

L1 – Servlet & JSP Lab Assessment

Name: Nagarjun N S

Emp ID: 12228

Please find the below case study which is assigned to your name and perform below action

Instructions:

1.

Analyze the given problem using **W3H** technique.

2.

Write an **algorithm** and draw the **flowchart** for the application.

3.

Prepare **UML diagrams** for the given scenario [**Class Diagram, Use Case Diagram & Sequence Diagram**].

4.

Create a web-based application for the given scenario using **Servlet and JSP**.

5.

Follow **MVC** design pattern.

6.

Use **MySQL** as your database.

7.

Perform all the **CRUD** operations using **DAO** pattern.

8.

Make the application as user interactive and responsive.

9.

Apply style using **CSS / Bootstrap**.

10.

Perform the validations using **Javascript**.

11.

Maintain the **Code Quality** and the **Coding Standard**.

Question:

12228	Nagarjun Suresh Babu	Courier office Management System
-------	----------------------	----------------------------------

W3H:

PROJECT NAME: Courier office Management System	
What? 1	How? 2
<p>1. What are the modules are required? Ans: a) Admin (System) Module b) User Module</p> <p>2. What are the responsibilities of Admin (System)? Ans: a) Admin (System) will handle all the CRUD operations of the user database and the courier database.</p> <p>3. What can the user do? Ans: a) The user can be able to track their courier status. b) User can be able to add a notes to their courier package. c) User can be able to see their Bill. d) User can be able to see their courier package details. e) User can be able to see their profile.</p> <p>4. What can the user see in their profile tab? Ans: a) User can be able to see their personal details. b) User can be able to see their courier history. c) They can update details of their account.</p> <p>5. What happens after the courier is delivered? Ans: Once the courier is delivered, the user is informed via SMS and the application notification.</p> <p>6. What types of couriers are available? Can a user be able to change it? Ans: There are 2 different types of couriers available. One is Speed courier, and another one is normal courier. User can be able to opt of their desires.</p>	<p>1) Admin (System): a) Login: Method 1: The admin can login using their Email and password. Method 2: The admin can login using their username and password. Method 3: The admin can login to their account using social media accounts.</p> <p>b) Adding users: Method 1: Admin can also add users by using their customerID, Name, Mobile number, city,emailID. Method 2: Admin can add users with their Name, email, ph.no and address.</p> <p>c) Update / delete / view user accounts: Method 1: Admin can be able to Update / delete / view user accounts by providing separate buttons. Method 2: Admin can be able to Update / delete / view user accounts by using the drop-down list.</p> <p>d) CRUD on Courier packages: Method 1: Admin can be able to perform CRUD operations based on the input of users by using the separate buttons for each operations. Method 2: Admin can be able to perform CRUD operations based on the input of users by using the drop-down list for each operations.</p> <p>2) User: a) Login: Method 1: The user can login by using their email and password. Method 2: The user can login by using mobile number and name and customer ID. Method 3: The user can login to their account using social media accounts.</p> <p>b) CRUD Operations: Method 1: User can perform CRUD Operations for their account and their courier packages with separate buttons. Method 2: User can be able to perform CRUD Operations for their account and their courier packages using a drop-down list and a confirm button.</p> <p>c) Check courier status: Method 1: User can be able to check their courier status using the “Track Courier” option in the menu. Method 2: User can check their courier status by calling the courier office.</p>
<p>1) Admin (System): a) Login: Method 1: The admin can login using their Email and password. (This method is used because the originality of the admin is verified).</p> <p>b) Adding users: Method 1: Admin can also add users by using their customerID, Name, Mobile number, city, emailID. (This method is used because the admin can also add the user in case of any server problems by providing the user’s basic details.)</p> <p>c) Update / delete / view user accounts: Method 1: Admin can be able to Update / delete / view user accounts by providing separate buttons. (Providing separate buttons with separate functionalities will help to use the application easily.)</p> <p>d) CRUD on Courier packages: Method 1: Admin can be able to perform CRUD operations based on the input of users by using the separate buttons for each operations. (If in any case the user needs to cancel that courier or if the user wants to change the address, then the admin only can be able to approve and change it by using the separate buttons, which is straight forward and easy one.)</p> <p>2) User: a) Login: Method 1: The user can login by using their email and password. (Login to the system by using their email and password will be a hassle-free experience and results in a secure way of authentication. Because the email ID is already a unique and verified one.)</p> <p>b) CRUD Operations: Method 1: User can perform CRUD Operations for their account and their courier packages with separate buttons. (Providing separate buttons with separate functionalities will help to use the application easily.)</p> <p>c) Check courier status: Method 1: User can be able to check their courier status using the “Track Courier” option in the menu. (By providing a separate tracking status for each courier will also help the customer to believe more about their package and the user doesn’t need to bother by thinking questions like “what happened to my package?”,etc.. By this, the tracking system will be very much easier.)</p>	<p>1) Admin: a) Login: Method 2: The admin can login using their username and password. (Admin can’t login using their username and password as the originality is not verified.)</p> <p>Method 3: The admin can login to their account using social media accounts. (Using Social media accounts to login will not be a feasible way and it is less secure).</p> <p>b) Adding user: Method 2: Admin can add users with their Name, email, ph.no and address. (If a user is added using the basic details, then the Originality of the user is not verified.)</p> <p>c) Update / delete / view user accounts: Method 2: Admin can be able to Update / delete / view user accounts by using the drop-down list. (The drop-down list feature is not feasible, and it is hard to use, and it is time-consuming function too).</p> <p>2) User: a) Login: Method 2: The user can login by using mobile number and name and customer ID. (User login using mobile number, name and customerID will not be much secure as email and password)</p> <p>Method 3: The user can login to their account using social media accounts. (Using social media accounts to login will be a security threat. Many personal details will be provided while sending a courier. Those details might have a chance to be leaked and unauthorized usage)</p> <p>b) CRUD Operations: Method 2: User can be able to perform CRUD Operations for their account and their courier packages using a drop-down list and a confirm button. (The drop-down list feature is not feasible, and it is hard to use, and it is time-consuming function too).</p> <p>c) Check courier status: Method 2: User can check their courier status by calling the courier office. (If a user needs to call the office to track the courier status, then there will be a huge traffic of calling to the office which results in loss of time and loss of productivity to the owner. So, it is not feasible.)</p>
Why? 3	Why not? 4

ALGORITHM:

Courier Management system:

ADMIN:

- Step 1: Start
- Step 2: Go to Login page
- Step 3: Enter the credentials
- Step 4: After completing step 3, click login
- Step 5: Go to homepage
- Step 6: Click on “Pending Couriers”
- Step 7: Get the courier details
- Step 8: Logout
- Step 9: Stop

USER:

Courier Sending:

- step 1: Start
- step 2: Login by entering your credentials
- step 3: Go to homepage
- step 4: Select “Send Courier”
- step 5: Enter destination details
- step 6: Enter personal details
- step 7: Click “Check Price”
- step 8: Make Payment
- step 9: Download Bill
- step 10: Logout and give the package to the courier office
- step 11: stop

