Automation of Travel Agency

A Project Report for Industrial Training

Submitted by

ANUJ DEEP, ALAMGIR ALAM, SAURABH RAJ, ARITRA DATTA

in partial fulfillment for the award of the degree of

B. TECH

At

Ardent Computech Pvt. Ltd.

26th December 2017 to 10th January 2018

Ardent Computech Pvt. Ltd.

BONAFIDE CERTIFICATE

Certified that this project work was carried out under my supervision

" ***Automation of Travel Agency***"**is the bonafide work of**

Name of the student: **ANUJ DEEP** ***Signature:***

Name of the student: **ALAMGIR ALAM** ***Signature:***

Name of the student: **SAURABH RAJ** ***Signature:***

Name of the student: **ARITRA DATTA** ***Signature:***

Name ***: JOYDEEP BANERJEE***
PROJECT MENTOR

SIGNATURE

Name:
EXAMINERS

Ardent Original Seal

Acknowledgement

I take this opportunity to express my deep gratitude and sincerest thank to my project mentor, Mr. Joydeep Banerjee for giving most valuable suggestion, helpful guidance and encouragement in the execution of this project work.

I will like to give a special mention to my colleagues. Last but not the least I am grateful to all the faculty members of Ardent Computech Pvt. Ltd. for their support.

Table of Contents

1. Abstract

The project Automation of Travel Agency helps all the citizens to book a cab easily through Internet at any time. The booking mechanism is fairly easy as the system becomes computerized.

2. Introduction and Objectives of the Project

1.1 Relevance of the Topic

The project is very much relevant in today's world. Customers can very easily book a cab/ride through the web portal.

1.2 Problem Definition

The project Automation of Travel Agencies surely going to make booking a cab/ride a lot easier. Users can now book a cab from the comforts of their home.

1.3 Project Objective

- To provide online booking facilities
- Support services in an integrated fashion so as to provide a simple and hassle free booking platform.

The system interacts to the users (customer, admin) by means of some front-end **form**. To enter the system a **sign in form** will appear where **username** and **Password** for both Admins and for the Customers will have to be entered to log into the system and then this will check the **Type** of that user. According to type the System is loaded.

After sign in customers enters into his profile. One can check the fare structure and book a ride.

After sign in admin can enter into his profile. Admin can view the Vehicle list, Driver List and Route List. Admin can also update/modify/add the details of Vehicle, Driver and Route.

New users must have to register before sign in.

Every stage data is stored in the database. This is why the system is very easy to use and every user can accept the system by thinking its utility in present and by applying the idea of Spiral Model the system can be developed with more features in future.

3. Project Category

Web Application

4. Tools/Platform, Hardware and Software Requirement specifications.

Tools

1. Adobe Dreamweaver CS6
2. WAMP Package
3. Ms-Office
4. Dia tools

Platform

1. Microsoft Windows 7/8

Hardware Requirement Specification

Client Machine		Server Machine	
HDD	200 MB	**HDD**	320 GB
Processor	Pentium 4 or newer processor that supports SSE2	**Processor**	Dual Core or newer processor
Memory	512 MB	**Memory**	2 GB

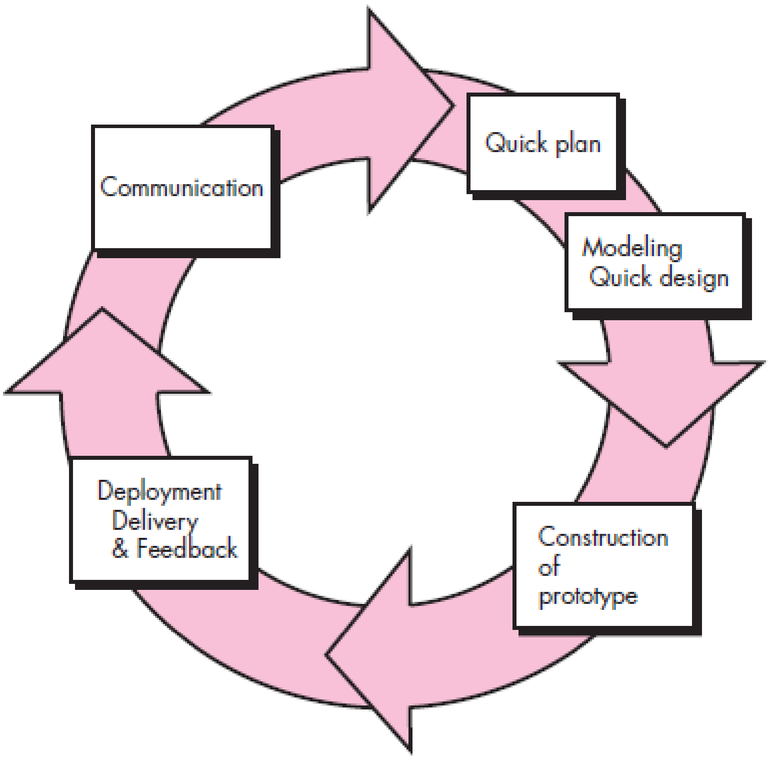
Software Requirement Specification

Client Machine		Server Machine	
Browser	Any standard browser with Javascript interpreter	**Software**	Apache
Client side mark up / scripting languages	HTML, Javascript	**Database Management System Software**	MySQL
		Specification	MySQL 4.1

5. Goals of Implementation

The implementation aims at seamless document sharing across the institution.

6. SDLC Process Applied



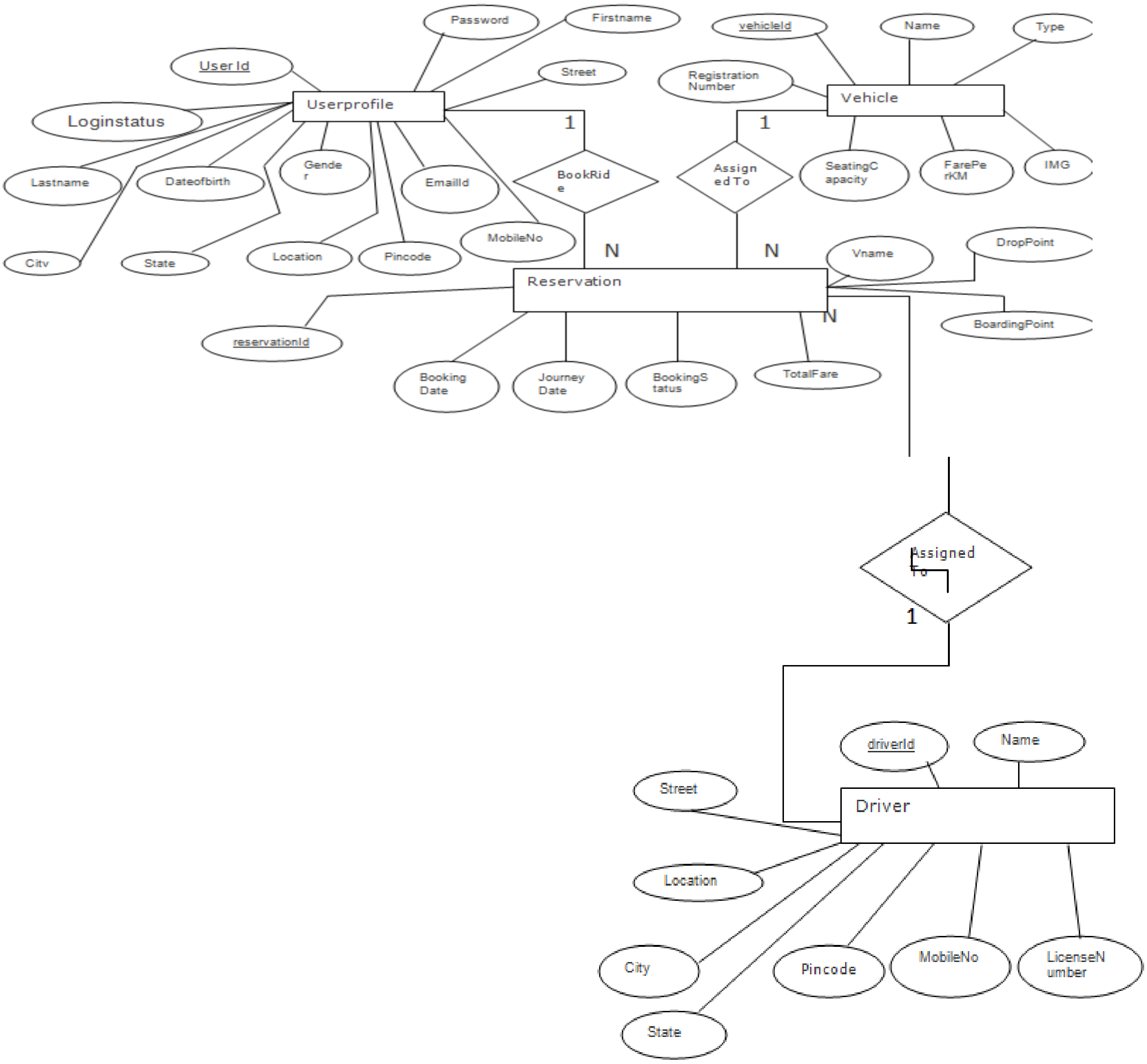
Often, a customer defines a set of general objectives for software but does not identify detailed input, processing, or output requirements. In other cases, the developer may be unsure of the efficiency of an algorithm, the adaptability of an operating system, or the form that human/machine interaction should take. In these, and many other situations, a prototyping paradigm may offer the best approach.

The prototyping paradigm begins with **requirements gathering**. Developer and customer meet and define the overall objectives for the software, identify whatever requirements are known, and outline areas where further definition is mandatory. A **"quick design"** then occurs. The quick design focuses on a representation of those aspects of the software that will be visible to the customer/user (e.g., input approaches and output formats). The quick design leads to the construction of a prototype. The prototype is evaluated by the customer/user and used to refine requirements for the software to be developed. Iteration occurs as the prototype is tuned to satisfy the needs of the customer, while at the same time enabling the developer to better understand what needs to be done.

