

EMPLOYEE TRANSPORT MANAGEMENT SYSTEM

Submitted in partial fulfillment of the requirements for
The award of
Bachelor of Engineering degree in Computer Science and Engineering

by

MALLIDI VEERESH (Reg. No. 37110422)

BURAGA BALAJI (Reg. No. 37110114)



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI
SALAI, CHENNAI – 600 119**

MARCH - 2021



SATHYABAMA



**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with “A” grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **MALLIDI VEERESH (Reg. No. 37110422)** and **BURAGA BALAJI (Reg. No.37110114)** who carried out the project entitled “**EMPLOYEE TRANSPORT MANAGEMENT SYSTEM** ” under my supervision from **August 2020** to **March 2021**.

Internal Guide

Ms. Ramya.G. Franklin, M.E.,

Head of the Department

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I **Mallidi Veeresh** hereby declare that the Project Report entitled “**EMPLOYEE TRANSPORT MANAGEMENT SYSTEM**” is done by me under the guidance of **Ms. Ramya.G. Franklin, M.E.**, Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D.,** and **Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms. Ramya.G. Franklin, M.E., Assistant Professor**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Transportation of the employees of any Industry is a herculean and important task in the operations. Nowadays, transportation is also important from the security aspect for the women employees.

Our project is aimed at the efficient management of the employee's transportation. It creates a complete ecosystem between employee's and transport administration. In addition to meeting the transportation needs, our system also takes care of the security of the employees.

Pickups and drop-offs are an essential service an organization provides transport for employees. It will provide a smooth flow of services like planning, scheduling and operations. Admin can view bookings and manage time slots by computerized system. Categories of employees can plan their transport trip based on gender.

They need specific requirements based on employee's choice, providing safety of employees by user confirmation, GPS or by call checking using IVRS (Interactive voice response system) especially for women.

INDEX

S.NO	CONTENTS	PG.NO
	LIST OF FIGURES	viii
1. INTRODUCTION		1
1.1 INTRODUCTION		1
1.2 OUTLINE OF PROJECT		1
1.3 AIM AND SCOPE		2
2. AIM AND SCOPE		3
2.1 AIM OF THE PROJECT		3
2.1.1 OBJECTIVES		3
2.2 SCOPE		3
2.3 HOW IT WORKS		4
2.3.1 WORKING OF GPS		4
2.3.2 WORKING OF IVRS		5
3. SYSTEM DESIGN AND METHODOLOGY		6
3.1 EXISTING SYSTEM		6
3.2 PREVIOUS PROPOSED SYSTEM		6
3.3 FEASIBILITY STUDY		6
4. SOFTWARE DEVELOPMENT METHODOLOGY		8
4.1 SYSTEM ARCHITECTURE		8
4.2 UML DIAGRAMS		8
4.2.1 Use Case Diagram		9
4.2.2 Sequence Diagram		11
4.2.3 Collaboration Diagram		12
4.2.4 Class Diagram		13
4.3 MODULES		16
4.3.1 Admin Module		16
4.3.2 Employee Module		16
4.3.3 Driver Module		17

4.4 DATABASE DESIGN	18
5. SYSTEM IMPLEMENTATION	21
5.1 Algorithm	21
5.2 Source code	22
6. TESTING	37
6.1 Unit Testing	37
6.2 Integration Testing	37
6.3 Functional Testing	38
6.4 System Testing	38
6.4.1 White Box Testing	38
6.4.2 Black Box Testing	38
6.5 Acceptance Testing	39
7. EXPERIMENTAL RESULTS	40
8. CONCLUSION	43
9. REFERENCE	44

LIST OF FIGURES

Fig No	Figure Description	PG.NO
2.3.1	Geo locations	4
4.1	System Architecture	8
4.2.1	Use Case Diagram	11
4.2.2	Sequence Diagram	12
4.2.3	Collaboration Diagram	13
4.2.4	Class Diagram	15
4.3.1	Admin Module	16
4.3.2	Employee Module	16
4.3.3	Driver Module	17
7.1	Output of Home Navigation bar	40
7.2	Output of Login page Navigation bar	40
7.3	Output of Search Navigation bar	41
7.4	Output of Employee Navigation bar	41
7.5	Output of Vehicle Navigation bar	42
7.6	Output of Schedule Navigation bar	42

1. INTRODUCTION

1.1 INTRODUCTION

The objective of this application is to automate the details of transportation (pick and drop) services provided by an organization to its employees and to manage the related information in a convenient manner. The purpose is to design a system that allows one to manage the relevant information. This system allows the user view all information of services provided by the organization. The User can make a request to Administrator for cab.

This system allows the user could make update request to cab. This system allows the administrator to maintain and update all information of cab details. The cabs are assigned by administrator to user cab request. This system allows user to make complaint for service and the administrator take action on complaint. The information of cab request details of day user and regular user are maintained by administrator and the administrator can view all or selected information. The purpose to design the system that allows search and retrieve related data easily. In addition to this we also providing the security for the employees by means of GPS tracking, IVRS automated system call. User can also give complaint against services. The administrator gets user's feedback.

1.2 OUTLINE OF PROJECT

Vehicle Management System is a software application which fulfills the requirement of a typical management of details of vehicles, staff, expenses and the fleet and reports in a company along with Booking option. It provides the interface to users in a graphical way to manage the daily transactions. Also provides the Management Reports like Vehicle Request details, Vehicle Request Status, Vehicle fuel consumption, Vehicle Distance Records, Vehicle Repair and Maintenance Records and User feedback.

The systems that are available now a days are manual ones in which users are maintaining ledgers, books etc to store the information like vehicles details, Booking details, Vehicle requests and approvals, Reports Management etc.

1.3 AIM AND SCOPE

The aim of this application is to reduce the manual effort needed to manage transactions in a company. Application provides an interface to users to view the details like the Vehicle details, Vehicle Booking, Profile Management, Repair and Maintenance schedules, Expenses Details, Daily Reports and feedback. Increases the profitability of the company by reducing the cost to purchase different software for different tasks to be performed.

2. AIM AND SCOPE

2.1 AIM OF THE PROJECT

The aim of this project is to facilitate the transportation to the employees by providing facility to book cab. It especially benefit women who are working in night shifts.

2.1.1.1 OBJECTIVE

In metro cities, plenty of people work in dynamic shifts, meet clients, and project deadlines. Most of them waste a significant amount of time in finding public transport or waiting for their pickups during office commute. Standing in queues for a long time or having no clue of the arrival of pickups leads to their anxiety and worries. As the employees are the most important for every company to drive its growth, they should be worry-free during their office time. Only then their productivity will be improved.

2.2 SCOPE

Convenient transport and safety are few of the important concerns of employees who work in shifts. So it designed for a secure tracking and transportation system in place, organization can be assured of the safe travel of employees.

There are different types of shift work schedules. Employers may assign workers to a first shift, second shift, or third shift. They may schedule employees to work regularly during that shift, called a fixed shift, or to work rotating shifts.

This is what we often refer to as traditional hours. Individuals who have the first shift usually work from 9 a.m. to 5 p.m., but may alternatively be on the job from 8 a.m. to 4p.m. or 10 a.m. to 6 p.m.

Second Shift: Relieving first shift workers are second shift workers who often start at 4p.m.or 5 p.m. and clock out at midnight or 1 a.m. The third shift is commonly referred to as the graveyard or overnight shift. Employees arrive for work between 10 p.m. and midnight and leave between 6 a.m. and 8 a.m.

