

# **1. INTRODUCTION**

## **1.1 OVERVIEW THE PROJECT**

Automobiles are synonyms for mobility and freedom. An amazing increase in the growth of population in this world leads to the rapid increase in the number of vehicles being used. With the growing number of vehicles and the consequent shortage of parking space, there is haphazard and totally unregulated parking of vehicles all over. In densely populated areas they are real challenge for city planners, architects and developers. The need to offer sufficient parking spaces is a task for specialists. This situation calls for the need for car automated parking system that not only regulates parking in a given area but also keeps the manual control to a minimum. Car Parking Management System is the sole solution to park as many cars as possible in as little space as possible.

Car Parking Management System that regulates the number of cars that can be parked in a given space at any given time based on the parking space availability. Automated parking is a method of parking and exiting cars. The entry and exit of vehicles are managed Status signals indicate whether a car is currently in the process of entering or leaving the parking space. This is more cost-effective than conventional garages, and allows more cars to be parked in less space. The automation technology is used to typically double to triple the capacity of conventional parking garages.

The manual handling of records about entering and exiting cars leads to long car queues causing traffic jams and safety concerns. Also, manual calculations of parking charges are bound to human errors. The car parking system has a very high error margin which cumulatively affects the regular customers to the parking site as well as other users of the parking site. The daily manual report making and the demand this puts to the storage resources add to the problem. In order to alleviate the problems, there is a need to develop a computerized information system that can handle the situation.

The Car Parking Management System determines the parking charges on hourly basis. For example, if one parks for duration of 1 hour and 5 minutes, the system will charge a duration of 2 hours, which has a very big error margin of 55 minutes. However, the Car Parking Management System allows the regular customers to pay for longer periods of a month or more,

but does not provide for cases where a car gets a mechanical breakdown that it takes a month without parking in the parking site.

These comprise of a purely paper-based system with lots of time delays and human errors coupled with frauds in the management of payments, that may need computerization to lift the situation.

**Dashboard:**

In these sections, admin can briefly view the number of vehicle entries in a particular period.

**Category:**

In this section, admin can manage category (add/update).

**Add Vehicle:**

In this section, admin add vehicle which is going to park.

**Manage Vehicle:**

In this section, admin can manage incoming and outgoing vehicle and admin can also add parking charges and his/her remarks.

**Reports:**

In this section admin can generate vehicle entries reports between two dates.

**Search:**

In this section, admin can search a particular vehicle by parking number. Admin can also update his profile, change the password and recover the password.

## **1.2 ORGANIZATION PROFILE**

SMC Technology was founded in 2016 with a specific goal. It is dedicated software research and development.

It is a startup company, or simply a startup, is an entrepreneurial venture in its early stages of operations typically aimed at resolving a real-life issue with an innovative product or service. Founders normally finance their startups and may attempt to attract outside investment before they get off the ground.

More responsibility and opportunities to learn are two. As SMC have fewer employees than large, established companies, employees tend to wear many hats, working in a variety of roles, which leads to more responsibility as well as opportunities to learn. The benefits of working at a startup include greater opportunities to learn, increased responsibility, flexible work hours, a relaxed work environment, increased employee interaction, good workplace benefits, and innovation.

## 2. SYSTEM STUDY

### 2.1 EXISTING SYSTEM

The system, which is followed at present, is a computerized system. That is, they are collected and store the customer details in the form of excel sheet. And they use tally for maintains the accounts details. But in the way they crossed many errors and doing rework and time taken. First the staffs should collect the customer details and item details in the form of printed paper after they will store in the excel sheet it is waste of time so now in the proposed system avoid this type of work. There are no high possibilities to commit errors and mistakes, which leads to produce the wrong statements to the management. Report keeping is also not an easy work.

#### 2.1.1 DRAWBACKS

1. **Manual working:** The existing system is handled manually without the touch of technology a large system becomes outdated.
2. **Restricted parking space:** Since there is no use of modern technology in the existing parking system available there is a lot of wastage of space. The existing system is not flexible in terms of utilizing the parking space.
3. **Loopholes in ticketing:** The current existing system uses the concept of ticketing done by employees which has loopholes.
4. **Manual ticketing:** The use of ticketing system is manual with no use of technology it makes the system inefficient and complex.
5. **Difficult to maintain records:** In the existing system it is difficult to maintain the records, database as there is no use of technologies.
6. **Less profit:** Since more space is utilized and less vehicles parked this makes the system prone to give profits less than expected.
7. **Lack of security:** There are minimum security checks in the current system without the use of lasers, sensors it makes the system less secure.
8. **Lack of customer satisfaction:** The customers are less satisfied with the existing system as it does not fulfill all the requirements of the customers.

## 2.2 PROPOSED SYSTEM

The proposed system over comes all the discrepancies of the existing system by making the at most use of technology. The proposed system will be fully integrated with the company's requirement and enhances the capabilities of the current system, since the proposed system is web-oriented application.

The proposed system allows security features by providing various right and privileges in according the data. All these aspects will enable the business to function as an independent system. It also provides the latest technology embedded in the system, which will yield the desired result.