Ideally, the prototype serves as a mechanism for identifying software requirements. If a working prototype is built, the developer attempts to use existing program fragments or applies tools (e.g., report generators, window managers) that enable working programs to be generated quickly.

7. Data Model

ER Diagram

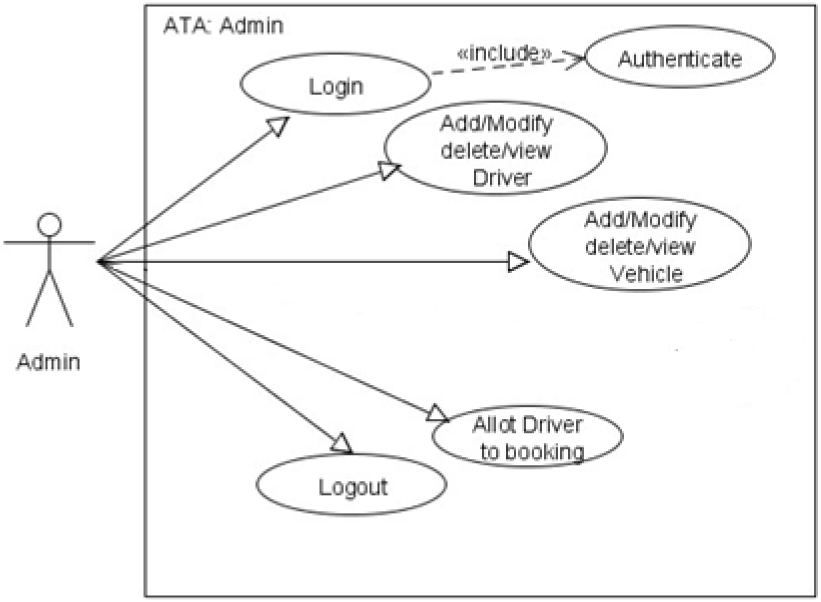


8. Functional Requirements

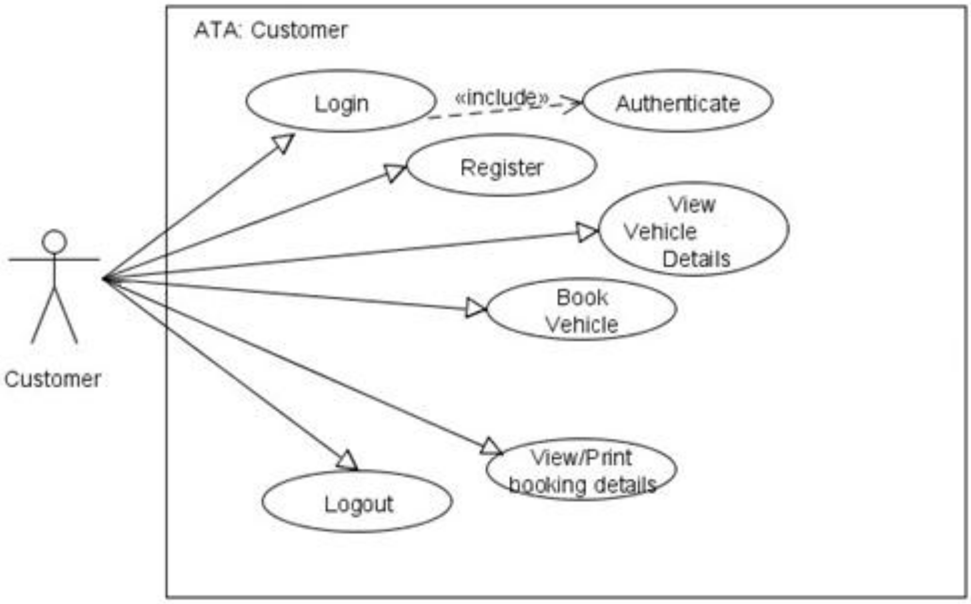
Functional Requirements are those that refer to the functionality of the system, i.e., what services it will provide to the user. Nonfunctional (supplementary) requirements pertain to other information needed to produce the correct system and are detailed separately.

Use Case Diagram:

Use Case Diagram for Admin



Use Case Diagram for Customer



Use Case Descriptions

Use Case Name:	Authentication
Priority	Essential
Trigger	Menu selection
Precondition	User is connected to the Internet and on the ATA home page
Basic Path	1. User enters username and password. 2. The username and password is matched with the record in the database. 3. If the authentication parameters are correct the user is directed to the user's main page, otherwise an error message is displayed.
Alternate Path	NA
Post Condition	The user is on the User Home Page
Exception Path	If there is a connection failure the server returns to the wait state

Use Case Name:	Submit Booking
Priority	Essential
Trigger	Menu selection
Precondition	User is connected to the Internet and on the user's main page
Basic Path	1. User selects a Car. 2. User clicks on the Booking button. 3. The server side program will give a reservation form in online.
Alternate Path	NA
Post Condition	The user uploads the document.
Exception Path	If there is a connection failure the server returns to the wait state

Use Case Name:	Browse document
Priority	Essential
Trigger	Menu selection
Precondition	User is connected to the Internet and on the user’s main page
Basic Path	1. User selects the browse document link. 2. The server side program returns the list of documents.
Alternate Path	NA
Post Condition	The user has the list of all documents.
Exception Path	If there is a connection failure the server returns to the wait state

Use Case Name:	Approve document
Priority	Essential
Trigger	Menu selection
Precondition	Admin is connected to the Internet and on the admin’s main page
Basic Path	1. The server program returns list of submitted non approved documents. 2. Admin can insert car, view booking or delete booking. 3. The end user can see all the car that admin inserted.
Alternate Path	NA
Post Condition	The admin approves a document.
Exception Path	If there is a connection failure the server returns to the wait state

9. Non Functional Requirements

In addition to the obvious features and functions that you will provide in your system, there are other requirements that don't actually DO anything, but are important characteristics nevertheless. These are called "non-functional requirements" or sometimes "Quality Attributes." For example, attributes such as performance, security, usability, compatibility.

aren't a "feature" of the system, but are a required characteristic. You can't write a specific line of code to implement them; rather they are "emergent" properties that arise from the entire solution. The specification needs to describe any such attributes the customer requires. You must decide the kind of requirements that apply to your project and include those that are appropriate.

Each requirement is simply stated in English. Each requirement must be objective and quantifiable; there must be some measurable way to assess whether the requirement has been met.

Often deciding on quality attributes requires making tradeoffs, e.g., between performance and maintainability. In the APPENDIX you must include an engineering analysis of any significant decisions regarding tradeoffs between competing attributes.

Here are some examples of non-functional requirements:

Performance requirements

Requirements about resources required, response time, transaction rates, throughput, benchmark specifications or anything else having to do with performance.

For better performance the application will restrict the document size to 5 MB.

Operating constraints

List any run-time constraints. This could include system resources, people, needed software,

The application must run without any manual intervention.

Platform constraints

Discuss the target platform. Be as specific or general as the user requires. If the user doesn't care, there are still platform constraints.

Since the application will be developed in PHP it is platform independent.

Accuracy and Precision

Requirements about the accuracy and precision of the data. (Do you know the difference?) Beware of 100% requirements; they often cost too much.

Modifiability

Requirements about the effort required to make changes in the software. Often, the measurement is personnel effort (person- months).

Minimal

Portability

The effort required to move the software to a different target platform. The measurement is most commonly person-months or % of modules that need changing.

Minimal

Reliability

Requirements about how often the software fails. The measurement is often expressed in MTBF (mean time between failures). The definition of a failure must be clear. Also, don't confuse reliability with availability which is quite a different kind of requirement. Be sure to specify the consequences of software failure, how to protect from failure, a strategy for error detection, and a strategy for correction.

Security

One or more requirements about protection of your system and its data. The measurement can be expressed in a variety of ways (effort, skill level, time, ...) to break into the system. Do not discuss solutions (e.g. passwords) in a requirements document.

Only secured users can access the application.

No one can go to any independent page without logging in.

Usability

Requirements about how difficult it will be to learn and operate the system. The requirements are often expressed in learning time or similar metrics.

Legal

There may be legal issues involving privacy of information, intellectual property rights, export of restricted technologies, etc.

10. Feasibility Study

You should provide a feasibility report in the following format:

- **Product:** A general statement of the product; give a brief description of what the proposed system will do, highlighting where the proposed system meets the specified business requirements of the organization.

- **Technical Feasibility:** Will the proposed system perform to the required specification? Outline technical systems options you propose to use, which will give a technical solution satisfying the requirements and constraints of the system, as outlined in the terms of reference.

- **Social Feasibility:** Consideration of whether the proposed system would prove acceptable to the people who would be affected by its introduction. Describe the effect on users from the introduction of the new system; consider whether there will be a need for retraining the workforce. Will there be a need for relocation of some of the workforce? Will some jobs become deskilled? Will the current workforce be able to perform effectively any new tasks introduced by the proposed system? Describe how you propose to ensure user co-operation before changes are introduced.

- **Economic Feasibility:** Consider the cost/benefits of the proposed system. Detail the costs that will be incurred by the organization adopting the new system; consider development costs and running costs. Detail benefits that the new system will bring, direct economic benefits such as reduced costs, and indirect benefits, such as improved management information and better customer service. Illustrate the cost/benefit of the new system by applying a suitable cost/benefit analysis method such as the payback method.

- **Market Research:** A comprehensive market research identifying a need for the product. Detail all market research you carried out, listing sources of information. Justify any conclusions you have drawn from your research. Identify the potential customer base for your product, together with evidence of customer need for the product. Describe how you propose to compete with similar products on the market.

- **Alternative Solution:** Consideration of alternative solutions should be documented. At least two alternative business or technical systems options should be considered. Detail the differences between these options and the proposed system. Justify your choice of the proposed system and the reasons for rejecting the alternative options.

At this point, all of the planning for the project has been done and if the feasibility study has shown that the project is likely to succeed within its constraints, then it only remains for us to start the requirements analysis and thus proceed with the project.

