Cases document. This document will be used to verify whether the functional requirements of the system have been met.

Phase 4: Develop the software

Develop the planned system.

Phase 5: Test the software using the prepared UTC/STC and Rework if needed

Run your software programs using the respective UTC to verify & test the software.

Phase 6: Demonstrate the S/W to users & Implement it

TOOLS / PLATFORM, HARDWARE AND SOFTWARE
REQUIREMENT SPECIFICATION

Software Requirement

Platform	:	Windows
The Operating System	:	Windows XP Professional sp2
Server	:	
Front-End Tool	:	Java Server Pages, JSF
Editing tool	:	Eclipse IDE 6.5.1
Browser	:	Internet Explorer
Database Server	:	MySQL Community

Hardware Used

Processor (GHz)	:	Intel Pentium 4 (3.06 GHz)
Memory	:	1 GB RAM.
Network Adapter	:	Ethernet Adaptor
Modem	:	56kbps Voice Fax Data
Secondary Storage	:	Seagate Hard disk (80 GB)

5.6 Scheduling



Projecthelp

6. CODING

Index.jsp

[Home](#) | [Registration](#) | [Sign In](#) | [Contact Us](#) | [Admin](#) | [Feedback](#)

ONLINE CAB SCHEDULING SYSTEM



Cab Traveling Management System is software which is helpful for Cab travelers, who wants to travel by Cab all over different routes. This system provide different routes for passengers, passengers can also have a look over how many trips are their available for specific route and what is the time table for different traveling. Passengers can book his ticket for his travel by Cab with a specific route and for a specific day. This system is useful for passenger who wants to know about Cab routes for traveling between two or many places, how much it will cost and how much time it would take for traveling. This system is also useful for employees like Cab drivers and conductors who are working for this system they can know about his work status information and his working time table with their route information he has to follow over his work.

Username

Password

[Create New Account](#)





start | Macromedia Dreamw... | BUS TRAVELING MGT... | cab traveling system... | 4:44 PM

```
<%@ page language="java" import="java.sql.*" %>
<%@ page session="true" %>
<html>
<head>
<title>ONLINE CAB SCHEDULING SYSTEM</title>
<style type="text/css">
<!--
.style2 {
    font-size: 16px;
    font-weight: bold;
    color: #990000;
}
.style3 {color: #0000FF}
.style4 {color: #990000}
.style5 {
    font-weight: bold;
    color: #990000;
}
-->
</style>
</head>
<body>
<form name="form1" method="post" action="log.jsp">

<table width="879" border="0" align="center" cellspacing="0">
<tr>
    <td colspan="2" height="154"><%@ include file="header1.jsp" %></td>
```

```

</tr>

<tr>
<td width="2" align="left" valign="top">&nbsp;</td>
<td width="903" align="left" valign="top"><table width="519" border="0" align="left"
cellpadding="0" cellspacing="0">
<tr>
<td width="34" height="230" align="left" valign="top"><p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p></td>
<td colspan="2" align="left" valign="top"><p align="justify" class="style5"> Cab
TRAVELLINGManagement System is software which is helpful for Cab travelers, who wants to
travel by Cab all over different routes. This system provide different routes for passengers,
passengers can also have a look over how many trips are their available for specific route and what is
the time table for different traveling. Passengers can book his ticket for his travel by Cab with a
specific route and for a specific day.

This system is useful for passenger who wants to know about Cab routes for TRAVELLINGbetween
two or many places, how much it will cost and how much time it would take for traveling. This
system is also useful for employees like Cab drivers and conductors who are working for this system
they can know about his work status information and his working time table with their route
information he has to follow over his work.</strong></p>
<p align="justify" class="style4">&nbsp;</p> </td>
</tr>

<tr>
<td align="left" valign="top">&nbsp;</td>
<td colspan="2" align="left" valign="top"> <% Class.forName("com.mysql.jdbc.Driver");

Connection connection = DriverManager.getConnection("jdbc:mysql://localhost/Cab","root","");
Statement statement = connection.createStatement();

String query="select * from CAB";
ResultSet rs = statement.executeQuery(query);
%>
</span>
<table width="613" border="1" height="68" bgcolor="#FFFFFF">
<tr>
<% int i=0;
while(rs.next())
{
if(i%3==0)
{
%>
</tr><tr>
<%}%>

<td width="522">

<p><img src='<%=rs.getString(7)%>' height="100" width="100" /> <br />
<span class="style3"><u><%=rs.getString(5)%></u></span><br><span
class="style3"><span class="style4"><%=rs.getString(6)%></span><br />
</span> <br />
</p> </td>

<br>
<%

i=i+1;} %>
</table></td>
</tr>

```

```

<tr>
  <td align="left" valign="top">&nbsp;</td>
  <td width="368" align="left" valign="top">&nbsp;</td>
  <td width="215" align="center" valign="middle">&nbsp;</td>
</tr>
</table>
<table width="200" height="191" border="0" align="center" cellpadding="0" cellspacing="0"
bordercolor="#0000FF">
  <tr>
    <td width="79">Username</td>
    <td width="105"><input name="user" type="text" id="user"></td>
  </tr>
  <tr>
    <td height="85" valign="top"><p>Password</p></td>
    <td valign="top"><input name="pass" type="password" id="pass">
      <br>
      <input type="submit" name="Submit" value="Submit"></td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td><a href="aboutus.jsp" class="style2">Create New Account </a></td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td><a href="forget password.jsp" class="style2"></a></td>
  </tr>
</table>
<p align="center"></p></td>
</tr>
</table>

</form>

</body>
</html>

```

Log.jsp

```

<%@ page language="java" import="java.sql.*"%>
<%@ page session="true" %>
<html>
<body>

<form name="form1" method="post" action="newmember.jsp" onSubmit="return
validate(this)">
<table width="731" border="0" align="center" cellspacing="0">
<tr>
  <td colspan="2" height="120"><%@ include file="header1.jsp" %></td>
</tr>

<tr>
<td width="150" align="left" valign="top">&nbsp;</td>
<td width="750" align="left" valign="top"><%

String str=request.getParameter("user");
String str1=request.getParameter("pass");

```

```
        Class.forName("com.mysql.jdbc.Driver");
        Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost/Cab","root","");
        Statement statement = connection.createStatement();
        String query="select * from login where usern='"+str+"' and passw='"+str1+"'";
        ResultSet x = statement.executeQuery(query);

if(x.next()==true)
{
session.setAttribute("user",str);
response.sendRedirect("user menu.jsp");
}
else
{
out.print("you are invalid user");
}

%></td>
</tr>
</table>
</form>
</body>
</html>
```

Projecthelpline.in

ONLINE CAB SCHEDULING SYSTEM



Username	<input type="text"/>
Password	<input type="text"/>
Email Id	<input type="text"/>
Security Question	<input type="text"/>
Answer	<input type="text"/>
Phone	<input type="text"/>
<input type="button" value="Submit"/>	

By clicking Register , you are indicating that you have read and agree to the Users Agreement and Privacy Policy.



```
<%@ page language="java" %>
<%@ page session="true" %>
<%@ page import="java.sql.*" %>
<html>
<head>
<title>Cab TRAVELLINGManagement System</title>
<style type="text/css">
<!--
.style1 {font-family: Verdana, Arial, Helvetica, sans-serif}
.style2 {
font-size: 12px;
color: #0000FF;
}
.style5 {color: #0000FF}
.style6 {font-style: italic; font-family: Verdana, Arial, Helvetica, sans-serif;}
-->
</style>
</head>
<body>

<script language="javascript">

function validate(form)
{
if (form.user.value=="")
{
alert("Username should not be blank. Please enter it.");
form.name.focus(); return false;
}
}
```

```

if (form.pass.value=="")
    {
        alert("Password should not be blank. Please enter it.");
        form.email.focus(); return false;
    }
if (form.email.value=="")
    {
        alert("Email should not be blank. Please enter it.");
        form.email.focus(); return false;
    }

if (form.address.value=="")
    {
        alert("Question should not be blank. Please enter it.");
        form.address.focus(); return false;
    }
if (form.country.value=="")
    {
        alert("Answer should not be blank. Please enter it.");
        form.country.focus(); return false;
    }

if (form.phone.value=="")
    {
        alert("Phone No. should not be blank. Please enter it.");
        form.phone.focus(); return false;
    }

if(isNaN(form.phone.value))
    {
        alert("Phone No. Should be a number ");
        form.phone.focus();
        return false;
    }

}

</script>

<form name="form1" method="post" action="newmember.jsp" onSubmit="return
validate(this)">
<table width="731" border="0" align="center" cellspacing="0">
<tr>
    <td colspan="2" height="120"><%@ include file="header1.jsp" %></td>
</tr>

<tr>
<td width="150" align="left" valign="top">&nbsp;</td>
<td width="750" align="left" valign="top"><table width="409" border="0"
align="left" cellpadding="5" cellspacing="0">
<tr>
    <td align="left" valign="top">&nbsp;</td>
    <td align="left" valign="top">&nbsp;</td>
    <td align="left" valign="top">&nbsp;</td>
</tr>

```

```

<tr>
  <td width="59" align="left" valign="top">&nbsp;</td>
  <td width="124" align="left" valign="top">Username</td>
  <td width="196" align="left" valign="top">
    <label>
      <div align="center">
        <input name="user" type="text" id="user">
      </div>
    </label> </td>
</tr>
<tr>
  <td align="left" valign="top">&nbsp;</td>
  <td align="left" valign="top">Password</td>
  <td align="center" valign="middle"><label>
    <input name="pass" type="password" id="pass">
  </label></td>
</tr>
<tr>
  <td align="left" valign="top">&nbsp;</td>
  <td align="left" valign="top">Email Id </td>
  <td align="center" valign="middle"><input name="email" type="text"
id="email"></td>
</tr>
<tr>
  <td align="left" valign="top">&nbsp;</td>
  <td align="left" valign="top">Security Question </td>
  <td align="center" valign="middle"><input name="address" type="text"
id="address"></td>
</tr>
<tr>
  <td align="left" valign="top">&nbsp;</td>
  <td align="left" valign="top">Answer</td>
  <td align="center" valign="middle"><input name="country" type="text"
id="country"></td>
</tr>
<tr>
  <td align="left" valign="top">&nbsp;</td>
  <td align="left" valign="top">Phone</td>
  <td align="center" valign="middle"><input name="phone" type="text"
id="phone"></td>
</tr>
<tr>
  <td align="left" valign="top">&nbsp;</td>
  <td align="left" valign="top">&nbsp;</td>
  <td align="center" valign="middle"><input type="submit" name="Submit"
value="Submit"></td>
</tr>
<tr>
  <td align="left" valign="top">&nbsp;</td>
  <td colspan="2" align="left" valign="top">
    <span class="style1">
      <label>

```

```

</label>
</span> <span class="style5">
<label></label>
</span>
<label>
<div align="justify" class="style2"><span class="style6">By clicking Register ,
you are indicating that you have read and agree to the Users Agreement and Privacy
Policy. </span></div>
</label> </td>
</tr>
</table></td>
</tr>
</table>
</form>

</body>
</html>

```

Newmember.jsp

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java"
import="java.sql.*" errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
</head>

<body>

<%
    String name=request.getParameter("user");
    String pass=request.getParameter("pass");
    String email=request.getParameter("email");
    String address=request.getParameter("address");
    String country=request.getParameter("country");
    String phone=request.getParameter("phone");

    Class.forName("com.mysql.jdbc.Driver");
    Connection connection =
    DriverManager.getConnection("jdbc:mysql://localhost/Cab","root","1234");
    Statement statement = connection.createStatement();
    String query1="select * from login";
    ResultSet rs = statement.executeQuery(query1);
    int i=1;
    while(rs.next()==true)
    {
        i=i+1;
    }
    String id="M00"+i;

```

```
String query="insert into login
values(""+id+"",""+name+"",""+pass+"",""+email+"",""+address+"",""+country+"",""+ph
one+"")";
//String query="delete from login where usern=""+name+""";
//String query="update login set
passw=""+pass+"",email=""+email+"",address=""+address+"",country=""+country+"",
phone=""+phone+" where usern=""+name+""";

statement.executeUpdate(query);

session.setAttribute("mid",id);
response.sendRedirect("login.jsp");
%>

</body>
</html>
```

Login.jsp

[Home](#)|[Registration](#) | [Sign In](#) | [Contact Us](#) | [Admin](#) | [Feedback](#)

ONLINE CAB SCHEDULING SYSTEM



Existing Users Login Here

Don't Have an Account Yet

Username

Password

Sign In

Get connected to incredible excess inventory deals from the world most trusted brands.

Register



```
<%@ page language="java" %>
<%@ page session="true" %>
<html>
<head>
<title>Online Cab Scheduling System</title>
<style type="text/css">
<!--
.style3 {font-family: "Courier New", Courier, monospace; font-size: 14px; }
-->
</style>
</head>
<body>
```

