CheapRides

Online Car Rental System from UTM

1.1 Background

Car rental or a car hire agency is a small shop which rent cars or automobiles for short period of time for a fee to their customer. In Malaysia, car rental service increasingly becomes the preferred option for most people, especially among students in campuses and universities. This occurs because not all students can afford having their own vehicle and perhaps the university bus service doesn’t always help. Besides, the raising taxi fares and inconsistent bus arrivals in Malaysia continue to discourage people from taking up the public transport. Car rental service continues to grow in Malaysia; hence it required an improvement and good monitoring system. However, some car rental agencies still use a manual system to manage rental car operations by spreading of their available car to local resident. This method wasting money and time for both rental person and car rental owner. Therefore, it is proposed to have a system that can be used to provide booking and management to make easier for both of them. This system takes information from the rental person and car rental owner through filing their details. A user being registered in the website has the facility to book a vehicle as required.

1.2 Problem Statement

There are bunch of rental cars that owned by different owner. Some of the car that the owner provided are different from other and each of the car have their own advantages and disadvantages. There is various type of car will give burden for user to choose which car is the best for them. Besides that, manual system does not allow user to booking online and hard to keep track on the record of rental cars. Instead, the car owner only spread the word about their available car to a local resident only. User must contact a car rental owner and contract out for a vehicle and this will delay the process of renting car. This method consume time for both car rental and car rental owner.

1.3 Objective

The main objective of this project is to develop a system that allow the car rental owner to advertise their cars and allow user to search any type of car in thissystem. In order to achieve the above-mentioned aim, below is the objectives of this project.

i) To propose a system to manage the car rental business.
ii) To apply technique for user's criteria preference.
Iii) To test of system functionality of the proposed technique.

1.4 Scope

The scopes for this project are identified to make the system development process
easier. The scope will be explained from user aspect of view.

I. Admin

a) Manage and monitor the application
b) Admin can view report

II. Owner

a) Register and login profile
b) Register, update, and delete car details
c) Approve booking status

III. User

a) Register and login profile
b) Search for car base on criteria chosen
c) Book car
d) Rating car

1.4 Use Case Diagram

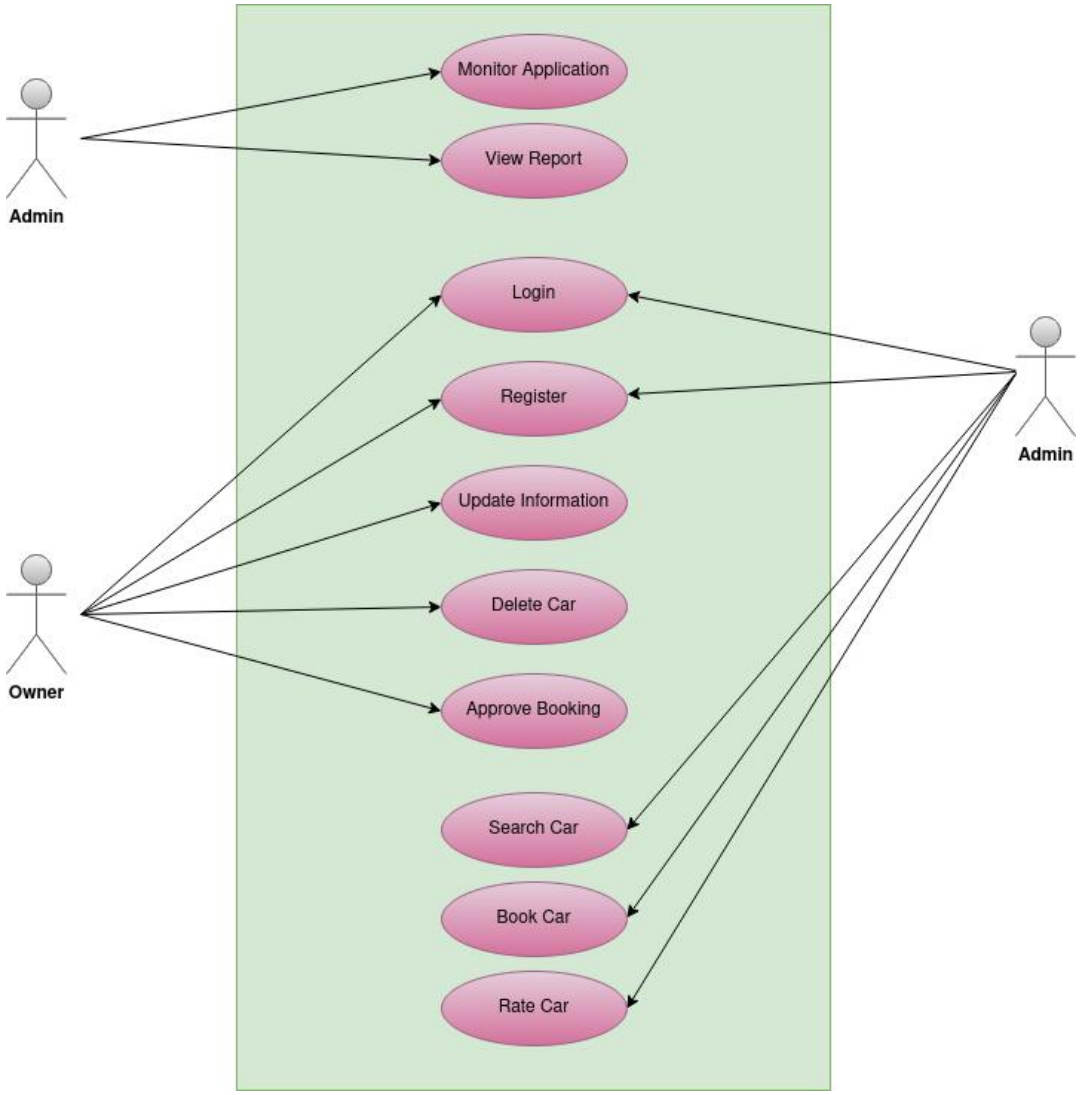

Fig 1. Use case diagram for Online Booking Rentals

1.5 Context Diagram

This system involves three main entities which is admin, user, and owner. User can bring user detail, book detail, and rating detail to the system. While owner can register car details to be presented in the application and give the approve booking into the system. As admin can monitor the system and view the reports.