Feasibility Study	
System: ATA	Date: 02/01/2018
Author: ALAMGIR ALAM	Page: 1
Product	
The project requires a web application to be developed that will allow online Booking.	
Technical Feasibility	
The web application will be developed using PHP and MySQL. The team is competent in that.	
Social Feasibility	
Some training for the users/admin are required but all users are IT literate.	
Market Research	
Market research says that this application would be useful for the users as it could seamlessly help them to share documents.	
Economic Feasibility	
The application can be developed within budget.	
Alternate Solution	
Could be a desktop system but that would not allow documents to be shared online.	

11. Project Planning

Project planning is concerned with identifying the following for every project:

- Activities
- Milestones
- Deliverables.

A plan must be drawn up to guide the development towards the project goal. A plan is drawn up at the start of a project. This plan should be used as the driver for the project. The initial plan is not static, and must be modified as the project progresses.

Planning is required for development activities from specification through to delivery of the system.

12. Project Scheduling

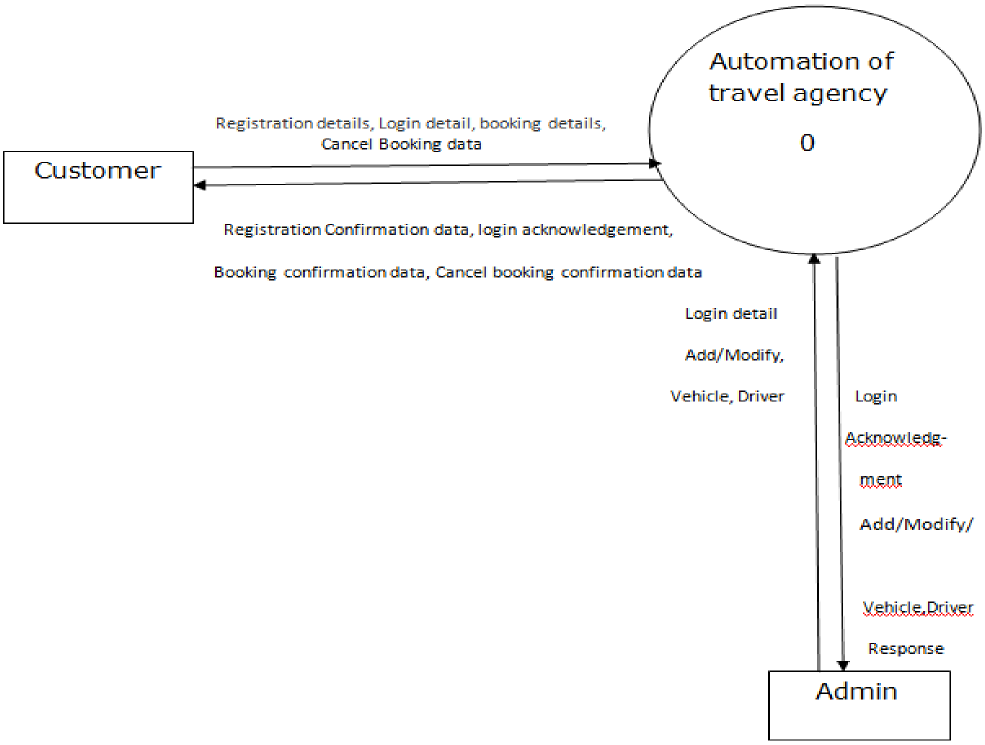
GANTT chart

Task	Person(s) Responsible	Week 1	Week 2	Week 3	Week 4
Communication		▨			
Quick Plan		▨			
Modeling Quick Design			▨		
Construction of Prototype			▨	▨	▨
Deployment, Delivery and Feedback					▨

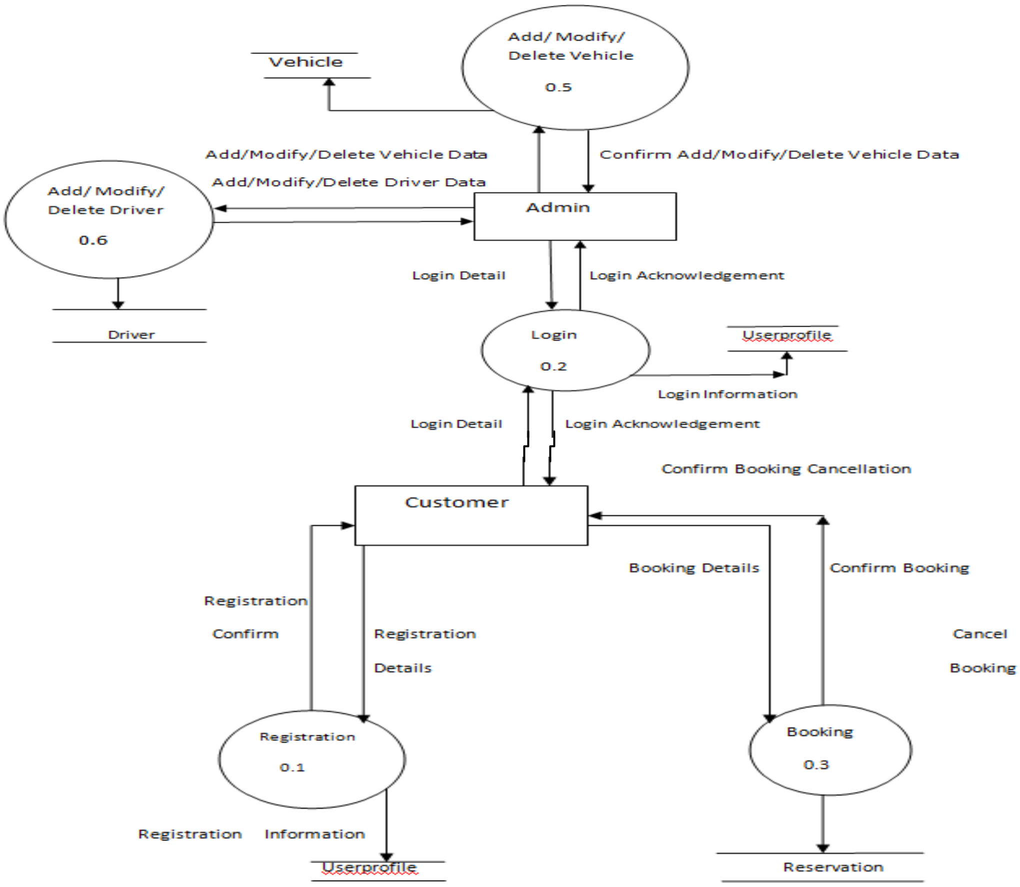
13. Software Engineering Paradigm Applied

Data Flow Diagrams

Level 0


Automation of travel agency
0
Customer
Registration details, Login detail, booking details, Cancel Booking data
Registration Confirmation data, login acknowledgement,
Booking confirmation data, Cancel booking confirmation data
Login detail
Add/Modify,
Vehicle, Driver
Login
Acknowledg-
ment
Add/Modify/
Vehicle,Driver
Response
Admin

Level 1



14. Schema/Database Design

1. Table: USERPROFILE

This table contains User specific details entered during User Registration.

Field Name	Data Type	Description
userid*	INT	Primary Key
firstname	TEXT	Not Null
lastname	TEXT	Not Null
email	TEXT	Not Null
gender	TEXT	Not Null
phone	TEXT	Not Null
address	TEXT	Not Null

* First 2 letters of First Name followed by 4 digits auto generated number

2. Table: VEHICLE

This table contains Vehicle specific information.

Field Name	Data Type	Description
vehicleId*	INT	Primary Key
Name	TEXT	Not Null
Type	TEXT	Not Null

RegistrationNumber	TEXT	Not Null
SeatingCapacity	TEXT	Not Null
FarePerKM	TEXT	Not Null

* Id should be First 2 letters of VehicleName followed by 4 digits auto generated number

3. Table: DRIVER

This table contains Driver details like Name, address & License number etc.

Field Name	Data Type	Description
driverId	INT	Primary Key
Name	TEXT	Not Null
Street	TEXT	Not Null
Location	TEXT	Not Null
City	TEXT	Not Null
State	TEXT	Not Null
Pincode	TEXT	Not Null
MobileNo	TEXT	Not Null
LicenseNo	TEXT	Not Null

4. Table: ROUTE

This table contains Route details like source, destination & distance etc.

Field Name	Data Type	Description
RouteId	INT	Primary Key
Source	TEXT	Not Null
Destination	TEXT	Not Null
Distance	TEXT	Not Null

5. Table: CONTACT

This table contains Contact details like name, email & message etc.

Field Name	Data Type	Description
ConatctId	INT	Primary Key
Name	TEXT	Not Null
Email	TEXT	Not Null
Message	TEXT	Not Null

6. Table: ADMIN

This table contains Admin details like email & password etc.