```

<script language="javascript">

function validate(form)
{
if (form.user.value=="")
    {    alert("Username should not be blank. Please enter it.");
        form.user.focus(); return false;
    }

if (form.pass.value=="")
    {    alert("Password should not be blank. Please enter it.");
        form.pass.focus(); return false;
    }

}

}

</script>

<form action="log.jsp" method="post" onsubmit="return validate(this)">
<table width="731" border="0" align="center" cellspacing="0">
<tr>
    <td colspan="2" height="120"><%@ include file="header1.jsp" %></td>
</tr>

<tr>
<td width="150" align="left" valign="top">&nbsp;</td>
<td width="750" align="left" valign="top"><table width="653" border="0"
align="left" cellpadding="5" cellspacing="0">
<tr>
    <td align="left" valign="top" bgcolor="#FFFFFF"><p>&nbsp;</p>
    <p>&nbsp;</p>
    <p>&nbsp;</p></td>
    <td colspan="2" align="left" valign="top" bordercolor="#CC9999"
bgcolor="#FFFFFF">&nbsp;</td>
    <td align="left" valign="top" bgcolor="#FFFFFF">&nbsp;</td>
    <td colspan="2" align="left" valign="top" bgcolor="#FFFFFF">&nbsp;</td>
</tr>
<tr>
    <td align="left" valign="top">&nbsp;</td>
    <td colspan="2" align="left" valign="top" bordercolor="#CC9999"
bgcolor="#CC6699"><span class="style3">Existing Users Login Here
</span></td>
    <td align="left" valign="top">&nbsp;</td>
    <td colspan="2" align="left" valign="top" bgcolor="#CCCCCC"><span
class="style3">Don't Have an Account Yet </span></td>
</tr>
<tr>
    <td width="95" height="84" align="left" valign="top">&nbsp;</td>
    <td width="109" align="left" valign="top"><p>&nbsp;</p>
    <p>Username</p>

```

```

        <p>Password</p></td>
<td width="160" align="left" valign="top"><label>
    <br>
    <br>
    <input name="user" type="text" id="user">
    <br>
    <br>
    <input name="pass" type="password" id="pass">
</label> </td>
<td width="2" align="left" valign="top">&nbsp;</td>
<td colspan="2" align="left" valign="top"><p>&nbsp;</p>
    <p>Get connected to incredible excess inventory deals from the world most
trusted brands. </p></td>
</tr>
<tr>
<td align="left" valign="top">&nbsp;</td>
<td align="left" valign="top">&nbsp;</td>
<td align="left" valign="middle"><label><a href="log.jsp">
    <input type="submit" name="Submit" value="Sign In">
<%
try
{
String mid=(String)session.getAttribute("mid");
if(mid.equals(""))
{
out.print("");
}
else
{
out.println("Your Id " + mid);
}
}
catch (Exception e){
}
%></a></label></td>
<td align="center" valign="middle">&nbsp;</td>
<td width="110" align="center" valign="middle">&nbsp;</td>
<td width="118" align="center" valign="middle"><a href="aboutus.jsp"></a></td>
</tr>

<tr>
<td align="left" valign="top">&nbsp;</td>
<td align="left" valign="top">&nbsp;</td>
<td align="center" valign="middle"><label></label></td>
<td align="center" valign="middle">&nbsp;</td>
<td align="center" valign="middle">&nbsp;</td>
<td align="center" valign="middle">&nbsp;</td>
</tr>
</table></td>
</tr>
</table>

```

```
<form>

</body>
</html>
```

Login.jsp

```
<%@ page language="java" import="java.sql.*"%>
<%@ page session="true" %>
<html>
<body>
```

```
<form name="form1" method="post" action="newmember.jsp" onSubmit="return
validate(this)">
<table width="731" border="0" align="center" cellspacing="0">
<tr>
    <td colspan="2" height="120"><%@ include file="header1.jsp" %></td>
</tr>
```

```
<tr>
<td width="150" align="left" valign="top">&nbsp;  </td>
<td width="750" align="left" valign="top"><%
```

```
String str=request.getParameter("user");
String str1=request.getParameter("pass");
```

```
        Class.forName("com.mysql.jdbc.Driver");
        Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost/Cab","root","");
        Statement statement = connection.createStatement();
        String query="select * from login where usern='"+str+"' and passw='"+str1+"'";
        ResultSet x = statement.executeQuery(query);
```

```
if(x.next()==true)
{
session.setAttribute("user",str);
response.sendRedirect("user menu.jsp");
}
else
{
out.print("you are invalid user");
}
```

```
%></td>
</tr>
</table>
</form>
</body>
```

</html>

Contactus.jsp

[Home](#)|[Registration](#) | [Sign In](#) | [Contact Us](#) | [Admin](#)| [Feedback](#)

ONLINE CAB SCHEDULING SYSTEM



[Contact Us](#)

For your Suggetions contact us at
Email: contactus@cabtraveling.com
Phone : 9856434580,9854335464



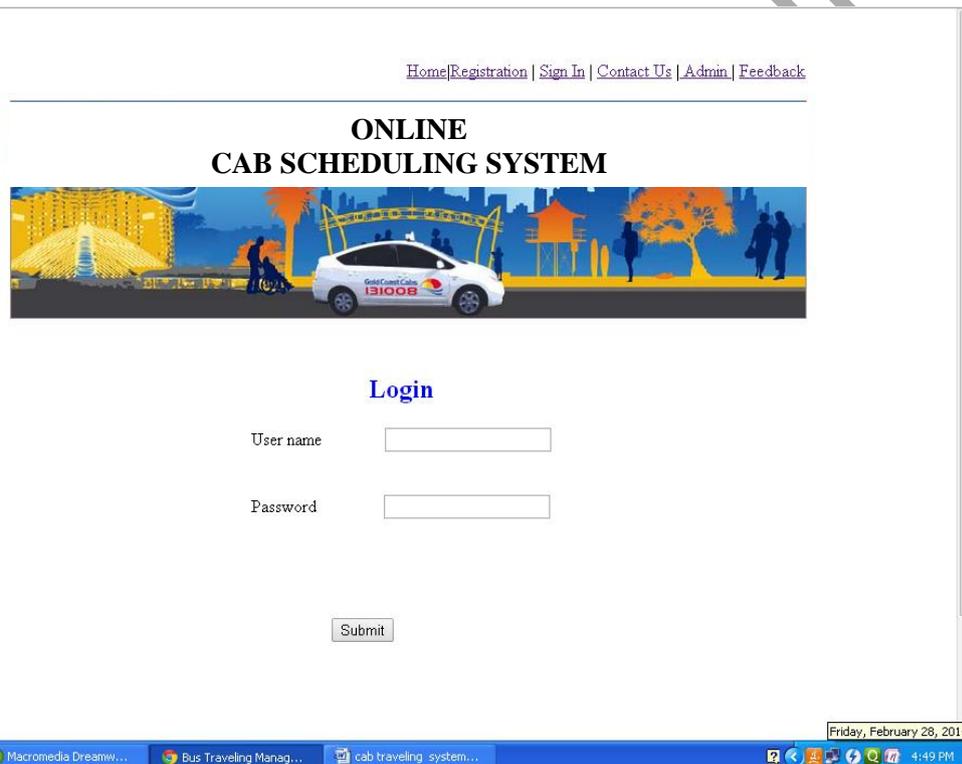
```
<%@ page language="java" %>
<%@ page session="true" %>
<html>
<head>
<title> Online Cab Scheduling System </title>
<style type="text/css">
<!--
.style2 {font-family: Arial, Helvetica, sans-serif}
.style4 {
font-size: 16px;
font-family: Arial, Helvetica, sans-serif;
font-weight: bold;
color: #0000FF;
}
.style6 {font-family: Arial, Helvetica, sans-serif; font-weight: bold; }
-->
</style>
</head>
<body>
<table width="731" border="0" align="center" cellspacing="0">
<tr>
  | |
```

```

<BR __designer:dtid="2251804108652553" />
<span class="style6">For your Suggetions contact us at<BR
__designer:dtid="2251804108652554" />
Email:contactus@cabtraveling.com<BR __designer:dtid="2251804108652555"
/>
Phone : 9856434580,9854335464</span></p>
<p>&nbsp;</p>
<p><span class="style2"><BR __designer:dtid="2251804108652556" />
</span></p></td>
</tr>
</table>
</body>
</html>

```

Admin.jsp



```

<%@ page language="java" %>
<%@ page session="true" %>
<html>
<head>
<title> Online Cab Scheduling System </title>
<style type="text/css">
<!--
.style7 {
font-family: Arial, Helvetica, sans-serif;
font-size: 16px;
font-weight: bold;
color: #0000FF;
}
.style8 {
font-size: 18px;

```



```

out.print("you are invalid user");
}
%>
</body>
</html>

```

Adminmenu.jsp

```

<%@ page language="java" import="java.sql.*" %>
<%@ page session="true" %>
<html>
<head>
<title> Online Cab Scheduling System </title>
<style type="text/css">
<!--
.style7 {
font-family: Arial, Helvetica, sans-serif;
font-size: 16px;
font-weight: bold;
color: #0000FF;
}
.style8 {
font-size: 18px;
color: #CC3300;
}
.style10 {font-size: 24px; color: #0000FF;}
.style12 {
font-size: 16px;
font-weight: bold;
}
.style13 {
color: #0000FF;
font-weight: bold;

```

```

}
.style14 {color: #0000FF}
.style15 {font-size: 16px; font-weight: bold; color: #FF0000; }
-->
</style>
</head>
<body>
<table width="731" border="0" align="center" cellspacing="0">
<tr>
<td colspan="2" height="120"><%@ include file="header1.jsp" %></td>
</tr>

<tr>
<td width="150" align="left" valign="top">&nbsp;</td>
<td width="750" align="left" valign="top" bgcolor="#FFFFFF"><form
name="form1" method="get" action="">
<table width="671" height="562" border="0">
<tr>
<td colspan="2"><div align="center" class="style10">
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>Administrator Menu </p>
<p>&nbsp;</p>
</div></td>
</tr>
<tr>
<td width="285"><p align="center" class="style15">&nbsp;</p>
<p align="center" class="style12">&nbsp;</p></td>
<td width="376"><span class="style15">Reports Available </span></td>
</tr>
<tr>
<td><div align="center" class="style13"><a href="ticket.jsp">Ticket
Reservation</a> </div></td>
<td><span class="style13"><a href="repo_users.jsp">Users
Details</a></span></td>
</tr>
<tr>
<td><div align="center" class="style13"></div></td>
<td>&nbsp;</td>
</tr>
<tr>
<td><div align="center" class="style13">
<p><a href="ticketcancel.jsp">Ticket Cancelation </a></p>
<p><a href="employee.jsp">Add Employee Details </a> </p>
</p>
</div></td>
<td align="left" valign="top" background="repo_items.jsp"><p><a
href="repo_Cab.jsp"><strong>Cabes Details and Timesheet </strong></a> </p>
<p><a href="user_work.jsp"><strong>Drivers and Conductors
Timings</strong></a><strong> </strong></p></td>
</tr>
<tr>

```

```

        <td><div align="center" class="style13"><a href="route.jsp">Add Route
Details </a> </div></td>
        <td><a href="repo_route.jsp"><strong>Route Details</strong></a> </td>
    </tr>
    <tr>
        <td align="center" valign="top"><div align="center" class="style13">
            <p>&nbsp;</p>
            <p><a href="Cab.jsp">Add Cab Details</a> </p>
        </div>
        <a href="workstatus.jsp"><span class="style13">Add Employee
WorkstatusDetails </span></a></td>
        <td><p class="style13">&nbsp;</p>
            <p class="style13"><a href="feedbackdet.jsp">Feedback Details</a></p>
            <p class="style13"><a href="repo_emp.jsp">Employee Details</a>
</p></td>
    </tr>
    <tr>
        <td><p align="center" class="style13">&nbsp;</p>
            <p align="center" class="style13">&nbsp;</p>
            <p align="center">&nbsp;</p>
            <p align="center" class="style14">&nbsp;</p>
            <p align="center" class="style14">&nbsp;</p></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    </table>
</form>
<h1 class="style8">&nbsp;</h1>
<p>&nbsp;</p>
<h1>&nbsp;</h1> <p class="style7"
__designer:dtid="281479271677962">&nbsp;</p> </td>
</tr>
</table>
</body>
</html>

```

Ticket.jsp

[Home](#) | [Registration](#) | [Sign In](#) | [Contact Us](#) | [Admin](#) | [Feedback](#)

**ONLINE
CAB SCHEDULING SYSTEM**



Add Ticket Details

Passanger Name	<input type="text"/>
Address	<input type="text"/>
Contact No.	<input type="text"/>
Sex	<input type="text"/>
Seat No.	<input type="text"/>
Ticket ID	<input type="text" value="T03"/>
Route ID	<input type="text" value="R01"/>
Ticket Cost	<input type="text"/>
Date	<input type="text"/>
Bus ID	<input type="text" value="B01"/>

Windows taskbar: start, Macromedia Dreamw..., Bus Traveling Mgt. Sy..., cab traveling system..., 4:51 PM

