

## **1.1 Introduction:**

With growing, Car parking increases with the number of vehicle users. With the increased use of smartphones and their applications, users prefer mobile phone-based solutions. This paper proposes the Smart Parking Management System (SPMS) that depends on Arduino parts, Android applications, and based on IoT. This gave the client the ability to check available parking spaces and reserve a parking spot. IR sensors are utilized to know if a car park space is allowed. Its area data are transmitted using the WI-FI module to the server and are recovered by the mobile application which offers many options attractively and with no cost to users and lets the user check reservation details. With IoT technology, the smart parking system can be connected wirelessly to easily track available locations.

## **1.2 Overview:**

Parking is an essential component of the transportation system. Vehicles must park at every destination. A typical automobile is parked 23 hours each day, and uses several parking spaces each week. Parking convenience affects the ease of reaching destinations and therefore affects overall accessibility.

Parking facilities are a major cost to society, and parking conflicts are among the most common problems facing designers, operators, planners and other officials. Such problems can be often defined either in terms of supply (too few spaces are available, somebody must build more) or in terms of management (available facilities are used inefficiently and should be better managed).

### **Problem definition:**

- Manual process of keeping vehicles parking record for managing employee is very difficult.
- There are various problems managing parking lots and vehicles.
- Go and look for the parking place
- Which plot are free and available for that have to go to that place and check.

### **Proposed solution:**

- Reduced development costs and increased affordability.
- More compact, multi-modal community planning (smart growth).
- Encourage use of alternative modes and reduce motor vehicle use (thereby reducing traffic
- congestion, accidents and pollution).
- Improved user options and quality of service, particularly for non-drivers.
- Improved design flexibility, creating more functional and attractive communities.
- Ability to accommodate new uses and respond to new demands.
- Reduced impervious surface and related environmental and aesthetic benefits.

### **Request Analysis**

The request was studied carefully to access what are exactly required. The focus being on the task of determining precisely what originator wants.

The analysis includes as what will be the fields, tables the tools to be used as it is a detailed study of the various operations performed by a system and their relationship within and outside the system. As it points question of what must be done

to solve the problem? One aspect of analysis is defining the boundaries of the system and determining whether or not a candidate system should consider other related systems. During analysis data are collected on the available files, decision points and transaction handled by the present system.

The project is being analyzed with every great care possible.

---

### 3. Feasibility Study

After doing the project Online Parking management system, study and analyzing all the existing or required functionalities of the system, the next test is to do the feasibility study for the project. All projects are feasible – given unlimited resources and infinite time.

Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements.

#### **Economical Feasibility:**

This is a very important aspect to be considered while developing a project. We decided the technology based on minimum possible cost factor.

- All hardware and software cost has to be borne by the organization.
- Overall we have estimated that the benefits the organization is going to receive from the proposed system will surely overcome the initial costs and the later on running cost for system.

#### **Technical Feasibility:**

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provide in the system, as described in the System Requirement Specification(SRS), and checked if everything was possible using different type of frontend and backend platform.

Technical feasibility centers on the existing computer system (hardware, software etc.)and to what extent it can support the proposed addition. This phase involves financial considerations to accommodate technical enhancements. If the budget is a serious constraint, then the project is judged not feasible. Our project is technical feasible as the technologies required are already available software required are VB, MS Access. Hardware requirement are 2GB RAM, 4GB storage Pentium IV processor correct.

The technical issue usually raised during the feasibility stage of the investigation include the followings:

- Does the necessary technology exist to do what is suggested?

- Do the proposed equipment 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 there technical guarantees of accuracy reliability case of access and data security?

### **Operational Feasibility:**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational 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 the operational feasibility of a project includes the following:

- Is there sufficient support for the management form 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?

### **Behavioral Feasibility:**

People are inherently resistant to change, and computers have been known to facilitate change. An estimate should be made of how strong a reaction the user is likely to have towards the development of a system.

### **Time Feasibility:**

Time feasibility check whether the project can be completed within the time frame suggest by customer. As per this project was assigned for four months duration and also study suggest that this much time is enough to complete the project therefore the project is also feasible as far as time is considered.

---

## 4. Requirement Gathering

The software requirement specification is produced at the culmination of the analysis task. The function and performance allocated to the software as a part of the system engineering are refined by establishing and complete information description, a detail functional description, a representation of system behavior, an indication of performance requirement and design constraints appropriate of validation criteria. The introduction of the software requirement specification states the goals and objectives of the software, describing the context of the computer base system.

### RESOURCE REQUIRED

#### HARDWARE SPECIFICATION

**RAM:** 1GB or more.

**Processor:** i3 processor or above.

**Hard-Disk:** 2GB or more.

**System Type:** 64-bit Operating System.

#### SOFTWARE SPECIFICATION

**Operating System:** Windows family.

**Web Technology:** PHP, HTML, CSS.

**Database:-** MySQL5.0.

**Server:** Wamp Server.

**Design Tool:** Sublime Text3.

**Documentation Tool:** Microsoft PowerPoint.

**UML Diagram:** StarUML.

## **TECHNOLOGY USED**

### **PHP:**

PHP stands for PHP: Hypertext Preprocessor. PHP is a server-side scripting language, like ASP. PHP supports many databases (MySQL, Informix, Mysql, Sybase, Solid, PostgreSQL, Generic ODBC, etc.). PHP is an open source software. PHP is free to download and use. PHP sits between your browser and the web server. When you type in the URL of a PHP website in your browser, your browser sends out a request to the web server. PHP is a server-side, cross-platform, HTML-embedded scripting language. There are over half a million domains running PHP and it is freely available for download.

### **HTML:**

HTML stands for Hyper Text Markup Language, which is most widely used language on web to develop web pages. HTML refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a web page is called Hypertext.

### **CSS:**

CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

### **Tools Used:**

MySQL is an open source RDBMS that relies on SQL for processing the data in database. MySQL provides APIs for the languages like C, C++, JAVA, PHP and Python. Users can quickly and easily create and access their fir in a secure and easy way. MySQL is most commonly used for web applications and for embedded applications and has become a popular alternative to proprietary database system because of its speed and reliability. MySQL can run on UNIX, Windows and Mac OS.

### **DATA FLOW DIAGRAMS**

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notation Yourdon,gane and sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose . The development of DFD's is done in several level . Each process in lower level diagram can be broken down into a more detailed DFD in the next level . The lop-level diagram is often called context diagram . It consist a single process bit, which plays vital role in studying the current system . The process in the context level diagram is exploded into other process at the first level DFD to understand the process.

Larry the idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst Constantine first developed the DFD as a way of expressing system requirement in a graphical from this lead to the modular design.

A DFD is also known as a “bubble chart ” has the purpose of clarifying system requirement and identifying major transformation that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consist of a series bubbles joined by data flow in the system.

#### **DFD SYMBOLS:**

In the DFD, there are four symbols:

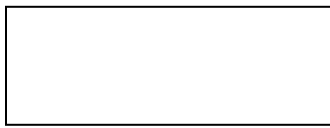
1. A square defines a source or destination of system data.
2. An arrow identifiers data flow. It is the pipeline through which the information flows.



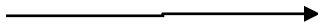
3. A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
4. An open rectangle is a data store, data at rest or a temporary repository of data.