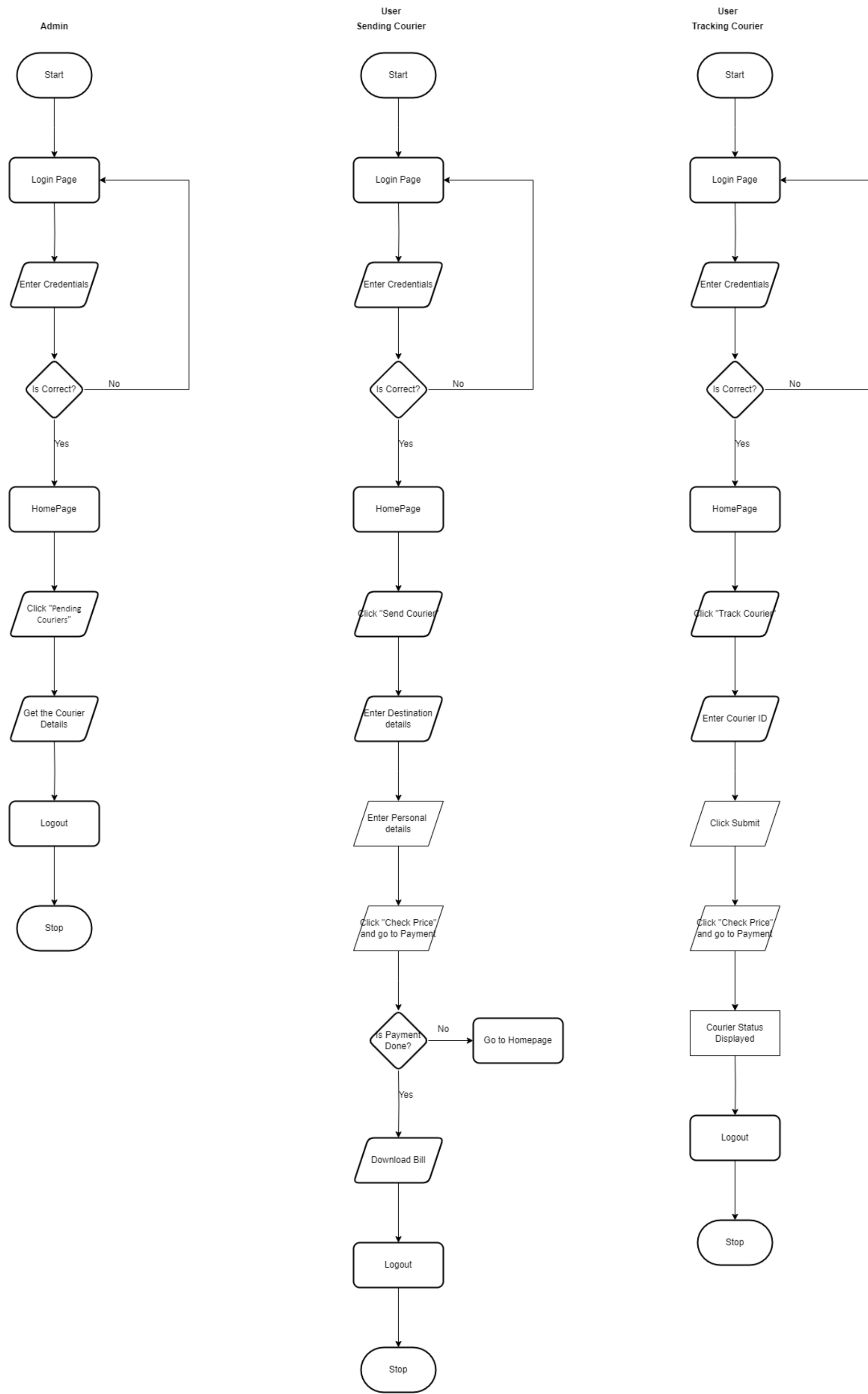
Update Courier Details:

- step 1: Start
- step 2: Login by entering your credentials
- step 3: Go to homepage
- step 4: select “Change Courier Details”
- step 5: Enter courier ID
- step 6: Enter your Mobile number
- step 7: Enter OTP
- step 8: Click Submit
- step 9: Update Courier details
- step 10: Confirm it
- step 11: Success msg displayed
- step 12: Logout
- step 13: Stop

Track Courier:

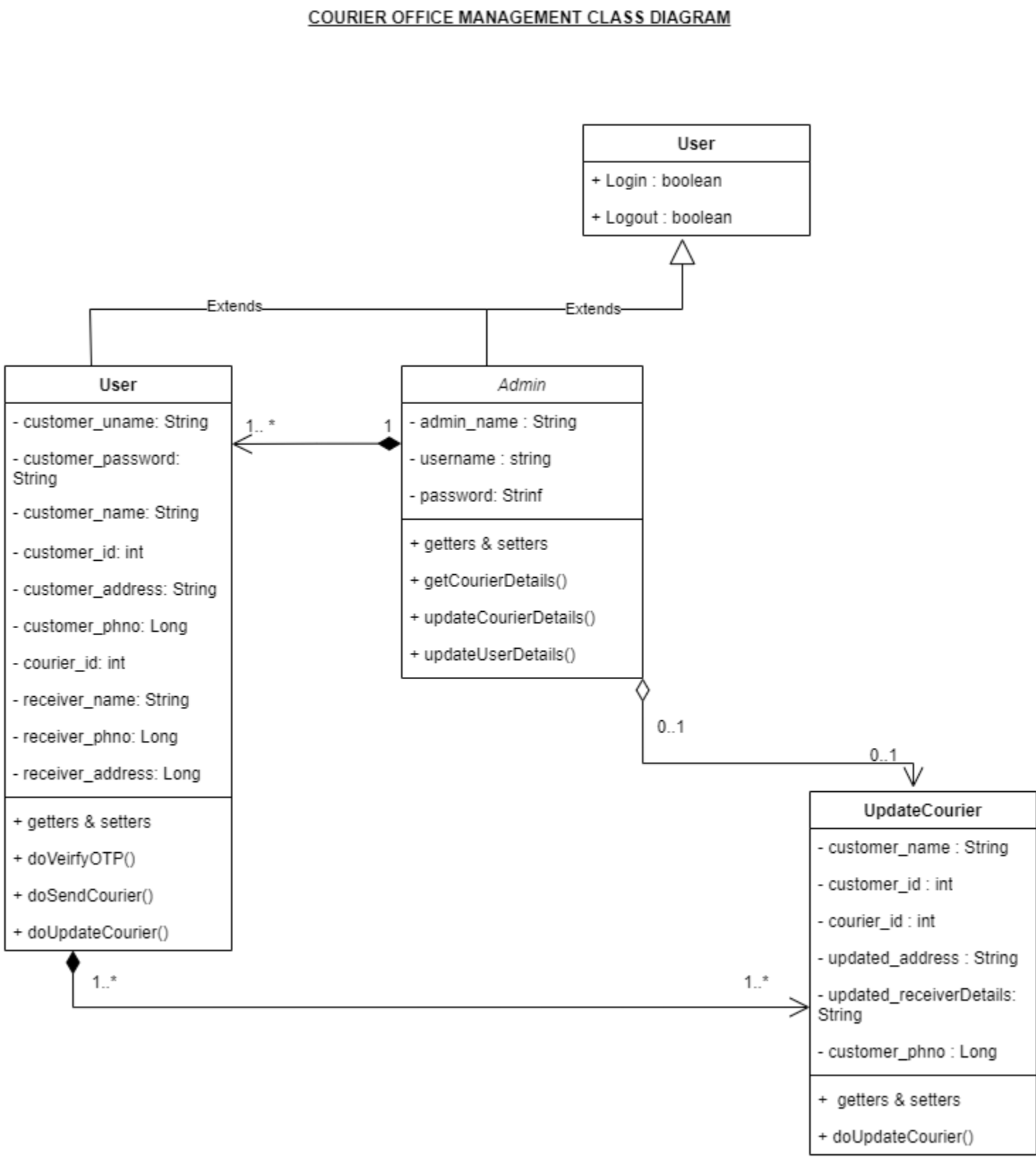
- step 1: Start
- step 2: Login by entering your credentials
- step 3: Go to homepage
- step 4: Select "Track Courier”
- step 5: Enter the Courier ID
- step 6: Click Ok
- step 7: Courier Status will be displayed
- step 8: Logout
- step 9: Stop

FLOWCHART:



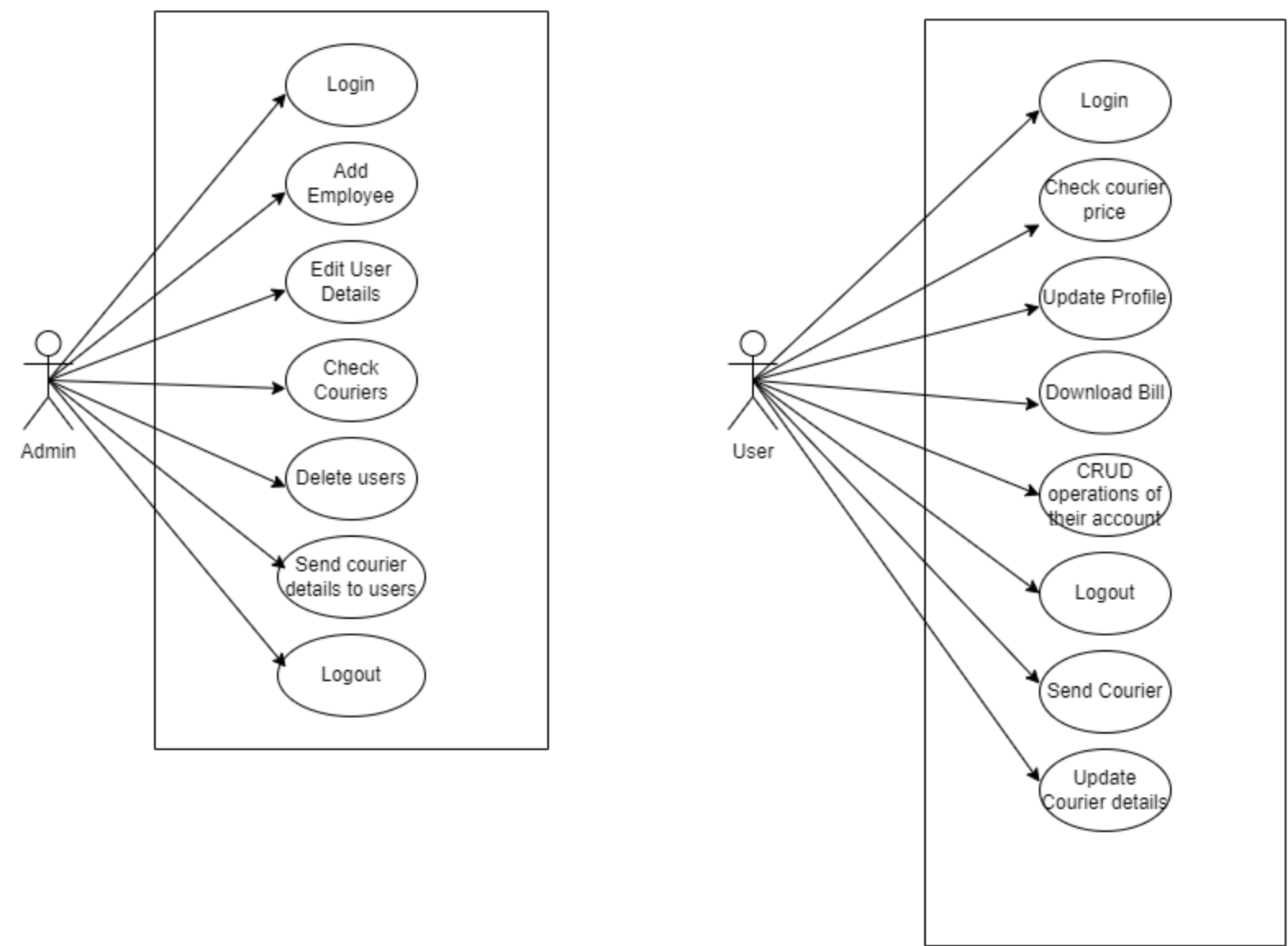
UML DIAGRAMS:

CLASS DIAGRAM:

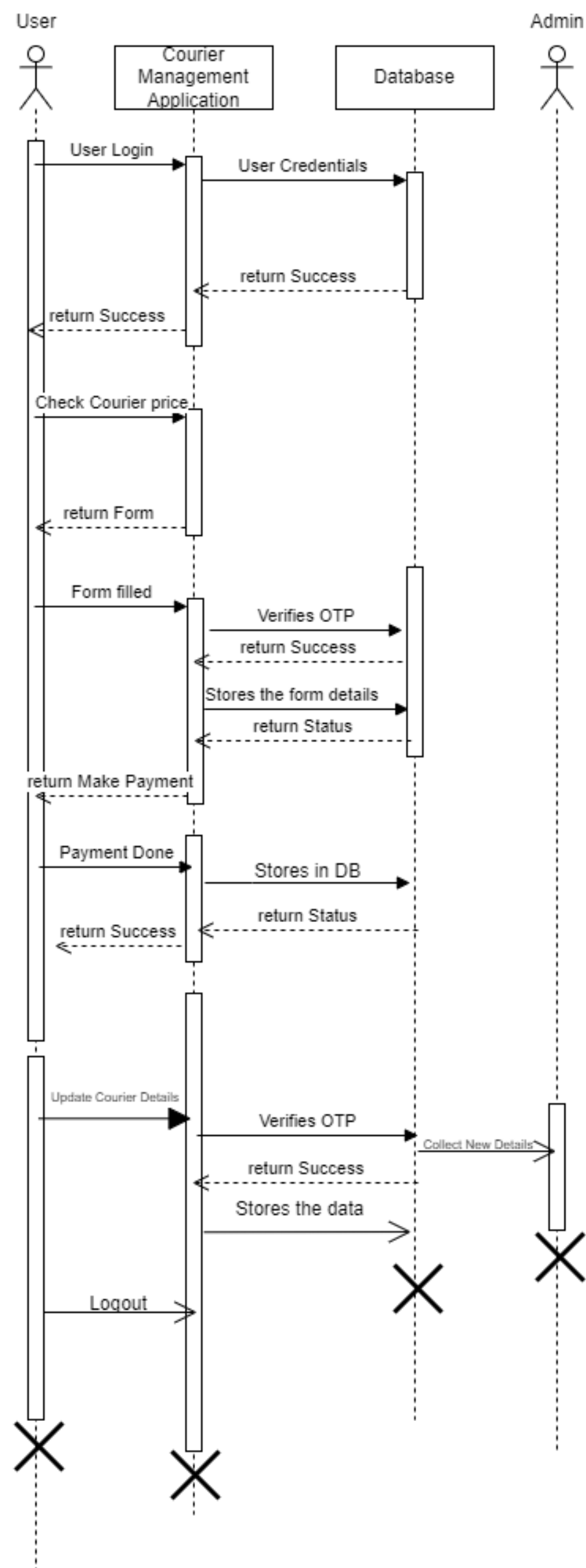


Use-Case Diagram:

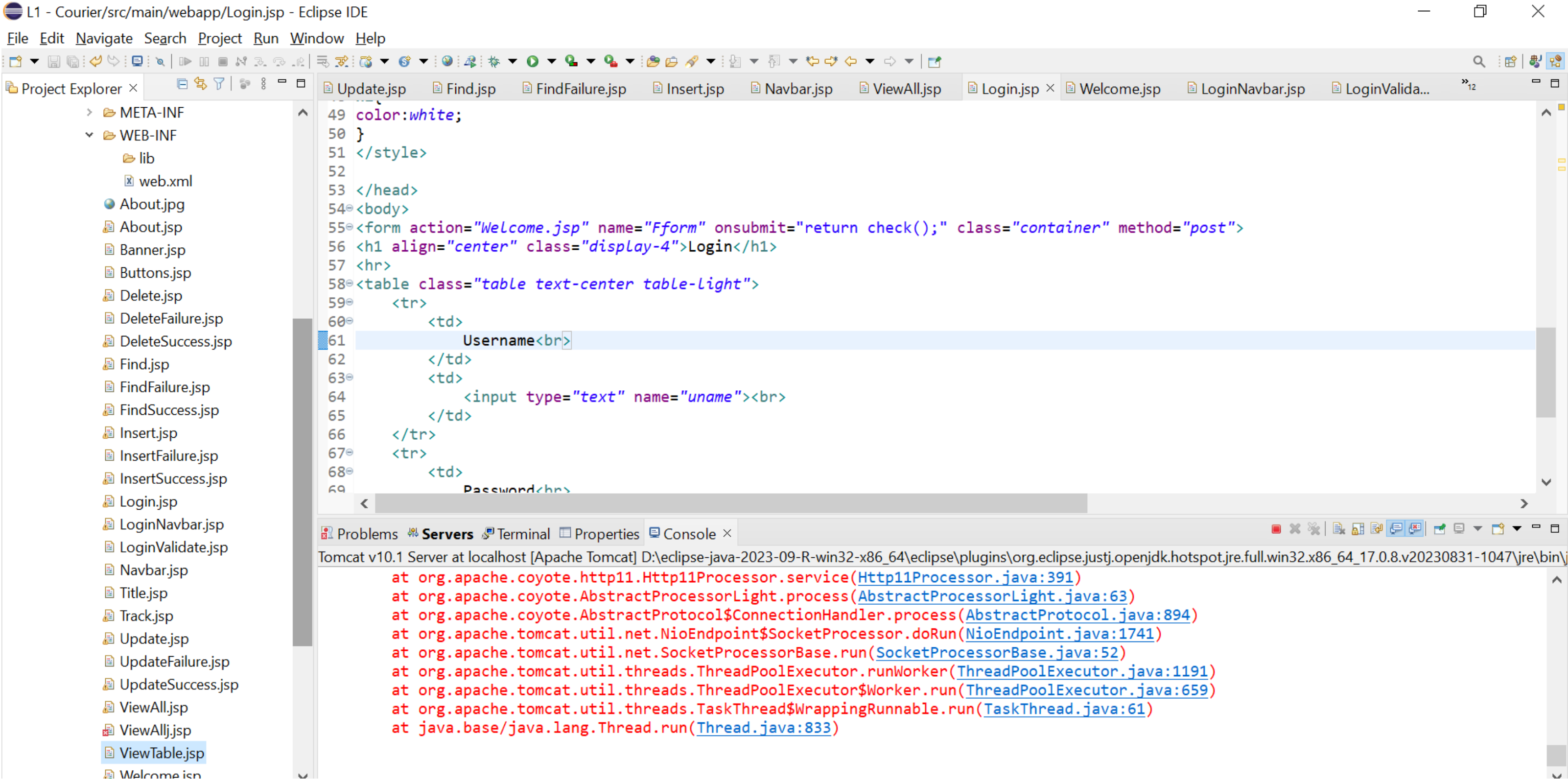
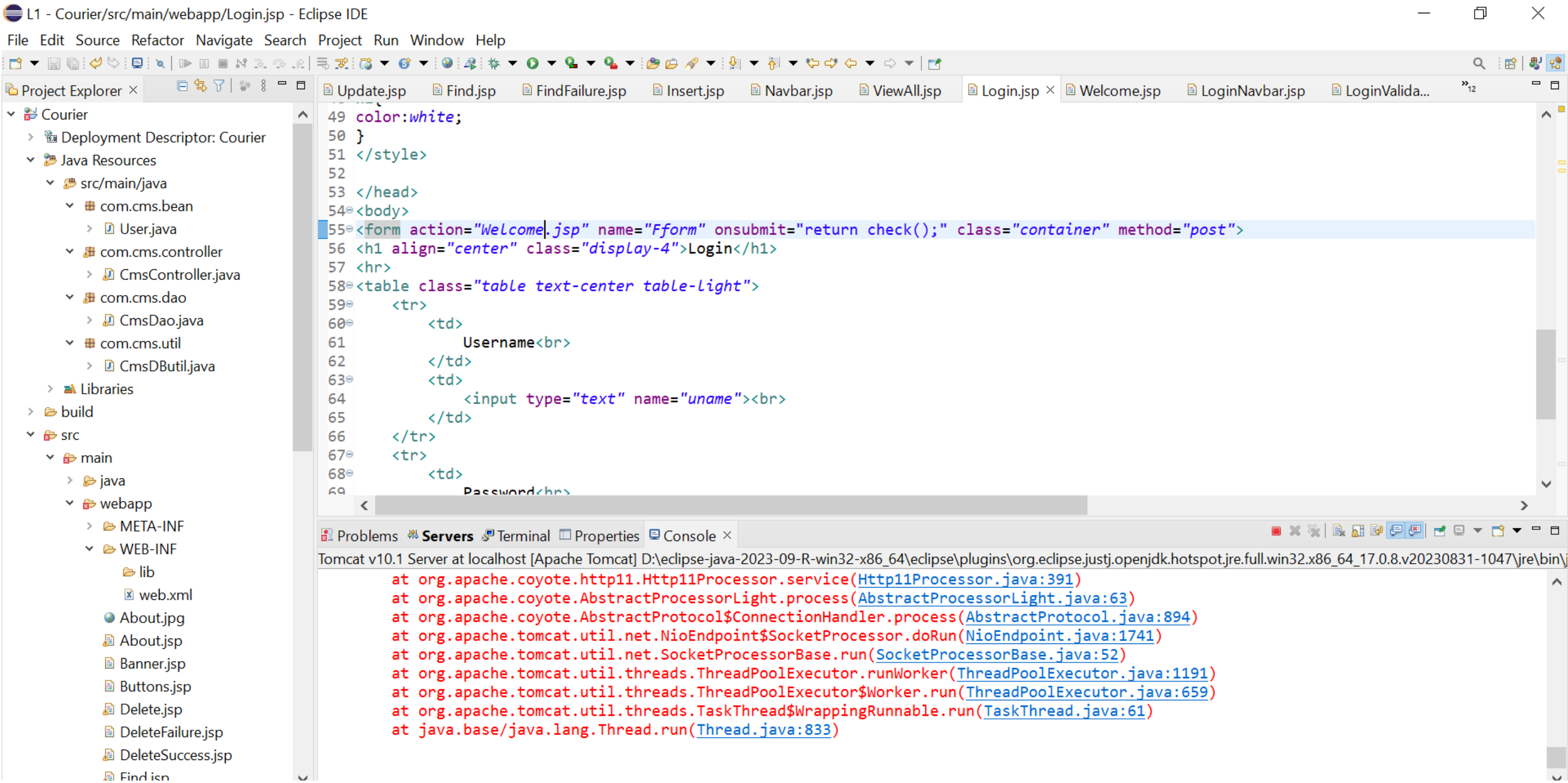
USE CASE DIAGRAM



SEQUENCE DIAGRAM:



FOLDER STRUCTURE:



CODE:

com.cms.bean

User.java:

```
package com.cms.bean;

public class User {
    int uid;
    String uname;
    int cid;
    String uadd;
    String desti;
    String rec_name;

    public User() {
        super();
    }

    public User(int uid, String uname, int cid, String uadd, String desti, String rec_name) {
        super();
        this.uid = uid;
        this.uname = uname;
        this.cid = cid;
        this.uadd = uadd;
        this.desti = desti;
        this.rec_name = rec_name;
    }

    public int getUid() {
        return uid;
    }

    public void setUid(int uid) {
        this.uid = uid;
    }

    public String getUname() {
        return uname;
    }

    public void setUname(String uname) {
        this.uname = uname;
    }

    public int getCid() {
        return cid;
    }

    public void setCid(int cid) {
        this.cid = cid;
    }

    public String getUadd() {
        return uadd;
    }

    public void setUadd(String uadd) {
        this.uadd = uadd;
    }

    public String getDesti() {
        return desti;
    }

    public void setDesti(String desti) {
        this.desti = desti;
    }

    public String getRec_name() {
        return rec_name;
    }

    public void setRec_name(String rec_name) {
        this.rec_name = rec_name;
    }

}
```