During the night shifts employees face many problems in transportation like no cab facility for them. To overcome this problem we implement this system.

A basis for remote vehicle monitoring is a small hardware device mounted inside the car/cab. The device is actually an On board Computer with GPS and wireless communication capabilities that transfers all relevant information from vehicle to the

fleet management center. It gives you straight answers to the questions like: where did the vehicle drive for the last, say, 10 days, where did it stop, how fast did it drive, was there over speeding moments, how many kilometers did it make, and much more.

2.3 HOW IT WORKS?

2.3.1 Working of GPS

The most basic use of GPS tracking is to keep track of where our vehicles are on the map .this enables you to see which vehicle is in motion and towards which direction. With GPS tracking technology, you can monitor vehicle speed, routes, engine start up and shut down, idling and routes. You can tell whether the vehicle is going to the assigned destination, the routes were taken, and when the vehicle is resting.



Fig 2.3.1 Geo locations

Shortest path algorithm is used to find the shortest distance between the source and destination. After the admin assign the cab to the employee according to the distance and location of the employee.

2.3.2 Working of IVRS

In addition to GPS we also provide the IVRS system. The IVR system automatically calls to employee whether he reached or not. If the employee reached their destinations then he send the message to the system. If the employee not reached then we take further actions.

3. SYSTEM DESIGN AND METHODOLOGY

3.1 EXISTING SYSTEM

The present systems organize the entire information in file and ledgers. It is difficult to search and retrieve relevant data, when required. It is difficult to view any particular information quickly. Lot of time is need to search the data. Often the data is mismanaged and this leads to loss of data It is difficult to track the vehicle details.

3.2 PREVIOUS PROPOSED SYSTEM

The proposed system is a web based application which maintains a centralized repository of all necessary information. This allows the users to access the information easily. The system allows tracking and managing all information through well-defined interfaces.

3.3 FEASIBILITY STUDY

Preliminary investigation exam in project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system.

There are aspects in the feasibility study portion of the preliminary investigation:

- a) Technical Feasibility
- b) Operational Feasibility

- **Technical Feasibility**

The technical issue usually raised during the feasibility stage of the investigation includes the following:

Does the necessary technology exist to do what is suggested?

Do the proposed equipment's have the technical capacity to hold the data required to use the new System?

Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?

Can the system be upgraded if developed?

Are the technical guarantees of accuracy, reliability, ease of access and data

security?

- **Operational Feasibility**

Proposed project are beneficial only if they can turned out into information system. That will be meet the organization's operating feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test

Operational feasibility project includes of the following:

Is the sufficient support for the management from the users?

Will the system be used and work properly if it is being developed and implemented?

Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the Management issues and user and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits.

4. SOFTWARE DEVELOPMENT METHODOLOGY

System design phase follows system analysis phase. Design is maintaining record proof design divisions and providing a blue print for the implementation phase. Design is the bridge between system analysis and system implementation. System design transition from user oriented, document oriented to programmers or database personnel. The design is a solution, a “how to” approach to the creation a new system. This is composed of several steps.

4.1 SYSTEM ARCHITECTURE

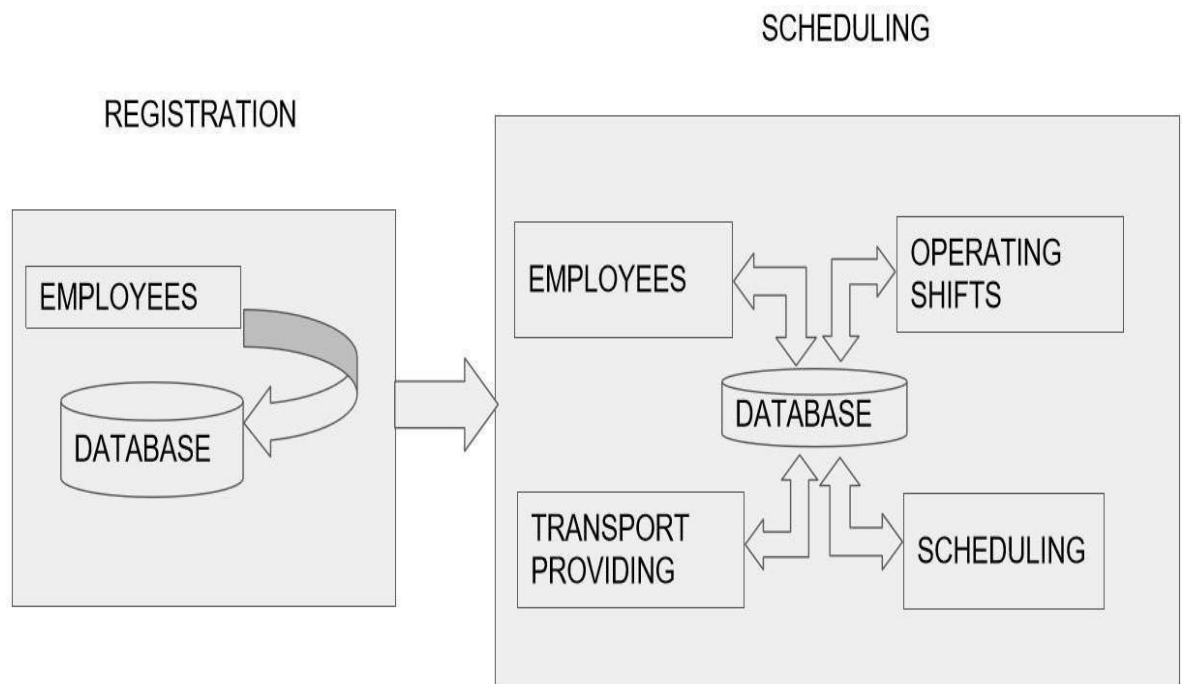


Fig 4.1 System architecture

4.2 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the ObjectManagement Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

4.2.1 Use case diagram

To model a system the most important is to capture the dynamic behavior. To clarify a bit in details, dynamic behavior means the behavior of the system

When it is running operating. In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Now to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction. These internal and external agents are known as actors. So, use case diagrams are consists of actors, use cases and their relationships. The diagram is used to model the system of an application.

PURPOSE OF USE CASE DIAGRAM

The purpose of use case diagram is to capture the dynamic aspect of a system. But this definition is too generic to describe the purpose. Use case diagrams are used to gather the requirements of a system including internal or external influences. Now when the initial task is complete use case diagrams are modeled to present the outside view.

- Used to gather requirements of a system.
- Used to get an outside view of a system.

- Identify internal and external factors influencing the system.
- Show the interacting among the requirements are actors.

Use case diagrams are considered for high level requirements analysis of a system. So when the requirements of a system are analyzed the functionalities are captured in use case manner.

Now the second things which are relevant to the use cases are the actors. Actors can be defined as something that interacts with the system. The actors can be human user. So in a brief when we are planning to draw a use case diagram, it should have the following items identified

- Functionalities to be represented as an use case
- Actors
- Relationships among the use cases and actors.

Use case diagrams are drawn to functional requirements of a system. So after identifying the above items the following guidelines to draw an efficient use case diagram.

- The name of a use case is very important. So the name should be chosen in such a way so that it can identify the functionalities performed.
- Do not try to include all types of relationships. Because the main purpose of the diagram is to identify requirements
- Use note whenever required to clarify some important points.