Forms data flow



Source or destination of



Data flow

## Data Flow

1. Data flow has only direction of flow between symbols . It may flow in both directions between a process and a data store to show read before an update .The latter is usually indicated however by two separate arrows since these happen at different type.
2. A join in DFD means that exactly the same data comes from any of two or more different processes data store or sink to a common location.
3. Data flow cannot go directly back to the same process it leads . There must be at least one other process that handles the data flow produce some other data flow returns the original data into the beginning process.
4. A Data flow to a data store means update (delete or change).
5. A data flow from a data store means retrieve or use.

A data flow has a noun phrase label more than one data flow noun phrase can appear on a single arrow as long as all of the flows on the same arrow move together as one package.

Flow charts are easy to understand diagram showing how steps in a process fit together . This makes them useful tools for communicating how processes work , and for clearly documenting how a particular job is done. Furthermore , the act of mapping a process out in flow chart format helps you cleaify your understanding of the process , and helps you think about where the process can be improved. A flow chat therefore be used to:

- Define and analyze processes.
- Build a step by step picture of the process for analysis , discussion, or communication .
- Define standardize or find areas for improvement in a process.

### How to use the tool:

Most flow charts are made up of three main types of symbol:

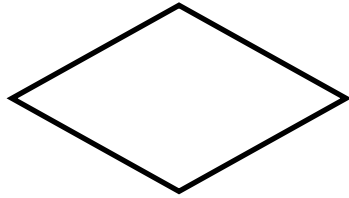
- Elongated circles, which signify the start or end of a process.



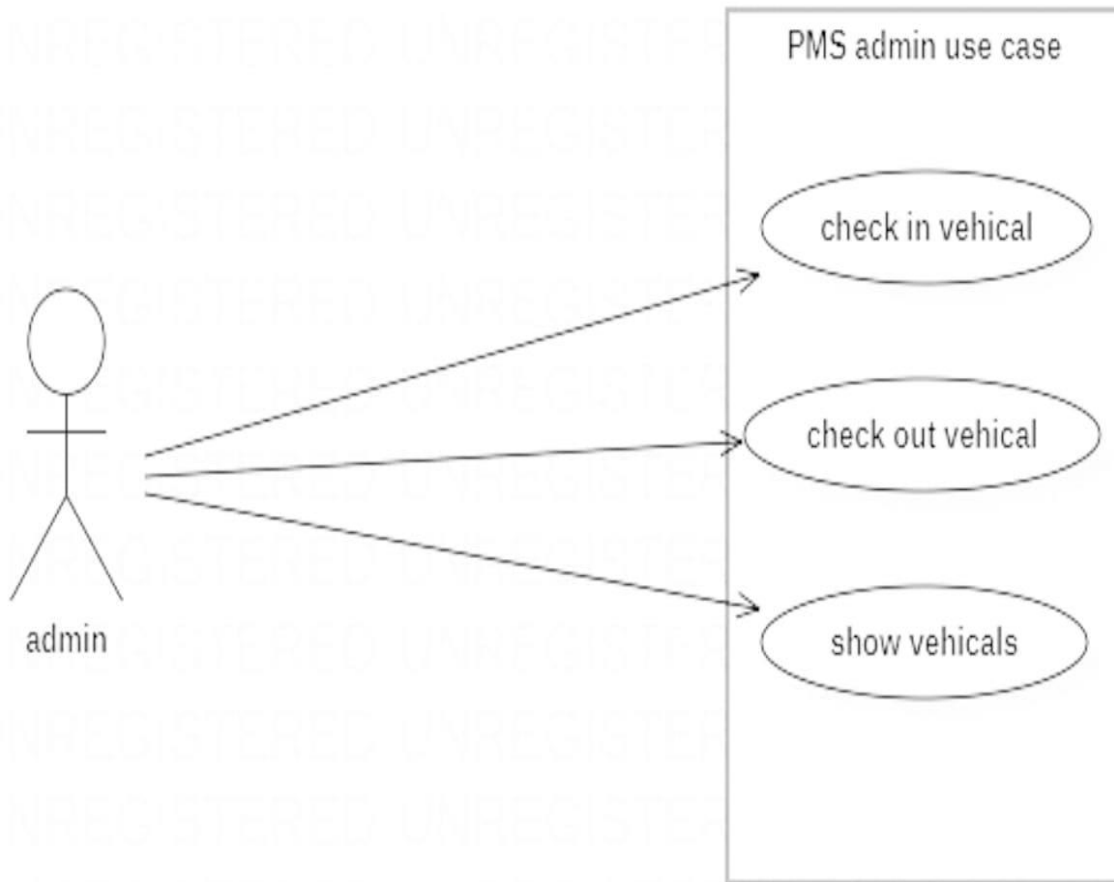
- Rectangles , which show instructions or actions



- Diamonds , which show decision that must be made



### Use Case Model (Admin):



Use case diagram (admin)

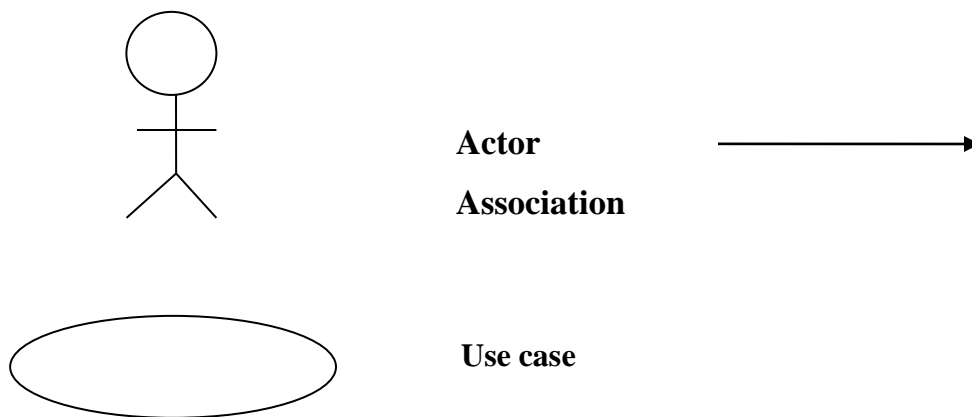
## Use Case diagram description:

A use case diagram at its simplest is a representation of user interaction with the system and depicting the specifications of use case. A use case diagram can portray the different type of user of a system and the various ways that they interact with the system.