Field Name	Data Type	Description
AdminId	INT	Primary Key
Email	TEXT	Not Null
Password	TEXT	Not Null

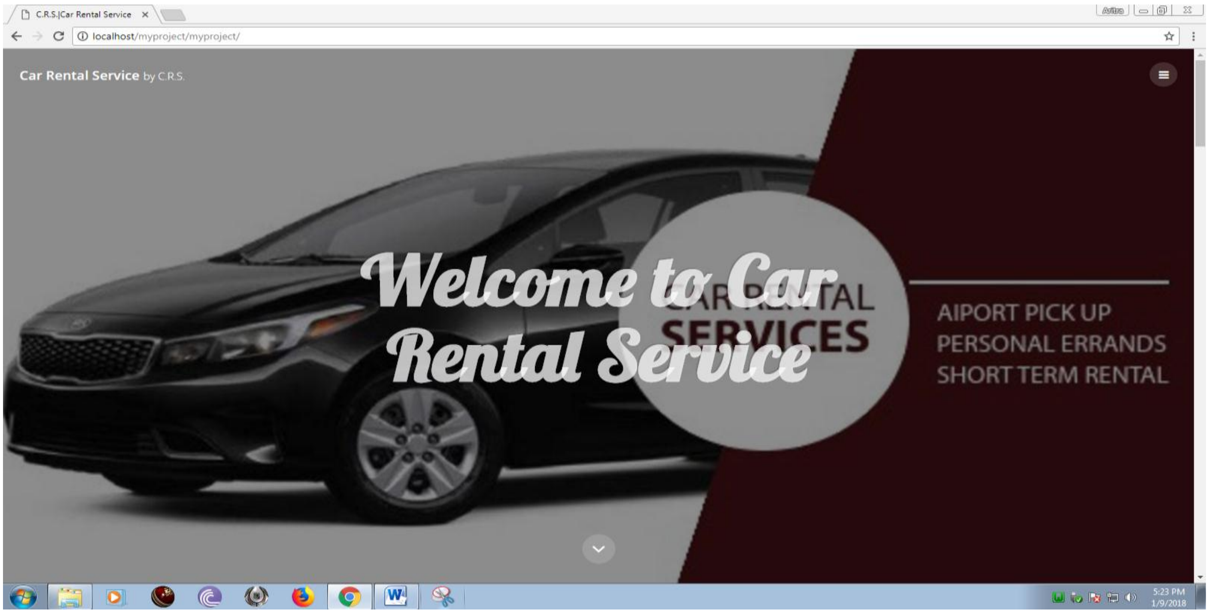
7. Table: BOOKING

This table contains Booking details like BookingId, VehicleId, DriverId, Totalpayment, RouteId, Status etc.

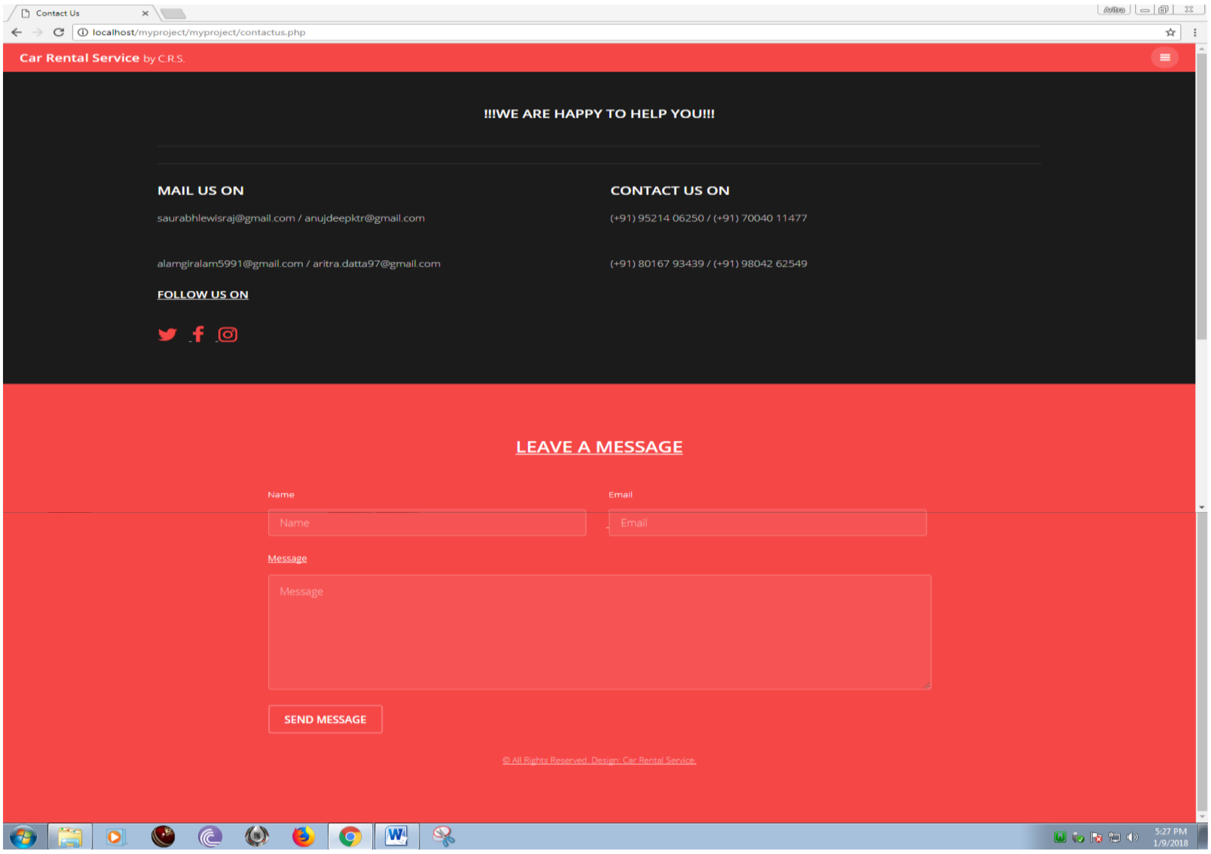
Field Name	Data Type	Description
BookingId	INT	Primary Key
VehicleId	INT	Not Null
DriverId	INT	Not Null
Totalpayment	TEXT	Not Null
RouteId	INT	Not Null
Status	TEXT	Not Null

15. User Interface Design

HOME :


C.R.S.|Car Rental Service
localhost/myproject/myproject/
Car Rental Service by C.R.S.
Welcome to Car Rental Service
CAR RENTAL SERVICES
AIPORT PICK UP
PERSONAL ERRANDS
SHORT TERM RENTAL
5:23 PM
1/9/2018

CONTACT US:


Contact Us
localhost/myproject/myproject/contactus.php
Car Rental Service by C.R.S.

!!!WE ARE HAPPY TO HELP YOU!!!