Once the details are fed into the computer there is no need for various persons to deal with separate sections. Only a single person is enough to maintain all the reports. The security can also be given as per the requirement of the users

### 2.2.1 ADVANTAGE OF PROPOSED SYSTEM

The inclusion of technologies in the system increases the efficiency as follows:

- 1. Computerized working:** The involvements of technologies increase the efficiency of the system by decreasing the load done manually.
- 2. Increased parking space:** Since the Car parking Management System is used in the companies which provide multi-storied parking facilities it increases the parking to maximum utilization.
- 3. Computerized allotment:** The Car parking Management System provides a computerized ticketing system which helps in maintaining inventory.
- 4. Maintenance of records:** Since the whole working of the system of the system is computerized so it easy to maintain the database, records, profit charts, inventory etc.
- 5. Security services:** the use of cameras, laser scans, sensors etc. makes the customer more satisfied and the system more secure.

### **3. SYSTEM SPECIFICATION**

#### **3.1 HARDWARE REQUIREMENTS:**

System	: Pentium IV 2.4 GHz.
Hard Disk	: 40 GB.
Floppy Drive	: 1.44 MB.
Monitor	: 15 VGA Colour.
Mouse	: Logitech.
RAM	: 256 GB.

#### **3.2 SOFTWARE REQUIREMENTS:**

Operating system	: - Windows.
Front End	: - PHP
Database	: - MYSQL
Tools	: - XAMPP SERVER

## 3.3 SOFTWARE DESCRIPTION

### HYPER TEXT MARKUP LANGUAGE

HTML stands for “Hypertext Markup Language”. HTML is a SGML (Standard Generalized Markup Language) application widely used to create web pages. It is basically a formatting language and not a programming language. HTML is a language that is easy to write, easy to understand and highly portable. HTML is not a compiled language and is directly interpreted by a browser. HTML is the set of instructions. Each instruction is called as an element or Markup. It is used to structure and format documents for presentation on the web. HTML enhances ASCII files with markup tags that permit the display of a variety of fonts, images, and highlighting options. It also designates structural elements such as headers, lists, and paragraphs, and provides hypertext links to other documents on the Internet.

#### **Interactive HTML:**

**Html Tag** The first and last tags in a document should always be the HTML tags. These are the tags that tell a Web browser where the HTML in your document begins and ends. The absolute most basic of all possible Web documents is:

```
<Html >
```

```
</Html>
```

That’s it. If we were to load such a page into a Web browser, it wouldn’t do anything except give us a blank screen, but it is technically a valid Web page.

#### **Head Tag**

The HEAD tags contain all of the document’s header information. When I say “header,” I don’t mean what appears at the top of the browser window, but things like the document title and so on. Speaking of which...

#### **Body Tag**

BODY comes after the HEAD structure. Between the BODY tags, all of the stuff that gets displayed in the browser window is found. All of the text, the graphics, and links, and so on these things occur between the BODY tags.

#### **Forms Tag**

Forms provide a unique feature to HTML. Forms allow you to collect data from the end user and return that data to an executable code. The <FORM> element is used to start a form. The <FORM> element specifies what program to run when the form is submitted and how the data is to be transferred.

## **Attributes Tag**

**ACTION**-The ACTION attribute specifies what program or HTML file is to be called when the submit button is pressed. The ACTION is specified as a URL.

## **Method**

The METHOD attribute specifies the protocol to be used when the client sends data to the server. There are two methods. GET (the default) or POST. Using GET method data is attached to the URL mentioned in ACTION attribute. The data that can be sent here is limited here. Where as in POST the data is posted through environment variables and unlimited amounts of data can be sent.

## **Name**

The NAME attribute specifies the name of the form with which the elements of the form can be referred to later.

## **The <INPUT> Element**

The<INPUT> tag provides some type of data entry in the form depending on the value of its type attribute.

## **Attributes**

**HREF** – HREF stands for “Hypertext Reference,” which is another way of saying, “The location of the file I want to load.” Most anchors are in the form <A HREF=”URL”>, where URL is the location of the resource to which you want the link to point. So, the HREF attribute of the Anchor element specifies a URL.

## **JAVA SCRIPT**

Java Script is Netscape’s cross-platform, object-based scripting language for client server application. JavaScript is mainly used as a client-side scripting language. This means that JavaScript code is written into an HTML page. When a user requests an HTML page with JavaScript in it, the script is sent to the browser and it’s up to the browser to do something with it. JavaScript can be used in other contexts than a Web browser. Netscape created server-side JavaScript as a CGI-language that can do roughly the same as Perl or ASP.

Fortunately, most browsers can handle JavaScript nowadays, but of course some browsers do not support some bits of script.



## **Types of JavaScript**

- a. Navigator Java Script also called client-side Java Script.
- b. Live Wire Java Script also called server-side Java Script.

## **Features of JavaScript (JS)**

- ✓ Browser interprets JavaScript.
- ✓ JavaScript is object based and uses built-in, extensible objects and have no classes or inheritance
- ✓ JavaScript is loosely typed language
- ✓ In JavaScript object reference are checked at runtime
- ✓ JavaScript is designed to supplement the capabilities of HTML with script that are capable of responding to web pages events. JSP has access to some extent of aspects of the web browser window.
- ✓ JavaScript control browser and content but cannot draw graphics or perform networking.