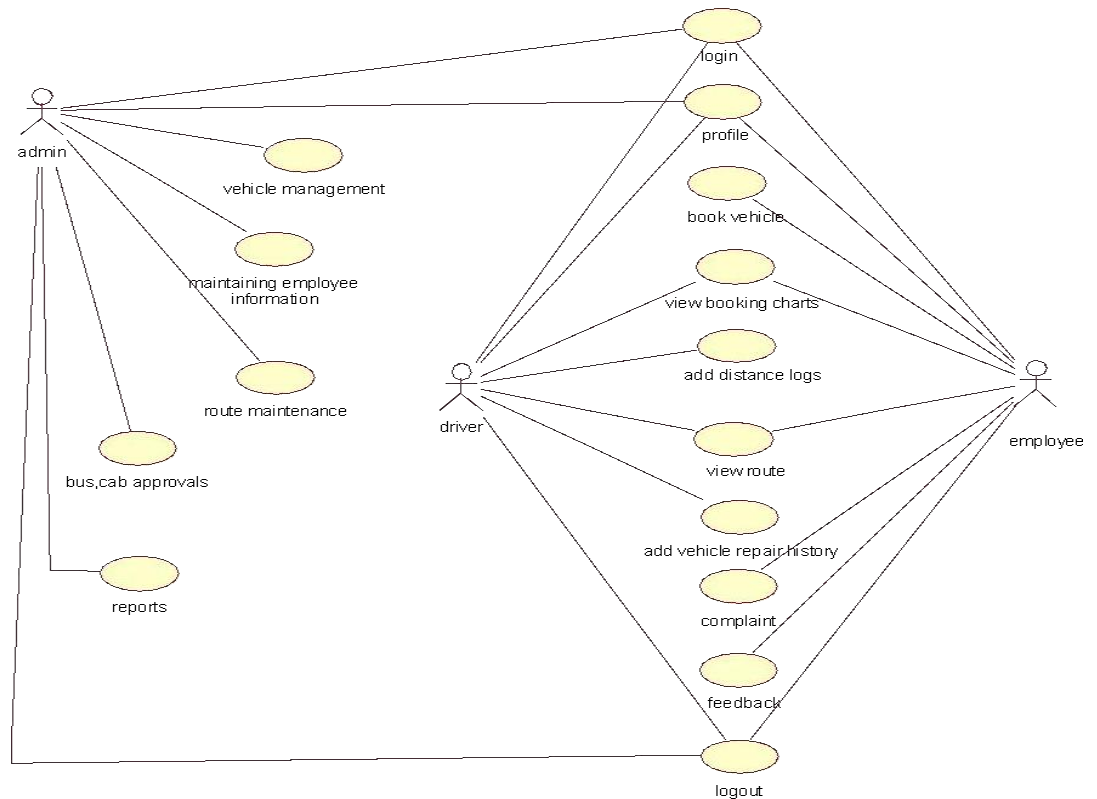


Figure 4.2.1 Use case Diagram

4.2.2 Sequence diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple run time scenarios in a graphical manner.

PURPOSE OF SEQUENCE DIAGRAM

- Model high-level interaction between active objects in a system
- Model the interaction between object instances within a collaboration that realizes a use case

- Model the interaction between objects within a collaboration that realizes an operation
- Either model generic interactions (showing all possible paths through the interaction) or specific instances of an interaction.

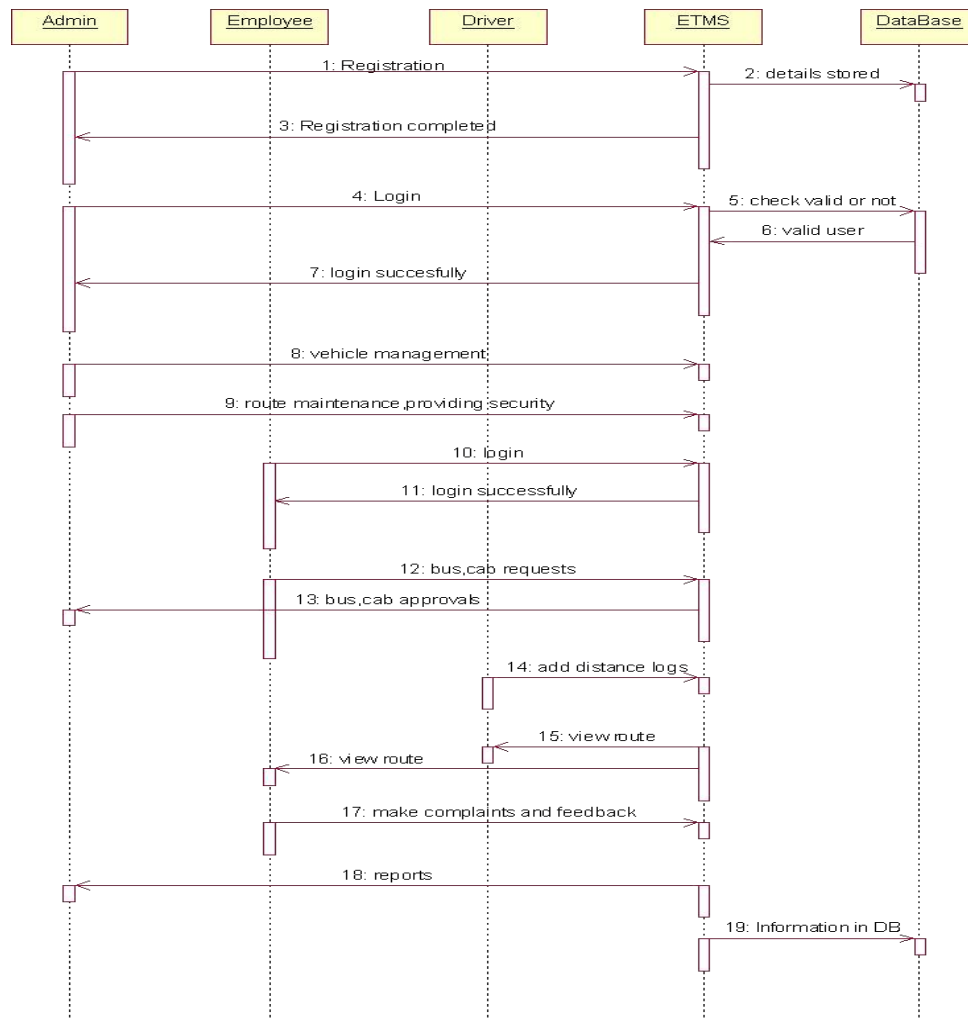


Figure 4.2.2 Sequence Diagram

4.2.3 Collaboration diagram

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the method are called one after another. We have taken the same order management system to describe the collaboration diagram

The method calls are similar to that of a sequence diagram. But the difference is that the sequence does not describe the object organization whereas the collaboration diagram shows the object organization

Now to choose between these two diagrams the main emphasis is given on the type of requirement. If the time sequence is important then sequence diagram is used and if organization is required then collaboration diagram is used.