MAIL US ON

saurabhlewisraj@gmail.com / anujdeepktr@gmail.com

alamgiralam5991@gmail.com / aritra.datta97@gmail.com

CONTACT US ON

(+91) 95214 06250 / (+91) 70040 11477

(+91) 80167 93439 / (+91) 98042 62549

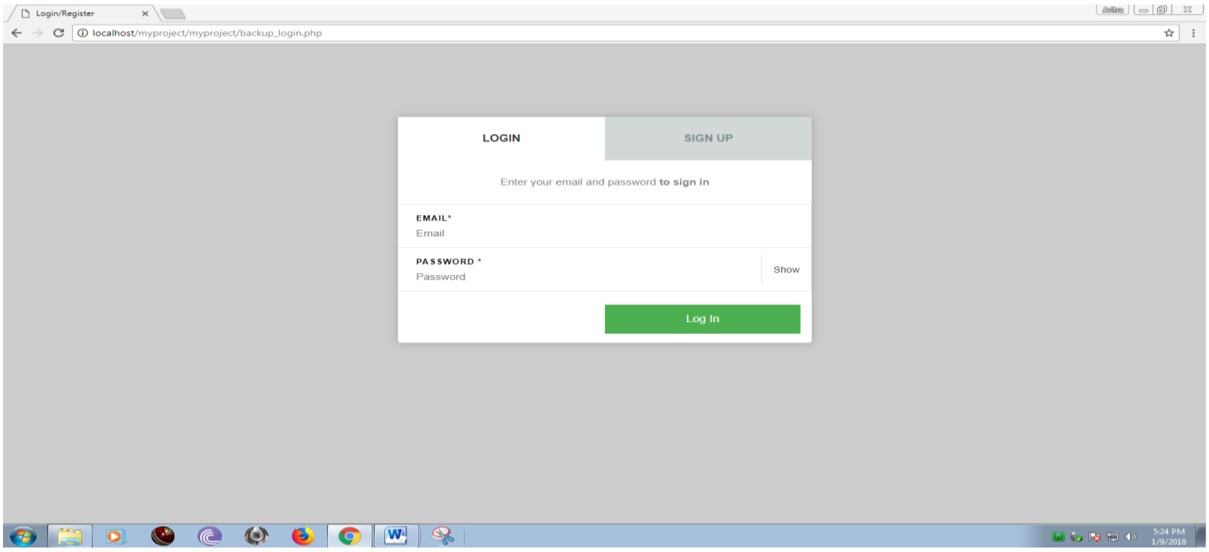
FOLLOW US ON

LEAVE A MESSAGE

Name
Name
Email
Email
Message
Message
SEND MESSAGE

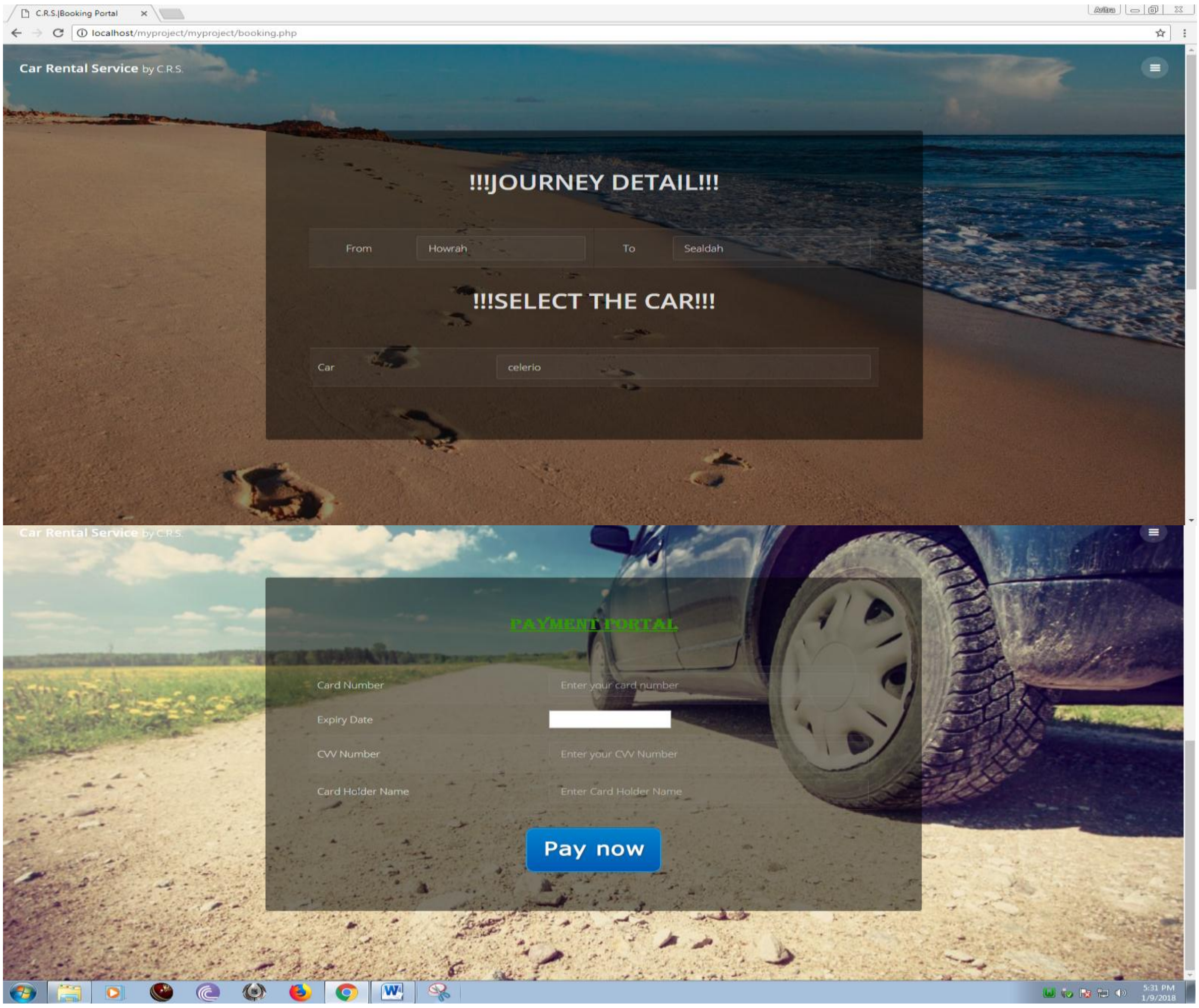
© All Rights Reserved. Design: Car Rental Service.

ONLINE REGISTRATION & LOGIN:



Enter your email and password **to sign in**

BOOKING SECTION :



C.R.S.|Booking Portal

localhost/myproject/myproject/booking.php

Car Rental Service by C.R.S.

!!!JOURNEY DETAIL!!!

From: Howrah

To: Sealdah

!!!SELECT THE CAR!!!

Car: celerio

Car Rental Service by C.R.S.

PAYMENT PORTAL

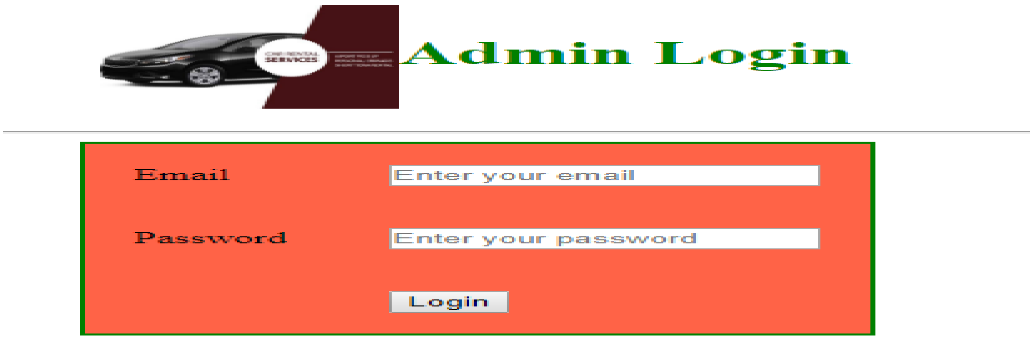
Card Number: Enter your card number

Expiry Date:

CVV Number: Enter your CVV Number

Card Holder Name: Enter Card Holder Name

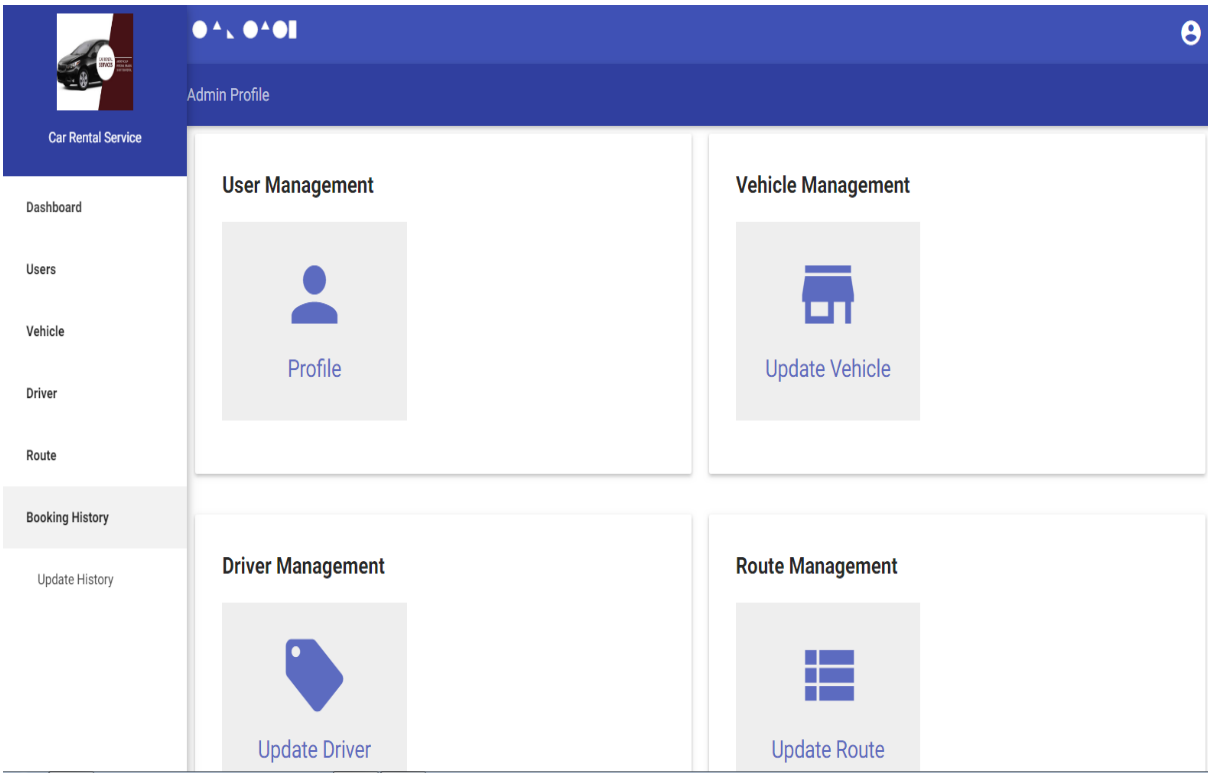
ADMIN:



Admin Login

Email: Enter your email

Password: Enter your password



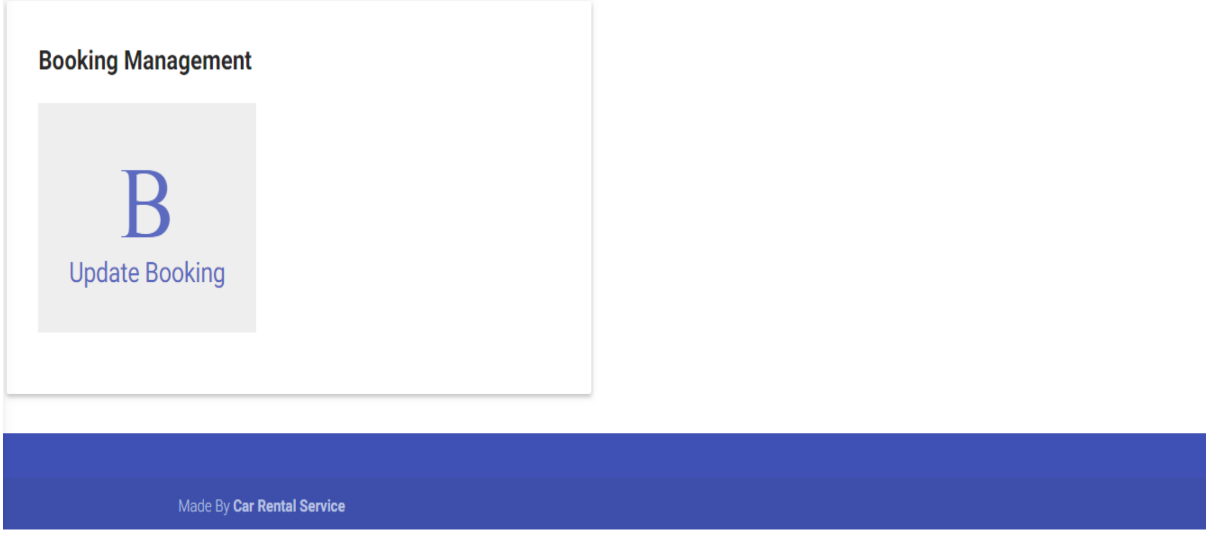
Admin Profile

User Management

Vehicle Management

Driver Management

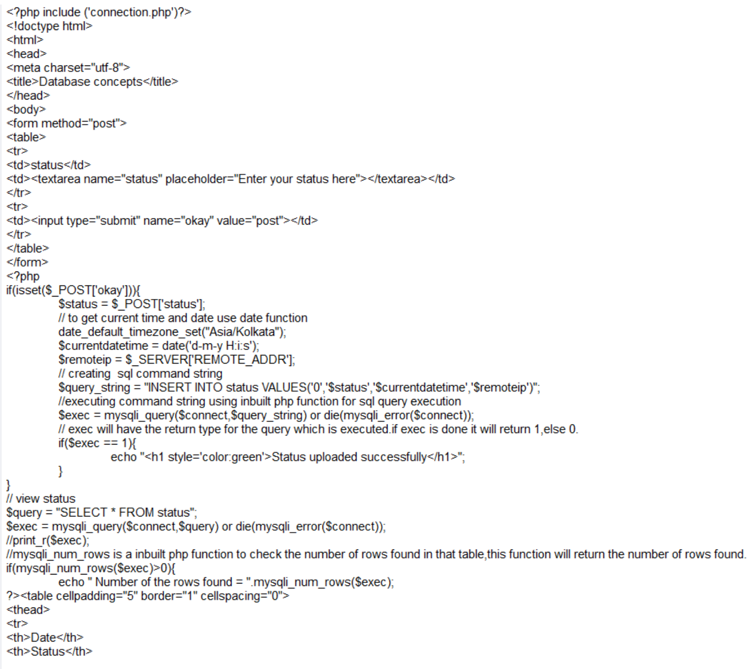
Route Management



Booking Management

Made By **Car Rental Service**

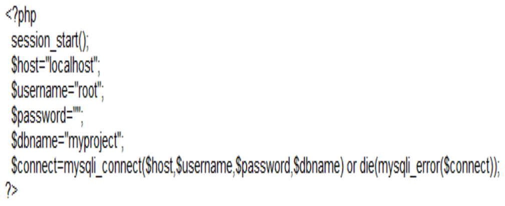
16. *Coding*



```
<?php include ('connection.php')?>
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Database concepts</title>
</head>
<body>
<form method="post">
<table>
<tr>
<td>status</td>
<td><textarea name="status" placeholder="Enter your status here"></textarea></td>
</tr>
<tr>
<td><input type="submit" name="okay" value="post"></td>
</tr>
</table>
</form>
<?php
if(isset($_POST['okay'])){
		$status = $_POST['status'];
		// to get current time and date use date function
		date_default_timezone_set("Asia/Kolkata");
		$currentdatetime = date('d-m-y H:i:s');
		$remoteip = $_SERVER['REMOTE_ADDR'];
		// creating  sql command string
		$query_string = "INSERT INTO status VALUES('0','$status','$currentdatetime','$remoteip')";
		//executing command string using inbuilt php function for sql query execution
		$exec = mysqli_query($connect,$query_string) or die(mysqli_error($connect));
		// exec will have the return type for the query which is executed.if exec is done it will return 1,else 0.
		if($exec == 1){
				echo "<h1 style='color:green'>Status uploaded successfully</h1>";
		}
}
// view status
$query = "SELECT * FROM status";
$exec = mysqli_query($connect,$query) or die(mysqli_error($connect));
//print_r($exec);
//mysqli_num_rows is a inbuilt php function to check the number of rows found in that table,this function will return the number of rows found.
if(mysqli_num_rows($exec)>0){
		echo " Number of the rows found = ".mysqli_num_rows($exec);
?><table cellpadding="5" border="1" cellspacing="0">
<thead>
<tr>
<th>Date</th>
<th>Status</th>
```

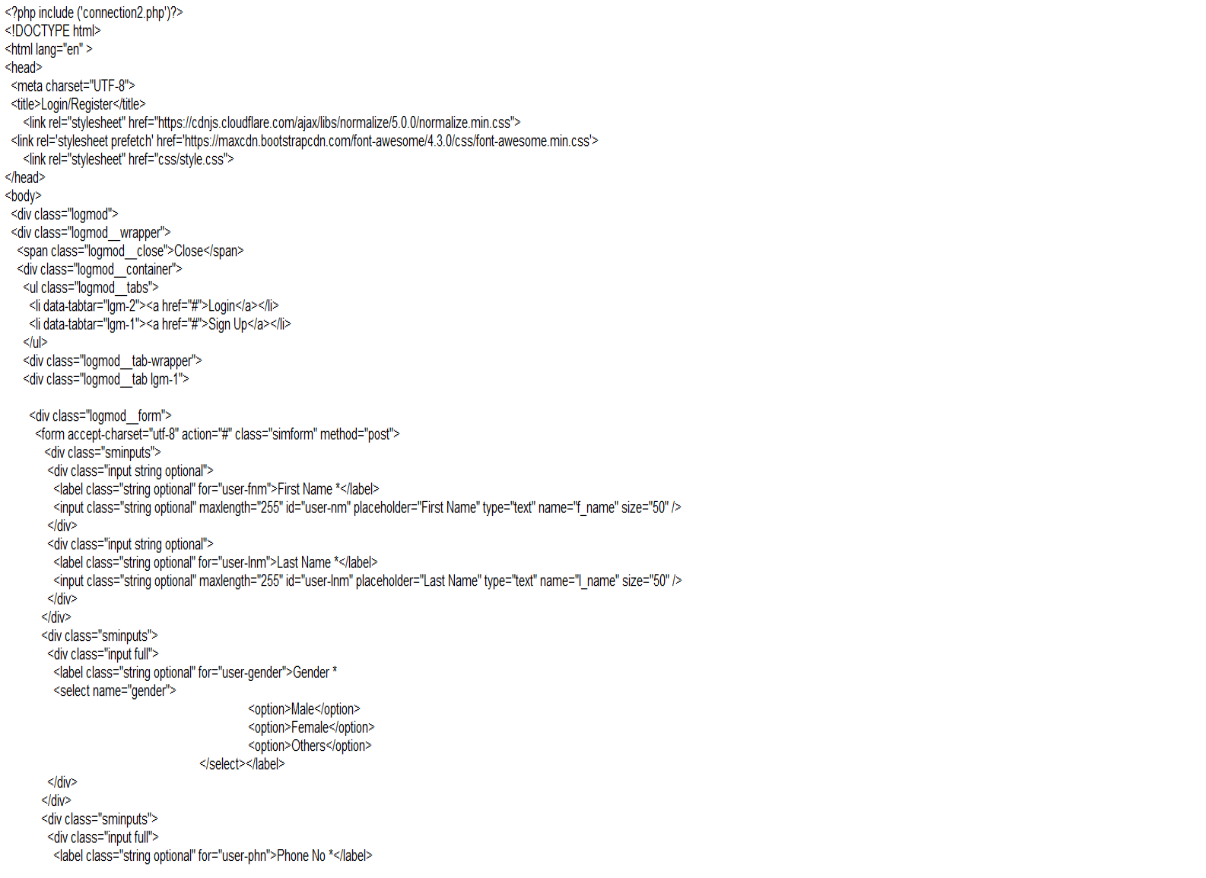
```
</tr>
</thead>
<tbody>
<?php while($each_status = mysqli_fetch_array($exec)){?>
<tr>
<td><?php echo $each_status['date']?></td>
<td><?php echo $each_status['status']?></td>
</tr>
<?php } ?>
</tbody>
</table><?php
}else{
		echo "No status found";
}
?>
</body>
</html>
```

Connection Code

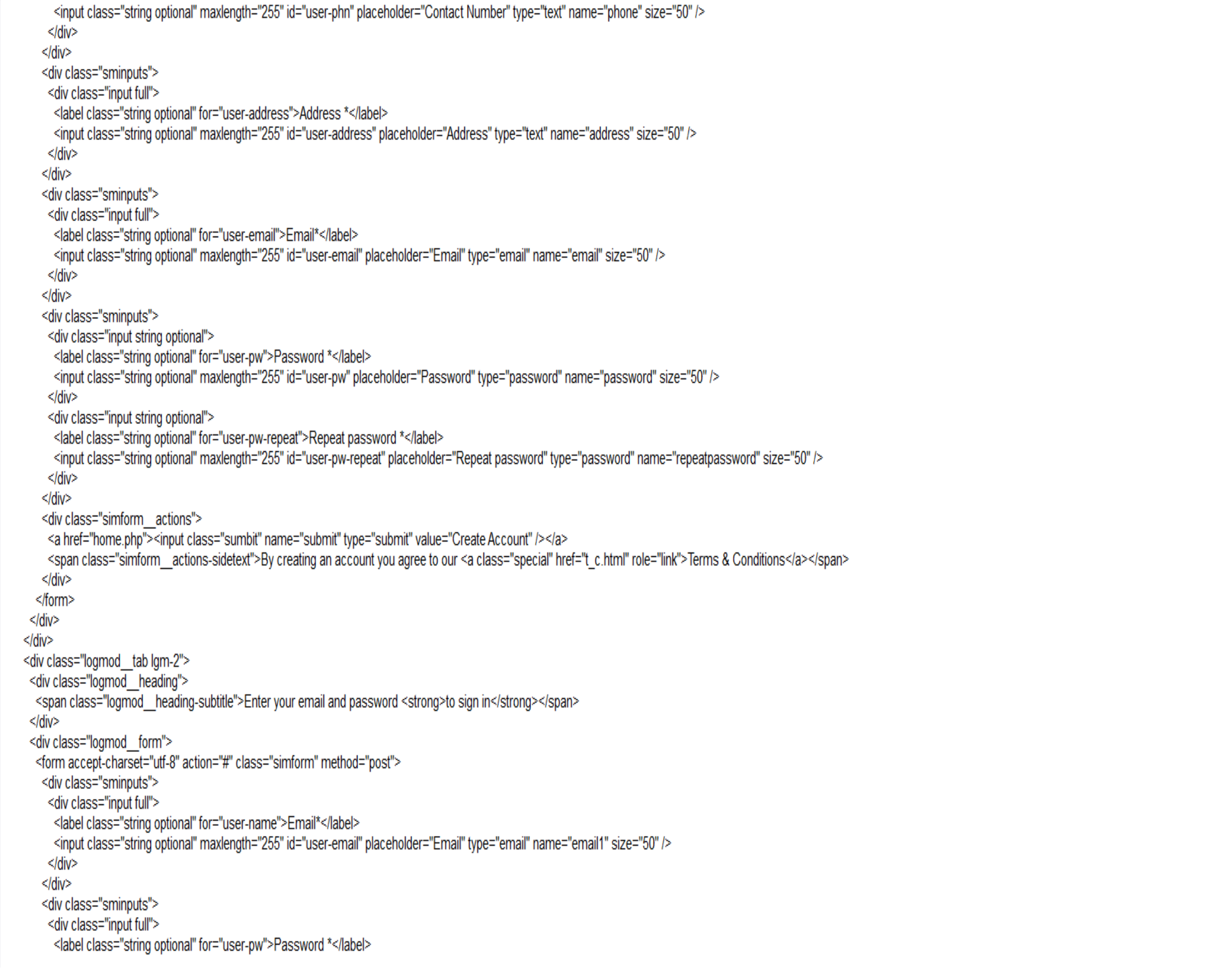


```
<?php
 session_start();
 $host="localhost";
 $username="root";
 $password="";
 $dbname="myproject";
 $connect=mysqli_connect($host,$username,$password,$dbname) or die(mysqli_error($connect));
?>
```