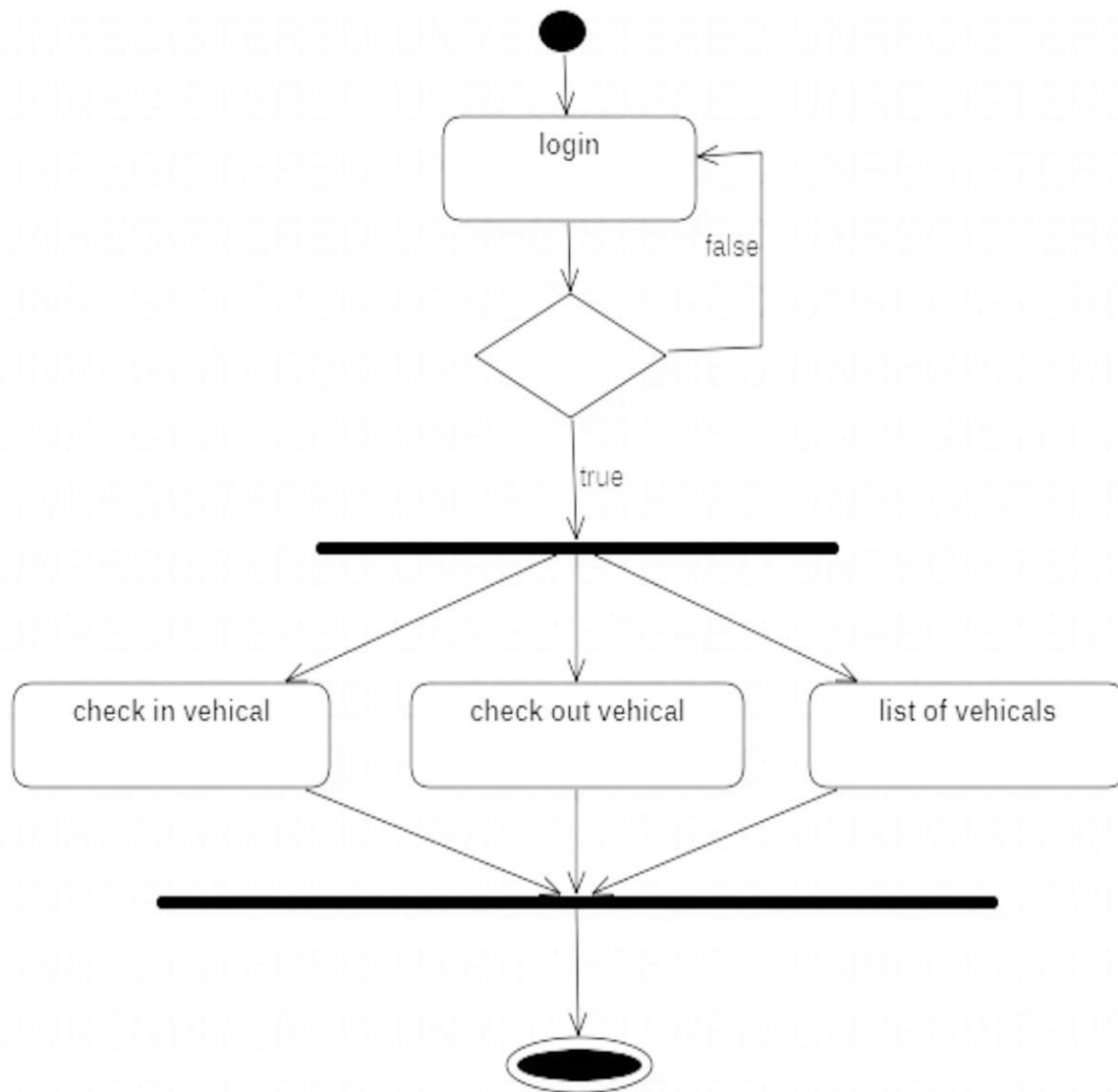
The type of diagram is typically used in conjunction with the textual use case and will often to be accomplished by other type of diagrams as well.

With regards to use case diagrams, that are exactly what they are meant to do, while a use case itself might drill into a lot of detail about every possibility, a use-case diagram can help provide a higher-level view of the system. It has been said before that “Use case diagrams are the blueprints for the system”. They provide the simplified and graphical representation of what the system must actually do.

- Always structure and organize the use case diagram from the perspective of the actor.
- Use case should start off simple and at the highest view possible. Only they can be refined and detailed further.
- Use case diagrams are based upon functionality and thus should focus on the “WHAT” and not the “HOW”.



### Activity Diagram (Admin):



Activity diagram (admin)

## **Activity diagram description:**

Activity diagrams are graphical representations of workflows of stepwise activity and actions with support for choice, iteration and concurrency. In the unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows). Activity diagrams show the overall flow control. The most important shape types:

- Rounded rectangles represent actions.
- Diamonds represent decisions.
- Bars represent the start (split) or end (join) of concurrent activities.
- The black circle represents the start (initial state) of the workflow.
- An encircled black circle represents the end (final state).

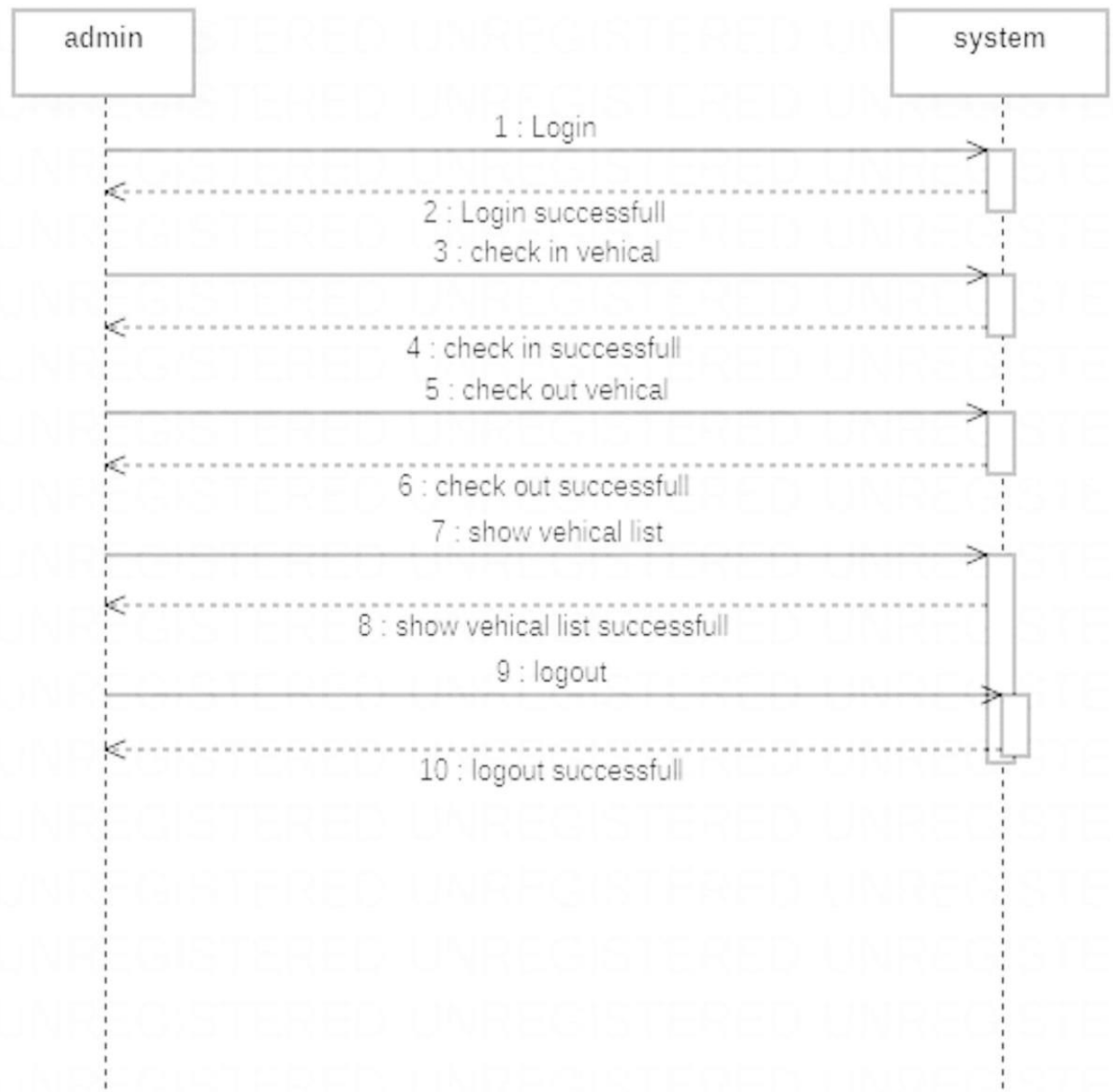
Arrows run from the start towards the end represent the order in which activities happen.