com.cms.controller

CmsController.java:

```
package com.cms.controller;

import jakarta.servlet.RequestDispatcher;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.Scanner;

import com.cms.bean.User;
import com.cms.dao.CmsDao;

/**
 * Servlet implementation class LmsController1
 */
public class CmsController extends HttpServlet {

    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public CmsController() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        response.getWriter().append("Served at: ").append(request.getContextPath());
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
}
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

    String action = request.getParameter("button");

    PrintWriter out = response.getWriter();

    CmsDao dao = new CmsDao();

    if (action.equals("Insert")) {

        User e = new
        User(Integer.parseInt(request.getParameter("uid")),request.getParameter("uname"),Integer.parseInt(request.getParameter("cid")),request.getParameter("uadd"),request.g
        etParameter("dest"),request.getParameter("recname"));

        int n=0;

        try {

            n = dao.doInsert(e);

        }

        catch(SQLException ex) {

            ex.printStackTrace();

        }

        if(n>0) {

            response.sendRedirect("InsertSuccess.jsp");

        }

        else {

            response.sendRedirect("InsertFailure.jsp");

        }

    }

    else if(action.equals("Update")) {

        User up = new
        User(Integer.parseInt(request.getParameter("uid")),request.getParameter("uname"),Integer.parseInt(request.getParameter("cid")),request.getParameter("uadd"),request.g
        etParameter("dest"),request.getParameter("recname"));

        int n=0;

        try {

            n = dao.doUpdate(up);

        }

        catch(SQLException e) {

            e.printStackTrace();

        }

        if(n>0) {

            response.sendRedirect("UpdateSuccess.jsp");

        }

        else {
```

```
response.sendRedirect("UpdateFailure.jsp");
}
}

else if(action.equals("Delete")) {
int n=0;
try {
n = dao.doDelete(Integer.parseInt(request.getParameter("uid")));
} catch (SQLException e) {
e.printStackTrace();
}
if(n>0) {
response.sendRedirect("DeleteSuccess.jsp");
}
else {
response.sendRedirect("DeleteFailure.jsp");

}
}

else if(action.equals("Find")) {
LinkedList<User> list= new LinkedList<>();
int id = Integer.parseInt(request.getParameter("uid"));
try {
list = dao.doFind(id);
}
catch(Exception e) {
e.printStackTrace();
}
if(!list.isEmpty()) {
RequestDispatcher rd= request.getRequestDispatcher("FindSuccess.jsp");
request.setAttribute("list", list);
rd.forward(request, response);
}
else{
RequestDispatcher rd= request.getRequestDispatcher("FindFailure.jsp");
rd.forward(request, response);
}
}

else if (action.equals("ViewAll")) {
RequestDispatcher rd = request.getRequestDispatcher("ViewSuccess.jsp");
```

```
LinkedList<User> list = new LinkedList<>();

try {
list = dao.viewAll();
} catch (SQLException e) {
e.printStackTrace();
}

request.setAttribute("list", list);
rd.forward(request, response);

}

}

}
```

Com.cms.dao

CmsDao.java:

package com.cms.dao;

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.util.LinkedList;
```

```
import com.cms.bean.User;
import com.cms.util.CmsDBUtil;
```

```
public class CmsDao {
public LinkedList viewAll() throws SQLException {
```

```
Connection con1 = CmsDBUtil.getConnection();
String sql = "select * from user";
PreparedStatement ps = con1.prepareStatement(sql);
ResultSet rs = ps.executeQuery();
```

```
LinkedList<User> list = new LinkedList<User>();
while(rs.next()) {
User e = new User();
e.setUid(rs.getInt(1));
e.setUname(rs.getString(2));
e.setCid(rs.getInt(3));
```

```
e.setUadd(rs.getString(4));  
e.setDesti(rs.getString(5));  
e.setRec_name(rs.getString(6));
```

```
list.add(e);  
}  
return list;  
}
```

```
public int doInsert(User e) throws SQLException {  
    Connection con1 = CmsDButil.getConnection();  
    String sql = "insert into user values(?,?,?, ?, ?, ?)";
```

```
    PreparedStatement ps = con1.prepareStatement(sql);
```

```
    ps.setInt(1, e.getUid());  
    ps.setString(2, e.getUname());  
    ps.setInt(3, e.getCid());  
    ps.setString(4, e.getUadd());  
    ps.setString(5, e.getDesti());  
    ps.setString(6, e.getRec_name());
```

```
    int n=ps.executeUpdate();  
    con1.close();  
    return n;  
}
```

```
public int doUpdate    (User e) throws SQLException {  
    Connection con1 = CmsDButil.getConnection();  
    String sql = "Update user set uname=?,courier_id=?,user_address=?,destination_address=?,receiver_name=? where uid=?";  
    PreparedStatement ps = con1.prepareStatement(sql);  
    ps.setString(1, e.getUname());  
    ps.setInt(2, e.getCid());  
    ps.setString(3, e.getUadd());  
    ps.setString(4, e.getDesti());  
    ps.setString(5, e.getRec_name());  
    ps.setInt(6, e.getUid());  
    int n=ps.executeUpdate();  
    con1.close();  
    return n;  
}
```

```
public int doDelete    (int id) throws SQLException {  
    Connection con1 = CmsDBUtil.getConnection();  
    String sql = "delete from user where uid=?";  
    PreparedStatement ps = con1.prepareStatement(sql);  
    ps.setInt(1, id);  
    int n=ps.executeUpdate();  
    con1.close();  
    return n;  
}
```

```
public LinkedList<User> doFind (int id) throws SQLException {  
    Connection con1 = CmsDBUtil.getConnection();  
    String sql = "select * from user where uid=?";  
    PreparedStatement ps = con1.prepareStatement(sql);  
    ps.setInt(1, id);  
    ResultSet rs = ps.executeQuery();  
    LinkedList<User> list = new LinkedList<User>();  
    while(rs.next()) {  
        User u = new User();  
        u.setUid(rs.getInt(1));  
        u.setUname(rs.getString(2));  
        u.setCid(rs.getInt(3));  
        u.setUadd(rs.getString(4));  
        u.setDesti(rs.getString(5));  
        u.setRec_name(rs.getString(6));  
  
        list.add(u);  
    }  
    return list;  
  
}  
}
```

com.cms.util:

CmsDBUtil.java:

```
package com.cms.util;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
import java.sql.SQLException;
```

```
public class CmsDBUtil {
```

```
public static Connection getConnection() throws SQLException{
    final String URL="jdbc:mysql://localhost:3306/courier";
    final String user="root";
    final String pass="root";
    Connection con = DriverManager.getConnection(URL,user,pass);
    return con;
}
}
```

About.jsp:

[illegible]

- Planning and optimizing delivery routes
- Courier tracking and scheduling
- Analyzing courier performance
- Package tracking during delivery service
- Updating customers

Manual planning and routing are time-consuming processes. It's hard to isolate the optimal route across cities with thousands of different roads. It's nearly impossible to do this well when you need to calculate hundreds or thousands of stops. This is known as the traveling salesman problem and it's what courier management systems set out to solve.


```
</p>
</div>
```

```
</div>
</div>
</body>
</html>
```

DELETE.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@include file="Navbar.jsp" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3LHB60NNkmXc5s9fDVZLESaAA55NDz0xhy9GkcIdsLK1eN7N6jIeHz" crossorigin="anonymous"></script>
<meta charset="ISO-8859-1">
<style>
h1{
color:white
}
body{
background-image: linear-gradient(#0f0c29,#302b63,#24243e);
background-repeat:no-repeat;
background-size: cover;
height:440px;
background-attachment: fixed;
}

#b1:hover,#b2:hover,#b3:hover,#b4:hover,#b5:hover{
transform:scale(1.01);
}
@keyframes ani{
from{
opacity:0;
}
to{
opacity:1;
}
}
}
#cont{
animation-name: ani;
animation-duration: 3s;
animation-timing-function: ease;
animation-direction: forwards;
animation-iteration-count: 1;
}
</style>
<script>

function check(){
var no = document.forms["Delform"]["uid"].value;
if(no === ""){
alert("ID is required!!!");
event.preventDefault();
}
}
</script>
<title>Insert title here</title>
</head>
<body>
<form action="CmsController" class="container" name="Delform" onsubmit="return check();" method="post">
<h1 align="center" class="display-4" >Removing Details</h1>
<hr>
<table class="table text-center table-light">
<tr>
<td>
Enter the User ID to delete:
</td>
<td>
<input type="number" name="uid">
</td>
</tr>
</table>
<div class="text-center">
<input type="submit" class="btn btn-danger" name="button" value="Delete">
</div>
</form>
</body>
</html>
```