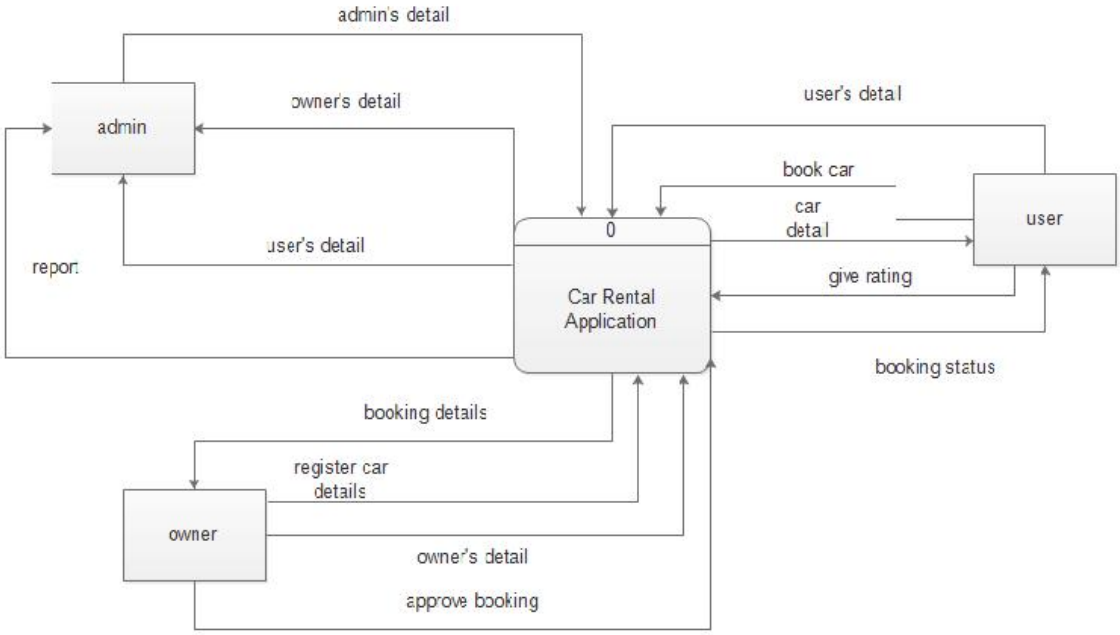

Fig 2. Context Diagram

1.6 Interface

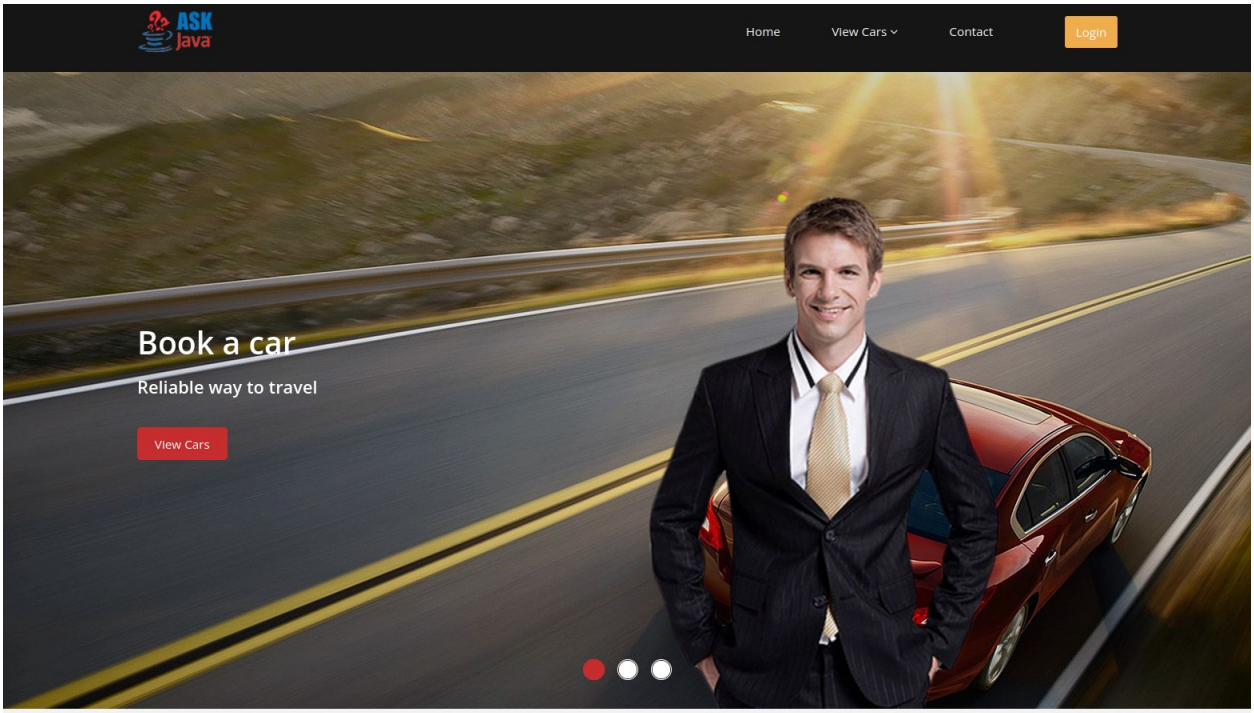

Fig 3. Home Interface

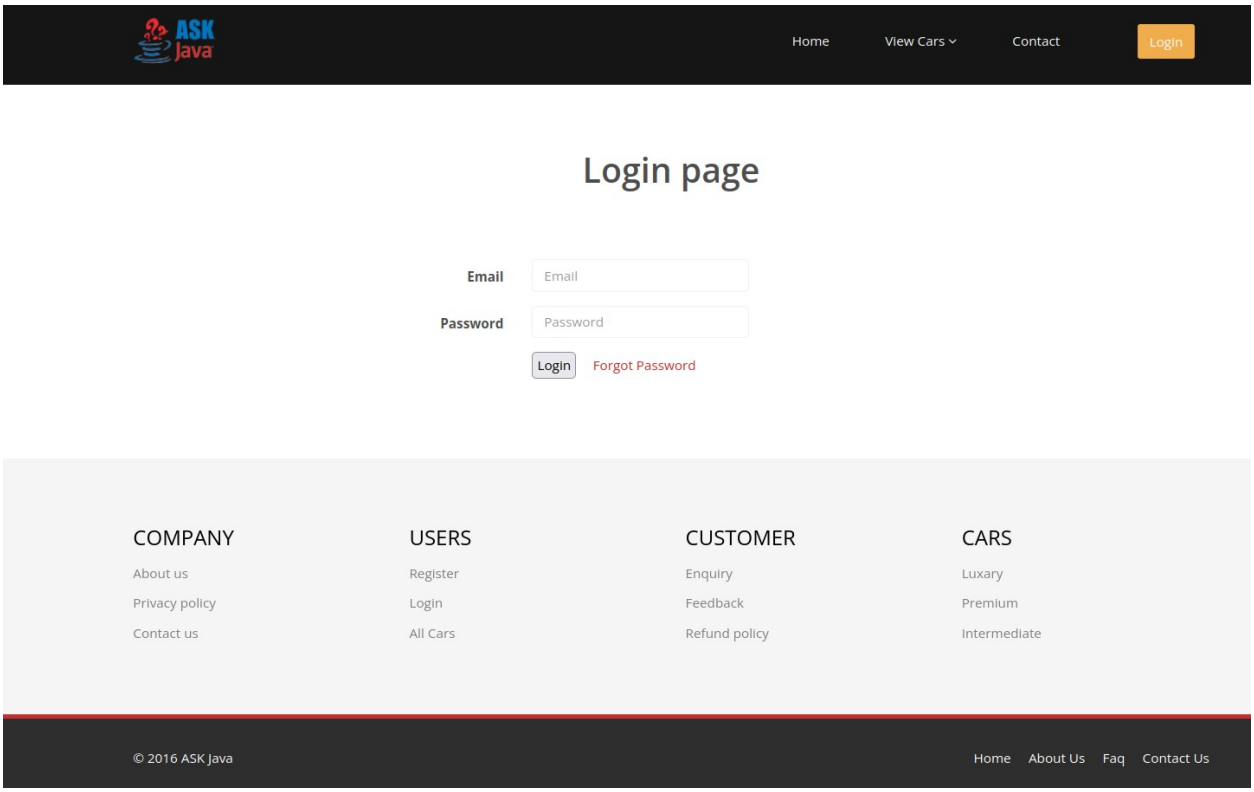

Fig 4. User Login Interface

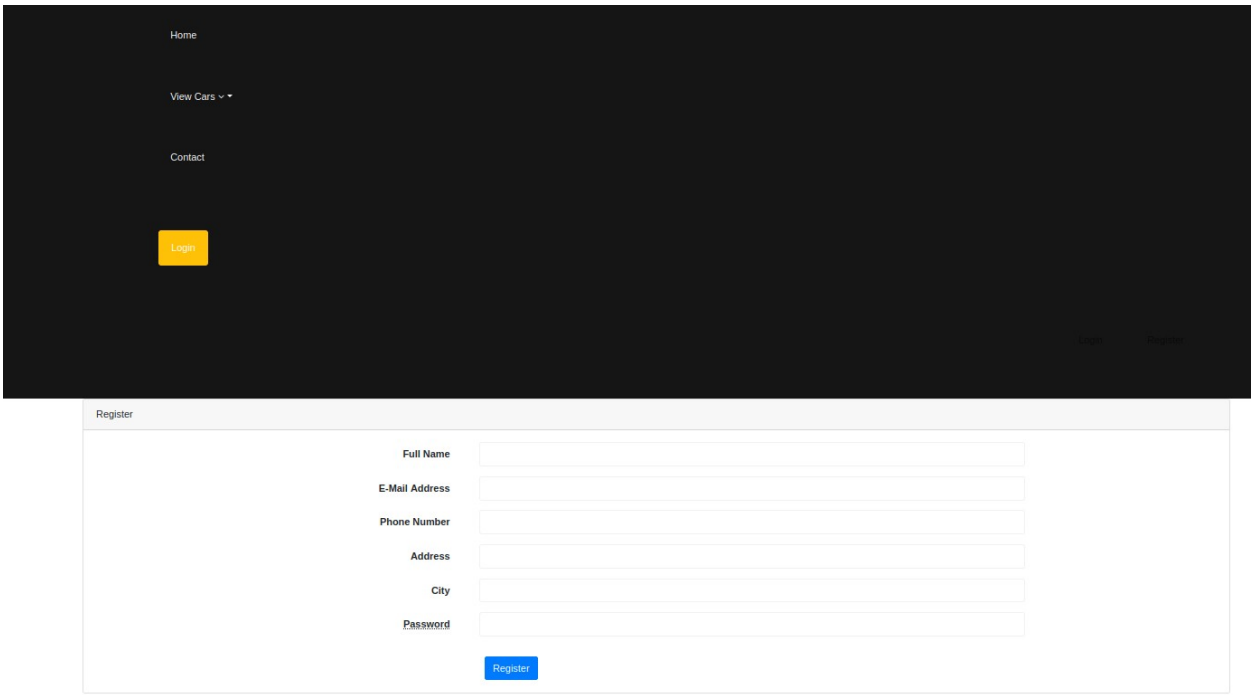

Fig 5. Register Interface for User

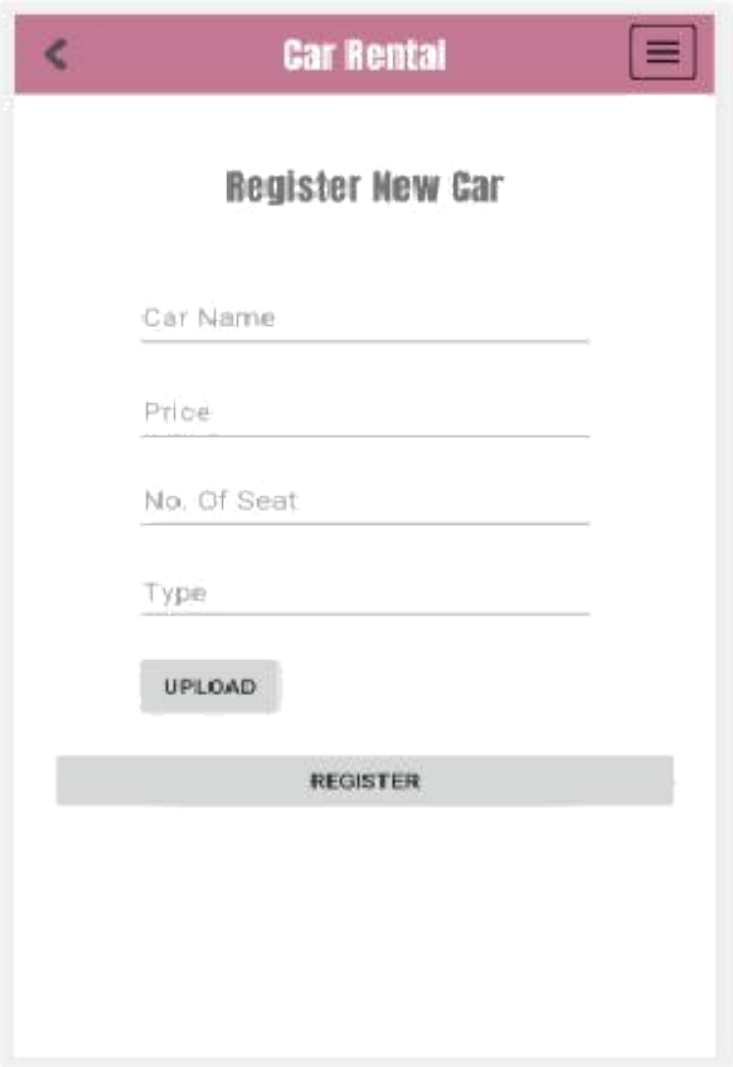

Fig 6. Register New Car Interface

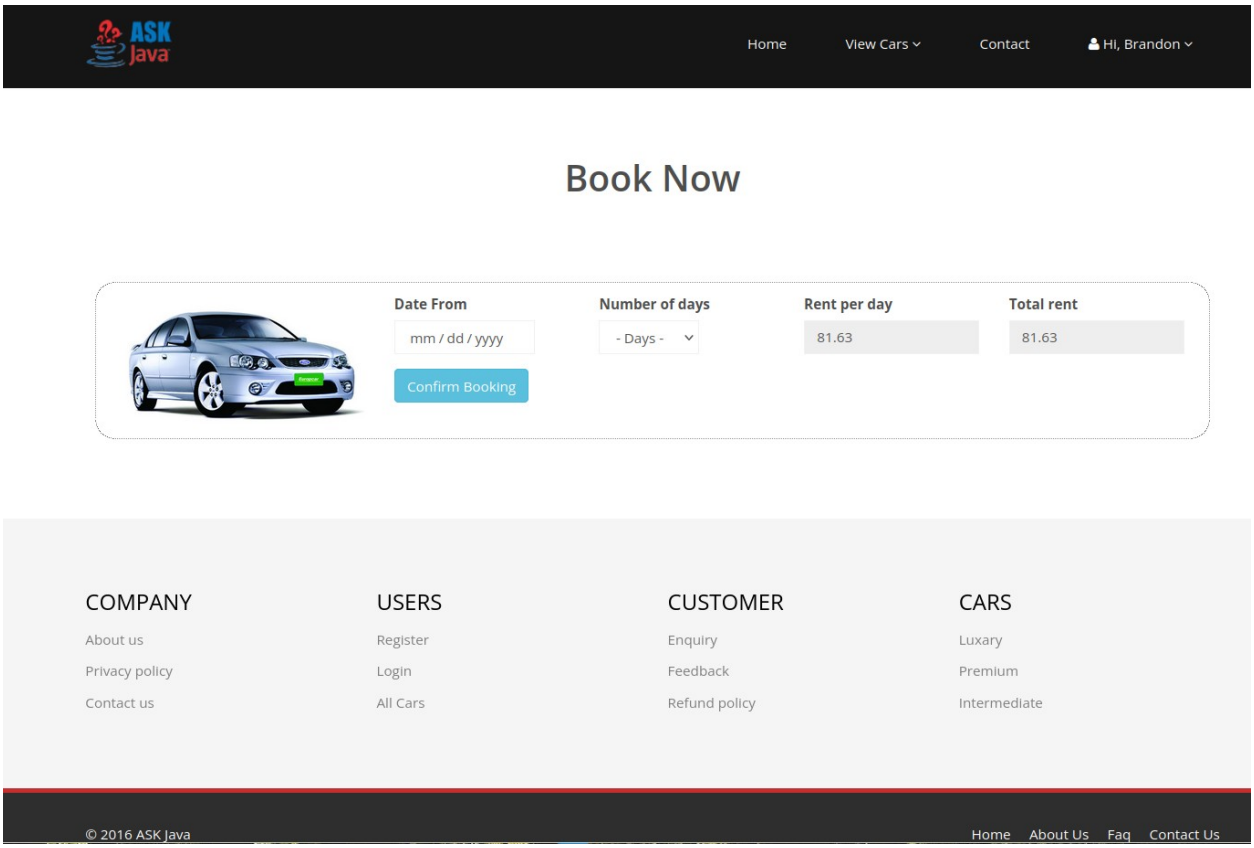

Fig 7. Booking Interface

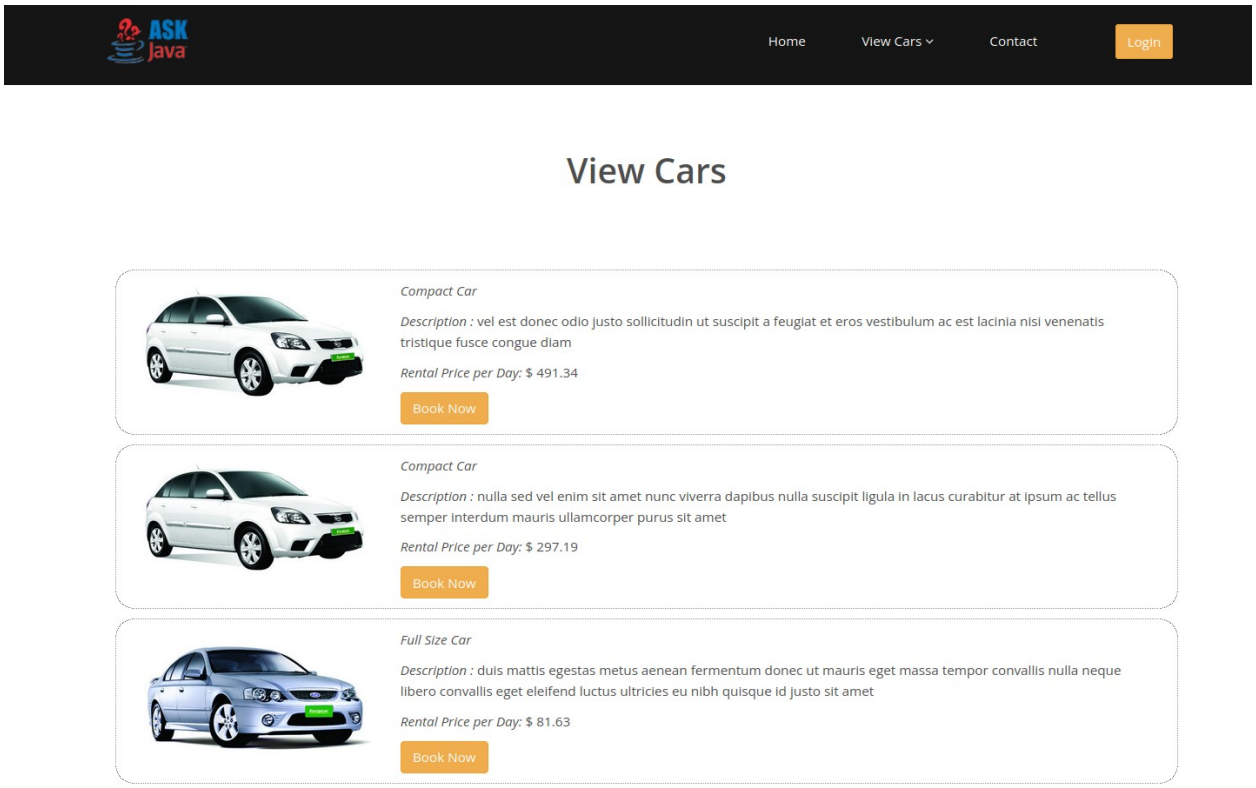

Fig 8. User Display Car Interface

Database Tables

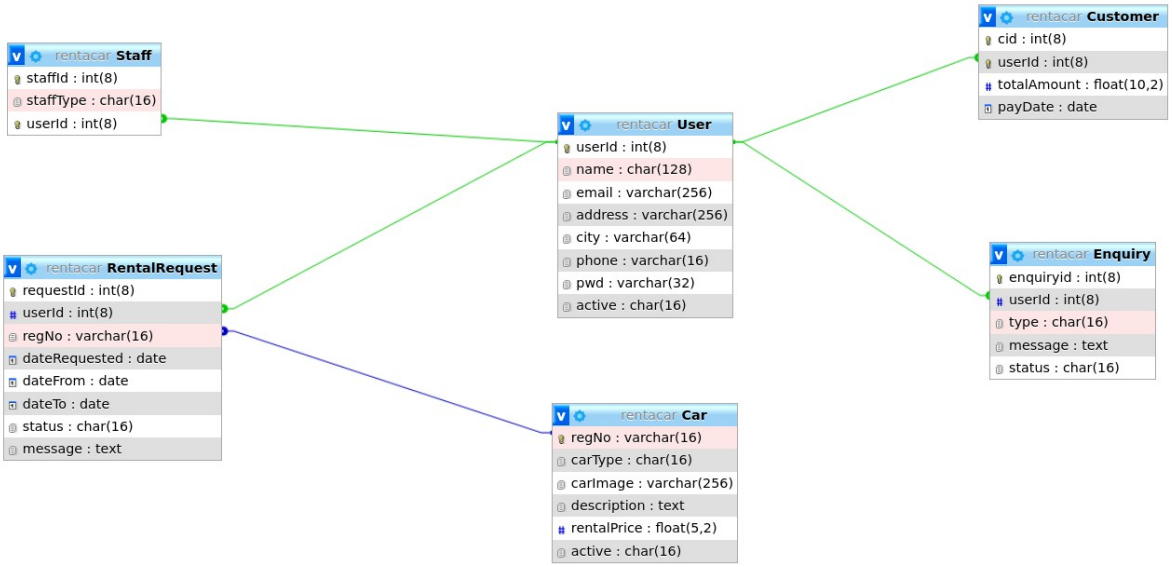

Fig 9. Database

Login Page

```
30
31      <%@ include file="header.jsp" %>
32
33      <section id="blog" class="container">
34          <div class="center">
35              <h2>Login page</h2>
36          </div>
37          <div class="blog">
38              <div class="row">
39                  <div class="col-md-offset-3 col-md-8">
40                      <form class="form-horizontal" method="POST" action="loginProcess.jsp" data-toggle="validator">
41                        <div class="form-group">
42                          <label for="inputEmail3" class="col-sm-2 control-label">Email</label>
43                          <div class="col-sm-4">
44                            <input type="email" name="email" class="form-control" placeholder="Email" required>