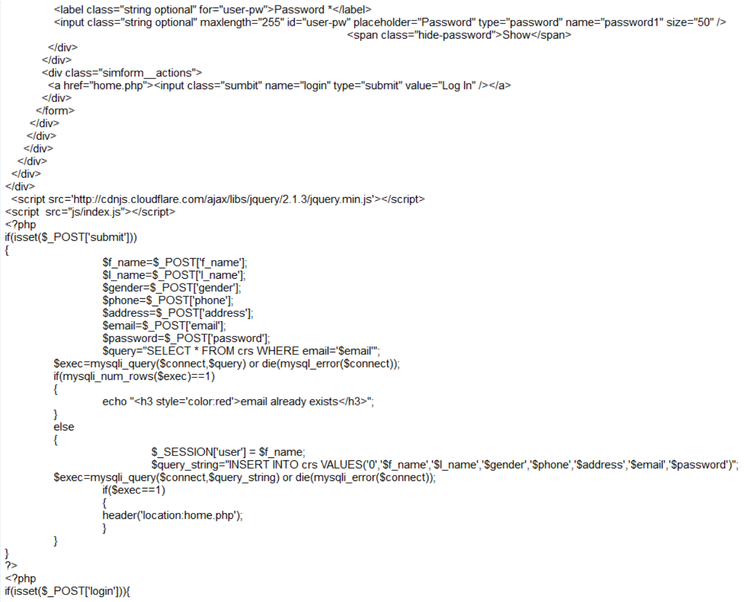
FOR ADMIN



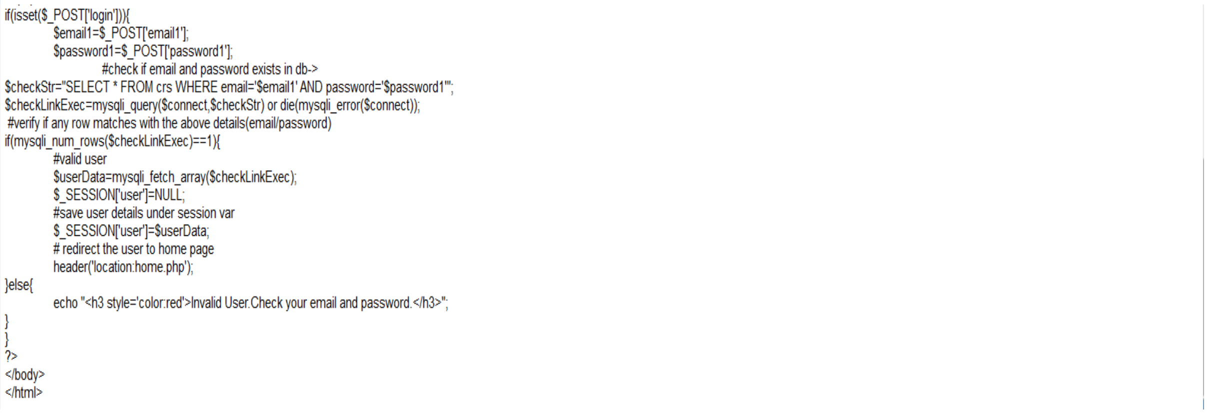
```
<?php include ('connection2.php')?>
<!DOCTYPE html>
<html lang="en" >
<head>
  <meta charset="UTF-8">
  <title>Login/Register</title>
      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/normalize/5.0.0/normalize.min.css">
  <link rel='stylesheet prefetch' href='https://maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css'>
      <link rel="stylesheet" href="css/style.css">
</head>
<body>
  <div class="logmod">
  <div class="logmod__wrapper">
    <span class="logmod__close">Close</span>
    <div class="logmod__container">
      <ul class="logmod__tabs">
        <li data-tabtar="lgm-2"><a href="#">Login</a></li>
        <li data-tabtar="lgm-1"><a href="#">Sign Up</a></li>
      </ul>
      <div class="logmod__tab-wrapper">
      <div class="logmod__tab lgm-1">

        <div class="logmod__form">
          <form accept-charset="utf-8" action="#" class="simform" method="post">
             <div class="sminputs">
              <div class="input string optional">
                <label class="string optional" for="user-fnm">First Name *</label>
                <input class="string optional" maxlength="255" id="user-nm" placeholder="First Name" type="text" name="f_name" size="50" />
              </div>
              <div class="input string optional">
                <label class="string optional" for="user-lnm">Last Name *</label>
                <input class="string optional" maxlength="255" id="user-lnm" placeholder="Last Name" type="text" name="l_name" size="50" />
              </div>
            </div>
            <div class="sminputs">
              <div class="input full">
                <label class="string optional" for="user-gender">Gender *
                <select name="gender">
                                                                    <option>Male</option>
                                                                    <option>Female</option>
                                                                    <option>Others</option>
                                                        </select></label>
              </div>
            </div>
            <div class="sminputs">
              <div class="input full">
                <label class="string optional" for="user-phn">Phone No *</label>
```



```
                <input class="string optional" maxlength="255" id="user-phn" placeholder="Contact Number" type="text" name="phone" size="50" />
              </div>
            </div>
            <div class="sminputs">
              <div class="input full">
                <label class="string optional" for="user-address">Address *</label>
                <input class="string optional" maxlength="255" id="user-address" placeholder="Address" type="text" name="address" size="50" />
              </div>
            </div>
            <div class="sminputs">
              <div class="input full">
                <label class="string optional" for="user-email">Email*</label>
                <input class="string optional" maxlength="255" id="user-email" placeholder="Email" type="email" name="email" size="50" />
              </div>
            </div>
            <div class="sminputs">
              <div class="input string optional">
                <label class="string optional" for="user-pw">Password *</label>
                <input class="string optional" maxlength="255" id="user-pw" placeholder="Password" type="password" name="password" size="50" />
              </div>
              <div class="input string optional">
                <label class="string optional" for="user-pw-repeat">Repeat password *</label>
                <input class="string optional" maxlength="255" id="user-pw-repeat" placeholder="Repeat password" type="password" name="repeatpassword" size="50" />
              </div>
            </div>
            <div class="simform__actions">
              <a href="home.php"><input class="sumbit" name="submit" type="submit" value="Create Account" /></a>
              <span class="simform__actions-sidetext">By creating an account you agree to our <a class="special" href="t_c.html" role="link">Terms & Conditions</a></span>
            </div>
          </form>
        </div>
      </div>
      <div class="logmod__tab lgm-2">
        <div class="logmod__heading">
          <span class="logmod__heading-subtitle">Enter your email and password <strong>to sign in</strong></span>
        </div>
        <div class="logmod__form">
          <form accept-charset="utf-8" action="#" class="simform" method="post">
            <div class="sminputs">
              <div class="input full">
                <label class="string optional" for="user-name">Email*</label>
                <input class="string optional" maxlength="255" id="user-email" placeholder="Email" type="email" name="email1" size="50" />
              </div>
            </div>
            <div class="sminputs">
              <div class="input full">
                <label class="string optional" for="user-pw">Password *</label>
```



```
            <label class="string optional" for="user-pw">Password *</label>
            <input class="string optional" maxlength="255" id="user-pw" placeholder="Password" type="password" name="password1" size="50" />
                                                              <span class="hide-password">Show</span>
          </div>
        </div>
        <div class="simform__actions">
          <a href="home.php"><input class="sumbit" name="login" type="submit" value="Log In" /></a>
        </div>
      </form>
     </div>
    </div>
   </div>
  </div>
 </div>
</div>
 <script src='http://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.min.js'></script>
<script  src="js/index.js"></script>
<?php
if(isset($_POST['submit']))
{
                        $f_name=$_POST['f_name'];
                        $l_name=$_POST['l_name'];
                        $gender=$_POST['gender'];
                        $phone=$_POST['phone'];
                        $address=$_POST['address'];
                        $email=$_POST['email'];
                        $password=$_POST['password'];
                        $query="SELECT * FROM crs WHERE email='$email'";
            $exec=mysqli_query($connect,$query) or die(mysql_error($connect));
            if(mysqli_num_rows($exec)==1)
            {
                        echo "<h3 style='color:red'>email already exists</h3>";
            }
            else
            {
                                    $_SESSION['user'] = $f_name;
                                    $query_string="INSERT INTO crs VALUES('0','$f_name','$l_name','$gender','$phone','$address','$email','$password')";
            $exec=mysqli_query($connect,$query_string) or die(mysqli_error($connect));
                        if($exec==1)
                        {
                        header('location:home.php');
                        }
            }
}
?>
<?php
if(isset($_POST['login'])){
```



```
if(isset($_POST['login'])){
            $email1=$_POST['email1'];
            $password1=$_POST['password1'];
                        #check if email and password exists in db->
$checkStr="SELECT * FROM crs WHERE email='$email1' AND password='$password1'";
$checkLinkExec=mysqli_query($connect,$checkStr) or die(mysqli_error($connect));
 #verify if any row matches with the above details(email/password)
if(mysqli_num_rows($checkLinkExec)==1){
            #valid user
            $userData=mysqli_fetch_array($checkLinkExec);
            $_SESSION['user']=NULL;
            #save user details under session var
            $_SESSION['user']=$userData;
            # redirect the user to home page
            header('location:home.php');
}else{
            echo "<h3 style='color:red'>Invalid User.Check your email and password.</h3>";
}
}
?>
</body>
</html>
```

17. Testing

Team Interaction

The following describes the level of team interaction necessary to have a successful product.