```
<%@ page language="java" import="java.sql.*" %>
<%@ page session="true" %>
<html>
<head>
<title>Online Cab Scheduling System</title>
<style type="text/css">
<!--
.style2 {font-family: Arial, Helvetica, sans-serif}
-->
</style>
</head>
<body>
<script language="javascript">

function validate(form)
{
if (form.textfield.value=="")
{
    alert("ID should not be blank. Please enter it.");
    form.textfield.focus(); return false;
}

if (form.textfield2.value=="")
{
    alert("Field should not be blank. Please enter it.");
    form.textfield2.focus(); return false;
}
if (form.textfield3.value=="")
{
    alert("Field should not be blank. Please enter it.");
    form.textfield3.focus(); return false;
}
}
```

```

        if (form.textfield4.value=="")
        {
            alert("Field should not be blank. Please enter it.");
            form.textfield4.focus(); return false;
        }
    }
    </script>

<form action="ticketins.jsp" method="post" onS ubmit="return validate(this)">

<table width="731" border="0" align="center" cellspacing="0">
<tr>
    <td colspan="2" height="120"><%@ include file="header1.jsp" %></td>
</tr>

<tr>
<td width="150" align="left" valign="top">&nbsp;</td>
<td width="750" align="center" valign="top"><p>&nbsp;</p>
<table width="380" border="0">
<tr>
    <td height="32" colspan="2"><strong>Add Ticket Details </strong></td>
    <td width="85">&nbsp;</td>
</tr>

<tr>
<td width="137">&nbsp;</td>
<td width="144">&nbsp;</td>
<td>&nbsp;</td>
</tr>
<tr>
<td><strong>Passanger Name

</strong></td>
<td><input name="textfield" type="text" id="textfield"></td>
<td>&nbsp;</td>
</tr>
<tr>
<td><strong>Address</strong></td>
<td><input name="textfield2" type="text" id="textfield2"></td>
<td>&nbsp;</td>
</tr>
<tr>
<td><strong>Contact No.</strong></td>
<td><input type="text" name="textfield3"></td>
<td>&nbsp;</td>
</tr>
<tr>
<td><strong>Sex</strong></td>
<td><input type="text" name="textfield4"></td>
<td>&nbsp;</td>
</tr>

```

```

<tr>
  <td><strong>Seat No.</strong> </td>
  <td><input type="text" name="textfield6"></td>
  <td>&nbsp;</td>
</tr>
<tr>
  <td><strong>Ticket ID
  <%

      Class.forName("com.mysql.jdbc.Driver");
      Connection connection =
      DriverManager.getConnection("jdbc:mysql://localhost/Cab","root","");
      Statement statement = connection.createStatement();

      String query="select * from ticket";
      ResultSet rs = statement.executeQuery(query);
      int i=0;
      while(rs.next())
      {
        i=i+1;
      }
      String id="T0"+(i+1);
%>
  </strong></td>
  <td><input name="textfield5" type="text" value="<%=id%>"></td>
  <td>&nbsp;</td>
</tr>
<tr>
  <td><strong>Route ID </strong></td>
  <td><select name="select" id="select">
    <%String query2="select * from route";
    ResultSet rs2 = statement.executeQuery(query2);
    while(rs2.next())
    {
%>
      <option><%=rs2.getString(1)%></option>
      <%
    }
%>
  </select></td>
  <td>&nbsp;</td>
</tr>
<tr>
  <td><strong>Ticket Cost

</strong></td>
  <td><input name="textfield7" type="text" id="textfield7"></td>
  <td>&nbsp;</td>
</tr>
<tr>

```

```

        <td><strong>Date</strong></td>
        <td><input type="text" name="textfield8"></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Cab ID </strong></td>
        <td><select name="select2" id="select2">
            <%String query3="select * from Cab";
            ResultSet rs3 = statement.executeQuery(query3);
            while(rs3.next())
            {
            %>
                <option><%=rs3.getString(1)%></option>
            <%
            }
            %>
        </select></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td><input type="submit" name="Submit" value="Submit"></td>
        <td>&nbsp;</td>
    </tr>
</table>
<p><BR __designer:dtid="2251804108652552" />
<span class="style2"><BR __designer:dtid="2251804108652556" />
</span></p>
</td>
</tr>
</table>
</form></body>
</html>

```

Ticketins.jsp

```
<%@ page contentType="text/html; charset=iso-8859-1" language="java"
import="java.sql.*" errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Online Cab Scheduling System</title>
</head>

<body>

<%
    String p1=request.getParameter("textfield");
String p2=request.getParameter("textfield2");
String p3=request.getParameter("textfield3");
String p4=request.getParameter("textfield4");
String p5=request.getParameter("textfield6");
String p6=request.getParameter("textfield5");
String p7=request.getParameter("select");
String p8=request.getParameter("textfield7");
String p9=request.getParameter("textfield8");
String p10=request.getParameter("select2");

Class.forName("com.mysql.jdbc.Driver");
Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost/Cab","root","1234");
Statement statement = connection.createStatement();

String query="insert into ticket
values('"+p1+"','"+p2+"','"+p3+"','"+p4+"','"+p5+"','"+p6+"','"+p7+"','"+p8+"','"+p
9+"','"+p10+"')";

statement.executeUpdate(query);

response.sendRedirect("ticket.jsp");
%>

</body>
</html>
```

Ticketcancel.jsp

[Home](#) | [Registration](#) | [Sign In](#) | [Contact Us](#) | [Admin](#) | [Feedback](#)



Ticket Cancellation

Ticket ID



```
a<%@ page language="java" import="java.sql.*" %>
<%@ page session="true" %>
<html>
<head>
<title>Online Cab Scheduling System</title>
<style type="text/css">
<!--
.style2 {font-family: Arial, Helvetica, sans-serif}
-->
</style>
</head>
<body>
<script language="javascript">

function validate(form)
{
if (form.textfield.value=="")
    {
        alert("ID should not be blank. Please enter it.");
        form.textfield.focus(); return false;
    }

if (form.textfield2.value=="")
    {
        alert("Field should not be blank. Please enter it.");
        form.textfield2.focus(); return false;
    }
if (form.textfield3.value=="")
    {
        alert("Field should not be blank. Please enter it.");
        form.textfield3.focus(); return false;
    }
}
```

```

        if (form.textfield4.value=="")
        {
            alert("Field should not be blank. Please enter it.");
            form.textfield4.focus(); return false;
        }
    }

    </script>

    <form action="ticketcdel.jsp" method="post" onSubmit="return validate(this)">

    <table width="731" border="0" align="center" cellspacing="0">
    <tr>
        <td colspan="2" height="120"><%@ include file="header1.jsp" %></td>
    </tr>

    <tr>
    <td width="150" align="left" valign="top">&nbsp;</td>
    <td width="750" align="center" valign="top"><p>&nbsp;</p>
    <table width="380" border="0">
        <tr>
            <td height="32" colspan="2"><strong>Ticket Cancelation </strong></td>
            <td width="85">&nbsp;</td>
        </tr>

        <tr>
            <td colspan="2">&nbsp;</td>
            <td>&nbsp;</td>
        </tr>

        <tr>
            <td><strong>Ticket ID
            <%

                Class.forName("com.mysql.jdbc.Driver");
                Connection connection =
                DriverManager.getConnection("jdbc:mysql://localhost/Cab","root","");
                Statement statement = connection.createStatement();

                String query="select * from ticket";
                ResultSet rs = statement.executeQuery(query);
                int i=0;
                while(rs.next())
                {
                    i=i+1;
                }
                String id="T0"+(i+1);
            %>
            </strong></td>

```

```

        <td><select name="select" id="select">
            <%String query2="select * from route";
ResultSet rs2 = statement.executeQuery(query2);
while(rs2.next())
{
%>
        <option><%=rs2.getString(5)%></option>
        <%
    }
%>
    </select></td>
    <td>&nbsp;</td>
</tr>

<tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
</tr>
<tr>
    <td>&nbsp;</td>
    <td><input type="submit" name="Submit" value="Submit"></td>
    <td>&nbsp;</td>
</tr>
</table>
<p><BR __designer:dtid="2251804108652552" />
    <span class="style2"><BR __designer:dtid="2251804108652556" />
    </span></p>
</td>
</tr>
</table>
</form></body>
</html>

```

Ticketcdel.jsp

```

<%@ page language="java" import="java.sql.*" %>
<%@ page session="true" %>
<html>
<head>
<title>Online Cab Scheduling System</title>
<style type="text/css">
<!--
.style2 {font-family: Arial, Helvetica, sans-serif}
-->
</style>
</head>
<body>
<script language="javascript">

function validate(form)
{

```

```

if (form.textfield.value=="")
    {
        alert("ID should not be blank. Please enter it.");
        form.textfield.focus(); return false;
    }

if (form.textfield2.value=="")
    {
        alert("Field should not be blank. Please enter it.");
        form.textfield2.focus(); return false;
    }
if (form.textfield3.value=="")
    {
        alert("Field should not be blank. Please enter it.");
        form.textfield3.focus(); return false;
    }
if (form.textfield4.value=="")
    {
        alert("Field should not be blank. Please enter it.");
        form.textfield4.focus(); return false;
    }
}

</script>

<form action="ticketdel.jsp" method="post" onSubmit="return validate(this)">

<table width="731" border="0" align="center" cellspacing="0">
<tr>
    <td colspan="2" height="120"><%@ include file="header1.jsp" %></td>
</tr>

<tr>
<td width="150" align="left" valign="top">&nbsp;</td>
<td width="750" align="center" valign="top"><p>&nbsp;</p>
<table width="380" border="0">
<tr>
    <td height="32" colspan="2"><strong>Ticket Cancellation </strong></td>
    <td width="85">&nbsp;</td>
</tr>

<tr>
    <td colspan="2">&nbsp;</td>
    <td>&nbsp;</td>
</tr>
<tr>
    <td><strong>Ticket ID
    <%
        String tid=request.getParameter("tid");
        Class.forName("com.mysql.jdbc.Driver");
Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost/Cab","root","");
Statement statement = connection.createStatement();

```

```
String query="delete from ticket where tid='"+tid+"'";
statement.executeUpdate(query);
```

```
%>
```

```
<td>&nbsp;</td>
</tr>
```

```
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
```

```
<tr>
<td>&nbsp;</td>
<td><input type="submit" name="Submit" value="Submit"></td>
<td>&nbsp;</td>
</tr>
```

```
</table>
```

```
<p><BR __designer:dtid="2251804108652552" />
<span class="style2"><BR __designer:dtid="2251804108652556" />
</span></p>
```

```
</td>
```

```
</tr>
```

```
</table>
```

```
</form></body>
```

```
</html>
```

Employee.jsp

[Home](#) | [Registration](#) | [Sign In](#) | [Contact Us](#) | [Admin](#) | [Feedback](#)

ONLINE CAB SCHEDULING SYSTEM



Add Route Details

Employee ID	<input type="text" value="E02"/>
Name	<input type="text"/>
Address	<input type="text"/>
Contact No.	<input type="text"/>
Designation	<input type="text"/>
Joining Date	<input type="text"/>
Salary	<input type="text"/>
	<input type="button" value="Submit"/>



```
<%@ page language="java" import="java.sql.*" %>
<%@ page session="true" %>
<html>
<head>
<title>Online Cab Scheduling System</title>
<style type="text/css">
<!--
.style2 {font-family: Arial, Helvetica, sans-serif}
-->
</style>
</head>
<body>
<script language="javascript">
function validate(form)
{
if (form.textfield.value=="")
{
alert("ID should not be blank. Please enter it.");
form.textfield.focus(); return false;
}

if (form.textfield2.value=="")
{
alert("Field should not be blank. Please enter it.");
form.textfield2.focus(); return false;
}
if (form.textfield3.value=="")
{
alert("Field should not be blank. Please enter it.");
form.textfield3.focus(); return false;
}
}
```

```

        if (form.textfield4.value=="")
        {
            alert("Field should not be blank. Please enter it.");
            form.textfield4.focus(); return false;
        }
    }
</script>

<form action="empins.jsp" method="post" onSubmit="return validate(this)">

<table width="731" border="0" align="center" cellspacing="0">
<tr>
    <td colspan="2" height="120"><%@ include file="header1.jsp" %></td>
</tr>

<tr>
<td width="150" align="left" valign="top">&nbsp;</td>
<td width="750" align="center" valign="top"><p>&nbsp;</p>
<table width="380" border="0">
    <tr>
        <td height="32" colspan="2"><strong>Add Route Details </strong></td>
        <td width="85">&nbsp;</td>
    </tr>

    <tr>
        <td width="137">&nbsp;</td>
        <td width="144">&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Employee ID
            <%
                Class.forName("com.mysql.jdbc.Driver");
                Connection connection =
                DriverManager.getConnection("jdbc:mysql://localhost/Cab","root","");
                Statement statement = connection.createStatement();

                String query="select * from employee";
                ResultSet rs = statement.executeQuery(query);
                int i=0;
                while(rs.next())
                {
                    i=i+1;
                }
                String id="E0"+(i+1);
            %>
            </strong></td>

```