DeleteFailure.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@include file="Delete.jsp" %>
<!DOCTYPE html>
<html>
```

```
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1 align="center"><font class="display-4" color=red>Record Deletion failed!!!</font></h1>
</body>
</html>
```

DeleteSuccess.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@include file="ViewALL.jsp" %>

<!DOCTYPE html>
<html>
<head>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3LHB60NNkmXc5s9fDVZLESaAA55NDz0xhy9GkcIdsLK1eN7N6jIeHz" crossorigin="anonymous"></script>
<meta charset="ISO-8859-1">
<style>
h1{
color:white
}
body{
background-image: linear-gradient(#0f0c29,#302b63,#24243e);
background-repeat:no-repeat;
background-size: cover;
height:440px;
background-attachment: fixed;
}

#b1:hover,#b2:hover,#b3:hover,#b4:hover,#b5:hover{
transform:scale(1.01);
}
@keyframes ani{
from{
opacity:0;
}
to{
opacity:1;
}
}
#cont{
animation-name: ani;
animation-duration: 3s;
animation-timing-function: ease;
animation-direction: forwards;
animation-iteration-count: 1;
}
</style>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1 align="center" class="display-4"><font color=green>Record Deleted Successfully!!!</font></h1>
</body>
</html>
```

FIND.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@include file="Navbar.jsp" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3LHB60NNkmXc5s9fDVZLESaAA55NDz0xhy9GkcIdsLK1eN7N6jIeHz" crossorigin="anonymous"></script>
<meta charset="ISO-8859-1">
<style>
h1{
color:white
}
body{
background-image: linear-gradient(#0f0c29,#302b63,#24243e);
background-repeat:no-repeat;
background-size: cover;
height:440px;
background-attachment: fixed;
}

#b1:hover,#b2:hover,#b3:hover,#b4:hover,#b5:hover{
transform:scale(1.01);
}
@keyframes ani{
from{
opacity:0;
}
}
```

```
to{
opacity:1;
}
}
#cont{
animation-name: ani;
animation-duration: 3s;
animation-timing-function: ease;
animation-direction: forwards;
animation-iteration-count: 1;
}
</style>
<script>
function check(){
var id = document.forms["Fform"]["uid"].value;
if(id === "" || id === " " || id === null){
alert("ID is required!!!");
event.preventDefault();
}
}
</script>
</head>

<body>
<form action="CmsController" name="Fform" onsubmit="return check();" class="container" method="post">
<h1 align="center" class="display-4">Find Courier Details</h1>
<hr>
<table class="table text-center table-light">
<tr>
<td>
Enter the User ID to view details:
</td>
<td>
<input type="text" name="uid">
</td>
</tr>
</table>
<div class="text-center">
<input type="submit" class="btn btn-info" name="button" value="Find">
</div>
</form>
</body>
</html>
```

Findfailure.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@include file="Find.jsp" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1 align="center"><font color=red class="display-4">Record not found!!!</font></h1>
</body>
</html>
```

FindSuccess.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@page import="com.cms.bean.User,java.util.*" %>
<%@include file="Find.jsp" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhJY6hW+ALEwIH" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3LHB60NNkMc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>
<meta charset="ISO-8859-1">
<style>
h1{
color:white
}
body{
background-image: linear-gradient(#0f0c29,#302b63,#24243e);
background-repeat:no-repeat;
background-size: cover;
height:440px;
background-attachment: fixed;
}

#b1:hover,#b2:hover,#b3:hover,#b4:hover,#b5:hover{
transform:scale(1.01);
}
@keyframes ani{
from{
opacity:0;
```

```
}
to{
opacity:1;
}
}
#cont{
animation-name: ani;
animation-duration: 3s;
animation-timing-function: ease;
animation-direction: forwards;
animation-iteration-count: 1;
}
</style>
</head>
<body>
<form class="container">
<%
LinkedList<User> list = (LinkedList<User>) request.getAttribute("list");

out.println("<h1 align=center class=display-4>User Details</h1><hr>");

out.println("<center><table border=1 class=table text-center container table-dark><tr><th>User ID</th><th>User Name</th><th>Courier ID</th><th>User Address</th><th>Destination Address</th><th>Receiver Name</th></tr>");
for(User e : list){
out.print("<tr>");
out.print("<td>"+e.getId()+"</td><td>"+e.getUname()+"</td><td>"+e.getCid()+"</td><td>"+e.getUadd()+"</td><td>"+e.getDesti()+"</td><td>"+e.getRec_name()+"</td></tr>");
}
out.println("</table></center>");
%>
</form>
</body>
</html>
```

Insert.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@include file="Navbar.jsp" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3LHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>

<style>
h1{
color:white;
}
body{
background-image: linear-gradient(#0f0c29,#302b63,#24243e);
background-repeat:no-repeat;
background-size: cover;
background-attachment: fixed;
}
@keyframes ani{
from{
opacity:0;
}
to{
opacity:1;
}
}
td{
color:white;}
a{
text-decoration:none;
color:white;
}
a:hover{
transform:scale(1.01);
}
</style>
<script>

function check(){
var uid = document.forms["Insform"]["uid"].value;
var uname = document.forms["Insform"]["uname"].value;
var cid = document.forms["Insform"]["cid"].value;
var uadd = document.forms["Insform"]["uadd"].value;
var dest = document.forms["Insform"]["dest"].value;
var rname = document.forms["Insform"]["recname"].value;
if(uid === "" || uname === "" || cid === "" || uadd === "" || dest === "" || rname === ""){
alert("All fields are required!!!");
event.preventDefault();
}
}
</script>
</head>
<body>
<form action="CmsController" onsubmit="return check();" name="Insform" class="container" method="post">
<h1 class="display-4 table-head" align="center">Registration Form</h1>
<hr>
```

```
<table class="table text-center table-light">
<tr>
<td>
User ID:
</td>
<td>
<input type="number" name="uid">
</td>
</tr>
<tr>
<td>
User Name:
</td>
<td>
<input type="text" name="uname">
</td>
</tr>
<tr>
<td>
Courier ID:
</td>
<td>
<input type="number" name="cid">
</td>
</tr>
<tr>
<td>
User Address:
</td>
<td>
<input type="text" name="uadd">
</td>
</tr>
<tr>
<td>
Destination Address:
</td>
<td>
<input type="text" name="dest">
</td>
</tr>
<tr>
<td>
Receiver Name:
</td>
<td>
<input type="text" name="recname">
</td>
</tr>
</table>
<div class="text-center">
<input type="submit" onclick="check()" class="btn btn-success" name="button" value="Insert">
</div>
```

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

Login.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@include file="LoginNavbar.jsp" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
<title>Courier Management System</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3LHB60NNkMxc5s9fDVZLESaAA55NDz0xhy9GkcIdsLK1eN7N6jIeHz" crossorigin="anonymous"></script>
<style>
body{
background-image: linear-gradient(#0f0c29,#302b63,#24243e);
background-repeat:no-repeat;
background-size: cover;
background-attachment:fixed;
}
@keyframes ani{
from{
opacity:0;
}
to{
opacity:1;
}
}
}
body{
text-decoration:none;
color:white;
animation-name: ani;
animation-duration: 3s;
animation-timing-function: ease;
animation-direction: forwards;
animation-iteration-count: 1;
```



```
}
p{
text-decoration:none;
color:white;
animation-name: ani;
animation-duration: 2s;
animation-timing-function: ease;
animation-direction: forwards;
animation-iteration-count: infinite;
}
a:hover{
transform:scale(1.01);
}
h1,td{
color:white;
}
</style>