## **The Client-Side JavaScript also has the following features**

- ✓ Controls Document's appearance and content
- ✓ Control the browser
- ✓ Interact with the HTML forms
- ✓ Interact with the user
- ✓ Read and write client state with cookies
- ✓ Server- Side JavaScript Features
- ✓ Embedded in HTML page
- ✓ Executed at the server
- ✓ Pre-compiled for faster response
- ✓ Access to Server-side objects
- ✓ Encapsulation of the request

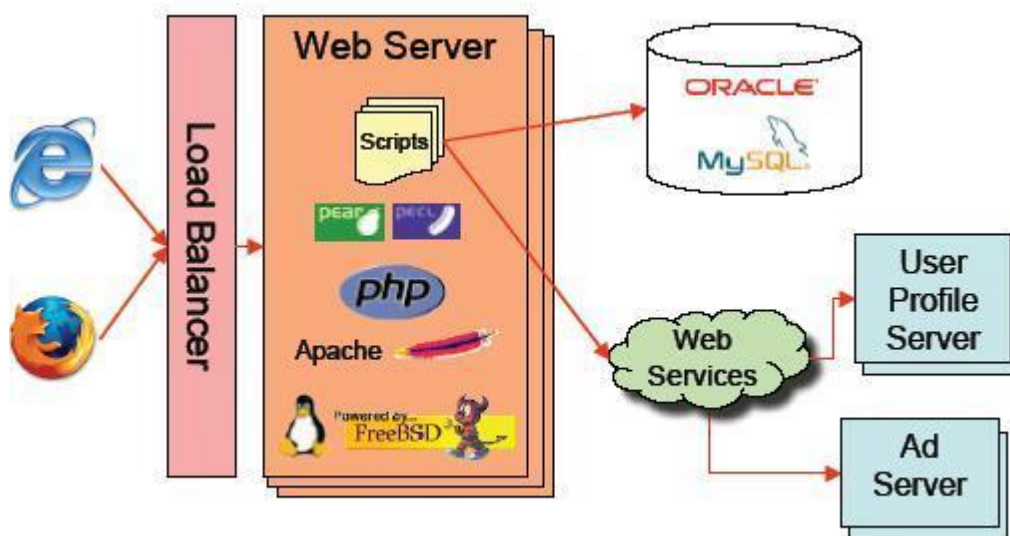
## jQuery:

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript. Support from our corporate members makes it possible for the jQuery Foundation to continue our work on our JavaScript libraries and pushing the open web forward with events and participation in the standards process.

## PHP

PHP was originally an acronym for Personal Home Pages, but is now a recursive acronym for PHP: Hypertext Preprocessor. PHP was originally developed by the Danish Greenlander Rasmus Lerdorf, and was subsequently developed as open source. PHP is not a proper web standard but an open-source technology. PHP is neither real programming language – but PHP lets you use so called scripting in your documents. To describe a PHP page, we could say that it is a file with the extension. PHP that contains a combination of HTML tags and scripts that run on a web server.

### Architecture of PHP:



## **How does PHP work?**

The best way to explain how PHP works is by comparing it with standard HTML. Imagine you type the address of an HTML document (e.g. `http://www.mysite.com/page.htm`) in the address line of the browser. This way you request an HTML page.

The server first reads the PHP file carefully to see if there are any tasks that need to be executed. Only when the server has done what it is supposed to do, the result is then sent to the client. It is important to understand that the client only sees the result of the server's work, not the actual instructions. This means that if you click "view source" on a PHP page, you do not see the PHP codes only basic HTML tags.

## **Comments in PHP:**

A comment in PHP code is a line that is not read/executed as part of the program. Its only purpose is to be read by someone who is editing the code.

### **Comments are useful for:**

Comments let other programmers understand what you were doing in each step. Most programmers have experienced coming back to their own work a year or two later and having to re-figure out what they did. Comments can remind you of what you were thinking when you wrote the code.

## **UNIQUE FEATURES OF PHP:**

PHP language has support features of other languages like c, Perl and etc. It also has some unique features of its own. Some of them are listed below.

- ✓ In PHP there is no need to specify data type for variable declaration. Rather, it can be determined at the time of execution depends on the value of the variable. So that, PHP is called as loosely typed language.
- ✓ PHP provides cross platform compatibility, unlike some other server-side scripting language.
- ✓ PHP has set of predefined variables called super global which will be start by `_`. Some of the examples are, `$_GET`, `$_POST`, `$_COOKIE`, `$_SESSION`, `$_SERVER` and etc. So, any variable except super global, that are start with `_` will cause error.
- ✓ PHP programming structure includes variable variables; that is, the name of the variable can be change dynamically.

- ✓ This language contains access monitoring capability to create logs as the summary of recent accesses.
- ✓ And then, it includes several magic methods that begins with `__` character which will be defined and called at appropriate instance. For example, `__clone ()` will be called, when the clone keyword is used.
- ✓ Predefined error reporting constants are available to generate a warning or error notice. For example, when `E_STRICT` is enabled, a warning about deprecated methods will be generated.
- ✓ PHP supports extended regular expression that leads extensive pattern matching with remarkable speed.
- ✓ And then, properties like, nowdocs and heredocs are used to delimit some block of context which should not be sent for parsing.
- ✓ Since PHP is a single inheritance language, the parent class methods can be derived by only one directly inherited sub class. But, the implementation of traits concept, reduce the gap over this limitation and allow to reuse required method in several classes.

### **Advantages of PHP:**

Open source: It is developed and maintained by a large group of PHP developers, this will help in creating a support community, abundant extension library.