```

        <td><input name="textfield" type="text" id="textfield"
value='<%=id%>'></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Name</strong></td>
        <td><input name="textfield2" type="text" id="textfield2"></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Address</strong></td>
        <td><input type="text" name="textfield3"></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Contact No. </strong></td>
        <td><input type="text" name="textfield4"></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Designation</strong></td>
        <td><input type="text" name="textfield5"></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Joining Date </strong></td>
        <td><input type="text" name="textfield6"></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Salary</strong></td>
        <td><input name="textfield7" type="text" id="textfield7"></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td><input type="submit" name="Submit" value="Submit"></td>
        <td>&nbsp;</td>
    </tr>
</table>
<p><BR __designer:dtid="2251804108652552" />
<span class="style2"><BR __designer:dtid="2251804108652556" />
</span></p>
</td>
</tr>
</table>
</form></body>
</html>

```

Empins.jsp

```
<%@ page contentType="text/html; charset=iso-8859-1" language="java"
import="java.sql.*" errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Online Answering.com</title>
</head>

<body>

<%
    String p1=request.getParameter("textfield");
String p2=request.getParameter("textfield2");
    String p3=request.getParameter("textfield3");
String p4=request.getParameter("textfield4");
    String p5=request.getParameter("textfield5");
String p6=request.getParameter("textfield6");
    String p7=request.getParameter("textfield7");

    Class.forName("com.mysql.jdbc.Driver");
        Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost/Cab","root","1234");
        Statement statement = connection.createStatement();

String query="insert into employee
values(""+p1+"",""+p2+"",""+p3+"",""+p4+"",""+p5+"",""+p6+"");

    statement.executeUpdate(query);

    response.sendRedirect("employee.jsp");
%>

</body>
</html>
```

Route.jsp

[Home](#) | [Registration](#) | [Sign In](#) | [Contact Us](#) | [Admin](#) | [Feedback](#)

ONLINE CAB SCHEDULING SYSTEM



Add Route Details

Route ID	<input type="text" value="R02"/>
	<input type="text"/>
Route Description	<input type="text"/>
Start Time	<input type="text"/>
End Time	<input type="text"/>
Total Cost for the route	<input type="text"/>
	<input type="button" value="Submit"/>



```
<%@ page language="java" import="java.sql.*" %>
<%@ page session="true" %>
<html>
<head>
<title>Online Cab Scheduling System</title>
<style type="text/css">
<!--
.style2 {font-family: Arial, Helvetica, sans-serif}
-->
</style>
</head>
<body>
<script language="javascript">

function validate(form)
{
if (form.textfield.value=="")
{
    alert("ID should not be blank. Please enter it.");
    form.textfield.focus(); return false;
}

if (form.textfield2.value=="")
{
    alert("Field should not be blank. Please enter it.");
    form.textfield2.focus(); return false;
}
if (form.textfield3.value=="")
{
    alert("Field should not be blank. Please enter it.");
    form.textfield3.focus(); return false;
}
}
```

```

        if (form.textfield4.value=="")
        {
            alert("Field should not be blank. Please enter it.");
            form.textfield4.focus(); return false;
        }
    }

</script>
<form action="routeins.jsp" method="post" onSubmit="return validate(this)">

<table width="731" border="0" align="center" cellspacing="0">
<tr>
    <td colspan="2" height="120"><%@ include file="header1.jsp" %></td>
</tr>

<tr>
<td width="150" align="left" valign="top">&nbsp;</td>
<td width="750" align="center" valign="top"><p>&nbsp;</p>
<table width="380" border="0">
<tr>
    <td height="32" colspan="2"><strong>Add Route Details </strong></td>
    <td width="85">&nbsp;</td>
</tr>

<tr>
<td width="137">&nbsp;</td>
<td width="144">&nbsp;</td>
<td>&nbsp;</td>
</tr>
<tr>
<td><strong>Route ID
    <%
        Class.forName("com.mysql.jdbc.Driver");
        Connection connection =
        DriverManager.getConnection("jdbc:mysql://localhost/Cab","root","");
        Statement statement = connection.createStatement();

        String query="select * from route";
        ResultSet rs = statement.executeQuery(query);
        int i=0;
        while(rs.next())
        {
            i=i+1;
        }
        String id="R0"+(i+1);
    %>
    </strong></td>
    <td><input name="textfield" type="text" id="textfield"
    value='<%=id%>'></td>

```

```

        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Cab No. </strong></td>
        <td><input name="textfield2" type="text" id="textfield2"></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Route Desription </strong></td>
        <td><textarea name="textfield3"></textarea></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Start Time </strong></td>
        <td><input type="text" name="textfield4"></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>End Time </strong></td>
        <td><input type="text" name="textfield5"></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Total Cost for the route </strong></td>
        <td><input type="text" name="textfield6"></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td><input type="submit" name="Submit" value="Submit"></td>
        <td>&nbsp;</td>
    </tr>
</table>
<p><BR __designer:dtid="2251804108652552" />
    <span class="style2"><BR __designer:dtid="2251804108652556" />
    </span></p>
</td>
</tr>
</table>
</form></body>
</html>

```

Routeins.jsp

```

<%@ page language="java" import="java.sql.*" %>
<%@ page session="true" %>
<html>
<head>
<title>Online Cab Scheduling System</title>
<style type="text/css">
<!--
.style2 {font-family: Arial, Helvetica, sans-serif}
-->

```

```

</style>
</head>
<body>
<script language="javascript">

function validate(form)
{
if (form.textfield.value=="")
    {    alert("ID should not be blank. Please enter it.");
      form.textfield.focus(); return false;
    }

    if (form.textfield2.value=="")
    {    alert("Field should not be blank. Please enter it.");
      form.textfield2.focus(); return false;
    }
    if (form.textfield3.value=="")
    {    alert("Field should not be blank. Please enter it.");
      form.textfield3.focus(); return false;
    }
    if (form.textfield4.value=="")
    {    alert("Field should not be blank. Please enter it.");
      form.textfield4.focus(); return false;
    }
}

}

</script>

<form action="routeins.jsp" method="post" onSubmit="return validate(this)">

<table width="731" border="0" align="center" cellspacing="0">
<tr>
<td colspan="2" height="120"><%@ include file="header1.jsp" %></td>
</tr>

<tr>
<td width="150" align="left" valign="top">&nbsp;</td>
<td width="750" align="center" valign="top"><p>&nbsp;</p>
<table width="380" border="0">
<tr>
<td height="32" colspan="2"><strong>Add Route Details </strong></td>
<td width="85">&nbsp;</td>
</tr>

<tr>
<td width="137">&nbsp;</td>
<td width="144">&nbsp;</td>

```

```

        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Route ID
        <%

                Class.forName("com.mysql.jdbc.Driver");
Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost/Cab","root","");
                Statement statement = connection.createStatement();

String query="select * from route";
ResultSet rs = statement.executeQuery(query);
int i=0;
while(rs.next())
{
i=i+1;
}
String id="R0"+(i+1);
%>
        </strong></td>
        <td><input name="textfield" type="text" id="textfield"
value='<%=id%>'></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Cab No. </strong></td>
        <td><input name="textfield2" type="text" id="textfield2"></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Route Description </strong></td>
        <td><textarea name="textfield3"></textarea></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Start Time </strong></td>
        <td><input type="text" name="textfield4"></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>End Time </strong></td>
        <td><input type="text" name="textfield5"></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Total Cost for the route </strong></td>
        <td><input type="text" name="textfield6"></td>
        <td>&nbsp;</td>
    </tr>
    <tr>

```

```

<td>&nbsp;</td>
<td><input type="submit" name="Submit" value="Submit"></td>
<td>&nbsp;</td>
</tr>
</table>
<p><BR __designer:dtid="2251804108652552" />
<span class="style2"><BR __designer:dtid="2251804108652556" />
</span></p>
</td>
</tr>
</table>
</form></body>
</html>

```

Cab.jsp

```

<%@ page language="java" import="java.sql.*" %>
<%@ page session="true" %>
<html>
<head>
<title>Online Cab Scheduling System</title>
<style type="text/css">
<!--
.style2 {font-family: Arial, Helvetica, sans-serif}
-->
</style>
</head>
<body>
<script language="javascript">

```

```
function validate(form)
```

```

{
if (form.textfield.value=="")
    {
        alert("ID should not be blank. Please enter it.");
        form.textfield.focus(); return false;
    }

    if (form.textfield2.value=="")
    {
        alert("Field should not be blank. Please enter it.");
        form.textfield2.focus(); return false;
    }
    if (form.textfield3.value=="")
    {
        alert("Field should not be blank. Please enter it.");
        form.textfield3.focus(); return false;
    }
    if (form.textfield4.value=="")
    {
        alert("Field should not be blank. Please enter it.");
        form.textfield4.focus(); return false;
    }
}

}

</script>

<form ENCTYPE="multipart/form-data" ACTION="picins.jsp"
METHOD="POST" onSubmit="return validate(this)">

<table width="731" border="0" align="center" cellspacing="0">
<tr>
    <td colspan="2" height="120"><%@ include file="header1.jsp" %></td>
</tr>

<tr>
<td width="150" align="left" valign="top">&nbsp;</td>
<td width="750" align="center" valign="top"><p>&nbsp;</p>
<table width="380" border="0">
<tr>
    <td height="32" colspan="2"><strong>Add Cab Details </strong></td>
    <td width="85">&nbsp;</td>
</tr>

<tr>
    <td width="137">&nbsp;</td>
    <td width="144">&nbsp;</td>
    <td>&nbsp;</td>
</tr>
<tr>
<td><strong>Cab ID
    <%

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

```

```

Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost/Cab","root","");
Statement statement = connection.createStatement();

```

```

String query="select * from Cab";
ResultSet rs = statement.executeQuery(query);
int i=0;
while(rs.next())
{
i=i+1;
}
String id="B0"+(i+1);
%>
</strong></td>
<td><input name="textfield" type="text" id="textfield"
value='<%=id%>'></td>
<td>&nbsp;</td>
</tr>
<tr>
<td><strong>Cab No. </strong></td>
<td><input name="textfield2" type="text" id="textfield2"></td>
<td>&nbsp;</td>
</tr>
<tr>
<td><strong>Available Seats </strong></td>
<td><input type="text" name="textfield3"></td>
<td>&nbsp;</td>
</tr>
<tr>
<td><strong>Reserved Seats </strong></td>
<td><input type="text" name="textfield4"></td>
<td>&nbsp;</td>
</tr>
<tr>
<td><strong>Cab Type </strong></td>
<td><input type="text" name="textfield5"></td>
<td>&nbsp;</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td><textarea name="textfield6"></textarea></td>
<td>&nbsp;</td>
</tr>
<tr>
<td><strong>Image</strong></td>
<td><input type="file" name="file"></td>
<td>&nbsp;</td>
</tr>
<tr>
<td>&nbsp;</td>
<td><input type="submit" name="Submit" value="Submit"></td>

```

```

        <td>&nbsp;</td>
    </tr>
</table>
<p><BR __designer:dtid="2251804108652552" />
    <span class="style2"><BR __designer:dtid="2251804108652556" />
    </span></p>
</td>
</tr>
</table>
</form></body>
</html>

```

Picins.jsp

```

<%@ page language="java"
import="java.io.*,java.util.*,java.sql.*,org.apache.commons.fileupload.*,org.apache.c
ommons.fileupload.servlet.ServletFileUpload,org.apache.commons.fileupload.disk.*"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
    <title>Cab TRAVELLINGMgt System</title>

    <meta http-equiv="pragma" content="no-cache">
    <meta http-equiv="cache-control" content="no-cache">
    <meta http-equiv="expires" content="0">
    <meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
    <meta http-equiv="description" content="This is my page">
    <!--
    <link rel="stylesheet" type="text/css" href="styles.css">
    -->

</head>

<body>

    <center><h1>Picture has been Uploaded</h1></center>
    <%
        String p1="";
String p2="";
String p3="";
String p4="";
String p5="";
String p6="";

    %>
    <%
boolean isMultipart = ServletFileUpload.isMultipartContent(request);
if (!isMultipart) {
} else {
    FileItemFactory factory = new DiskFileItemFactory();
    ServletFileUpload upload = new ServletFileUpload(factory);

```

```

List items = null;
try {
items = upload.parseRequest(request);
} catch (FileUploadException e) {
e.printStackTrace();
}
Iterator itr = items.iterator();
while (itr.hasNext())
{
FileItem item = (FileItem) itr.next();
if (item.isFormField())
{
String name = item.getFieldName();
String value = item.getString();
if(name.equals("textfield"))
{
p1=value;
}
if(name.equals("textfield2"))
{
p2=value;
}
if(name.equals("textfield3"))
{
p3=value;
}
if(name.equals("textfield4"))
{
p4=value;
}
if(name.equals("textfield5"))
{
p5=value;
}
if(name.equals("textfield6"))
{
p6=value;
}
}
} else
{
try {
String saveFile = item.getName();
saveFile = saveFile.substring(saveFile.lastIndexOf("\\")+1);

```