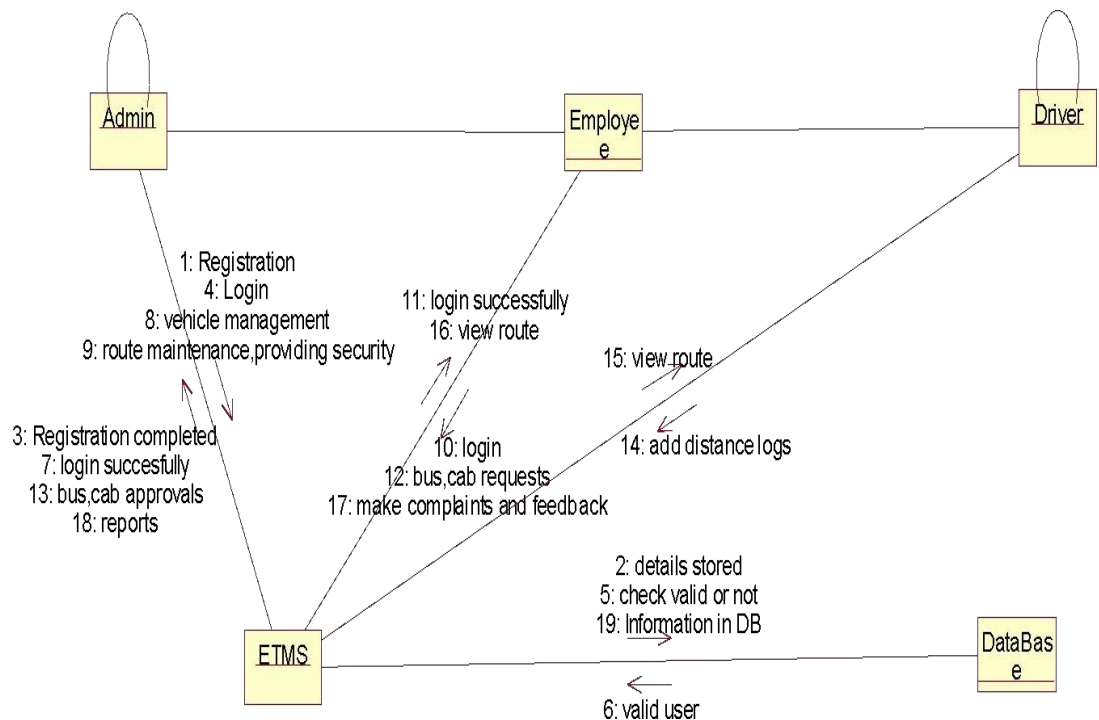


Fig 4.2.3 Collaboration Diagram

4.2.4 Class diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system.

The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

PURPOSE OF CLASS DIAGRAM

The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction. UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application, however class diagram is a bit different. It is the most popular UML diagram in the coder community.