45                          </div>
46                        </div>
47                        <div class="form-group">
48                          <label for="inputPassword3" class="col-sm-2 control-label">Password</label>
49                          <div class="col-sm-4">
50                            <input type="password" name="pwd" class="form-control" placeholder="Password" required>
51                          </div>
52                        </div>
53                        <div class="form-group">
54                          <div class="col-sm-offset-2 col-sm-4">
55                            <button type="submit" class=""> Login </button>
56                               <a href="forgot.html">Forgot Password</a>
57                          </div>
58                        </div>
59                      </form>
60                  </div><!--/.col-md-8-->
61              </div><!--/.row-->
62          </div>
63      </section><!--/#blog-->
64
65      <%@ include file="footer.jsp" %>
66
67      <script src="js/jquery.js"></script>
68      <script src="js/bootstrap.min.js"></script>
69      <script src="js/jquery.prettyPhoto.js"></script>
```

```
Test Results    Output ×
RentACar (run) ×    Payara Server ×
BUILD SUCCESSFUL (total time: 0 seconds)
```

Fig 10. Login page for User

Login-Process

```
30      <body>
32          <%@ include file="header.jsp" %>
33
34          <section id="blog" class="container">
35              <div class="center">
36                  <h1>Login Process Page</h1>
37                  <%-- Java class used to verify email and password stored in the database --%>
38                  <%@page import="Controller.LoginDao"%>
39                  <%-- Java class containing setter and getter methods for all the varibles of database fields --%>
40                  <jsp:useBean id="bean" class="model.User"/>
41                  <%-- Create an object of LoginBean  --%>
42                  <jsp:setProperty property="*" name="bean"/>
43
44                  <%
45                      //  A string that contains values retured from checkLogin function
46                      String str = LoginDao.checkLogin(bean);
47                      // If there is error
48                      if (str == "error") { %>
49                  <div class="alert alert-danger" role="alert">
50                      <span class="glyphicon glyphicon-exclamation-sign" aria-hidden="true"></span>
51                      <strong> Error:</strong> Email address or password do not match
52                  </div>
53                  <%  } // If there is an exception