- ✓ **Speed:** It is relative fast since it uses much system resource.
- ✓ **Easy to use:** It uses C like syntax, so for those who are familiar with C, it's very easy for them to pick up and it is very easy to create website scripts.
- ✓ **Stable:** Since it is maintained by many developers, so when bugs are found, it can be quickly fixed.
- ✓ **Powerful library support:** You can easily find functional modules you need such as PDF, Graph etc.
- ✓ **Built-in database connection modules:** You can connect to database easily using PHP, since many websites are data/content driven, so we will use database frequently, this will largely reduce the development time of web apps.
- ✓ Can be run on many platforms, including Windows, Linux and Mac, it's easy for users to find hosting service providers.

## **4. SYSTEM DESIGN AND DEVELOPMENT**

### **4.1 DESCRIPTION OF MODULES**

#### **✓ Administrator functions:**

- Generating receipts of one-time customers/scan receipts
- Updating database
- Maintain the structure
- View parking slot generation charts to allocate location to customers
- Payment of the employees
- Checking the inventory
- Attendance of employees
- Generation of payment slips of employees

#### **✓ Customer login facility:**

- View parking company details
- View rule of the parking
- View parking rates
- Create account

#### **✓ Parker's details:**

In this module car parkers details that is user details and their car number and personal details are stored. Access to the Park Management System is controlled by the use of a user name and password. This provides the necessary security for parkers allowing them the ability to control their car park requests and vehicle details. Those members of parkers who currently have an annual car parking permit will already have been sent a user name and password. However, those members of parker who do not currently have access to the system will be required to register on the system prior to using it. If you do not have a current car park permit user should register their details. If you have already registered, (this will include all current parkers permit holders) please proceed to the Login for Registered.

### ✓ **Manage parking spaces /ALLOTMENT DETAILS**

The car parking management system can schedule both staff parking and visitor parking in one, shared car park. And the maintain the slot of the parked and unparked areas are maintained properly and it helps us to maximize the parking space. If no spaces are available for the time slot chosen in a particular car park the car park name will not be displayed. Only car parks with available spaces will be displayed on this list. No other visitor spaces will be provided if all visitor car parking bays are used. If none are available the user can re-request for a different timeslot if required.

Having selected the car park required the user should now complete the name of the person visiting and the name of the person being visited. These fields are mandatory and no space will be allocated until they have been completed.

### ✓ **Billing details**

The **car parking management software** may be configured to enforce you may wish to limit how many hours a day or a week someone may park and monthly bill and bill by parking can be generated. Payment candidate can pay both ways online and offline.

## **4.2 DATA FLOW DIAGRAM**

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.

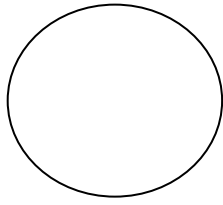
The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations.

Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical form, this led to the modular design. A DFD is also known as a “bubble Chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So, it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

### **CONSTRUCTING A DFD**

Several rules of thumb are used in drawing DFD'S

- ✓ Process should be named and numbered for an easy reference. Each name should be representative of the process.
- ✓ The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way to indicate this is to draw long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD, it is marked with a short diagonal.