The purpose of the class diagram can be summarized as –

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

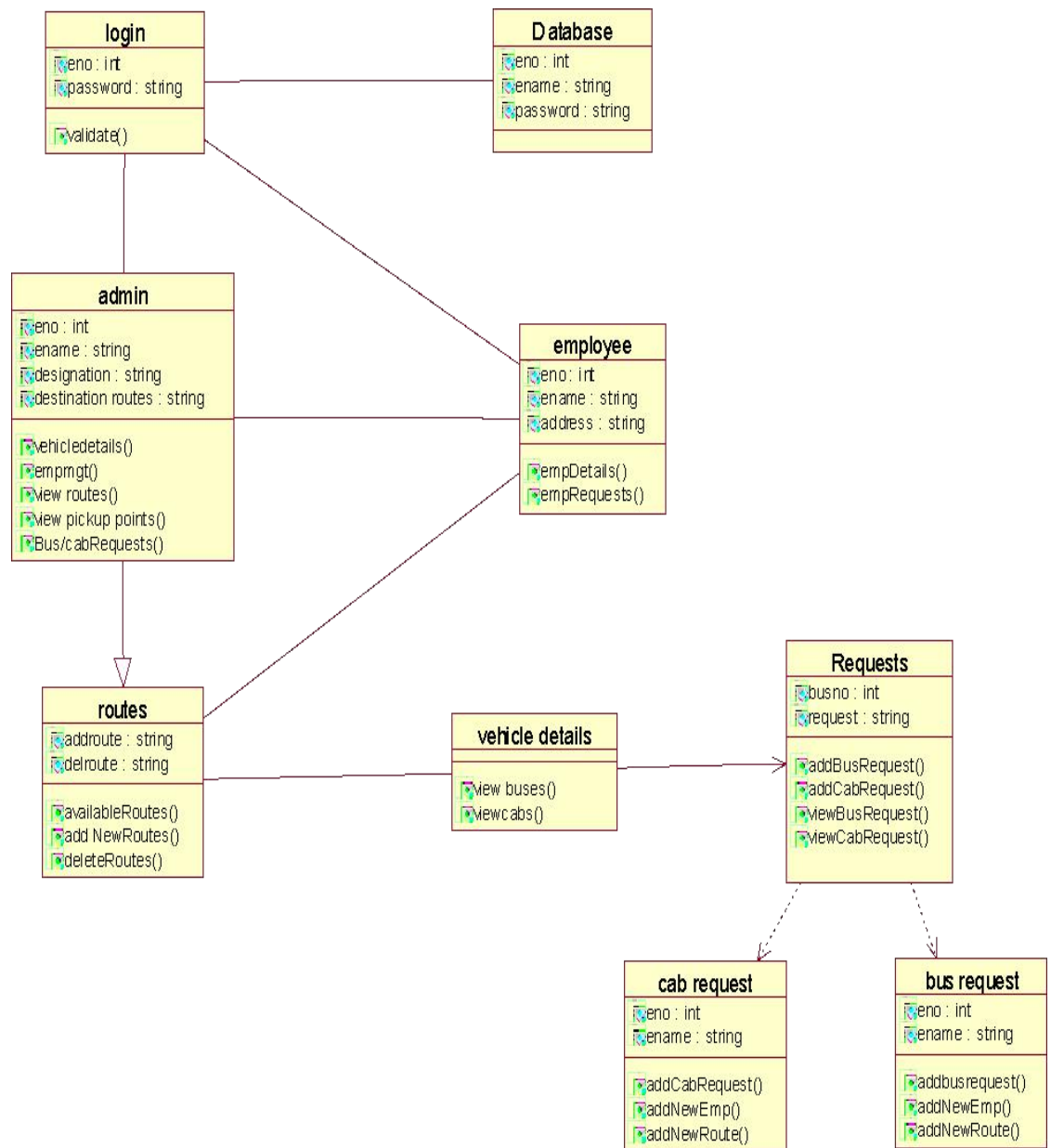


Fig 4.2.4 Class Diagram

4.3 MODULES

4.3.1 Admin module

View/Edit/Delete vehicle, employee details

Add new employee/driver

Assign the cabs

View booking charts and reports

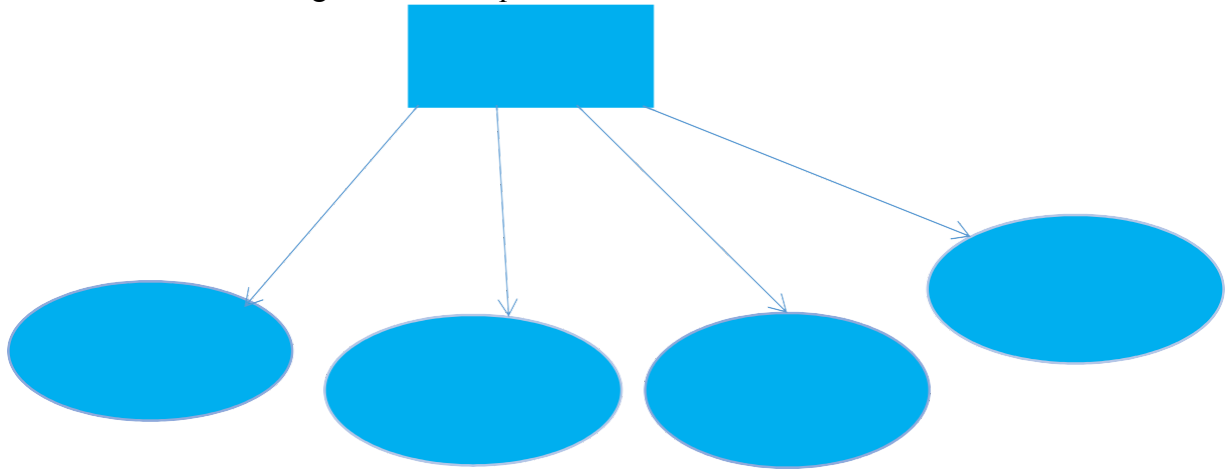


Fig. 4.3.1 Admin module

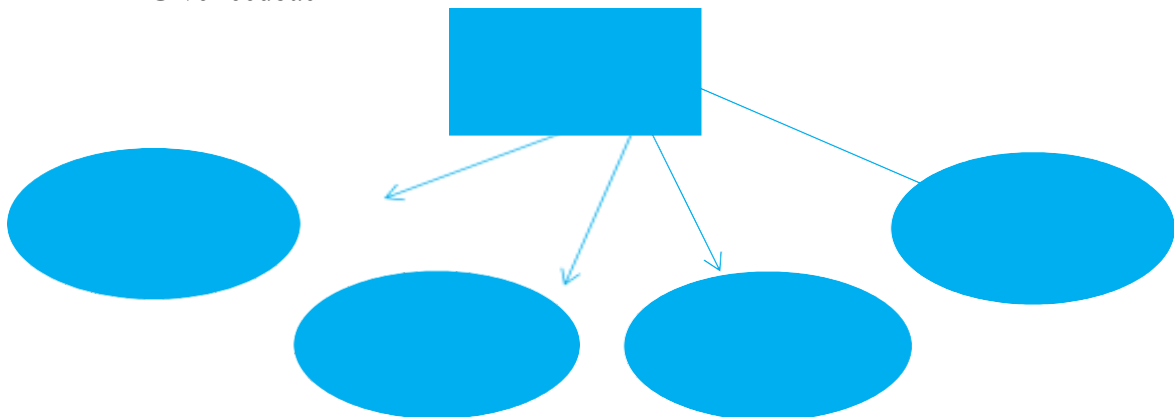
4.3.2 Employee module

View/Update Employee profile

Book vehicle

Make a compliant

Give feedback



4.3.3 Driver module

View booking charts

Add distance logs, vehicle repair history

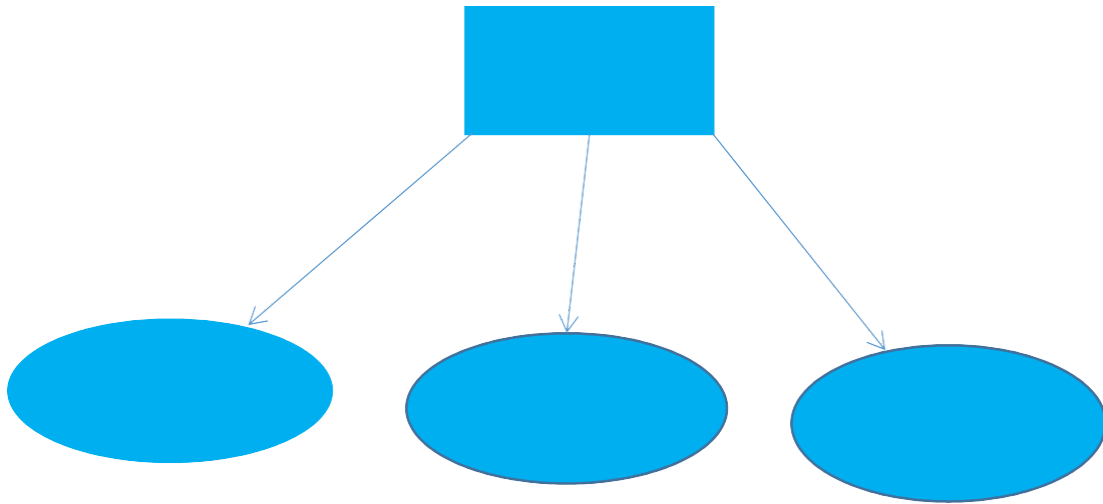


Fig 4.3.3 Driver module

4.4 DATABASE DESIGN

DATABASE TABLES

Employee table: The Employee database contains the Employee information such as Employee id, Gender and Employee address.

Field name	Data type	comments
Emp_id	number(10)	Primary key
Ename	varchar2(80)	Not null
Gender	Char(6)	Not null
DOB	DATE	Not null
Phno	number(10)	Not null
Joiningdate	DATE	Not null
Createdate	DATE	Not null
Address	varchar2(20)	Not null
Geolocation Latitude Longitude	number(10) number(10)	Not null Not null

Organization table: The Organization database contains the Organization information such as Organization id, Organization name and Organization location.

Field name	Data type	comments
Org_id	number(10)	Primary key
Orgname	varchar2(20)	Not null
Orgloc Latitude Longitude	number(10) number(10)	Not null Not null

Division table: The Division database contains the Division id and division location, number of employees in the division.

Field name	Data type	comments
Div_id	Number(10)	Primary key
Divname	Varchar2(20)	Not null
Divloc		
Latitude	Number(10	Not null
Longitude	)	Not null
	Number(10	
	)	
No.of_ems	Number(3)	Not null

Transportation request table: This database table contains the request id and employee id, vehicle type.

Field name	Data type	comments
Req_id	Number(3)	Primary key
Emp_id	Number(15)	Primary key
Fromloc	Varchar2(30)	Not null
Vehicle_type	Varchar2(10)	Not null
End_date	Varchar2(10)	Not null

Scheduled table: This database contains the vehicle_id, Route_id and beginning point of the employee.

Field name	Data type	comments
Vehicle_id	Number(10)	Primary key
Route_id	Number(10)	Primary key
Beginningpoint	Varchar2(20)	Not null

Route planning:

Routes table: In this database Route id, Path and Destination is stored.

Field name	Data type	comments
Route_id	Number(4)	Primary key
Routename	Varchar2(20)	Not null
Path	Varchar2(20)	Not null
Destination	Varchar2(20)	Not null

Vehicle table: In this database vehicle data is stored.

Field name	Data type	comments
Vehicleid	Number(10)	Primary key
Vehiclename	Varchar2(10)	Not null
Location	Varchar2(30)	Not null
Capacity	number(2)	Not null
Driverid	Number(10)	Primary key

Driver table: In this database driver details are stored.

Field name	Data type	comments
Driverid	Number(10)	Primary key
Drivername	Varchar2(20)	Not null
Vehicleid	Number(10)	Primary key

5. SYSTEM IMPLEMENTATION

5.1 ALGORITHM

Dijkstra's algorithm is used to find the shortest path from one node to another node in a graph. Dijkstra's algorithm is also known as a single source shortest path algorithm. It is applied only on positive weights. In Global Positioning System (GPS) is used for adding a new functionality in Dijkstra's algorithm. Using Global Positioning System (GPS) the position parameter is added in the Dijkstra's algorithm. From this current position is retrieved at any point. By using this current position, the distance can be determined from one node to another node.