54                  else if (str == "exception") { %>
55                  <div class="alert alert-danger" role="alert">
56                      <span class="glyphicon glyphicon-exclamation-sign" aria-hidden="true"></span>
57                      <strong> Exception:</strong> Exception in executing SQL query
58                  </div>
59                  <% } else if (str == "In-Active") { %>
60                  <div class="alert alert-danger" role="alert">
61                      <span class="glyphicon glyphicon-exclamation-sign" aria-hidden="true"></span>
62                      <strong> Alert:</strong> You account has been deactivated by Administrator
63                  </div>
64                  <% } else { %>
65                  <div class="alert alert-success" role="alert">
66                      <span class="glyphicon glyphicon-ok" aligned="true"></span>
67                      <strong> Success:</strong> You have been successfully logged-in !
html > body >
Test Results   Output ×
RentACar (run) ×   Payara Server ×
BUILD SUCCESSFUL (total time: 0 seconds)
```

Fig 11. Login process

LoginDAO

```java
import model.User;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.Arrays;



public class LoginDao {
    static String returnString = null;

    public static String checkLogin(User bean) {
        try {
            // Get connection object from ConnectionProvider.java
            Connection conn = ConnectionProvider.getConnection();
            // Prepare SQL query
            PreparedStatement ps = conn.prepareStatement("SELECT * FROM User WHERE email=? AND pwd=?");

            ps.setString(1, bean.getEmail());   // set first parameter to email
            ps.setString(2, bean.getPwd()); // set second parameter to password

            ResultSet rs = ps.executeQuery(); // get the result of the SQL query
            if(rs.next()) {
                // Result set will contain string in following numbers
                // 1 - id, 2 - name, 3 - password, 4- email, 5 - country
                if(rs.getString(8).equals("In-Active"))
                    returnString = "In-Active";
                else
                    returnString = String.valueOf(rs.getInt(1)) + "-" + rs.getString("name").substring(0, rs.getString("name").indexOf(" ")) ;
            } else {
                returnString = "error";
            }
        } catch (Exception ex) {
            returnString = "exception";
        }
        return returnString;
    }
}
```

```
Test Results    Output
RentACar (run)    Payara Server
BUILD SUCCESSFUL (total time: 0 seconds)
```