```

File savedFile1 = new File("c:\\program files\\Apache software
foundation\\tomcat 5.5\\webapps\\Cab traveling\\images\\"+saveFile);

item.write(savedFile1);
out.print(saveFile);
    %><center>
    <img border="2" src=images/<%=saveFile %> width="137" height="140">
    <%

        out.println("<br><b>Cab ID :<b>"+p1);

            out.println("<br><b>Cab Description :<b>"+p6);

                // out.println("<br><b>Picture ID :<b>"+picid);

String s="images/"+saveFile;

Class.forName("com.mysql.jdbc.Driver");
    Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost/Cab","root","1234");
    Statement statement = connection.createStatement();
String query="insert into Cab
values("+p1+", "+p2+", "+p3+", "+p4+", "+p5+", "+p6+", "+s+")";

statement.executeUpdate(query);
out.println("<br><br>Record inserted");

    } catch (Exception e) {
    out.print(e.getMessage());
    }
    }
    }
    }
    %>

</body>
</html>

```

ONLINE CAB SCHEDULING SYSTEM



Add Workstatus of employees

Emp ID	<input type="text" value="001"/>
Start Timing	<input type="text"/>
End Timings	<input type="text"/>
Route ID	<input type="text" value="R01"/>
<input type="button" value="Submit"/>	



```
<%@ page language="java" import="java.sql.*" %>
<%@ page session="true" %>
<html>
<head>
<title>Online Cab Scheduling System</title>
<style type="text/css">
<!--
.style2 {font-family: Arial, Helvetica, sans-serif}
-->
</style>
</head>
<body>
<script language="javascript">
function validate(form)
{
if (form.textfield.value=="")
{
alert("ID should not be blank. Please enter it.");
form.textfield.focus(); return false;
}

if (form.textfield2.value=="")
{
alert("Field should not be blank. Please enter it.");
form.textfield2.focus(); return false;
}
if (form.textfield3.value=="")
{
alert("Field should not be blank. Please enter it.");
form.textfield3.focus(); return false;
}
}
```

```

        if (form.textfield4.value=="")
        {
            alert("Field should not be blank. Please enter it.");
            form.textfield4.focus(); return false;
        }
    }

</script>

<form action="workins.jsp" method="post" onSubmit="return validate(this)">

<table width="731" border="0" align="center" cellspacing="0">
<tr>
    <td colspan="2" height="120"><%@ include file="header1.jsp" %></td>
</tr>

<tr>
<td width="150" align="left" valign="top">&nbsp;</td>
<td width="750" align="center" valign="top"><p>&nbsp;</p>
<table width="380" border="0">
    <tr>
        <td height="32" colspan="2"><strong>Add Workstatus of employees
</strong></td>
        <td width="85">&nbsp;</td>
    </tr>

    <tr>
        <td width="137">&nbsp;</td>
        <td width="144">&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Emp ID </strong></td>
        <td><select name="select" id="select">
            <%
                Class.forName("com.mysql.jdbc.Driver");
                Connection connection =
                DriverManager.getConnection("jdbc:mysql://localhost/Cab","root","");
                Statement statement = connection.createStatement();

                String query2="select * from employee";
                ResultSet rs2 = statement.executeQuery(query2);
                while(rs2.next())
                {
                    %>
                    <option><%=rs2.getString(1)%></option>
                    <%
                }
            </td>
    </tr>
</table>
</td>
</tr>

```

```

%>
  </select></td>
  <td>&nbsp;</td>
</tr>
<tr>
  <td><strong>Start Timing </strong></td>
  <td><input name="textfield2" type="text" id="textfield2"></td>
  <td>&nbsp;</td>
</tr>
<tr>
  <td><strong>End Timings </strong></td>
  <td><input type="text" name="textfield3"></td>
  <td>&nbsp;</td>
</tr>
<tr>
  <td><strong>Cab No. </strong></td>
  <td><input type="text" name="textfield4"></td>
  <td>&nbsp;</td>
</tr>

<tr>
  <td><strong>Route ID </strong></td>
  <td><select name="select2" id="select2">
    <%String query="select * from route";
ResultSet rs1 = statement.executeQuery(query);
while(rs1.next())
{
%>
  <option><%=rs1.getString(1)%></option>
  <%
}
%>
  </select></td>
  <td>&nbsp;</td>
</tr>

<tr>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
</tr>
<tr>
  <td>&nbsp;</td>
  <td><input type="submit" name="Submit" value="Submit"></td>
  <td>&nbsp;</td>
</tr>
</table>
<p><BR __designer:dtid="2251804108652552" />
  <span class="style2"><BR __designer:dtid="2251804108652556" />
  </span></p>

```

```
</td>
</tr>
</table>
</form></body>
</html>
```

Workins.jsp

```
<%@ page contentType="text/html; charset=iso-8859-1" language="java"
import="java.sql.*" errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Online Answering.com</title>
</head>

<body>

<%
    String p1=request.getParameter("select");
    String p2=request.getParameter("textfield2");
    String p3=request.getParameter("textfield3");
    String p4=request.getParameter("textfield4");
    String p5=request.getParameter("select2");

    Class.forName("com.mysql.jdbc.Driver");
        Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost/Cab","root","1234");
        Statement statement = connection.createStatement();

String query="insert into workstatus
values("+p1+"",""+p2+"",""+p3+"",""+p4+"",""+p5+"");

statement.executeUpdate(query);

response.sendRedirect("workstatus.jsp");
%>

</body>
</html>
```



```

<th width="645" align="left" valign="top" scope="col"><p>&nbsp;</p>
<table width="618" border="0" cellspacing="0" cellpadding="0">
<tr>
<th width="618" height="375" align="left" valign="top" scope="col"><h2
align="center" class="style8">Cabes Details</h2>
<%

```

```

Class.forName("com.mysql.jdbc.Driver");
Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost/Cab","root","");
Statement statement = connection.createStatement();

```

```

String query="select * from Cab";
ResultSet rs = statement.executeQuery(query);
%>

```

```

<table width="582" border="1" align="center" cellspacing="0" >
<tr>
<td width="35"><span class="style9">Cab Id</span></td>
<td width="37"><span class="style9">Cab No. </span></td>
<td width="79"><span class="style9">Available Seats </span></td>
<td width="77"><span class="style9">Seats Reserved </span></td>
<td width="50"><span class="style9">Cab Type </span></td>
<td width="170"><span class="style9">Description </span></td>
</tr>
<%
while(rs.next())
{
%>
<tr>
<td height="104"><%=rs.getString(1)%></td>
<td><%=rs.getString(2)%></td>
<td><%=rs.getString(3)%></td>
<td><%=rs.getString(4)%></td>
<td><%=rs.getString(5)%></td>
<td><%=rs.getString(6)%></td>
<td width="104"><img src='<%=rs.getString(7)%>' height="100" width="100"
/></td>
</tr>
<%=}%>
</table>

```

```

<p align="center" class="style8">&nbsp;</p></th>
</tr>
</table>
<h2 align="center" class="style8">&nbsp;</h2>
</th>
</tr>
</table>
<p>&nbsp;</p>

```


Repo_route.jsp

Routes Available

Route Id	: No.	Route Description	Arrival Time	Departure Time	Cost of Ticket
R01	DL12 7896	5 Hours	3:00p.m.	6:00a.m.	800

```
start Macromedia Dreamw... Bus Traveling Mgt Sy... cab traveling system... 5:00 PM
<strong></strong><%@ page contentType="text/html; charset=iso-8859-1"
language="java" import="java.sql.*" errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title> Online Cab Scheduling System </title>
<style type="text/css">
<!--
.style8 {color: #0000FF}
-->
</style>
</head>

<body>
<table width="200" border="0" cellspacing="0" cellpadding="2">
<tr>
<td>&nbsp;</td>
</tr>
</table>
<table width="931" border="0" cellspacing="0" cellpadding="0">
<tr>
<th width="180" height="67" align="left" valign="top"
scope="col">&nbsp;</th>
<th width="106" scope="col">&nbsp;</th>
<th width="645" align="left" valign="top" scope="col"><p>&nbsp;</p>
<table width="618" border="0" cellspacing="0" cellpadding="0">
<tr>
```

```

        <th width="618" height="375" align="left" valign="top" scope="col"><h2
align="center" class="style8">Routes Available </h2>
        <%

```

```

                Class.forName("com.mysql.jdbc.Driver");
Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost/Cab","root","");
                Statement statement = connection.createStatement();

```

```

String query="select * from route";
ResultSet rs = statement.executeQuery(query);
%>

```

```

        <table width="569" border="1" align="center" cellspacing="0" >
<tr>
<td width="58"><strong>Route Id</strong></td>
<td width="59"><strong>Cab No. </strong></td>
<td width="108"><strong>Route Description </strong></td>
<td width="70"><strong>Arrival Time</strong></td>
<td width="73"><strong>Departure Time</strong></td>
<td width="79"><strong>Cost of Ticket</strong></td>

</tr>
<%
while(rs.next())
{
%>
<tr>
<td><%=rs.getString(1)%></td>
<td><%=rs.getString(2)%></td>
<td><%=rs.getString(3)%></td>
<td><%=rs.getString(4)%></td>
<td><%=rs.getString(5)%></td>
<td><%=rs.getString(6)%></td>

</tr>
<%=}%>
</table>

```

```

        <p align="center" class="style8">&nbsp;</p></th>
</tr>
</table>
        <h2 align="center" class="style8">&nbsp;</h2>
</th>
</tr>
</table>
<p>&nbsp;</p>
<p>&nbsp;</p>

```


Repo_emp.jsp

Employees Details

Emp Id	Name	Contact No	Designation	Joining Date	Salary	
001	sa	wz23	9876331632	null	null	null

```
start Macromedia Dreamw... Bus Traveling Mgt Sy... cab traveling system... 5:01 PM
<strong></strong><%@ page contentType="text/html; charset=iso-8859-1"
language="java" import="java.sql.*" errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title> Online Cab Scheduling System </title>
<style type="text/css">
<!--
.style8 {color: #0000FF}
-->
</style>
</head>

<body>
<table width="200" border="0" cellspacing="0" cellpadding="2">
<tr>
<td>&nbsp;</td>
</tr>
</table>
<table width="931" border="0" cellspacing="0" cellpadding="0">
<tr>
<th width="180" height="67" align="left" valign="top"
scope="col">&nbsp;</th>
<th width="106" scope="col">&nbsp;</th>
<th width="645" align="left" valign="top" scope="col"><p>&nbsp;</p>
<table width="618" border="0" cellspacing="0" cellpadding="0">
<tr>
```

```
<th width="618" height="375" align="left" valign="top" scope="col"><h2 align="center" class="style8">Employees Details</h2>
<%
```

```
Class.forName("com.mysql.jdbc.Driver");
Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost/Cab","root","");
Statement statement = connection.createStatement();
```

```
String query="select * from employee";
ResultSet rs = statement.executeQuery(query);
%>
```

```
<table width="473" border="1" align="center" cellspacing="0" >
<tr>
<td width="76"><strong>Emp Id</strong></td>
<td width="99"><strong>Name </strong></td>
<td width="161"><strong>Contact No </strong></td>
<td width="119"><strong>Designation </strong></td>
<td width="161"><strong>Joining Date </strong></td>
<td width="119"><strong>Salary </strong></td>

</tr>
<%
while(rs.next())
{
%>
<tr>
<td><%=rs.getString(1)%></td>
<td><%=rs.getString(2)%></td>
<td><%=rs.getString(3)%></td>
<td><%=rs.getString(4)%></td>
<td><%=rs.getString(5)%></td>
<td><%=rs.getString(6)%></td>
<td><%=rs.getString(7)%></td>