- The Test Team will work closely with the Development Team to achieve a high quality design and user interface specifications based on customer requirements. The Test Team is responsible for visualizing test cases and raising quality issues and concerns during meetings to address issues early enough in the development cycle.

- The Test Team will work closely with Development Team to determine whether or not the application meets standards for completeness. If an area is not acceptable for testing, the code complete date will be pushed out, giving the developers additional time to stabilize the area.

- Since the application interacts with a back-end system component, the Test Team will need to include a plan for integration testing. Integration testing must be executed successfully prior to system testing.

Test Objective

The objective our test plan is to find and report as many bugs as possible to improve the integrity of our program. Although exhaustive testing is not possible, we will exercise a broad range of tests to achieve our goal. We will be testing a Binary Search Tree Application utilizing a pre-order traversal format. There will be eight key functions used to manage our application: load, store, clear, search, insert, delete, list in ascending order, and list in descending order. Our user interface to utilize these functions is designed to be user-friendly and provide easy manipulation of the tree. The application will only be used as a demonstration tool, but we would like to ensure that it could be run from a variety of platforms with little impact on performance or usability.

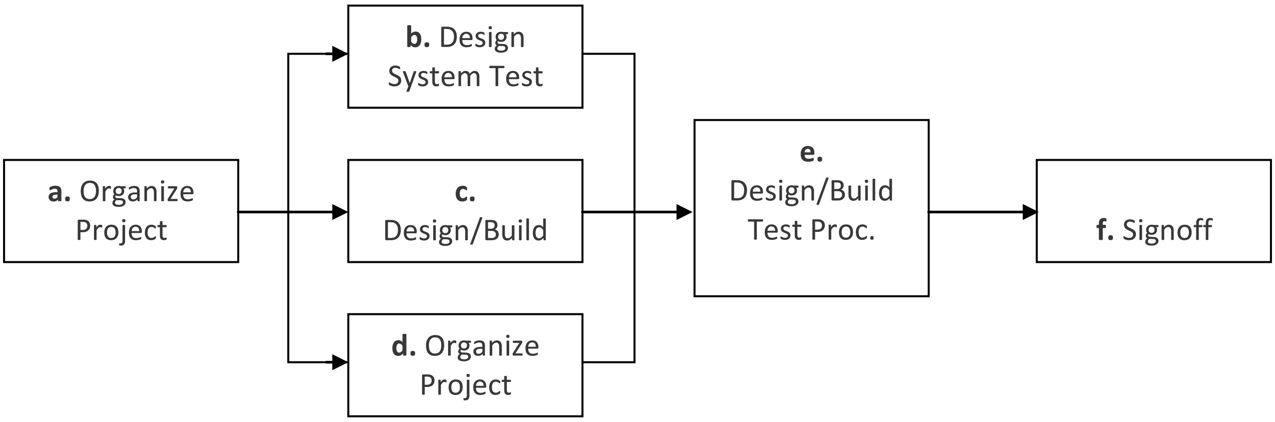
Process Overview

The following represents the overall flow of the testing process:

1. Identify the requirements to be tested. All test cases shall be derived using the current Program Specification.
2. Identify which particular test(s) will be used to test each module.
3. Review the test data and test cases to ensure that the unit has been thoroughly verified and that the test data and test cases are adequate to verify proper operation of the unit.
4. Identify the expected results for each test.
5. Document the test case configuration, test data, and expected results.
6. Perform the test(s).

7. Document the test data, test cases, and test configuration used during the testing process. This information shall be submitted via the Unit/System Test Report (STR).
8. Successful unit testing is required before the unit is eligible for component integration/system testing.
9. Unsuccessful testing requires a Bug Report Form to be generated. This document shall describe the test case, the problem encountered, its possible cause, and the sequence of events that led to the problem. It shall be used as a basis for later technical analysis.
10. Test documents and reports shall be submitted. Any specifications to be reviewed, revised, or updated shall be handled immediately.

Testing Process



The diagram above outlines the Test Process approach that will be followed.

a. **Organize Project** involves creating a System Test Plan, Schedule & Test Approach, and assigning responsibilities.

b. **Design/Build System Test** involves identifying Test Cycles, Test Cases, Entrance & Exit Criteria, Expected Results, etc. In general, test conditions/expected results will be identified by the Test Team in conjunction with the Development Team. The Test Team will then identify Test Cases and the Data required. The Test conditions are derived from the Program Specifications Document.

c. **Design/Build Test Procedures** includes setting up procedures such as Error Management systems and Status reporting.

d. **Build Test Environment** includes requesting/building hardware, software and data set-ups.

e. **Execute System Tests –** The tests identified in the Design/Build Test Procedures will be executed. All results will be documented and Bug Report Forms filled out and given to the Development Team as necessary.

f. **Signoff** - Signoff happens when all pre-defined exit criteria have been achieved.

Testing Strategy

The following outlines the types of testing that will be done for unit, integration, and system testing. While it includes what will be tested, the specific use cases that determine how the testing is done will be detailed in the Test Design Document. The test cases that will be used for designing use cases is shown in Figure 2.1 and onwards.

Tested By:	SAURABH RAJ & ALAMGIR ALAM
Test Type	Unit Testing
Test Case Number	1
Test Case Name	User Identification
Test Case Description	The user should enter his/ her accurate userid and password so that he/she can able to go for the further options. The test case will check the application for the same since a user can only login with the correct userid , password.
Item(s) to be tested	
1	Verification of the userid and password with the record in the database.
Specifications	
Input	**Expected Output/Result**

1) Correct User id and password 2) Incorrect Id or Password	1) Successful login 2) Failure Message

Tested By:	ANUJ DEEP & ARITRA DATTA	
Test Type	Unit Testing	
Test Case Number	2	
Test Case Name	Submit Document	
Test Case Description	The User submit document. For document submission the must login.	
Item(s) to be tested		
1	Check whether the user id logged in.	
2	Check if the user has selected a document and the size of that is < 5 MB.	
Specifications		
Input		**Expected Output/Result**
1) Trying to submit document without logging in. 2) Check whether document is selected. 3) Check whether the size of the document is < 5 MB.		1) The user is redirected to the login page. 2) A message asks the user to select a document. 3) Return an error message.

Test Cases

Unit Testing

Unit Testing is done at the source or code level for language-specific programming errors such as bad syntax, logic errors, or to test particular functions or code modules. The unit test cases shall be designed to test the validity of the programs correctness.

White Box Testing

In white box testing, the UI is bypassed. Inputs and outputs are tested directly at the code level and the results are compared against specifications. This form of testing ignores the function of the program under test and will focus only on its code and the structure of that code. Test case designers shall generate cases that not only cause each condition to take on all possible values at least once, but that cause each such condition to be executed at least once. To ensure this happens, we will be applying Branch Testing. Because the functionality of the program is relatively simple, this method will be feasible to apply.

Each function of the binary tree repository is executed independently; therefore, a program flow for each function has been derived from the code.

Black Box Testing

Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would. We have decided to perform Equivalence Partitioning and Boundary Value Analysis testing on our application.

System Testing

The goals of system testing are to detect faults that can only be exposed by testing the entire integrated system or some major part of it. Generally, system testing is mainly concerned with areas such as performance, security, validation, load/stress, and configuration sensitivity. But in our case well focus only on function validation and performance. And in both cases we will use the black-box method of testing.

18. System Security measures (Implementation of security for the project developed)

- Only authorized users are allowed.
- Without signing in users are not allowed to go an intermediate page by typing an URL. For all such efforts, users will be redirected to the home page.

19. Database/Data security

- Database is present in remote machine.
- Oracle’s default securities are applied.

20. Creation of User profiles and access rights

- The admin must create users manually
- The admin can create more admins

21. Cost Estimation of the Project along with Cost Estimation Model

Analogous estimate of effort or cost

Used for Early Estimate or Individual Activity Estimate

Sample example shown below is for two major deliverables of a software project. You use a previous project as a benchmark for analogous estimation. Using your experience you will estimate a multiplier.

Multipliers:

1. Prototyping: 0.75.
2. Testing: 0.5
3. Deployment: 0.5

Finally, if you want to convert to cost, you would use current rates for the resource.

WBS ID	Previous Similar Project Activity	Previous Effort	Current Project Estimate	Multiplier	Effort (Previous Effort * 0.75)	Cost (Rs. 500/hr.)
1	Prototyping	40 Work-Hours	Prototyping	0.75	30 Work-hours	Rs. 15000/-
2	Testing	20 Work-Hours	Testing	0.50	10 Work-Hours	Rs. 5000/-
Total					**40 Work-Hours**	**Rs. 20000/-**

Note: Effort is also called Size and unit of estimation is called either Work-Hour, person-hours.

22. Future scope and further enhancement of the Project

ATA has lot of enhancement options. In future documents may be classified category-wise. It may try to analyze the user behavior and preferences and accordingly suggest document categories. AI concepts can be applied to make ATA intelligent.

23. Bibliography

1. Roger S. Pressman. Software Engineering: A Practioner's Approach (Sixth Edition, International Edition). McGraw-Hill, 2005.
2. Ian Sommerville. Software Engineering (Seventh Edition). Addison-Wesley, 2004.
3. Frederick P. Brooks. The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition. Addison-Wesley Pub Co; 1st edition (August 2, 1995).
4. PHP and MySQL Web Development by Luke Welling
5. Learning PHP, MySQL, JavaScript, CSS... by Robin Nixon
6. PHP and MySQL for Beginners by Mark A Lassoff
7. Database System Concepts by Abraham Silberschatz, Henry F. Korth, S. Sudarshan
8. Database System, written by Connolly & Begg
9. HTML BY Davinder Singh Minhas
10. HTML and CSS: Design and Build Websites, Author: Jon Duckett