Step 1: Consider the all positions on GPS as nodes. S is the starting node and n is the number of nodes.

Step 2: Finding the different possible paths from source to destination node and then goto step 3.

Step 3: By using shortest path algorithm (Dijkstra) we are finding the shortest path between the source locations to destination location from step 2.

Step 4: Apply the below Dijkstra

formula for start_node to end_node

do

for i=1 to end_node do

For j=1 to end_node do

$Dist[i,j] = \min \{dist[i,j], dist[i, end_node] + dist[end_node, j]\}$

done

Step 5: Stop the process.

5.2 Source code

```
//default.aspx
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.master"
AutoEventWireup="true"
    CodeBehind="Default.aspx.cs" Inherits="MapClusters._Default" %>
<asp:Content                ID="HeaderContent"
runat="server" ContentPlaceHolderID="HeadContent">
</asp:Content>
<asp:Content                ID="BodyContent"
runat="server" ContentPlaceHolderID="MainContent">
<h2>
    Welcome to Employee Transport Management System!
</h2>
<p>
    Transportation of the employees of any industry is a herculean and important
task in the operations.
Now a days,the transportation is also important from the security aspect for the
women employees.
</p>
<p>
Our project is aimed at the efficient management of the employee's transportation.
It creates a complete ecosystem between employee's and transport administration.
In addition to meeting the transportation needs, our system also takes care of the
security of the employees.</p>
<p>
Pickups and drop-offs are an essential service an organization provides transport
for employees. It will provide a smooth flow of services like planning, scheduling
and operations. Admin can view bookings and manage time slots by computerized
system. Categories of employees can plan their transport trip based on gender.
They need specific requirements based on employee's choice, providing safety of
employees by user confirmation, GPS or by call checking using IVRS (Interactive
Voice Response
System) especially for women.By using Routing Mapping, we can monitor the
```

system of current address.

</p>

</td>

<td>Gender


```

        <asp:Radiobutton id="rdoMale" checked="true"
GroupName="Gender" Text="Male" runat="server"></asp:Radiobutton>
        <asp:Radiobutton id="rdoFemale"
GroupName="Gender" Text="Female"
runat="server"></asp:Radiobutton>
    </td>
</tr>
<tr>
<td>Address-Street<br />
    <asp:TextBox id="txtAdd" runat="server"></asp:TextBox>

</td>
<td>City<br />
    <asp:TextBox id="txtCity" runat="server"></asp:TextBox>
</td>
    <td>Bus Route<br />
    <asp:TextBox id="txtBus" runat="server"></asp:TextBox>
</td>
</tr>
<tr>
<td>District<br />
    <asp:TextBox id="txtDist" runat="server"></asp:TextBox>

</td>
<td>State<br />
    <asp:TextBox id="txtState" runat="server"></asp:TextBox>
</td>
    <td>Latitude<br />
    <asp:TextBox id="txtLat" runat="server"></asp:TextBox>
</td>
</tr>
<tr>
<td>Pin Code<br />

```

<asp:TextBox id="txtPin" runat="server"></asp:TextBox>

[illegible]

```

<tr>
    <td colspan="2">
        <asp:GridView ID="dgEmp" runat="server"></asp:GridView>
    </td>
</tr>
</table>
</div>
</asp:Content>

```

```
//mapdisplay
```

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="MapDisplay.aspx.cs"
Inherits="MapClusters.MapDisplay" %>

```

```

<asp:Content ID="Content1"
ContentPlaceHolderID="HeadContent" runat="server">
    <link href="Styles/Site.css" rel="stylesheet" type="text/css" />
</asp:Content>

```

```

<asp:Content ID="Content2"
ContentPlaceHolderID="MainContent" runat="server">
    <div id="main">
        <div id="sidebar" style="width:30%;">
            <strong>Division<br />
            </strong>
            <asp:CheckBox ID="chkBR1" runat="server" Text="CSE" /><br />
            <asp:CheckBox ID="chkBR2" runat="server" Text="ECE" /><br />
            <asp:CheckBox ID="chkBR3" runat="server" Text="MEC" /><br />
            <br />
            <strong>Gender<br />
            </strong>
            <asp:CheckBox ID="chklt3k" runat="server" Text="Male" /><br />
            <asp:CheckBox ID="chklt5k" runat="server" Text="Female" /><br />
            <br />
            <strong>Route</strong>
            <br />

```

```

        <asp:CheckBox ID="chkIndHouse" runat="server" Text="Route #1" /><br
    />
    <asp:CheckBox ID="chkApartment" runat="server" Text="Route #2" /><br
    />
    <br />
    <strong>Shift</strong>
    <br />
    <asp:CheckBox ID="chkDowntown" runat="server" Text="Regular" /><br
    />
    <asp:CheckBox ID="chkMidTown" runat="server" Text="Morning" /><br
    />
    <asp:CheckBox ID="chkSuburban" runat="server" Text="Night" /><br />
    <br />
    <br />
</div>

<asp:TextBox
                    ID="txtSearch"
                    style="width:60%"
runat="server"></asp:TextBox>

<asp:Button ID="btnSubmit" runat="server"
    onclick="btnSubmit_Click" Text="Search" /><br />
<div id="map" style="width: 70%; height: 450px"></div>
</div>
<script type="text/javascript">
    var locations = [
<asp:Repeater ID="rptMarkers" runat="server" EnableViewState="false">
<ItemTemplate>{ regdno: '<%=#Eval("regdno")%>',
    lat: <%=#Eval("slat")%>,
    lng: <%=#Eval("slong")%>,
    name:
    "<%=#Eval("sname")%>",
    gender: '<%=#Eval("sgender")%>',
    branch:'<%=#Eval("sbranch")%>',
    year:'<%=#Eval("scuryear")%>',
    busroute:'<%=#Eval("sbusroute")%>'
}</ItemTemplate>

```

```
<SeparatorTemplate>,</SeparatorTemplate>  
</asp:Repeater>
```

```

];
function initMap() {

    //alert(locations.length);
    var map = new google.maps.Map(document.getElementById('map'), {
        zoom: 11,
        center: { lat: 16.55, lng: 81.52 }
    });
    // Create an array of alphabetical characters used to label the markers.
    var labels = 'MFMFMMFFMMFMFMFMFMFMFMFFMMMMFF'
    // Add some markers to the map.
    // Note: The code uses the JavaScript Array.prototype.map() method to
    // create an array of markers based on a given "locations" array.
    // The map() method here has nothing to do with the Google Maps
    API. var iconBase = '/images/';
    var markers = locations.map(function (location, i) {
        return new google.maps.Marker({
            //position: new google.maps.LatLng(locations[i].lat, locations[i].lng),
            position: location,
            label: location.gender,
            icon: iconBase+(location.gender === 'M' ? 'student.png'
: 'student.png')
        });
    });

    var i;
    for (i = 0; i < markers.length; i++)
        { addInfoWindow(markers[i],
locations[i].year+'-'+locations[i].branch+'-'+locations[i].name+'('+locations[i].gender+'')');
    }
    // Add a marker clusterer to manage the markers.
    var markerCluster = new MarkerClusterer(map, markers,

```

```

        {imagePath:
'https://developers.google.com/maps/documentation/javascript/examples/markercl
usterer/m' });
    }

```