</tr>
<%}%>
</table>
```

```
<p align="center" class="style8">&nbsp;</p></th>
</tr>
</table>
<h2 align="center" class="style8">&nbsp;</h2>
</th>
</tr>
</table>
<p>&nbsp;</p>
<p>&nbsp;</p>
```


7. CODE EFFICIENCY

Reviewing of Code efficiency for a module is carried out after the module is successfully compiled and all the syntax errors eliminated. Code efficiency review is extremely cost-effective strategies for reduction in coding errors in order to produce high quality code. Normally, two types of efficiency are carried out on the code of a module - code optimization and code inspection. The procedure and final objective of these two efficiency techniques are very different as discussed below.

8. OPTIMIZATION OF CODE

Code optimization is an informal code analysis technique. In this technique, after a module has been coded, it is successfully compiled and all syntax errors are eliminated. Some members of the development team are given the code a few days before the optimization meeting to read and understand the code. Each member selects some test cases and simulates execution of the code by hand (i.e. trace execution through each statement and function execution). The main objectives of the optimization are to discover the algorithmic and logical errors in the code. The members note down their findings to discuss these in a optimization meeting where the coder of the module is also present.

Even though a code optimization is an informal analysis technique, several guidelines have evolved over the years for making this naïve technique more effective and useful. Of course, these guidelines are based on personal experience, common sense, and several subjective factors. Therefore are based on personal experience, common sense, and several subjective factors. Therefore, guidelines should be considered as examples rather than as rules to be applied dogmatically. Some of these guidelines are the following:

The team performing the code optimization should not be either too big or too small. Ideally, it should consist of three to seven members.

9. TESTING

Software Testing is an empirical investigation conducted to provide stakeholders with information about the quality of the product or service under test, with respect to the context in which it is intended to operate. This includes, but is not limited to, the process of executing a program or application with the intent of finding software bugs. It can also be stated as the process of validating and verifying that a software program/application/product meets the business and technical requirements that guided its design and development, so that it works as expected and can be implemented with the same characteristics.

A primary purpose for testing is to detect software failures so that defects may be uncovered and corrected. This is a non-trivial pursuit. Testing cannot establish that a product functions properly under all conditions but can only establish that it does not function properly under specific conditions.^[11] The scope of software testing often includes examination of code as well as execution of that code in various environments and conditions as well as examining the aspects of code: does it do what it is supposed to do and do what it needs to do. In the current culture of software development, a testing organization may be separate from the development team. There are various roles for testing team members. Information derived from software testing may be used to correct the process by which software is developed.

Defects and failures

Not all software defects are caused by coding errors. One common source of expensive defects is caused by requirements gaps, e.g., unrecognized requirements, that result in errors of omission by the program designer. A common source of requirements gaps is non-functional requirements such as testability, scalability, maintainability, usability, performance, and security.

Software faults occur through the following process. A programmer makes an error (mistake), which results in a defect (fault, bug) in the software source code. If this defect is executed, in certain situations the system will produce wrong results, causing a failure.^[12] Not all defects will necessarily result in failures. For example, defects in dead code will never result in failures. A defect can turn into a failure when the environment is changed. Examples of these changes in environment include the

software being run on a new hardware platform, alterations in source data or interacting with different software.^[12] A single defect may result in a wide range of failure symptoms.

Compatibility

A frequent cause of software failure is compatibility with another application, a new operating system, or, increasingly, web browser version. In the case of lack of backward compatibility, this can occur (for example...) because the programmers have only considered coding their programs for, or testing the software upon, "the *latest* version of" this-or-that operating system. The unintended consequence of this fact is that: their latest work might not be fully compatible with earlier mixtures of software/hardware, or it might not be fully compatible with *another* important operating system. In any case, these differences, whatever they might be, may have resulted in (unintended...) software failures, as witnessed by some significant population of computer users.

This could be considered a "prevention oriented strategy" that fits well with the latest testing phase suggested by Dave Gelperin and William C. Hetzel, as cited below ^[13].

Input combinations and preconditions

A very fundamental problem with software testing is that testing under *all* combinations of inputs and preconditions (initial state) is not feasible, even with a simple product. This means that the number of defects in a software product can be very large and defects that occur infrequently are difficult to find in testing. More significantly, non-functional dimensions of quality (how it is supposed to *be* versus what it is supposed to *do*) -- for example, usability, scalability, performance, compatibility, reliability -- can be highly subjective; something that constitutes sufficient value to one person may be intolerable to another.

Static vs. dynamic testing

There are many approaches to software testing. Reviews, walkthroughs or inspections are considered as static testing, whereas actually executing programmed code with a given set of test cases is referred to as dynamic testing. The former can be, (and unfortunately in practice often is...) omitted, whereas the latter takes place

when programs begin to be used for the first time - which is normally considered the beginning of the testing stage. This may actually begin before the program is 100% complete in order to test particular sections of code (modules or discrete functions). For example, Spreadsheet programs are, by their very nature, tested to a large extent "on the fly" during the build process as the result of some calculation or text manipulation is shown interactively immediately after each formula is entered

1. UNIT TESTING:

This is the smallest testable unit of a computer system and is normally tested using the white box testing. The author of the programs usually carries out unit tests.

2. INTEGRATION TESTING:

In integration testing, the different units of the system are integrated together to form the complete system and this type of testing checks the system as whole to ensure that it is doing what is supposed to do. The testing of an integrated system can be carried out top-down, bottom-up, or big-bang. In this type of testing, some parts will be tested with white box testing and some with black box testing techniques. This type of testing plays very important role in increasing the systems productivity. We have checked our system by using the integration testing techniques.

3. SYSTEM TESTING:

A part from testing the system to validate the functionality of software against the requirements, it is also necessary to test the non-functional aspect of the system. Some examples of non-functional tools include tests to check performance, data security, usability/user friendliness, volume, load/stress that we have used in our project to test the various modules.

System testing consists of the following steps:

1. Program(s) testing.
2. String testing.
3. System testing.
4. System documentation.
5. User acceptance testing.

4. FIELD TESTING:

This is a special type of testing that may be very important in some projects. Here the system is tested in the actual operational surroundings. The interfaces with other systems and the real world are checked. This type of testing is very rarely used. So far our project is concerned; we haven't tested our project using the field testing.

5. ACCEPTANCE TESTING:

After the developer has completed all rounds of testing and he is satisfied with the system, then the user takes over and re-tests the system from his point of view to judge whether it is acceptable according to some previously identified criteria. This is almost always a tricky situation in the project because of the inherent conflict between the developer and the user. In this project, it is the job of the bookstores to check the system that whether the made system fulfills the goals or not.

WHY SYSTEM TESTING?

Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. Inadequate testing results in two types of problems:

1. The time lag between the cause and the appearance of the problem.
2. The effect of system errors on the files and records within the system.

ACTIVITY NETWORK FOR SYSTEM TESTING

The test plan entails the following activities:

1. Prepare test plan.
2. Specify conditions for user acceptance testing.
3. Prepare test data for program testing.
4. Prepare test data for transaction path testing.
5. Plan user training.
6. Compile/assemble programs.
7. Prepare performance aids.
8. Prepare operational documents.

PREPARE TEST: A workable test plan must be prepared in accordance with established design specifications. It includes the following items:

- Outputs expected from the system.
- Criteria for evaluating outputs.
- A volume of test data.
- Procedure for using test data.
- Personnel and training requirements.

SPECIFY CONDITIONS FOR USER ACCEPTANCE TESTING

Planning for user acceptance testing calls for the analyst and the user to agree on conditions for the test.

PREPARE TEST DATA FOR PROGRAM TESTING

As each program is coded, test data are prepared and documented to ensure that all aspects of the program are properly tested.

PREPARE TEST DATA FOR TRANSACTION PATH TESTING

This activity develops the data required for testing every condition and transactions to be introduced into the system. The path of each transaction from origin to destination is carefully tested reliable results.

PLAN USER TRAINING

User training is designed to prepare the user for testing and converting the system. User involvement and training take place parallel with programming for three reasons:

- The system group has time available to spend on training while the programs are being written.
- Initiating a user-training program gives the systems group a clearer image of the user's interest in the new system.
- A trained user participates more effectively in system testing.

The training plan is followed by preparation of the user training manual and other text materials.

COMPILE / ASSEMBLE PROGRAMS

All programs have to be compiled / assembled for testing.

PREPARE PERFORMANCE AIDS

In this activity the materials to be used by personnel to run the system are specified and scheduled. This includes a display of materials.

PREPARE OPERATIONAL DOCUMENTS

During the test plan stage, all operational documents are finalized including copies of the operational formats required by the candidate system.

SYSTEMS TESTING

The computer department to ensure that the system functions as specified does this testing. This testing is important to ensure that a working system is handed over to the user for acceptance testing.

ACCEPTANCE TESTING

The user to ensure that the system functions, as the user actually wanted performs this testing. With prototyping techniques, this stage becomes very much a formality to check the accuracy and completeness of processing. The screen layouts and output should already have been tested during the prototyping phase.

An error in the program code can remain undetected indefinitely. To prevent this from happening the code was tested at various levels. To successfully test a system, each condition, and combinations of conditions had to be tested. Each program was tested and linked to other programs. This unit of program is tested and linked to other units and so on until the complete system has been tested.

The purpose of testing is to ensure that each program is fully tested. To do so a test plan had to be created. The test plan consists of a number of test runs such as the valid paths through the code, and the exception and error handling paths. For each test run there is a list of conditions tested, the test data used and the result expected. The test plan was then reviewed to check that each path through the code is tested correctly. It is the responsibility of the programmer to collect the data that will produce the required test condition.

Projecthelpline.in

10. TEST CASES

	TEST DATA Specifications for Online Cab Scheduling System user form1		
Test Date		Programmer name:	Online Cab Scheduling System
Tested By:		Project ID:	11108765

User ID	<input style="width: 100%;" type="text"/>	
		The fields are required.Can enter only numeric (Length upto 8 digit)
Name	<input style="width: 100%;" type="text"/>	
		The fields are required.Can enter only letters, spaces, hyphens, and apostrophes. No numeric & special characters are allowed(Length upto 32 characters)
Address	<input style="width: 100%; height: 100px;" type="text"/>	
		the fields are required.Can enter only letters, spaces, hyphens, and apostrophes. No numeric & special characters are allowed(Length upto 132 characters)
City	<input style="width: 100%;" type="text"/>	Select here
		the fields are required.Can enter only letters, spaces, allowed(Length upto 80 characters)
Email	<input style="width: 100%;" type="text"/>	@yahoo.com
		Use 4 to 32 characters and start with a letter. You may use letters, numbers, underscores, and one dot(.)

Password password length > than 6 Don't use your Name or ID alphanumeric

Re-Type Pasword password length > than 6 alphanumeric

Security Question Use 4 characters or more- not case sensitive

Answer

CREATE MY ACCOUNT

POSITIVE TEST CASES FOR REGISTRATION FORM

T. C ID	PRE-CONDITION	T.C DESCRIPTION	T.C DATA	EXPECTED	ACTUAL	RESULT
1	User should be on https://Online Cab Scheduling System /registration? And is on Work ID Field	Check the functionality of Work ID field	1234	Will accept only numeric upto 8 digit . Work ID are required	Ok	Pass
2	User should be on https://Online Cab Scheduling System/registration? And is on Name Field	Check the functionality of Name field	Shikha	Will accept only letters, spaces, hyphens, and apostrophes.Length upto 32 characters. name are required.	Ok	Pass