Process



Source or Destination of data



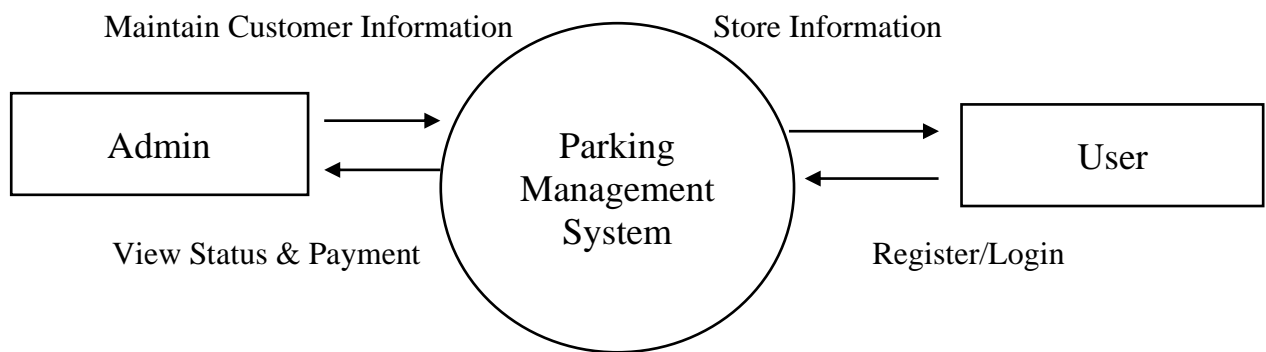
Data Flow



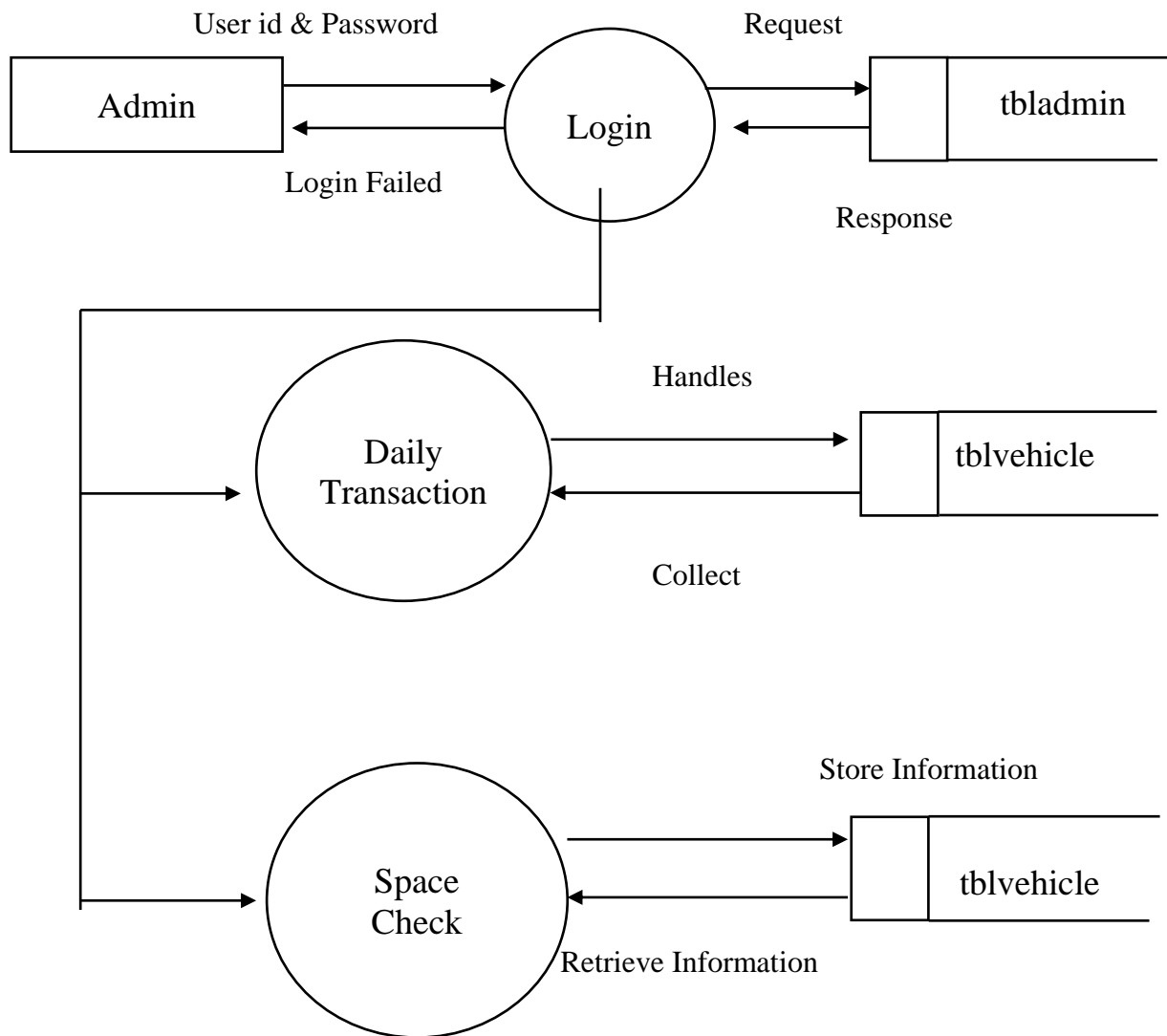
Database



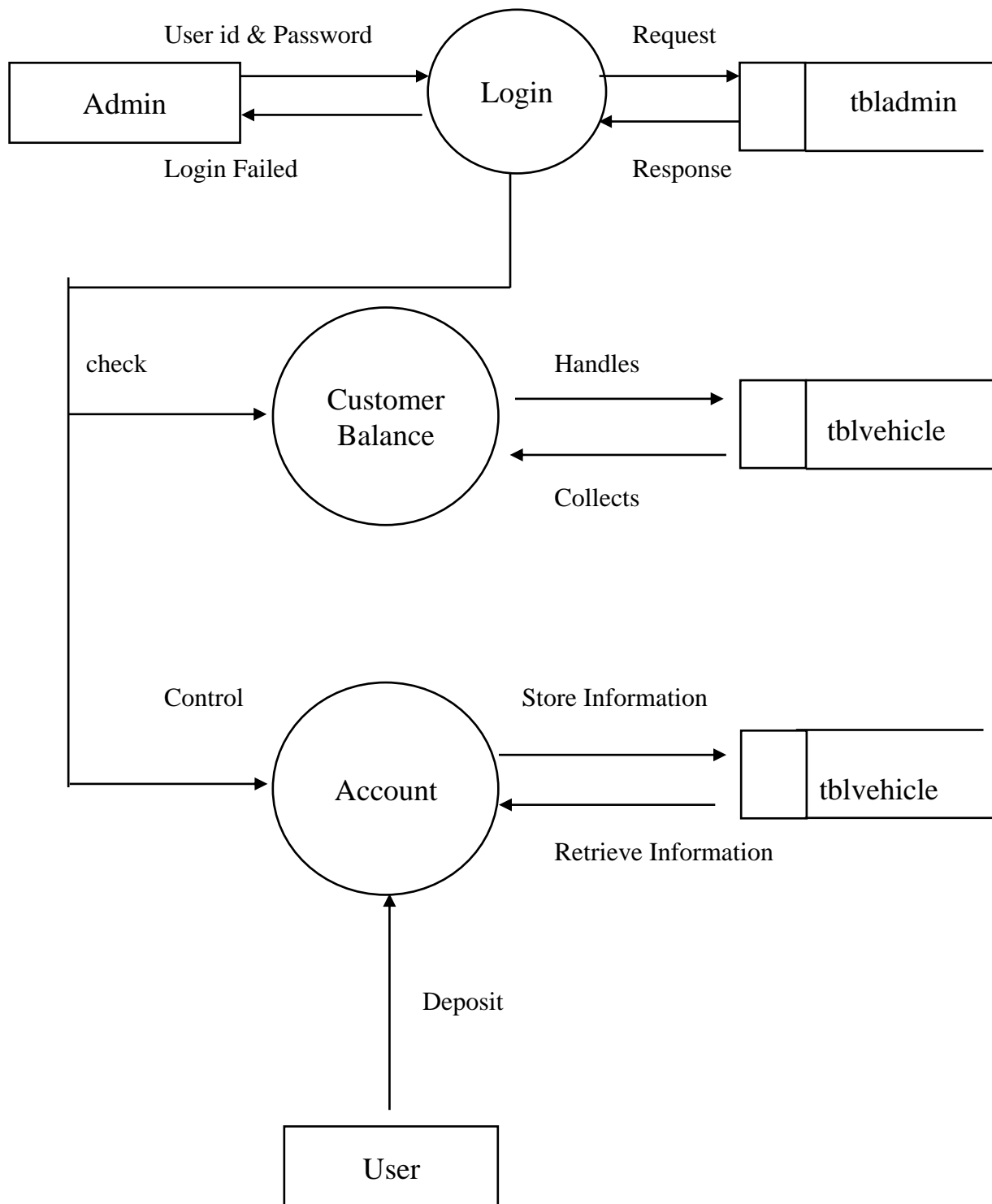
## Level 0



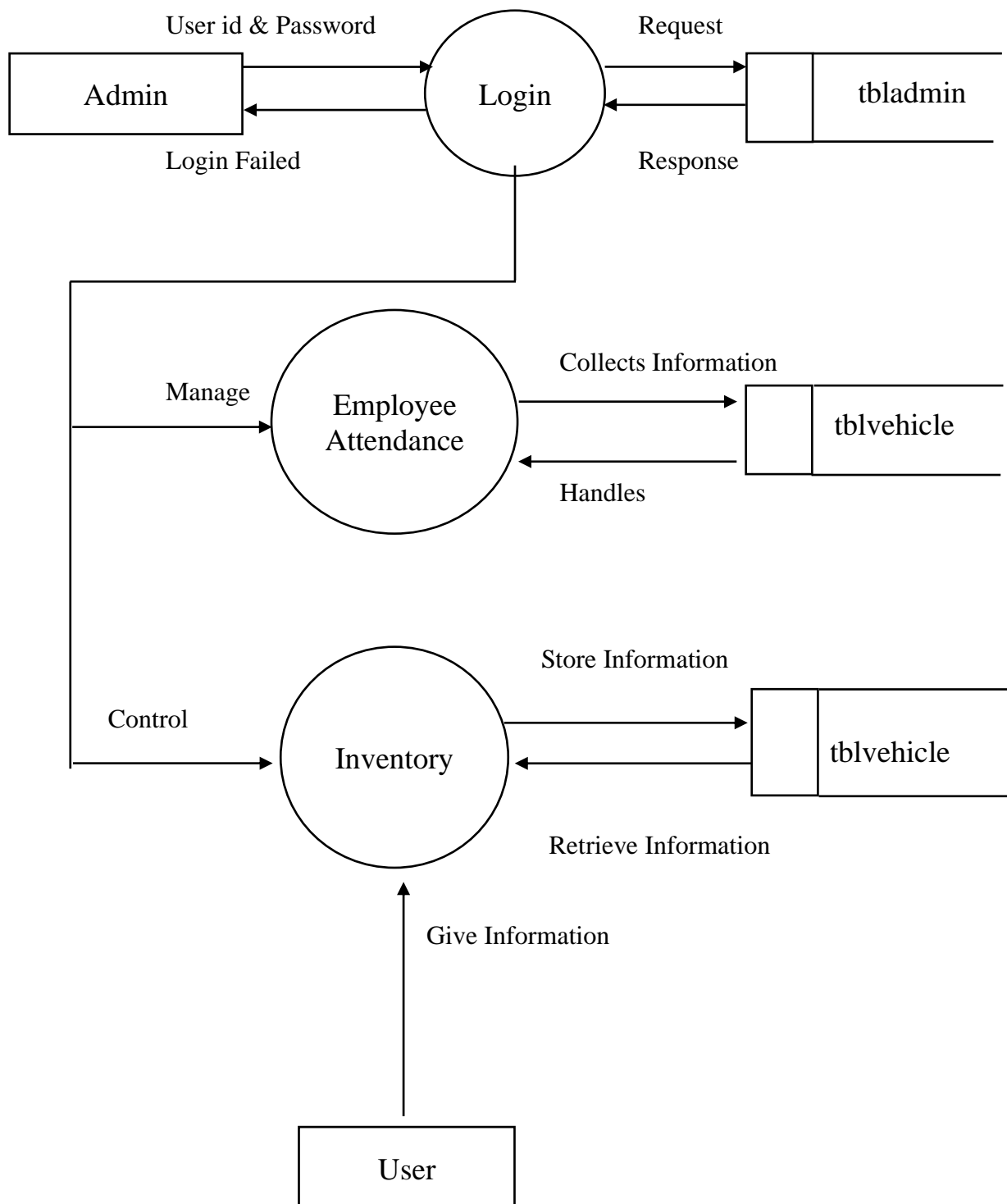
## Level 1



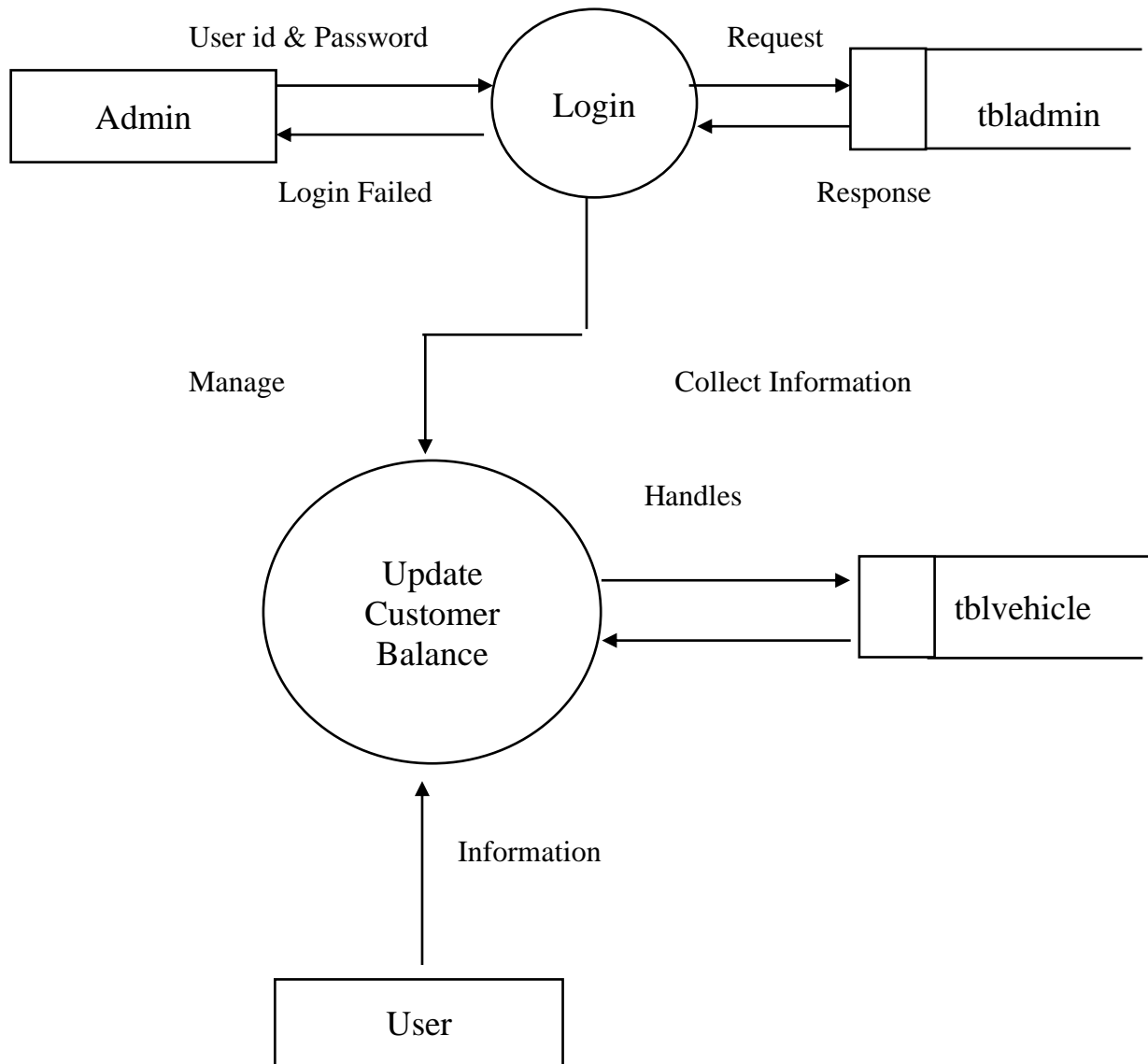
## Level 2



### Level 3



## Level 4



## 4.3 DATABASE DESIGN

A database management (DBMS) is computer software designed for the purpose of managing databases, a large set of structured data and run operations on the data requested by numerous users. Typical examples of DBMSs include Oracle, DB2, Microsoft SQL Server, Microsoft Access, Firebird, MySQL, FileMaker and Sybase Adaptive Server Enterprise. DBMSs are typically used by Database administrators in the creation of Database systems. Typical examples of DBMS use include accounting, human resources and customer support systems.

Originally found only in large companies with the computer hardware needed to support large data sets, DBMSs have more recently emerged as a fairly standard part of any company back office.

### **Description:**

A DBMS is a complex set of software programs that controls the organization, storage, management, and retrieval of data in a database. A DBMS includes:

- ✓ A modeling language to define the schema of each database hosted in the DBMS, according to the DBMS data model.
  - The four most common types of organizations are the hierarchical, network, relational and object models. Inverted lists and other methods are also used.
  - The dominant model in use today is the abhor one embedded in sql, despite the objections of purists who believe this model is a corruption of the relational model, since it violates several of its fundamental principles for the sake of practicality and performance. Many DBMSs also support the Open Database Connectivity API that supports a standard way for programmers to access the DBMS.
- ✓ Data structures optimized to deal with very large amounts of data stored on a permanent data storage device
- ✓ A database query language and report writer to allow users to interactively interrogate the database, analyze its data and update it according to the users' privileges on data.

- It also controls the security of the database.
- Data security prevents unauthorized users from viewing or updating the database. Using passwords, users are allowed access to the entire database or subsets of it called sub-schemas. For example, an employee database can contain all the data about an individual employee, but one group of users may be authorized to view only payroll data, while others are allowed access to only work history and medical data.