```

function addInfoWindow(marker, message) {

    var infoWindow = new google.maps.InfoWindow({
        content: message
    });

    google.maps.event.addListener(marker, 'click', function () {
        infoWindow.open(map, marker);
    });
}

```

```

</script>
<script type="text/javascript"
src="https://developers.google.com/maps/documentation/javascript/examples/mar
kerclusterer/markerclusterer.js">
</script>
<script type="text/javascript" async defer

src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBX66lTtkMfxS7lJlAP
F_O5TNiE5UnxDtg&callback=initMap">
</script>
</asp:Content>

```

```

//mapreduce.aspx.c
s using System;
using
System.Collections.Generic;
using System.Linq;

```

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

```

using
MySQL.Data.MySqlClient;
using System.Data;
using System.Configuration;

namespace MapClusters
{
    public partial class MapDisplay : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            string strSql = "select * from employee
            "; populateMarkers(strSql);
        }

        private void populateMarkers(string strSql)
        {
            DataTable dtMarkers;
            dtMarkers = GetDataTable(strSql);
            rptMarkers.DataSource = dtMarkers;
            rptMarkers.DataBind();

        }

        private DataTable GetDataTable(string strQuery)
        {
            DataTable dt = new DataTable();
            string constr=
ConfigurationManager.ConnectionStrings["constr"].ConnectionString;

            using (MySqlConnection conn = new MySqlConnection(constr))
            {
                MySqlCommand cmd =
                conn.CreateCommand(); cmd.CommandText =

```

```
strQuery;
```

```
conn.Open();
```

```

        using (MySqlDataReader dr = cmd.ExecuteReader())
        {
            dt.Load(dr);
        }
    }
    return dt;
}

```

```

protected void btnSubmit_Click(object sender, EventArgs e)
{
    string strBranchSel =
    ""; string strGenderSel
    = "";
    string strSql = "select regdno,sname,sgender,sbranch,trim(slat)
slat,trim(slong) slong,sbustrans,sbusroute from employee where 1= 1 ";

```

```

    if (strBranchSel != "")
        strSql = strSql + " and sbranch in
"+strBranchSel; if (strGenderSel != "")
        strSql = strSql + " and sgender in
"+strGenderSel; populateMarkers(strSql);

```

```

    }
}
}
//dbhelper.aspx

```

```

using System;
using
System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;

```

```
using  
MySQL.Data.MySqlClient;  
using System.Configuration;
```

```
namespace DEOUtility
```

```

{
    public class DBHelper
    {
        public string constr=
ConfigurationManager.ConnectionStrings["constr"].ConnectionString;
        public MySqlConnection con;

        //public void openDBConnection()
        //{
        //    constr =
@"Server=dallaswin200.arvixeshared.com;Database=mysqlpdf;Uid=pradmysql;
P wd=pradmysql;";
        //    //constr =
@"Server=127.0.0.1;Database=test;Uid=pnadmin;Pwd=pnadmin;
";

        //    if (con == null)
        //        con = new MySqlConnection(constr);
        //    if (con.State == System.Data.ConnectionState.Closed) {
        //        con.Open();
        //    }

        //}

        public void closeDBConnection()
        {
            if (con.State == System.Data.ConnectionState.Open)
            {
                con.Close()
                ; con = null;
            }
        }

        public DataTable GetDataTable(string strSql)
        {

```

```
DataTable dt = new DataTable();
```

```

using (MySqlConnection conn = new MySqlConnection(constr))
{
    MySqlCommand cmd =
        conn.CreateCommand(); cmd.CommandText =
            strSql;

    conn.Open();

    using (MySqlDataReader dr = cmd.ExecuteReader())
    {
        dt.Load(dr);
    }
}

return(dt);
}
public string GetSelectString(string strSql)
{

    string strReturn = "";

    using (MySqlConnection conn = new MySqlConnection(constr))
    {

        conn.Open();
        using (MySqlCommand cmd = conn.CreateCommand())
        {

            cmd.CommandText = strSql;
            cmd.CommandType = CommandType.Text;
            MySqlDataReader rdr = cmd.ExecuteReader();
            if (rdr.Read())
            {

```

```
    strReturn = rdr[0].ToString();  
}
```

```

    }

}

return (strReturn);
}

public string GetSelectString(string strSql,string strCmdType,
MySQLParameter[] param)
{

    string strReturn = "";

    using (MySQLConnection conn = new MySQLConnection(constr))
    {

        conn.Open();
        using (MySQLCommand cmd = conn.CreateCommand())
        {

            cmd.CommandText = strSql;
            if (strCmdType=="Procedure")
                cmd.CommandType = CommandType.StoredProcedure;
            else
                cmd.CommandType = CommandType.Text;
            foreach (MySQLParameter msp in param)
                cmd.Parameters.Add(msp);
            MySQLDataReader rdr = cmd.ExecuteReader();
            if (rdr.Read())
            {
                strReturn = rdr[0].ToString();
            }
        }
    }
}

```

```

    }

    return (strReturn);
}
public int ExecuteCmd(string strSql)
{

    int retVal = 0;

    using (MySqlConnection conn = new MySqlConnection(constr))
    {

        conn.Open();
        using (MySqlCommand cmd = conn.CreateCommand())
        {

            cmd.CommandText = strSql;
            cmd.CommandType = CommandType.Text;
            retVal = cmd.ExecuteNonQuery();
        }

    }

    return (retVal);
}
public int ExecuteStoredProcParam(string strSql, MySqlParameter[] param)
{

    int retVal = 0;

    using (MySqlConnection conn = new MySqlConnection(constr))
    {

```

```

        conn.Open();
        using (MySqlCommand cmd = conn.CreateCommand())
        {
            cmd.CommandText = strSql;
            cmd.CommandType = CommandType.StoredProcedure;
            foreach (MySqlParameter msp in param)
                cmd.Parameters.Add(msp);

            retVal = cmd.ExecuteNonQuery();
        }
    }
    return (retVal);
}

public int ExecuteStoredProc(string strSql)
{
    int retVal = 0;
    using (MySqlConnection conn = new MySqlConnection(constr))
    {
        conn.Open();
        using (MySqlCommand cmd = conn.CreateCommand())
        {
            cmd.CommandText = strSql;
            cmd.CommandType = CommandType.StoredProcedure;
            retVal = cmd.ExecuteNonQuery();
        }
    }