Activity diagrams may be regarded as a form of flow chart. Typical flowchart techniques lack constructs for expressing concurrency. However, the join and split symbols in activity diagrams only resolve this for simple cases; the meaning of the model is not clear when they are arbitrarily combined with decisions or loops.

It was a specialized form of state diagrams, and it was renormalized to be based on Petri net-like semantics, increasing the scope of situations that can be modeled using activity diagrams.

UML activity diagrams can be used in various domains, i.e. in design of embedded systems, it is possible to verify such a specification using model checking technique.

### Sequence Diagram (Admin):



Sequence diagram (admin)



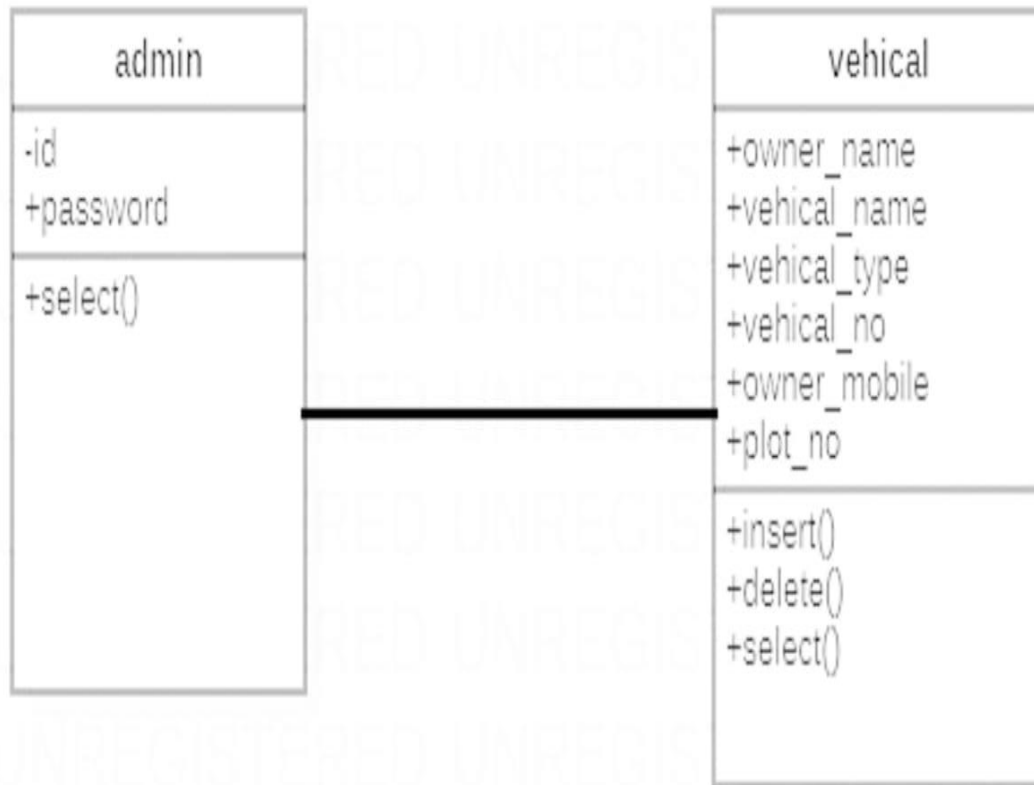
## **Sequence diagram description:**

Sequence diagrams descriptions among classes. These interactions are modeled as exchange of messages. These diagram focus on classes and the messages they exchange to accomplish some described behavior. Sequence diagrams type of interaction diagrams.

- Class roles, which represent roles that objects may play within the interaction.
- Lifelines, which represent the existence of an object over a period of time.
- Activations, which represent the time during which an object is performing an operation.
- Messages, which represent communication between objects.

A sequence diagram shows the participants in an interaction and the sequence of message among them. A sequence diagram shows the interaction of a system with its actor.

### Class Diagram:

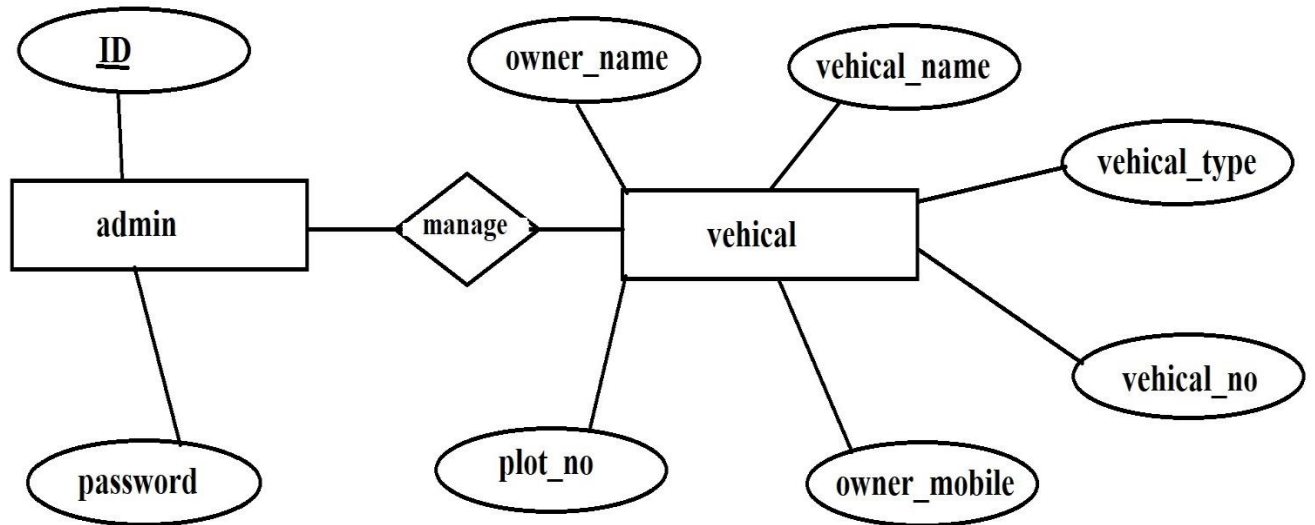


### Class Diagram

## **Class diagram description:**

In software engineering a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system classes, their attributes, operations (or methods), and the relationships among objects. The class diagram is the main building block of object orientated modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed.

## ER Diagram



## ER diagram

## **ER diagram description:**

In 1976, chain developed the entity relationship (ER) model, a high-level data model that is useful in a developing in a conceptual design for a database, creation of an ER Diagram, which is one of the first steps in designing a database helps the designer to understand and to specify the desired components. An ER model is a diagram containing entities or “items”, relationships among them, and attributes of the entities and relationships. The entity relationship model a high level data model that is useful in developing a conceptual design for a database. The overall logical expressed graphical by an ER-diagram. It consists a set of basic objects entities and of relationship among these objects. An entity is represented by the set of attributes. Attributes are descriptive properties possessed by each member of an entity set. The ER diagram consists of the following major components.