Fig 12. Data Access Object for Login

Booking

```
29      <body>
30          <%@ include file="header.jsp" %>
31          <section id="blog" class="container">
32              <div class="center">
33                  <h2>Book Now</h2>
34              </div>
35              <div class="blog">
36                  <%-- Java class used to retive car info stored in the database --%>
37                  <%@page import="Controller.BookCarDao"%>
38                  <%-- Java class containing setter and getter methods for all the varibles of database fields --%>
39                  <jsp:useBean id="bean" class="model.Car"/>
40                  <%-- Create an object of LoginBean  --%>
41                  <jsp:setProperty property="*" name="bean"/>
42                  <%
43                      Car c = BookCarDao.bookCar(bean, request.getParameter("id"));
44                  %>
45                  <form action="book.jsp" method="post" >
46                      <div class="row car-details">
47                          <div class="col-md-3">
48                              <%  out.println("<img class='img-responsive' src='" + c.getCarImage() + "'>");%>
49                          </div>
50                          <div class="col-md-9">
51                              <div class="row">
52                                  <div class="form-group col-md-3">
53                                      <label>Date From</label>
54                                      <div class='input-group date'>
55                                          <input type='date' name="from" id="dateFrom" class="form-control" required/>
56                                      </div>
57                                  </div>
58                                  <div class="form-group col-md-3">
59                                      <label>Number of days</label>
60                                      <div class='input-group'>
61                                          <select id="days" name="days" class="form-control" required>
62                                              <option selected disabled="disabled"> - Days - </option>
63                                              <% for (int a = 1; a <= 10; a++) {
64                                                      out.print("<option value='" + a + "'>" + a + "</option>");
65                                                  }
66                                              %>
67                                          </select>
```

html > body > section > div > form > div > div > div > div > div

Test Results | Output

RentACar (run) | Payara Server

BUILD SUCCESSFUL (total time: 0 seconds)

Fig 13. Booking process

BookCarDAO

```java
package Controller;
import model.Car;
import java.io.PrintWriter;
import java.io.StringWriter;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;

public class BookCarDao {

    static String returnString = null;

    public static Car bookCar(Car bean, String regNo) {
        Car temp = new Car();

        try {
            // Get connection object from ConnectionProvider.java
            Connection conn = ConnectionProvider.getConnection();
            // Prepare SQL query
            PreparedStatement ps = conn.prepareStatement("SELECT * FROM Car WHERE regNo=?");

            ps.setString(1, regNo);   // set first parameter to email

            ResultSet rs = ps.executeQuery(); // get the result of the SQL query

            rs.first();
            temp.setRegNo(rs.getString("regNo"));
            temp.setCarImage(rs.getString("carImage"));
            temp.setCarType(rs.getString("carType"));
            temp.setDescription(rs.getString("description"));
            temp.setRentalPrice(rs.getFloat("rentalPrice"));
            return temp;
        } catch (Exception ex) {
            StringWriter errors = new StringWriter();
            ex.printStackTrace(new PrintWriter(errors));
            returnString = "exception" + errors.toString();
        } finally {
            return temp;
        }
```

```
Test Results   Output ×
RentACar (run) ×   Payara Server ×
BUILD SUCCESSFUL (total time: 0 seconds)
```

Fig 14. Data access object for booking

View Cars

```
31
32      <body>
33
34          <%@ include file="header.jsp" %>
35
36          <section id="blog" class="container">
37              <div class="center">
38                  <h2>View Cars</h2>
39              </div>
40              <div class="blog">
41                  <%-- Java class used to retive car info stored in the database --%>
42                  <%@page import="Controller.ViewCarsDao"%>
43                  <%-- Java class containing setter and getter methods for all the varibles of database fields --%>
44                  <jsp:useBean id="bean" class="model.Car"/>
45                  <%-- Create an object of LoginBean  --%>
46                  <jsp:setProperty property="*" name="bean"/>
47                  <%
48                      ArrayList<Car> str = ViewCarsDao.viewCars(bean, request.getParameter("cars"));
49                     for (Car c : str) { %>
50                  <div class="row car-details">
51                      <div class="col-md-3">
52                          <%  out.println("<img class='img-responsive' src='" + c.getCarImage() + "'>"); %>
53                      </div>
54                      <div class="col-md-9">
55                          <%
56                              out.println("<p><em>" +c.getCarType() + " Car </em> </p>");
57                              out.println("<p><em>Description : </em>" + c.getDescription() + "</p>");
58                              out.println("<p><em>Rental Price per Day: </em>$ " + c.getRentalPrice() + "</p>");
59                              out.println("<a class='btn btn-warning' href='book-now.jsp?id=" + c.getRegNo() + "' role='button'>Book Now</a>");
60                          %>
61                      </div>
62                  </div>
63                  <% }%>
64              </div><!--/.row-->
65          </div>
66      </section><!--/#blog-->
67
68      <%@ include file="footer.jsp" %>
69
70      <script src="js/jquery.js"></script>
```

```
Test Results    Output ×
RentACar (run) ×    Payara Server ×
BUILD SUCCESSFUL (total time: 0 seconds)
```

Fig 15. View cars

ViewCarsDAO

```java
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;

public class ViewCarsDao {
    static String returnString = null;

    public static ArrayList viewCars(Car bean, String type) {
        if(type.equals("all"))
            type = "%";
        ArrayList<Car> cars = new ArrayList<Car>();
        try {
            // Get connection object from ConnectionProvider.java
            Connection conn = ConnectionProvider.getConnection();
            // Prepare SQL query
            PreparedStatement ps = conn.prepareStatement("SELECT * FROM Car WHERE carType like ? AND active=?");

            ps.setString(1, type);   // set first parameter to email
            ps.setString(2, "Active");   // set first parameter to email

            ResultSet rs = ps.executeQuery(); // get the result of the SQL query

            while(rs.next()) {
                Car temp = new Car();
                temp.setRegNo(rs.getString("regNo"));
                temp.setCarImage(rs.getString("carImage"));
                temp.setCarType(rs.getString("carType"));
                temp.setDescription(rs.getString("description"));
                temp.setRentalPrice(rs.getFloat("rentalPrice"));

                cars.add(temp);
//                returnString +=  "car type " + rs.getString("regNo");
            }
                return cars;
        } catch (Exception ex) {
            StringWriter errors = new StringWriter();
            ex.printStackTrace(new PrintWriter(errors));
            returnString = "exception" + errors.toString();
        }
//        return returnString;
```