    return (retVal);
}
}
}

```

6. TESTING

System testing is testing conducted on a complete integrated system to evaluate the system's compliance with its specified requirements. System testing takes, as its input, all of the integrated components that have passed integration testing. The purpose of integration testing is to detect any inconsistencies between the units that are integrated together called assemblages. System testing seeks to detect defects both within the "inter-assemblages" and also within the system as a whole. The actual result is the behavior produced or observed when a component or system is tested.

System testing is performed on the entire system in the context of either functional requirement specifications (FRS) or system requirement specification (SRS), or both. System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software or hardware requirements specification.

6.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that

arise from the combination of components.

6.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input: identified classes of valid input must be accepted.

Invalid Input: identified classes of invalid input must be rejected.

Functions: identified functions must be exercised.

Output: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

6.4 SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

6.4.1 White box testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

6.4.2 Black box testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document.

It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

6.5 ACCEPTANCE TESTING

In engineering and its various sub disciplines, acceptance testing is a test conducted to determine if the requirements of a specification are met. It involves performance tests and other type of tests. In systems engineering it may involve black-box testing performed on a system prior to its output. It is also known as functional testing, black-box testing, QA testing, application testing, confidence testing, or acceptance testing. Software developers often distinguish acceptance testing by the system provider from acceptance by the customer (the user or client) prior to acceptance transfer of ownership.

7. EXPERIMENTAL RESULTS

Our application is Employee transport management system. In our application Home, Search, Employee, Vehicles, Schedule navigation bars.

The following is the output for home navigation. It shows the agenda of our application.



Fig 7.1. Output of Home navigation bar

The following is the output for the Login page. It contains the username and password. If the user does not exist then he/she must register.

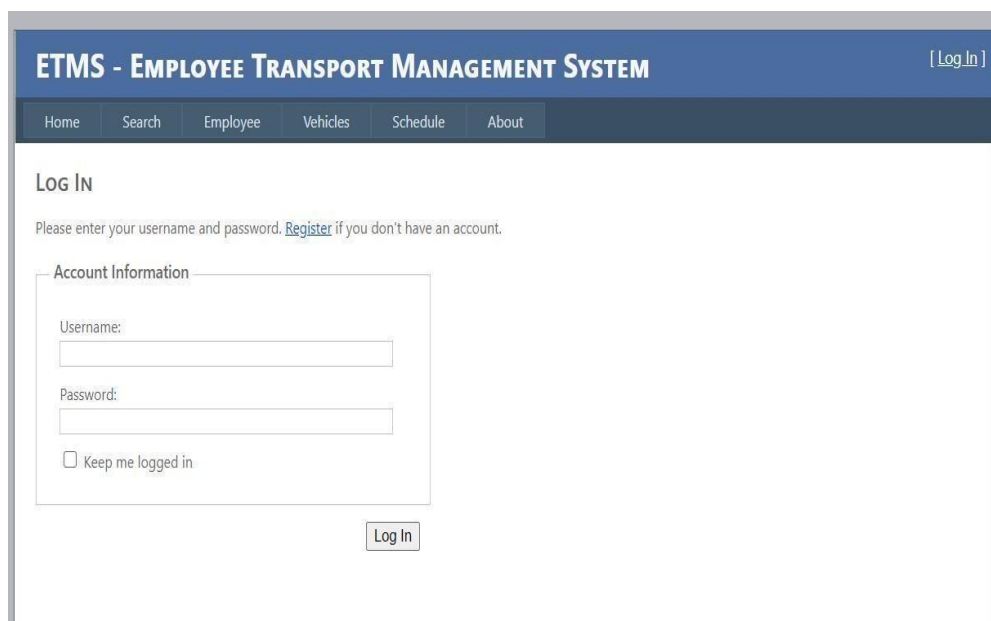


Fig 7.2 Output of Login page

When we click on the Search button it will show as follows. It also contains the Divisions, gender, Routes, shifts.

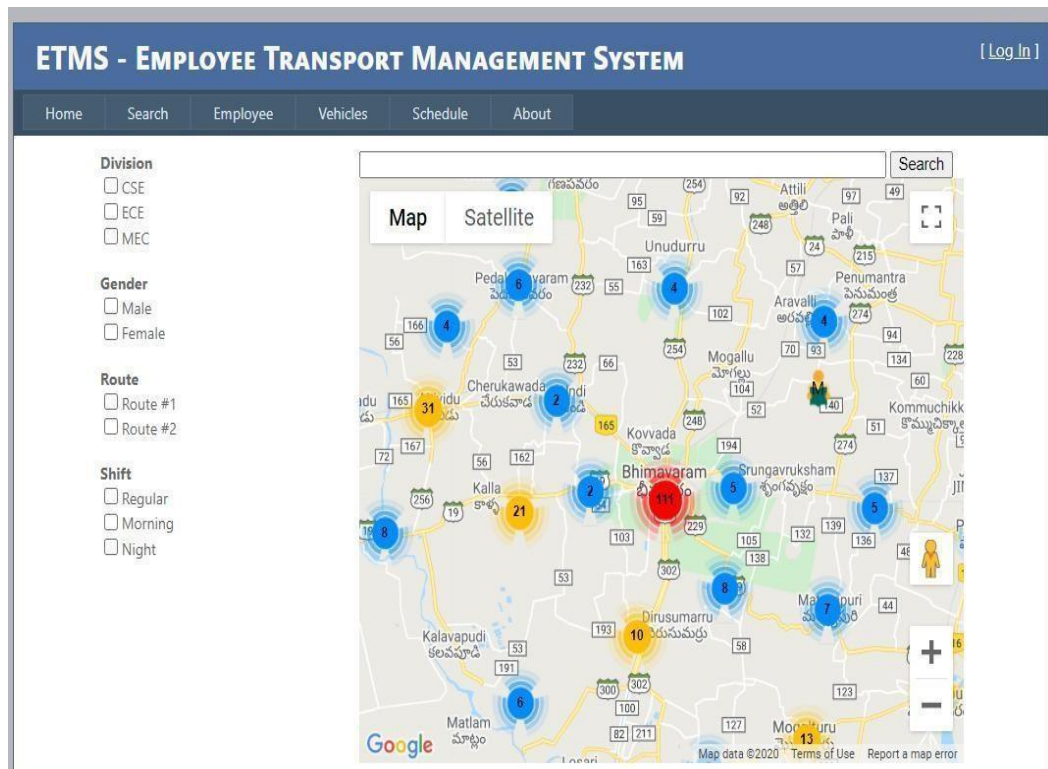


Fig 7.3 Output of Search navigation bar

The following figure contains the employee information like name and address, District, Latitude, and Longitude etc.

Employee Details [Save] [Add]

Name:

Address-Street:

District:

Pin Code:

Phone:

City:

State:

email:

Inactive Date:

Gender: ☒ Male ☐ Female

Bus Route:

Latitude:

Longitude:

Inactive: ☐

Calendar: July 2020

Mon	Tue	Wed	Thu	Fri	Sat	Sun
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Fig 7.4 Output of Employee navigation bar

ETMS - EMPLOYEE TRANSPORT MANAGEMENT SYSTEM

[Log In]

Home

Search

Employee

Vehicles

Schedule

About

Vehicle Details

Save

Add

Name

Driver Name

District

Pin Code

Phone

Regn No

State

email

Inactive Date

Inactive

<

July 2020

>

Mon	Tue	Wed	Thu	Fri	Sat	Sun
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Fig 7.5 Output of Vehicle navigation bar

ETMS - EMPLOYEE TRANSPORT MANAGEMENT SYSTEM

[Log In]

Home

Search

Employee

Vehicles

Schedule

About

Vehicle Schedule

Save

Add

Date

Vehicle Regdno

Start Time

Driver Name

Route

End Time

Inactive Date

Inactive

<

July 2020

>

Mon	Tue	Wed	Thu	Fri	Sat	Sun
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Fig 7.6 Output of Schedule nagation bar

8. CONCLUSION

The Transport Management System is an application that the admin launch the route, add time and cab for the route and make update for route, time and cab. Then the user come and view the route details and make a request for cab then the administrator assign the cab for the request. Then the user gets information about cab. We also providing the security for the employee's especially for women. Providing the security by means of gps tracking, IVRS automated system call. User can also give complaint against services .The administrator get user's feedback.

9. REFERENCES

www.google.com

www.wikipedia.com

www.gpsinsight.com/blog/gps-tracking-work/

<https://mobisoftinfotech.com/products/employee-transport-management-system>