The entity relationship diagram is graphical representation of the database logic and includes a detailed description of all entities, relationship and constraints. The ER diagram is based on perception of a real world that consists of a set of basic object called entities and of relationship among these objects. It is pictorial representation; it is easy to understand such a diagram of following main component-

- Rectangles represent entity set.
- Ellipses represent relationship set.
- Diamonds represent relationship set.
- Double ellipses represent multivalve attribute.

### Home.php

```
<?php
?>

<html>

<head>

<style>

body{

background-image:url("bg1.jpg");

background-position:center;

background-size:cover;

}

</style>

</head>

<body>

<h1                style="position:absolute;top:100px;left:350px;width:800px;font-
family:'algerian';color:black"><b/>welcome to parking managment system</h1>

<a                href="login                .php"><input                type="button"
style="position:absolute;top:250px;left:450px;font-family:'algerian';font-
size:20px;height:50px;width:450px;background-color:yellow" value="LOGIN"></a>

</body>

</html>
```

### Login.php

```
<html>

<head>

<style>
```

```

body{
background-image:url("bg1.jpg");
background-position:center;
background-size:cover;
}
</style>
</head>
<form method="POST">
<h1 style="position:absolute;top:100px;left:750px;font-family:'algerian';color:black">LOGIN FORM</h1>
<h1 style="position:absolute;top:190px;left:750px;color:black">USER ID</h1>
<input type="text" name="user_id" placeholder="enter your user id here" style="position:absolute;top:260px;left:750px;height:35px;width:400px;border-width:2px" required>
<h1 style="position:absolute;top:280px;color:black">PASSWORD</h1>
<input type="password" name="pwd" placeholder="enter your password" style="border-width:2px;position:absolute;top:345px;left:750px;height:35px;width:400px">
<input type="submit" value="LOGIN" style="position:absolute;top:410px;left:750px;height:40px;width:400px;color:blue;background-color:white;font-size:20px;font-family:algerian" required>
</form>
<body>
<button onclick="document.location='home.php'" style="background-color:yellow;width:150px">back</button>

</body>
</html>
<?php
include("conn.php");
$id=$_POST['user_id'];

```

```

$password=$_POST['pwd'];
if(empty($password) and empty($id))
{
    echo"<h3 style='color:red;position:absolute;left:750px;top:450px'><u>\".\"PLESAE
ENTER USER ID AND PASSWORD\".\"</u></h3>";
}
else
{
    $query="select * from admin where id='$id' and password='$password'";
    $q=mysqli_query($conn,$query);
    $num=mysqli_num_rows($q);
    if($num==1)
    {
        header('location:work.php');
    }
    else
    {
        echo"<h3 style='color:red;position:absolute;left:750px;top:450px'><u>\".\"PLEASE
ENTER CORRECT USERNAME AND PASSWORD\".\"</u></h3>";
    }
}
?>

```

**Work.php**

```

<?php
?>
<html>
<head>

```



```

<style>
body{
background-image:url("bg1.jpg");
background-position:center;
background-size:cover;
}
</style>
</head>
<body>
<h1 style="position:absolute;top:100px;left:550px;width:800px;font-
family:'algerian'"><b>welcome to parking managment system</b></h1>
<a href="vehical arrives.php"><input type="button"
style="position:absolute;top:250px;left:650px;font-family:'algerian';font-
size:20px;height:50px;width:450px;background-color:yellow" value="CHECK IN
VEHICAL"></a>
<a href="vehical released.php"><input type="button"
style="position:absolute;top:350px;left:650px;font-family:'algerian';font-
size:20px;height:50px;width:450px;background-color:yellow" value="CHECKOUT
VEHICAL"></a>
<a href="vehical list.php"><input type="button"
style="position:absolute;top:450px;left:650px;font-family:'algerian';font-
size:20px;height:50px;width:450px;background-color:yellow" value="SHOW
VEHICALS"></a>
<button onclick="document.location='home.php'" style="background-
color:yellow;width:150px">LOGOUT</button>
</body>
</html>

```

**Vehical arrives.php**

```

<html>
<head>

```

```

<style>
body{
background-image:url("bg1.jpg");
background-position:center;
background-size:cover;
}
</style>
</head>
<form method="POST">
<h1 style="position:absolute;top:120px;left:520px;color:black;font-
family:'algerian'">ADD VEHICAL DETAILS</h1>
<input type="text" name="vname" placeholder="ENTER VEHICAL NAME"
style="position:absolute;top:220px;left:500px;height:35px;width:215px" required>
<input type="text" name="oname" placeholder="ENTER OWNER NAME"
style="position:absolute;top:220px;left:800px;height:35px;width:215px" required>
<input type="text" name="vtype" placeholder="ENTER VEHICAL TYPE"
style="position:absolute;top:290;left:500px;height:35px;width:215px" required>
<input type="text" name="vnum" placeholder="ENTER VEHICAL NUMBER"
style="position:absolute;top:290;left:800px;height:35px;width:215px" required>
<input type="text" name="mobile" placeholder="ENTER OWNER NUMBER"
style="position:absolute;top:360;left:500px;height:35px;width:215px" required>
<input type="text" name="plot" placeholder="ENTER PARKING PLOT NUMBER"
style="position:absolute;top:360;left:800px;height:35px;width:215px" required>
<input type="submit" value="checkin"
style="position:absolute;top:430;left:500px;height:35px;width:520px;color:white;backgro-
und-color:blue;font-size:20px;font-family:'algerian'">
</form>
<body>
<button onclick="document.location='WorK.php'" style="background-
color:yellow;width:150px">admin page</button>
<button onclick="document.location='vehical list.php'" style="background-
color:yellow;width:150px">list of all vehicals</button>

```

```
<button    onclick='document.location='vehical    released.php''    style='background-color:yellow;width:150px'>checkout page</button>
```

```
</body>
```

```
</html>
```

```
<?php
```

```
include('conn.php');
```

```
$vname=$_POST['vname'];
```

```
$oname=$_POST['oname'];
```

```
$vtype=$_POST['vtype'];
```

```
$vnum=$_POST['vnum'];
```

```
$mobile=$_POST['mobile'];
```

```
$plot=$_POST['plot'];
```

```
if(empty($vname) or empty($oname) or empty($vtype) or empty($vnum) or empty($plot) or empty($mobile))
```

```
{
```

```
    echo'<h3                style='position:absolute;top:460px;left:480px;color:red;font-family:algerian'><u><blink>'. "please enter data required data with correct manner". "</blink></h3>";
```

```
}
```

```
else
```

```
{
```

```
$query1='select * from vehical where vehical_no='$vnum';';
```

```
$q1=mysqli_query($conn,$query1);
```

```
$num=mysqli_num_rows($q1);
```

```
    if($num>=1)
```

```
    {
```

```

        echo'<h3                style="position:absolute;top:460px;left:500px;color:red;font-
family:algerian"><u><blink>'. "the vehical is already in parking"."</blink></h3>";

    }

    else

    {

        $query="insert                into                vehical
values('$oname','$vname','$vtype','$vnum',$mobile,$plot)";

        $q=mysqli_query($conn,$query);

        if($q)

        {

            echo'<h3                style="position:absolute;top:460px;left:550px;color:red;font-
family:algerian"><u><blink>'. "vehical has been parked"."</blink></h3>";

        }

    }

}

?>