3	User should be on https://Online Cab Scheduling System/registration? And is on Name Field	Check the functionality of Name field	A S	Will accept only letters, spaces, hyphens, and apostrophes.Length upto 32 characters. name are required.	Ok	Pass
4	User should be on https://Online Cab Scheduling System/registration? And is on Name Field	Check the functionality of Name field	Shikha S	Will accept only letters, spaces, hyphens, and apostrophes.Length upto 32 characters.name are required.	Ok	Pass
5	User should be on https://Online Cab Scheduling System/registration?And is on Name Field	Check the functionality of Name field	pooja sharma	Will accept only letters, spaces, hyphens, and apostrophes.Length upto 32 characters.name are required.	Ok	Pass

6	User should be on https://Online Cab Scheduling System/registration?And is on Address Field	Check the functionality of Address field	vikas puri	Will accept only letters, spaces, hyphens, and apostrophes.Length upto 132 characters. name are required.	Ok	Pass
7	User should be on https://Online Cab Scheduling System/registration? And is on Address Field	Check the functionality of Address field	m-10/20 uttamnagar	Will accept only letters, spaces, hyphens, and apostrophes.Length upto 132 characters. name are required.	Ok	Pass
8	User should be on https://Online Cab Scheduling System/registration? And is on Address Field	Check the functionality of Address field	g7-57	Will accept only letters, spaces, hyphens, and apostrophes.Length upto 132 characters. name are required.	Ok	Pass
9	User should be on https://Online Cab Scheduling System/registration? And is on city Field	Check the functionality of City field	Lucknow	Will accept only letters, spaces.Length upto 80 characters. name are required.	Ok	Pass
10	User should be on https://Online Cab Scheduling System/registration?And is on city Field	Check the functionality of City field	Lucknow delta	Will accept only letters, spaces.Length upto 80 characters. name are required.	Ok	Pass
14	User should be on https://Online Cab Scheduling System/registration? And is on Email Field	Check the functionality of ID of sign up option	Shikha.s10@yahoo.com	Will accept only letters, numbers, underscores, and one dot (.)	Ok	pass
15	User should be on https://Online Cab Scheduling System/registration? And is on Email Field	Check the functionality of ID of sign up option	100s_shika@yahoo.com	Will accept only letters, numbers, underscores, and one dot (.)	Ok	Pass
16	User should be on https://Online Cab Scheduling System/registration? And is on Email Field	Check the functionality of ID of sign up option	h.shikha100@yahoo.com	Will accept only letters, numbers, underscores, and one dot (.)	Ok	Pass
17	User should be on https://Online Cab Scheduling System/registration? And is on Email Field	Check the functionality of ID of sign up option	ak.7_1345@yahoo.com	Will accept only letters, numbers, underscores, and one dot (.)	Ok	Pass
18	User should be on https://Online Cab Scheduling System/registration? And is on Password Field	Check the functionality of password text box	avhjkLhm	Will accept 6 to 32 characters.Capitalisation matters. and don't use your name ID.	Ok	Pass
19	User should be on https://Online Cab Scheduling System/registration? And is on Password Field	Check the functionality of password text box	66778884	Will accept 6 to 32 characters.Capitalisation matters. and don't use your name or ID.	Ok	Pass
20	User should be on https://Online Cab Scheduling System/registration?And is on Password Field	Check the functionality of password text box	2207**	Will accept 6 to 32 characters.Capitalisation matters. and don't use your name or ID.	Ok	Pass
21	User should be on https://Online Cab Scheduling System/registration? And is on Password Field	Check the functionality of password text box	Chinnu1312**	Will accept 6 to 32 characters.Capitalisation matters. and don't use your name or ID.	Ok	Pass

27	User should be on https://Online Cab Scheduling System/registration? And is on Security Question field	Check the functionality of Security Question	What was the make of your first car?	Either select one from drop list or type one	Ok	Pass
28	User should be on https://Online Cab Scheduling System/registration? And is on Answer field	Check the functionality of Answer text box	Santro	Will accept letters and a single space only	Ok	Pass

NEGATIVE TEST CASES FOR REGISTRATION FORM

T.C ID	PRE-CONDITION	T.C DESCRIPTION	T.C DATA	EXPECTED	ACTUAL	RESULT
1	User should be on https://Online Cab Scheduling System/registration?And is on Work ID Field	Check the functionality of Work ID		Will accept only numeric upto 8 digit . Work ID ID are required		
2	User should be on https://Online Cab Scheduling System/registration?And is on Name Field	Check the functionality of Name field	A	Will accept only letters, spaces, hyphens, and apostrophes.Length upto 32 characters. name are required.	Invalid	Fail
3	User should be on https://Online Cab Scheduling System/registration? And is on Name Field	Check the functionality of Name field	Shikha's	Will accept only letters, spaces, hyphens, and apostrophes.Length upto 32 characters. name are required.	Invalid	Fail
4	User should be on https://Online Cab Scheduling System/registration?And is on Name Field	Check the functionality of Name field	Satish	Will accept only letters, spaces, hyphens, and apostrophes.Length upto 32 characters.name are required.	Invalid	Fail
5	User should be on https://Online Cab Scheduling System/registration?And is on Name Field	Check the functionality of Name field	,	Will accept only letters, spaces, hyphens, and apostrophes.Length upto 32 characters. name are required.	Invalid	Fail
6	User should be on https://Online Cab Scheduling System/registration? And is on Address Field	Check the functionality of Address field		Will accept only letters, spaces, hyphens, and apostrophes.Length upto 132 characters. name are required.	Invalid	Fail

7	User should be on https://Online Cab Scheduling System/registration? And is on Address Field	Check the functionality of Address field	SPACE	Will accept only letters, spaces, hyphens, and apostrophes.Length upto 132 characters. name are required.	Invalid	Fail
8	User should be on https://Online Cab Scheduling System/registration? And is on city Field	Check the functionality of City field		Will accept only letters, spaces.Length upto 80 characters. name are required.	Invalid	Fail
9	User should be on https://Online Cab Scheduling System/registration? And is on city Field	Check the functionality of City field	SPACE	Will accept only letters, spaces.Length upto 80 characters. name are required.	Invalid	Fail
10	User should be on https://Online Cab Scheduling System/registration? And is on Email Field	Check the functionality of ID of sign up option	blank	Will accept only letters, numbers, underscores, and one dot (.).The ID must be available.	Invalid	Fail
11	User should be on https://Online Cab Scheduling System/registration? And is on Email Field	Check the functionality of ID of sign up option	Shikha@yahoo.com	Will accept only letters, numbers, underscores, and one dot (.).The ID must be available.	Invalid	Fail
12	User should be on https://Online Cab Scheduling System/registration? And is on Email Field	Check the functionality of ID of sign up option	#shikha@yahoo.com	Will accept only letters, numbers, underscores, and one dot (.).The ID must be available.	Invalid	Fail
13	User should be on https://Online Cab Scheduling System/registration? And is on Password Field	Check the functionality of password text box	2#	Will accept 6 to 32 characters.Capitalisation matters. and don't use your name or ID.	Invalid	Fail
14	User should be on https://Online Cab Scheduling System/registration? And is on Password Field	Check the functionality of password text box	a123	Will accept 6 to 32 characters.Capitalisation matters. and don't use your name or ID.	Invalid	Fail
15	User should be on https://Online Cab Scheduling System/registration? And is on Secret Question 2 field	Check the functionality of Secret Question Drop Down List	-Select one-	Either select one from drop list or type one	Invalid	Fail
16	User should be on https://Online Cab Scheduling System/registration? And is on Your Answer field	Check the functionality of Answer text box	blank	Will accept anything.But not blank	Invalid	Fail

11. VERIFICATION AND VALIDATION (V&V)

The objectives of verification, validity activities are to assess and improve the quality of the work products generated during development and modification of the software. Quality depends upon the various attributes like correctness, completeness, consistency, reliability, usefulness, usability, efficiency and conformance to standards.

The terms verification and validation are used synonymously. These are defined as under: -

Verification: “Are we building the product right?”

Validation: “Are we building the right product?”

Verification activities include proving, testing, and reviews. Validation is the process of evaluating software at the end of the software development to ensure compliance with the software requirements. Testing is a common method of validation. Clearly, for high reliability we need to perform both activities. Together, they are often called V&V activities.

The major V&V activities for software development are inspection, reviews, and testing (both static and dynamic). The V&V plan identifies the different V&V tasks for the different phases and specifies how these tasks contribute to the project V&V goals. The methods to be used for performing these V&V activities, the responsibilities and milestones for each of these activities, inputs and outputs for each V&V task, and criteria for evaluating the outputs are also specified.

The two major V&V approaches are testing and inspections. Testing is an activity that can be generally performed only on code. It is an important activity and is discussed in detail in a later chapter. Inspection is a more general activity that can be applied to any work product, including code. Many of the V&V tasks are such that for them, an inspection type of activity is the only possible way to perform the tasks (e.g. trace ability and document evaluation). Due to this, inspections play a significant role in verification.

12. POST IMPLEMENTATION MAINTENANCE AND REVIEW

As we know, creating software is one thing and the implementation of the created software is another. The process of implementing software is much difficult as compared to the task of creating the project. First we have to implement the software on a small scale for removing the bugs and other errors in the project and after removing them we can implement the software on a large scale.

Before we think in terms of implementing the Software on a large basis, we must consider the Hardware requirements.

Whenever we develop software or project a certain hardware and software is being used by the programmer for developing the project. The hardware and software to be used by the programmer for developing the project should be such that it would result in the development of a project, which would satisfy all the basic needs for which the project has been created by the programmer. The Hardware should be such that cost constraints of the Client should also be taken into account without affecting the performance.

12.1 Hardware Evaluation Factors

When we evaluate computer hardware, we should first investigate specific *physical and performance* characteristics for each hardware component to be acquired. These specific questions must be answered concerning many important factors. These *hardware evaluation factors* questions are summarized in the below figure.

Notice that there is much more to evaluating hardware than determining the fastest and cheapest computing device. For e.g. the question of possible obsolescence must be addressed by making a technology evaluation. The factor of *ergonomics* is also very important. Ergonomics is the science and technology that tries to ensure that computers and other technologies are "user-friendly", that is safe, comfortable and easy to use.

Connectivity is another important evaluation factor, since so many computer systems are now interconnected within wide area or local area telecommunications networks.

Hardware Evaluation Factors

- 1) Performance
- 2) Cost
- 3) Reliability
- 4) Availability
- 5) Compatibility
- 6) Modularity
- 7) Technology
- 8) Ergonomics
- 9) Connectivity
- 10) Environmental requirements
- 11) Software
- 12) Support

12.2 Software Evaluation Factors

Software can be evaluated according to many factors similar to the hardware evaluation. Thus the factors of *performance, cost, reliability, compatibility, modularity, technology, ergonomics, and support* should be used to evaluate proposed software acquisitions. In addition, however, *the software evaluation factors* are summarized in below figure. For e.g. some software packages require too much memory capacity and are notoriously slow, hard to use, or poorly documented. They are not a good selection for most end users, even if offered at attractive prices.

Software Evaluation Factors:

1. **EFFICIENCY:** is the software a well-written system of computer instructions that does not use much memory capacity or CPU time?
2. **FLEXIBILITY:** can it handle its processing assignments easily without major modifications?
3. **SECURITY:** does it provide control procedures for errors, malfunctions and improper use?

4. **LANGUAGE:** do our computer programmers and users write it in a programming language that is used?
5. **DOCUMENTATION:** is the s/w well documented? Does it include helpful user instructions?
6. **HARDWARE:** does existing hardware have the features required to best use this software?
7. Other characteristics of hardware such as its performance, what about the cost, how much is reliable and etc.

12.3 Conversion and Training

An important aspect of is to make sure that the new design is implemented to establish standards. The term implementation has different meanings, ranging from the conversion of a basic application to a complete replacement of a computer system. Implementation is used here to mean the process of converting a new or revise system into an operational one. Conversion is one aspect of implementation. Conversion means changing from one system to another. The objective is to put the tested system into operation while holding costs, risks, and personnel irritation to a minimum. It involves creating computer-compatible files, training the operation staff, and installing terminal and hardware. A critical aspect of conversion is not disrupting the functioning of the organization.

When a new system is used over and old, existing and running one, there are always compatibility errors. These errors are caused because of the lack of equipment or personnel to work the new system. Running any specified system at an organization does require some or other hardware or, in this case, software requirement as well.

Conversion is one aspect of implementation review & software maintenance.

There are three types of implementation:

1. Implementation of a computer system to replace a manual system. The problems encountered are converting files, training users, creating accurate files and verifying printouts for integrity.
2. Implementation of a new computer system to replace an existing one. This is usually a difficult conversion. If not properly planned there can be many problems. Some large computer systems have taken as long as year to convert.