</head>
<body>
<form action="Welcome.jsp" name="Fform" onsubmit="return check();" class="container" method="post">
<h1 align="center" class="display-4">Login</h1>
<hr>
<table align="center">
<tr>
<td class="display-6">
Username<br>
</td>
</tr>
<tr>
<td id="in">
<input type="text" name="uname"><br>
</td>
</tr>
<tr>
<td class="display-6">
Password<br>
</td>
</tr>
<tr>
<td id="in">
<input type="password" name="pass"><br>
</td>
</tr>
<tr>
<td class="display-6">
<div class="text-center">
<input type="submit" class="btn btn-success" name="button" value="Login">
</div>
</form>
</body>
</html>
```

Navbar.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3LHB60NNkmXc5s9fDVZLESaAA55NDz0xhy9GkcIdsLK1eN7N6jIeHz" crossorigin="anonymous"></script>
</head>
<body>
<nav class="navbar navbar-expand-lg bg-transparent">
<div class="container-fluid">
<a style="color:white" class="navbar-brand" href="Welcome.jsp">Courier Management System</a>
<button style="background-color:white;" class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse justify-content-end" id="navbarNavAltMarkup">
<div class="navbar-nav">
<a style="color:white" class="nav-link active" aria-current="page" href="Track.jsp">Track Courier</a>
<a style="color:white" class="nav-link " href="Insert.jsp">Insert</a>
<a style="color:white" class="nav-link " href="Update.jsp">Update</a>
<a style="color:white" class="nav-link " href="Delete.jsp">Delete</a>
<a style="color:white" class="nav-link " href="Find.jsp">Find</a>
<a style="color:white" class="nav-link " href="ViewALL.jsp">View All</a>
<a style="color:white" class="nav-link " href="About.jsp">About Us</a>
</div>
</div>
</div>
</nav>
</body>
</html>
```

Update.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@include file="Navbar.jsp" %>
```

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhJY6hW+ALEwIH" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3LHB60NNkmXc5s9fDVZLESaAA55NDz0xhy9GkcIdsLK1eN7N6jIeHz" crossorigin="anonymous"></script>
<meta charset="ISO-8859-1">
<script>

function check(){
var uid = document.forms["Uform"]["uid"].value;
var uname = document.forms["Uform"]["uname"].value;
var cid = document.forms["Uform"]["cid"].value;
var uadd = document.forms["Uform"]["uadd"].value;
var dest = document.forms["Uform"]["dest"].value;
var rname = document.forms["Uform"]["recname"].value;
if(uid === "" || uname === "" || cid === "" || uadd === "" || dest === "" || recname === ""){
alert("All fields are required!!!");
event.preventDefault();
}
}
</script>
<style>
td,h1{
color:white
}
body{
background-image: linear-gradient(#0f0c29,#302b63,#24243e);
background-repeat:no-repeat;
background-size: cover;
height:440px;
background-attachment: fixed;
}

#b1:hover,#b2:hover,#b3:hover,#b4:hover,#b5:hover{
transform:scale(1.01);
}
@keyframes ani{
from{
opacity:0;
}
to{
opacity:1;
}
}
#cont{
animation-name: ani;
animation-duration: 3s;
animation-timing-function: ease;
animation-direction: forwards;
animation-iteration-count: 1;
}
</style>
</head>
<body>

<form action="CmsController" name="Uform" onsubmit="return check();" class="container" method="post">
<h1 align="center" class="display-4">Updation Form</h1>
<hr>
<table class="table text-center table-light">
<tr>
<td>
User ID:
</td>
<td>
<input type="number" name="uid">
</td>
</tr>
<tr>
<td>
User Name:
</td>
<td>
<input type="text" name="uname">
</td>
</tr>
<tr>
<td>
Courier ID:
</td>
<td>
<input type="number" name="cid">
</td>
</tr>
<tr>
<td>
User Address:
</td>
<td>
<input type="text" name="uadd">
</td>
</tr>
<tr>
<td>
```

```
Destination Address:
</td>
<td>
<input type="text" name="dest">
</td>
</tr>
<tr>
<td>
Receiver Name:
</td>
<td>
<input type="text" name="recname">
</td>
</tr>
</table>
<div class="text-center">
<input type="submit" class="btn btn-warning" name="button" value="Update">
</div>
</form>
</body>
</html>
```

View All.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1" import="com.cms.dao.CmsDao,java.sql.ResultSet,java.util.LinkedList,com.cms.bean.User"%>

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNLYT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3LHB60NNkmXc5s9fDVZLESaAA55NDz0xhy9GkcIdsLK1eN7N6jIeHz" crossorigin="anonymous"></script>
<style>
h1{
color:white
}
body{
background-image: linear-gradient(#0f0c29,#302b63,#24243e);
background-repeat:no-repeat;
background-size: cover;
height:440px;
background-attachment: fixed;
}

#b1:hover,#b2:hover,#b3:hover,#b4:hover,#b5:hover{
transform:scale(1.01);
}
@keyframes ani{
from{
opacity:0;
}
to{
opacity:1;
}
}
#cont{
animation-name: ani;
animation-duration: 3s;
animation-timing-function: ease;
animation-direction: forwards;
animation-iteration-count: 1;
}
</style>
</head>
<body>
<a href="Welcome.jsp" style="color:white; text-decoration:none;"><b>Go Home</b></a>
<h1 align="center" class="display-4 text-center">User Details</h1>
<hr>
<form class="container">
<%
CmsDao dao = new CmsDao();
int c=0;
try{
LinkedList<User> list = dao.viewAll();
out.println("<center><table border=1 class=table text-center container table-dark><tr><th>User ID</th><th>User Name</th><th>Courier ID</th><th>User Address</th><th>Destination Address</th><th>Receiver Name</th></tr>");
for(User e : list){
out.print("<tr>");
out.print("<td>" + e.getId() + "</td><td>" + e.getUname() + "</td><td>" + e.getCid() + "</td><td>" + e.getUadd() + "</td><td>" + e.getDesti() + "</td><td>" + e.getRec_name() + "</td></tr>");
}
out.println("</table></center>");
}
catch(Exception e){
e.printStackTrace();
}
if(c<=0){
out.println("No Records Found!!!");
}
}%>
</form>
</body>
</html>
```


Welcome.jsp:

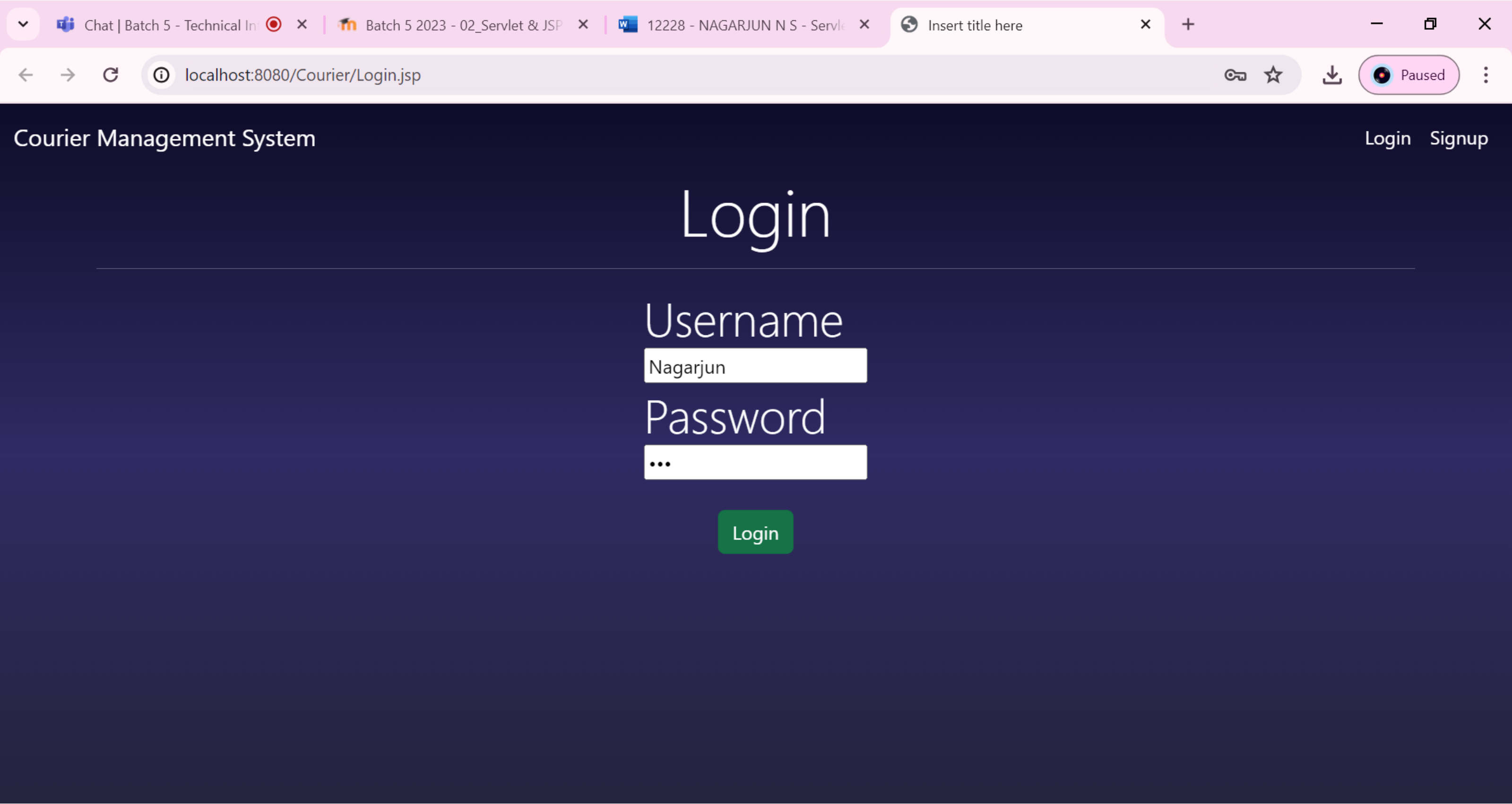
```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@include file="Navbar.jsp" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Courier Management System</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3LHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>
<style>
body{
background-image: linear-gradient(#0f0c29,#302b63,#24243e);
background-repeat:no-repeat;
background-size: cover;
background-attachment:fixed;
}
#content{
position: fixed;
inset: 0;
width: fit-content;
height: fit-content;
margin: auto;
}