Test Results Output ×

RentACar (run) × Payara Server ×

BUILD SUCCESSFUL (total time: 0 seconds)

Fig 16. Data access object for view cars

View inquiries

```jsp
<%@ include file="admin-header.jsp" %>

<section id="blog" class="container">
    <div class="center">
        <h2>All Enquiries </h2>
    </div>
    <div class="blog">
        <div class="row">
            <div class="col-md-12">
                <%
                    String error;
                    if (session.getAttribute("type").equals("Staff") || session.getAttribute("type").equals("Admin")) { %>
                <table class="table table-condensed table-bordered">
                    <thead> <tr>
                            <th>#</th>
                            <th>userId</th>
                            <th>type</th>
                            <th>message</th>
                            <th>status </th>
                        </tr> </thead>
                        <%
                            try {
                                // Get connection object from ConnectionProvider.java
                                Connection conn = ConnectionProvider.getConnection();
                                // Prepare SQL query
                                PreparedStatement ps = conn.prepareStatement("SELECT * FROM Enquiry WHERE type=?");
                                ps.setString(1, "Enquiry");
                                ResultSet rs = ps.executeQuery();
                                int count = 1;
                                while (rs.next()) {
                                    String statusClass = "";
                                    if (rs.getString("status").equals("Pending")) {
                                        statusClass = "danger";
                                    } else if (rs.getString("status").equals("Replied")) {
                                        statusClass = "warning";
                                    } else {
                                        statusClass = "success";
                                    }
                        %>
```

```
Test Results    Output
RentACar (run)    Payara Server
BUILD SUCCESSFUL (total time: 0 seconds)
```

Fig 17. View enquiries for Staff and Admin

Accept Booking

```
43                  <h2>Book</h2>
44                  <p class="lead">Booking accepted</p>
45              </div>
46              <jsp:useBean id="bean" class="Controller.RentalRequest"/>
47              <div class="blog">
48                  <div class="row">
49                      <div class="col-md-12">
50                          <%
51                              String error;
52                              if (session.getAttribute("type").equals("Staff") || session.getAttribute("type").equals("Admin")) {
53                              try {
54                                  // Get connection object from ConnectionProvider.java
55                                  Connection conn = ConnectionProvider.getConnection();
56                                  // Prepare SQL query
57                                  PreparedStatement ps = conn.prepareStatement("UPDATE rentalRequest SET status=? WHERE requestId=?");
58
                                    ps.setString(1, "Accepted");   // set first parameter to email
60                                  ps.setString(2, request.getParameter("rid"));   // set first parameter to email
61
62                                  ps.executeUpdate(); // get the result of the SQL query
63                              } catch (Exception ex) {
64                                  StringWriter errors = new StringWriter();
65                                  ex.printStackTrace(new PrintWriter(errors));
66                                  error = "exception" + errors.toString();
67                                  out.println(error);
68                              }
69                          %>
70                          <h4>Booking accepted</h4>
71                          <h4>Email is sent to the customer</h4>
72                          <jsp:forward page="rental-requests.jsp" />
73
74                          <% } else { %>
75                          <div class="alert alert-danger" role="alert">
76                              <span class="glyphicon glyphicon-ok" aria-hidden="true"></span>
77                              <strong> Please login: </strong> You must be logged-in to book car!
78                          </div>
79                          <% }%>
80
81                      </div><!--/.col-md-8
82                  </div><!--/.row-->
```

html > body > section > div > div > div

Test Results | Output

RentACar (run) | Payara Server

BUILD SUCCESSFUL (total time: 0 seconds)

Fig 18. Accept booking

1.6 Conclusion

Car Rental System has offered an advantage to both car rental as well as car rental owner to efficiently and effectively manage the business and satisfies user’s need.

Reflection

Ruhul Quddus Tamim

I was able to learn about a variety of topics about which I had no prior knowledge, such as MVC Tiers for Internet programming and the history of how coding has progressed through technology to lead to our modern society. We are now in a position to do things and grateful for the opportunity to lead a team and manage a group that this course has given me.

Md Yusuf Bin Forkan

When time and resources are scarce, the most obvious benefit of this course was the ability to collaborate effectively with others. Dr Norizam bin Katmon, our lecturer, was friendly and helpful. We were able to finish our semester effectively thanks to a variety of projects that were both interesting and enjoyable. With all the values, ethics, information, skills, and communication that I've gained throughout the course, I can confidently say that it has helped me become a better programmer.

Shafi Ahmed

We are proud that finally we created a website for our project. I gained a wide range of abilities, including leadership, adaptability, and time management, while working with my team. Despite the obstacles we encountered, I found this project to be the most time-consuming this semester. There were times when several of us considered abandoning our studies because of the difficulty of the material, but we persevered and were aided by our lecturer, who was always there to help. In spite of the fact that it was done online, we were able to complete this project in the most efficient manner feasible. It was an honour to work with the people on my team because we all helped each other out and figured out how to fix everything.

Syafiq Ibnu Ramadhan

1. What did you learn during this collaborative learning interaction?
   I learn to communicate better
   Implementing the knowledge learn in class and gain better understanding

2. What did you contribute?
   Trello Update
   Project's discussion
   Contribute on making proposal and project's progress presentation

3. What did the other members of the group contribute?
   A lot,we divide the task fairly and some of them did more than they should have

4. What is your assessment of how the group is functioning thus far?
   Good communication even though everything was conducted online
   Fair task portion
   And supportive team members
   Transparent work