3. Implementation of a modified application to replace an existing one, using the same computer. This type of conversion is relatively easy to handle, provided there are no major changes in the files.

12.4 Training Needs

Training needs refer to the gaining of knowledge required for running the system.

First of all the system is a computer based system therefore the person should have good knowledge about computer and its working.

He should know how to use software's on the computer.

For a better usage and working of the software the organization should appoint a person who has good knowledge of all the required software. The organization gets a person trained through different institutes present in the market. The training should be as per the above requirements.

13. COST ESTIMATION OF THE PROJECT

Cost in a project is due to the requirements for software, hardware, and human resources. Hardware resources are computer time, terminal time and memory required for the project. Software resources include the tools and compilers needed during development. The bulk of cost of software development is due to human resources needed. Cost estimates are determined in terms of person-months (PM).

Total No. Of Persons Involved In This Project:

1. Administrator
2. Senior Programmer
3. Junior Programmers
4. On line Users.

Since this Project will complete in 4 months

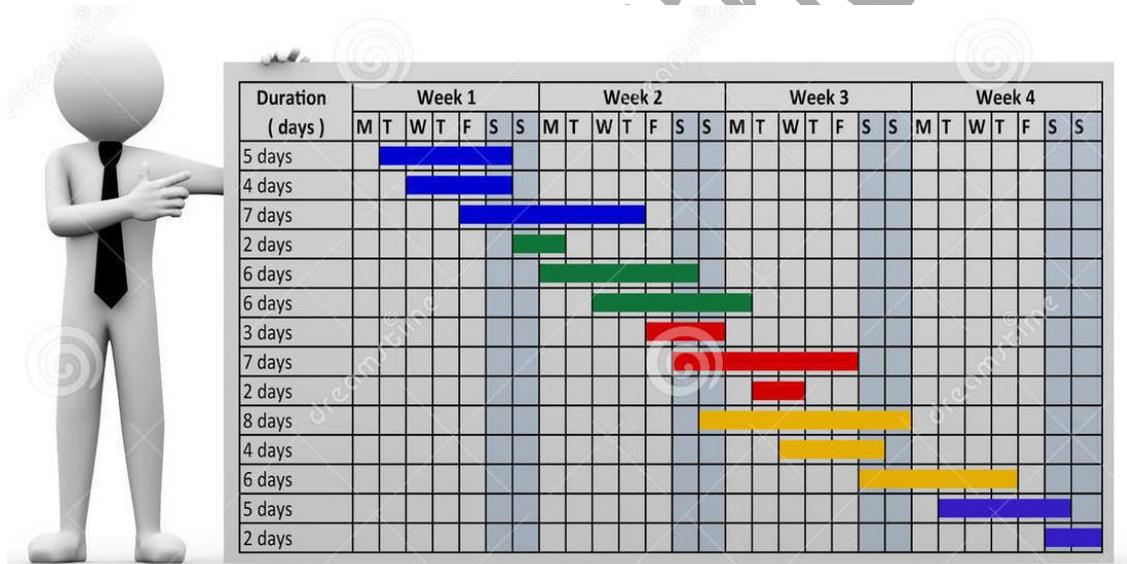
COST ESTIMATE: (Salary of Project Manager + Salary of Senior Programmer + 2 * Salary of Junior Programmer) * 2

GANTT & PERT CHART

GANTT CHART

Gantt charts mainly used to allocate resources to activities. The resources allocated to activities include staff, hardware, and software. Gantt charts (named after its developer Henry Gantt) are useful for resource planning. A Gantt chart is special type of bar chart where each bar represents an activity. The bars are drawn along a timeline. The length of each bar is proportional to the duration of the time planned for the corresponding activity.

Gantt chart is a project scheduling technique. Progress can be represented easily in a Gantt chart, by coloring each milestone when completed. The project will start in the month of November and end after 4 months at the end of February.



Pert Chart

PERT (Project Evaluation and Review Technique) charts consist of a network of boxes and arrows. The boxes represent activities and the arrows represent task dependencies.

PERT chart represents the statistical variations in the project estimates assuming a normal distribution. Thus in a PERT chart instead of making a single estimate for each task, *pessimistic*, *likely*, and *optimistic* estimates are also made. The boxes of PERT charts are usually annotated with the pessimistic, likely, and optimistic

estimates for every task. Since all possible completion times between the minimum and maximum durations for every task have to be considered, there are many critical paths, depending on the permutations of the estimates for each task. This makes critical path analysis in PERT charts very complex. A critical path in a PERT chart is shown by using thicker arrows. The PERT chart representation of the companies problem of Figure A. is shown in Figure B.

Task	ES	EF	LS	LF	ST
Specification Part	0	15	0	15	0
Design Database Part	15	60	15	60	0
Design GUI Part	15	45	90	120	75
Code Database Part	60	165	60	165	0
Code GUI Part	45	90	120	165	75
Integrate and Test	165	285	165	285	0
Write User Manual	15	75	225	285	210

Figure A : Different Tasks for the Online Cab Scheduling System are shown in above table.

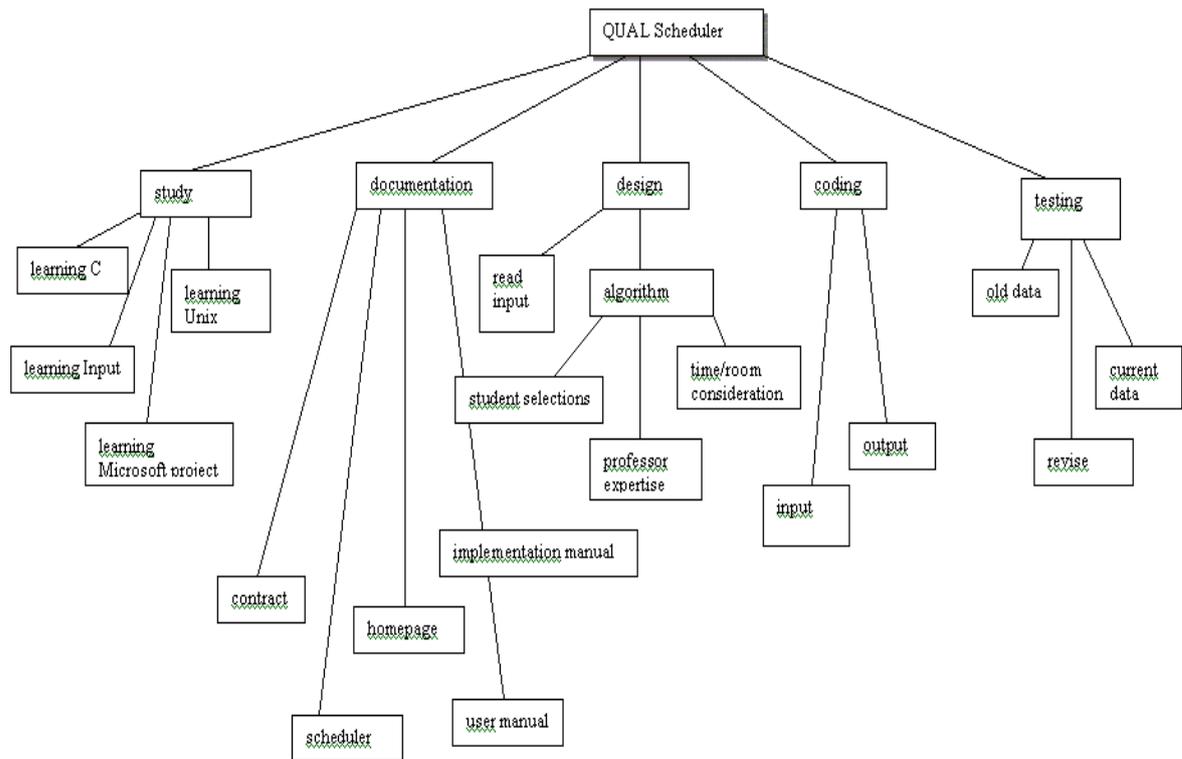


FIGURE B :PERT chart representation of the Online Cab Scheduling System.

PERT charts are a more sophisticated form of activity chart. In activity diagrams only the estimated task durations are represented. Since the actual durations might vary from the estimated durations, the utility of the activity diagrams is limited.

14. SECURITY AND VALIDATION CHECKS

Software's Vulnerability to Attack

Software development is not yet a science or a rigorous discipline, and the development process by and large is not controlled to minimize the vulnerabilities that attackers exploit.

The security of software is threatened at various points throughout its life cycle, both by inadvertent and intentional choices and actions taken by “insiders”—individuals closely affiliated with the organization that is producing, deploying, operating, or maintaining the software, and thus trusted by that organization—and by “outsiders” who have no affiliation with the organization. The software's security can be threatened

- 
- **during its development:** A developer may corrupt the software—intentionally or unintentionally—in ways that will compromise the software's dependability and trustworthiness when it is operational.
 - **during its deployment (distribution and installation):** If those responsible for distributing the software fail to tamperproof the software before shipping or uploading, or transmit it over easily intercepted communications channels, they leave the software vulnerable to intentional or unintentional corruption. Similarly, if the software's installer fails to “lock down” the host platform, or configures the software insecurely, the software is left vulnerable to access by attackers.
 - **during its operation:** Once COTS and open source software has gone operational, vulnerabilities may be discovered and publicized; unless security patches and updates are applied and newer supported versions (from which the root causes of vulnerabilities have been eliminated) are adopted, such software will become increasingly vulnerable. Non-commercial software and open source software (OSS) may also be vulnerable, especially as it may manifest untrustworthy behaviors over time due to changes in its environment that stress the software in ways that were not anticipated and simulated during its testing.
 - **during its sustainment:** If those responsible for addressing discovered vulnerabilities in released software fail to issue patches or updates in a timely

manner, or fail to seek out and eliminate the root causes of the vulnerabilities to prevent their perpetuation in future releases of the software, the software will become increasingly vulnerable to threats over time. Also, the software's maintainer may prove to be a malicious insider, and may embed malicious code, exploitable flaws, etc., in updated versions of the code.

The Challenge of Building Secure Software

1. **Dependability:** Dependable software executes predictably and operates correctly under all conditions, including hostile conditions, including when the software comes under attack or runs on a malicious host.
2. **Trustworthiness:** Trustworthy software contains few if any vulnerabilities or weaknesses that can be intentionally exploited to subvert or sabotage the software's dependability. In addition, to be considered trustworthy, the software must contain no malicious logic that causes it to behave in a malicious manner.
3. **Survivability (also referred to as "Resilience"):** Survivable—or resilient—software is software that is resilient enough to (1) either resist (i.e., protect itself against) or tolerate (i.e., continue operating dependably in spite of) most known attacks plus as many novel attacks as possible, and (2) recover as quickly as possible, and with as little damage as possible, from those attacks that it can neither resist nor tolerate.

Software Assurance

The main objective of software assurance is to ensure that the processes, procedures, and products used to produce and sustain the software conform to all requirements and standards specified to govern those processes, procedures, and products. Software security and secure software are often discussed in the context of software assurance. Software assurance in its broader sense refers to the assurance of any required property of software.

An increasingly agreed-upon approach for assuring the security of software is the software security assurance case, which is intended to provide justifiable confidence that the software under consideration (1) is free of vulnerabilities; (2) functions in the “intended manner,” and this “intended manner” does not compromise the security or any other required properties of the software, its environment, or the information it handles; and (3) can be trusted to continue operating dependably under all anticipated circumstances, including anomalous and hostile environmental and utilization circumstances—which means that those who build the software need to anticipate such circumstances and design and implement the software to be able to handle them gracefully. Such circumstances include

- the presence of unintentional faults in the software and its environment
- the exposure of the operational software to accidental events that threaten its security
- the exposure of the software to intentional choices or actions that threaten its security during its development, deployment, operation, or sustainment

Software is more likely to be assuredly secure when security is a key factor in the following aspects of its development and deployment:

- **development principles and practices:** The practices used to develop the software and the principles that governed its development are expressly intended to encourage and support the consideration and evaluation of security in every phase of the software’s development life cycle. Some secure development principles and practices for software are suggested later in this article.
- **development tools:** The programming language(s), libraries, and development tools used to design and implement the software are evaluated

and selected for their ability to avoid security vulnerabilities and to support secure development practices and principles.

- **testing practices and tools:** The software is expressly tested to verify its security, using tools that assist in such testing.
- **acquired components:** Commercial off-the-shelf (COTS) and OSS components are evaluated to determine whether they contain vulnerabilities, and if so whether the vulnerabilities can be remediate through integration to minimize the risk they pose to the software system.
- **deployment configuration:** The installation configuration of the software minimizes the exposure of any residual vulnerabilities it contains.
- **execution environment:** Protections are provided by the execution environment that can be leveraged to protect the higher level software that operates in that environment.

practitioner knowledge: The software's analysts, designers, developers, testers, and maintainers are provided with the necessary information (e.g., through training and education) to give them sufficient security awareness and knowledge to understand, appreciate, and effectively adopt the principles and practices that will enable them

15. SCOPE OF FUTURE APPLICATION

It is directly dependent on the lay stone of the project that is we will have to design a system which when the time passes having a better system initially should not become a joke later. The scope of my software is based on the idea of connecting people online. The next increment may scope changes suggested by a review of the preceding increment, but once the second increment commences, scope is again frozen temporarily. This approach enable the web application team to work without having to accommodate a continual stream of change but still recognizes the continuous evolution characteristics of most web application.

Besides that, the following basic quality in the software always safeguards the future scope of the software.

Correctness:-

When a program functions correctly according to their specification that it show the quality of correctness to the definition of correctness the specification of the system that determine unambiguous. Whether a program meets the specification

Reusability:

Reusability is possible as and when we required in this application. We can update its next version. Reusable software reduces design, coding and testing cost by amortizing effort over several designs. Reducing the amount of code also simplified understanding, which increases the likelihood that the code is correct. We followed up both types of reusability as sharing of newly written code within a project and reuse of previously written code on new projects.

Extensibility:

This application software is extended in ways that its original developers may not expect. The following principles enhance extensibility like hiding data structures, avoiding traversing multiple links or methods, avoiding case statements on object type and distinguishing public and private operations.

RoCabtness:

Its method is roCabt and it will not fail even if it receives improper parameters. There are some alert pages and messages are flashed out with some dialogue boxes to warn and inform the end user about the current processes going on. It also interacts with the user by alerting them about invalid parameters.

Understandability:

A method is understandable if anyone other than the developer of the method can understand the code (as well as the developer after a time-span).

Cost-effectiveness:

Its cost is under the budget and developed within given time period. It is always desirable to aim for a system with a minimum cost subject to the condition that it must satisfy all the requirements.

Projecthelping.in

15. CONCLUSION

The project "Online Cab Scheduling system" is based on managing Cab Travelling on a system. This system allow the project manager to maintain all needed details regarding Cab management, route management, passenger management and staff member management. The system provides a graphical user interface, which helps all the users to know the Cab routes and there related time tables. This system provides work status report for staff members who work with this system.

This system gives advantage by providing all information on a single click. All users who want to know information about Cab Travelling and those who want to reserve ticket are facilitated by this system. If any user want to know about his ticket status than this information is also provided by this system.

It also generates reports which give detailed information about the ticket status and employee's status according to this system.

Future enhancements for this project can be also created using Bar-charts by which the performance of each project can be better analyzed and by using this resource allocation can be done efficiently.



16. BIBLIOGRAPHY

[1] Herbert Scheldt, **Java Complete Reference**, Fifth Edition, Tata McGraw Hill Edition.

[2] Phil Hanna, **JSP 2.0: The Complete Reference**, Tata McGraw Hill Edition, 2003.

[3] Elmarsi and Navathe, **Fundamentals of Database System** (Third Edition), Addison Wesley.

[4] Ian Somerville, **Software Engineering**, Third Edition, Pearson Education.

[5] Ali Bahrami, **Object-Oriented System Development**, Third Edition, Tata McGraw Hill Edition.

[6] Ivan Bayross, **SQL, PL/SQL programming language of Oracle**, Second Edition, BPB Publication.

WEB REFERENCES

[1] www.google.com

[2] www.htmlcodetutorial.com