```

## Vehical realease.php

```

<html>

<head>

<style>

body{

background-image:url("bg1.jpg");

background-position:center;

background-size:cover;

}

</style>

```

```

</head>

<form method="POST">

<h1 style="position:absolute;top:100px;left:750px;font-
family:'algerian';color:black">REMOVE VEHICAL FROM PARKING</h1>

<h1 style="position:absolute;top:190px;left:150;color:black;left:750px">TYPE OF
VEHICAL</h1>

<input type="text" name="type" placeholder="enter vehical type
here" style="position:absolute;top:260px;left:750px;height:35px;width:400px;border-
width:2px" required>

<h1 style="position:absolute;top:280px;color:black;left:750px">VEHICAL
NUMBER</h1>

<input type="text" name="number" placeholder="enter vehical number" style="border-
width:2px;position:absolute;top:345px;left:750px;height:35px;width:400px" required>

<input type="submit" value="CHECK OUT"
style="position:absolute;top:410px;left:750px;height:40px;width:400px;color:blue;backgr
ound-color:white;font-size:20px;font-family:algerian" required>

</form>

<button onclick="document.location='WorK.php'" style="background-
color:yellow;width:150px">admin page</button>

<button onclick="document.location='vehical arrives.php'" style="background-
color:yellow;width:150px">checkin page</button>

<button onclick="document.location='vehical list.php'" style="background-
color:yellow;width:150px">list of all vehicals</button>

<body>

</body>

</html>

<?php
include('conn.php');

$type=$_POST['type'];

$number=$_POST['number'];

if(empty($type) and empty($number))

{

```

```

        echo"<h3 style='color:red;position:absolute;left:750px;top:450px'><u>\".PLESAE
ENTER VEHICAL NUMBER AND VEHICAL TYPE\".</u></h3>";
    }
    else
    {
        $query="select * from vehical where vehical_type='$type' and vehical_no='$number';";
        $q=mysqli_query($conn,$query);
        $num=mysqli_num_rows($q);
        if($num==1)
        {
            $q1="delete from vehical where vehical_type='$type' and vehical_no='$number';";
            $que=mysqli_query($conn,$q1);
            if($que)
            {
                echo"<h3
style='color:red;position:absolute;left:750px;top:450px'><u>\"VEHICAL    CHECKED
OUT FROM PARKING\".</u></h3>";
            }
        }
    }
    else
    {
        echo"<h3 style='color:red;position:absolute;left:750px;top:450px'><u>\"THIS
VEHICALIS NOT IN PARKING\".</u></h3>";
    }
}
?>

```

**Vehical list.php**

```

<html>
<head>
<style>
body{
background-image:url("bg1.jpg");
background-position:center;
background-size:cover;
}
table{
    position:absolute;
    top:150;
    left:600;
    background-color:white;
}
</style>
</head>
<body>
<h1
                                style="position:absolute;top:20;left:700px;font-
family:algerian;color:blue;width:340px"><u>here is the list of all vehicals
</u></h1></body></html>

<button      onclick="document.location='WorK.php'"      style="background-
color:yellow;width:150px">admin page</button>

<button      onclick="document.location='vehical   arrives.php'"      style="background-
color:yellow;width:150px">checkin page</button>

<button      onclick="document.location='vehical   released.php'"      style="background-
color:yellow;width:150px">checkout page</button>

<?php
include("conn.php");

```

```

$q1="select * from vehical;";

$query1=mysqli_query($conn,$q1);

echo"<center/><table border='1'>

<tr>

<th>owner_name</th>

<th>vehical_name</th>

<th>vehical_type</th>

<th>vehical_no</th>

<th>owners_mobile</th>

<th>plot_no</th>

</tr>";

while($row=mysqli_fetch_array($query1))
{

echo "<tr><td>".$row["owner_name"]."</td> <br/><td>";

echo $row["vehical_name"]." </td><br/><td>";

echo $row["vehical_type"]."</td><td>";

echo $row["vehical_no"]." </td><br/><td>";

echo $row["owners_mobile"]." </td><br/><td>";

echo $row["plot_no"]." </td><br/></tr>";

}

echo"</table>";

?>

```

Conn.php

```
<?php
```

```
error_reporting(E_ERROR);
```



```
$sn="localhost";  
$un="root";  
$pwd="";  
$db="PLMS";  
$conn=mysqli_connect($sn,$un,$pwd,$db);  
?>
```

---

## 8. Implementation & Maintenance

### **5.1 Implementation:**

System implementation is the process of making the newly designed systems fully operational. The system is implemented after careful testing. The primary goal of product implementation is development of source code that is easy to read and easy to understand. The term implementation has different meaning, ranging from the conversion of a basic application to a compatible replacement of a computer system. Implementation is used here to mean the process converting a new or a revised system design in to an operational one. During the implementation stage we convert the detailed code in a programming language. Clarity of source code eases debugging, testing and modification of a software product. the difficulties encountered during implementation is successful integration of source code components into a functioning system before a routine can be placed in the evolving system. It may be required that the routine be inspected by an inspection team. Or reviewed or tested to a given level of test coverage.

### Testing

Software testing is an important element of s/w quality assurance and represents the ultimate review of specification, design and coding. the increasing visibility of s/w as a system element and the costs associated with an s/w failure are motivating forces for well planned, through testing thus a series of testing are performed for the proposed system before the system is ready for user acceptance testing. testing is a set of activities that can be planned in advance and conducted systematically. testing is very important stages of a software include unit testing, integration testing and deployment testing.

#### **White box testing:**

Is a test case design philosophy that uses the control structure described as part of component-level design to derive test cases.

White box testing is also called glass-box testing or structural testing. It performs close examination of procedural details. They test the software test the software logical path by having test cases exercising specific sets of condition and loops. It examine the program status at various points to determine whether the expected status correspond to the actual one.

Internal working of the product test can be conducted to ensure that internal operational performs according to specification. And all component have been adequately exercised is called white-box testing.

Using white box testing method it can be ensured that:

- All independent parts within a module have been exercised at least one.
- Exercise all logical decision and their all false sides.
- Exercise all loop at their boundaries and within their operational bound.
- Exercise internal data structure to ensure that validity.

#### **Black box testing:**

It is also called functional testing or behavioral testing. It performs tests of software interfaces. They test the operation ability of software functional, acceptance of input and delivery of output and the maintainability of external information integrity.

It is testing without knowledge of the internal working of the item being tested. For example, when black box testing is applied to software engineering, the tester would only know the “legal” inputs and what the expected outputs should be, but now the program actually arrives at those outputs. It is because of this that black box testing can be considered testing with respect to the specifications, no other knowledge of program in necessary.

It examine some of the basic aspects of a system, having little regard for the internal logical software structure.

Black box testing is not an alternative to white box technique. It is complementary approach that is likely to uncover a different class than white box methods.

- Black box testing attempts to find error in the categories.
- Incorrect or missing function.
- Interface error.
- Errors in data structure or database access.
- Performance errors.
- Initialization and termination errors.