@keyframes ani{
from{
opacity:0;
}
to{
opacity:1;
}
}
a{
text-decoration:none;
color:white;
animation-name: ani;
animation-duration: 3s;
animation-timing-function: ease;
animation-direction: forwards;
animation-iteration-count: 1;
}
p{
text-decoration:none;
color:white;
animation-name: ani;
animation-duration: 2s;
animation-timing-function: ease;
animation-direction: forwards;
animation-iteration-count: infinite;
}
a:hover{
transform:scale(1.01);
}
</style>
</head>
<body>
<div align="center" id="content">

<a href="About.jsp"><h1 class="display-2 text-center">Courier Management System</h1></a>
<p class="fs-5 text-bold text-center">Delivering Excellence, Every Mile!</p>
</div>
</body>
</html>
```

SCREENSHOTS:

Login Page:



HOMEPAGE:

Chat | Batch 5 - Techni

Batch 5 2023 - 02_Servle

12228 - NAGARJUN N S

Insert title here

Insert title here

+

-

localhost:8080/Courier/Welcome.jsp

☆

↓

Paused

Courier Management System

Track CourierInsertUpdateDeleteFindView AllAbout Us

Courier Management System

Delivering Excellence, Every Mile!

INSERT:

Chat | Batch 5 - Technical In

Batch 5 2023 - 02_Servlet & JSP

12228 - NAGARJUN N S - Servi

Insert title here

+

-

localhost:8080/Courier/Insert.jsp

☆

↓

Paused

Courier Management System

Track CourierInsertUpdateDeleteFindView AllAbout Us

Registration Form

User ID:	2
User Name:	Senathipathi
Courier ID:	1234
User Address:	Coimbatore
Destination Address:	Madurai
Receiver Name:	Nagarjun N S

Insert

Go Home

User Details

User ID	User Name	Courier ID	User Address	Destination Address	Receiver Name
1	Nagarjun N S	1200	Madurai	Trichy	Suresh Babu N R
2	Senathipathi	1234	Coimbatore	Madurai	Nagarjun N S

No Records Found!!!

Record Inserted!!!

UPDATE:

Courier Management System

Track CourierInsertUpdateDeleteFindView AllAbout Us

Updation Form

User ID:	2
User Name:	Sena
Courier ID:	12340
User Address:	Kovai
Destination Address:	Virudhunagar
Receiver Name:	Arjun

Update

Go Home

User Details

User ID	User Name	Courier ID	User Address	Destination Address	Receiver Name
1	Nagarjun N S	1200	Madurai	Trichy	Suresh Babu N R
2	Sena	12340	Kovai	Virudhunagar	Arjun

No Records Found!!!

Record Updated!!!

FIND:

Courier Management System

Track CourierInsertUpdateDeleteFindView AllAbout Us

Find Courier Details

Enter the User ID to view details:

2

Find

Courier Management System

Track CourierInsertUpdateDeleteFindView AllAbout Us

Find Courier Details

Enter the User ID to view details:

Find

User Details

User ID	User Name	Courier ID	User Address	Destination Address	Receiver Name
2	Sena	12340	Kovai	Virudhunagar	Arjun

DELETE:

Courier Management System

Track CourierInsertUpdateDeleteFindView AllAbout Us

Removing Details

Enter the User ID to delete:

2

Delete

Go Home

User Details

User ID	User Name	Courier ID	User Address	Destination Address	Receiver Name
1	Nagarjun N S	1200	Madurai	Trichy	Suresh Babu N R

No Records Found!!!

Record Deleted Successfully!!!

VIEW ALL:

Go Home

User Details


User ID	User Name	Courier ID	User Address	Destination Address	Receiver Name
1	Nagarjun N S	1200	Madurai	Trichy	Suresh Babu N R

No Records Found!!!

ABOUT US:

Courier Management System

Track CourierInsertUpdateDeleteFindView AllAbout Us



A courier delivery management system, or CMS, is business software that simplifies courier management and routing. A CMS streamlines all of the following tasks: Planning and optimizing delivery routes Courier tracking and scheduling Analyzing courier performance Package tracking during delivery service Updating customers Manual planning and routing are time-consuming processes. It's hard to isolate the optimal route across cities with thousands of different roads. It's nearly impossible to do this well when you need to calculate hundreds or thousands of stops. This is known as the traveling salesman problem and it's what courier management systems set out to solve.

VALIDATIONS:
INSERT PAGE:

localhost:8080/Courier/Insert.jsp

Courier Management System

er Insert Update Delete Find View All About Us

localhost:8080 says

All fields are required!!!

OK

User ID:	45
User Name:	Kannan
Courier ID:	9894
User Address:	Tanjore
Destination Address:	Madurai
Receiver Name:	

Insert

UPDATE PAGE:

localhost:8080/Courier/Update.jsp

Courier Management System

er Insert Update Delete Find View All About Us

localhost:8080 says

All fields are required!!!

OK

User ID:	1
User Name:	
Courier ID:	
User Address:	
Destination Address:	
Receiver Name:	

Update

DELETE PAGE:

▼

Chat | Batch 5 - Technical In

Batch 5 2023 - 02_Servlet & JSP

12228 - NAGARJUN N S - Servi

Insert title here

+

—

📄

✕

←

→

🔄

📄

localhost:8080/Courier/Delete.jsp

☆

📄

Paused

⋮

Courier Management System

er Insert Update Delete Find View All About Us

localhost:8080 says
ID is required!!!

OK

Enter the User ID to delete:

Delete

FIND PAGE:

▼

Chat | Batch 5 - Technical In

Batch 5 2023 - 02_Servlet & JSP

12228 - NAGARJUN N S - Servi

Insert title here

+

—

📄

✕

←

→

🔄

📄

localhost:8080/Courier/Find.jsp

☆

📄

Paused

⋮

Courier Management System

er Insert Update Delete Find View All About Us

localhost:8080 says
ID is required!!!

OK

Enter the User ID to view details:

Find