### Admin Table

```
C:\wamp\bin\mysql\mysql5.6.12\bin\mysql.exe

mysql> desc admin;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id     | varchar(20) | YES |     | NULL |     |
| password | varchar(20) | YES |     | NULL |     |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.09 sec)

mysql>
```

### User Table

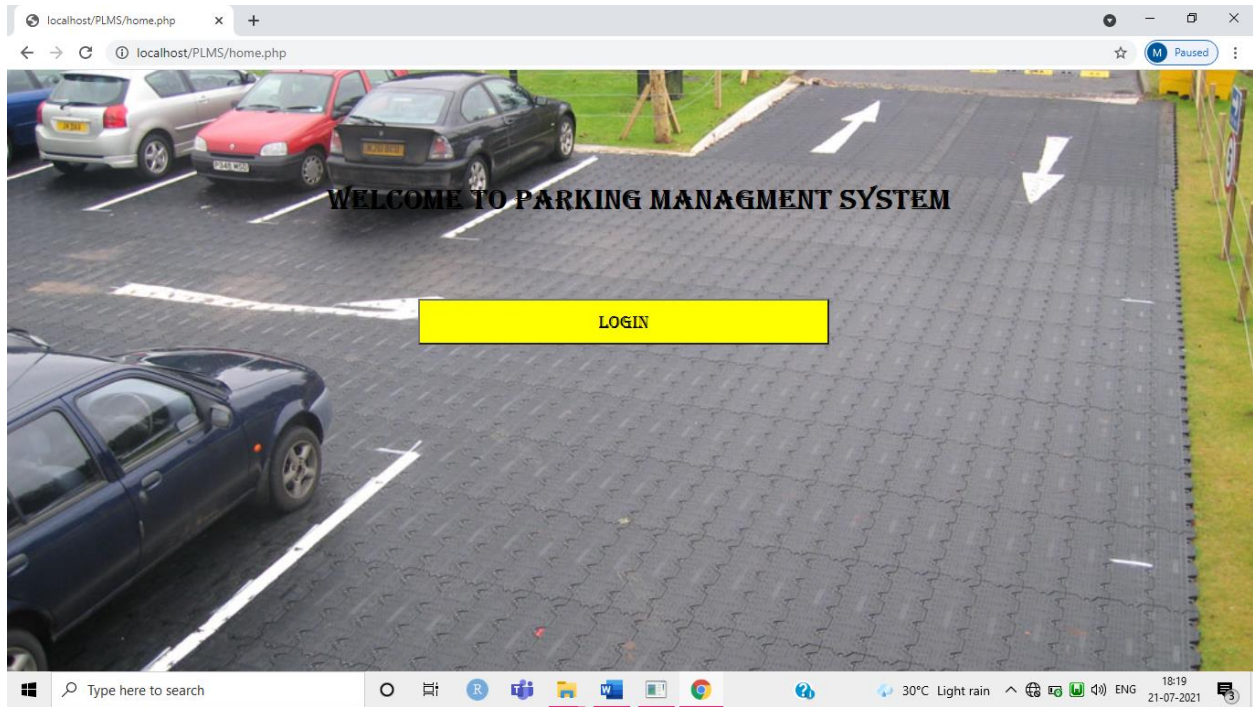
```
Select C:\wamp\bin\mysql\mysql5.6.12\bin\mysql.exe

mysql> desc vehical;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| owner_name | varchar(20) | YES |     | NULL |     |
| vehical_name | varchar(20) | YES |     | NULL |     |
| vehical_type | varchar(20) | YES |     | NULL |     |
| vehical_no | varchar(20) | YES |     | NULL |     |
| plot_no | int(10) | YES |     | NULL |     |
| owners_mobile | int(15) | YES |     | NULL |     |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.08 sec)

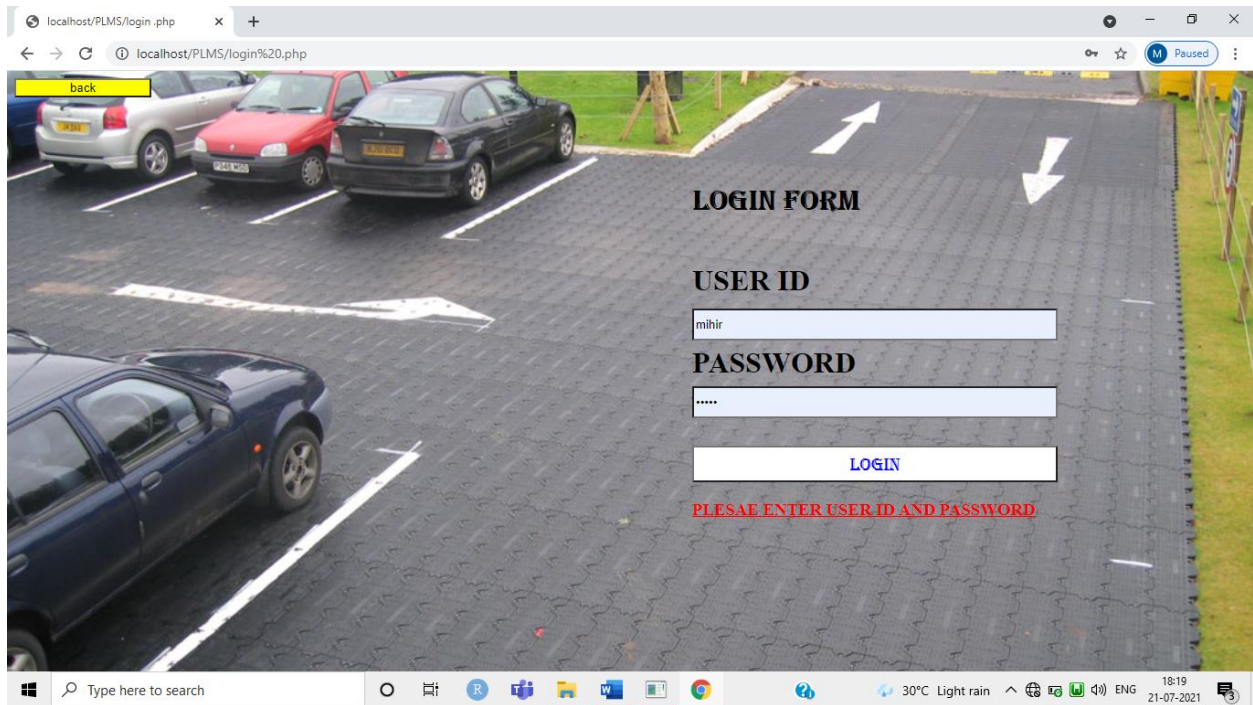
mysql>
```

## 11. Screenshots of Project

### Home Page Page

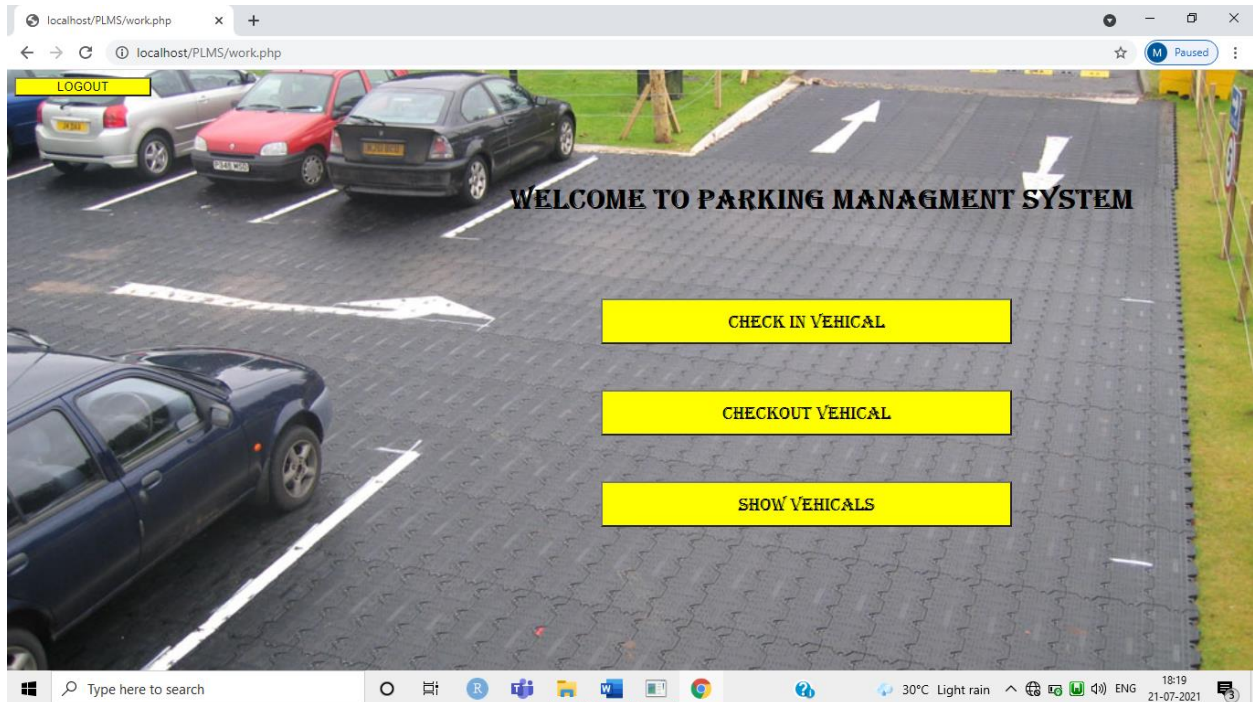


### login Page

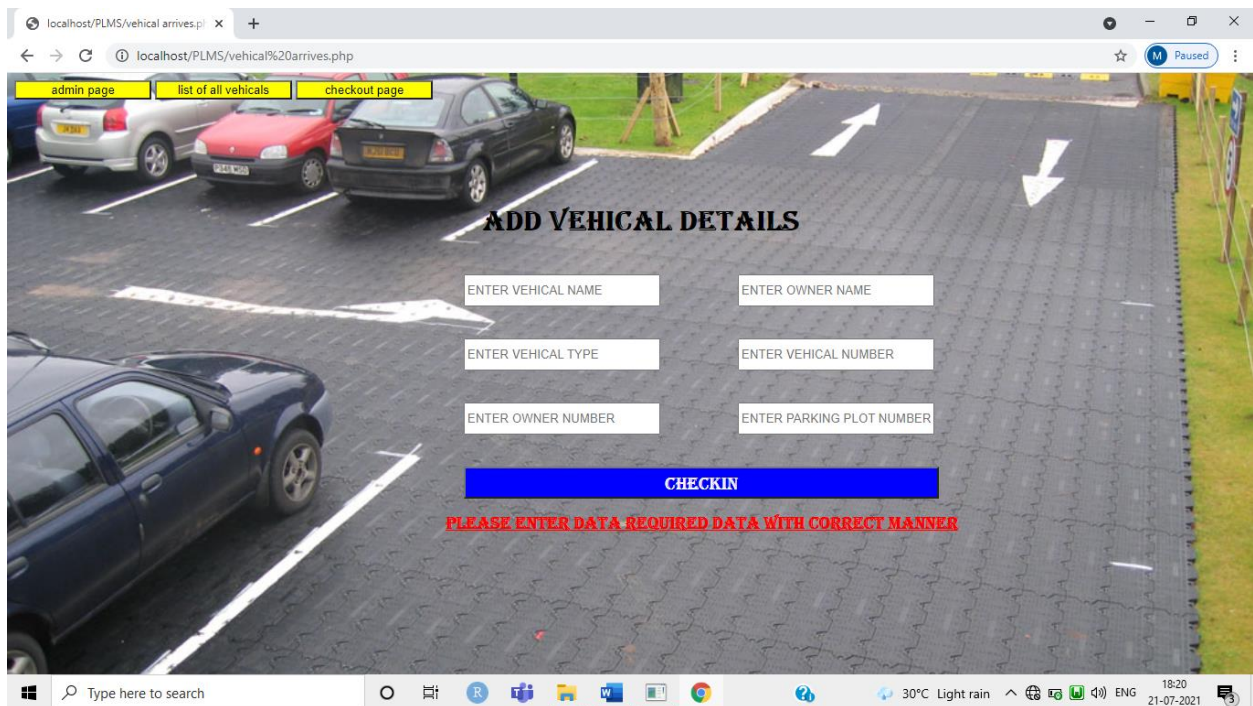




## work page



## Check in page



## Check out page

localhost/PLMS/vehical released. x +

localhost/PLMS/vehical%20released.php

admin page checkin page list of all vehicals

**REMOVE VEHICAL FROM PARKING**

**TYPE OF VEHICAL**

enter vehical type here

**VEHICAL NUMBER**

enter vehical number

**CHECK OUT**

**PLESAE ENTER VEHICAL NUMBER AND VEHICAL TYPE**

Type here to search

30°C Light rain 18:21 21-07-2021

## Vehical list page

localhost/PLMS/vehical list.php x +

localhost/PLMS/vehical%20list.php

admin page checkin page checkout page

**HERE IS THE LIST OF ALL VEHICALS**

owner_name	vehical_name	vehical_type	vehical_no	owners_mobile	plot_no
adaf	sgdgd	gfhg	244		244
adaf	sgdgd	gfhg	244		244
jhfisyu	mihuds	kjhgdfu	jhdhg		975947
jgfh	utgkmb	gjhjf	kmbjmbj	97	75686757

Type here to search

30°C Light rain 18:21 21-07-2021



---

## **12. Conclusion**

Current parking planning practices are inefficient, resulting in economically excessive parking supply, increased automobile traffic, and more dispersed destinations, contributing to various economic, social and environmental problems. There are many reasons to use management strategies that result in more efficient use of parking resources, in order to address parking problems without expanding supply. Parking facilities that serve multiple destinations and are efficiently regulated or priced to favor higher value users (for example, delivery vehicles and customers over commuters and residents) tend to be efficiently used. On-street metered parking and commercial parking are particularly suitable for this type of management, and so should be favored over unpriced, off-street parking that serves a single destination.

---

### **13. Future Scope & Further Enhancement of Project**

Nowadays parking is very important and hence it is necessary for every vehicle owner to park his or her car in a secure designated parking slot available. To escalate this particular system various parking owners have integrated themselves with sophisticated parking control systems, which are high tech and offers full fledged parking services.

## 14. References

---

- [www.w3school.com](http://www.w3school.com)
- [www.3pdf.com](http://www.3pdf.com)
- [www.project.com](http://www.project.com)
- [www.ukessay.com](http://www.ukessay.com)
- [www.elarninglarning.com](http://www.elarninglarning.com)