If the DBMS provides a way to interactively enter and update the database, as well as interrogate it, this capability allows for managing personal databases. However, it may not leave an audit trail of actions or provide the kinds of controls necessary in a multi-user organization. These controls are only available when a set of application programs are customized for each data entry and updating function.

A transaction mechanism that ideally would guarantee the ACID properties, in order to ensure data integrity, despite concurrent user accesses and faults.

- It also maintains the integrity of the data in the database.
- The DBMS can maintain the integrity of the database by not allowing more than one user to update the same record at the same time, The DBMS can help prevent duplicate records via unique index constraints; for example, no two customers with the same customer numbers can be entered into the database. See ACID properties for more information.

The DBMS accepts requests for data from the application program and instructs the operating system to transfer the appropriate data.

SQL is a database computer language designed for the retrieval and management of data in relational database. SQL stands for Structured Query Language. In the relational model, data is stored in structures called relations or tables. SQL, statements are issued for the purpose of:

**Data Definition:** Defining tables and structures in the database

**Data Manipulation:** Used to manipulate the data within those schema objects

A schema is a collection of database objects that can include: tables, views, indexes and sequences

List of SQL statements that can be issued against an Oracle database schema are:

**ALTER** : Change an existing table, view or index definition

**AUDIT** : Track the changes made to a table

**COMMENT** : Create new database objects such as tables or views

**DELETE** : Delete rows from a database table

**DROP** : Drop a database object such as table, view or index

**GRANT** : Allow another user to access database objects such as tables or views

**INSERT** : Insert new data into a database table

**NO AUDIT** : Turn off the auditing function

**REVOKE** : Disallow a user access to database objects such as tables and views

**ROLLBACK** : Undo any recent changes to the database

**SELECT** : Retrieve data from a database table

**TRUNCATE** : Delete all rows from a database table



## 4.4 TABLE STRUCTURE

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and effectively. After designing the input and output, the analyst must concentrate on database design or how data should be organized around user requirements. The general objective is to make information access, easy, quick, inexpensive and flexible for other users. Database is designed in such a way that it is used for efficiently storing and maintaining the bidding and user details, hence to provide proper service and to maintain auctions effectively.

**Table Name** : tbladmin

**Primary Key** : Id

S.No	Column Name	Data type	Description	Constraint
1	Id	int(10)	Contain id	Primary key
2	AdminName	varchar(120)	Contain full name	Not null
3	UserName	varchar(120)	Contain user name	Not null
4	MobileName	bigint(10)	Contain user Mobile number	Not null
5	Email	varchar(200)	Contain user email id	Not null
6	Password	varchar(120)	Contain user password	Not null
7	AdminRegdate	timestamp	Contain reg date and time	Not null

**Table Name** : tblcategory

**Primary Key** : Id

S,No	Column Name	Data Type	Description	Constraint
1	Id	int(10)	Contain vehicle id	Primary key
2	VehicleCat	varchar(120)	Contain vehicle category	Not null
3	CreationDate	timestamp	Contain creation date and time	Not null

**Table Name** : tblvehicle

**Primary Key** : Id

S.No	Name	Data Type	Description	Constraint
1	Id	int(10)	Contain id	Primary Key
2	ParkingNumber	varchar(120)	Contain parking number	Not null
3	VehicleCategory	varchar(120)	Contain vehicle category	Not null
4	VehicleCompanyName	varchar(120)	Contain vehicle company name	Not null
5	RegistrationNumber	varchar(120)	Contain Registration Number	Not null
6	OwnerName	varchar(120)	Contain owner name	Not null
7	OwnerContactNumber	bigint(10)	Contain owner contact number	Not null
8	InTime	timestamp	Contain vehicle in time	Not null
9	OutTime	timestamp	Contain vehicle out time	Not null
10	ParkingCharge	varchar(120)	Contain vehicle parking charge	Not null
11	Remark	mediumtext	Contain remark	Not null
12	Status	varchar(5)		Not null

## **4.5 INPUT DESIGN**

Input is any data or instructions entered into the memory of a computer. Two types of input are data and instructions. Data is a collection of unorganized items that can include words, numbers, pictures, sounds, and video. A computer processes data into information, which is organized, meaningful, and useful. Instructions can be in the form of programs, commands, or user responses. A program is a series of instructions that tells a computer how to perform the tasks necessary to process data into information. A command is an instruction given to a computer program.

### **Login Page**

By using username and password we can login to the dashboard

### **Profile Page**

In profile page we can update user details and click update button to update.

### **Add Category**

In category page we can add category and click add button to add category in the vehicle category table.

### **Update Category**

In update category page we can update vehicle category and click update button to update vehicle category in category table.

### **Add Vehicle**

In Add vehicle page we give inputs that is select, vehicle company, Reg No, Owner name and owner contact details and click add button to add vehicle.

### **Search Vehicle**

By entering vehicle reg no we can search vehicle.

## **4.6 OUTPUT DESIGN**

Output design will be done through reports in the system. The system's output will be generated either through reports or through system prompted messages. The system will be designed in such a way that output forms will be designed for displaying outputs to the user. Output will be generally displayed to the user through monitor as visual display. Output can also be an attached email or bill from the printer. Some of the outputs can be audio which will be delivered by the system. In the web-based auction project, outputs are designed in such a way that, it will be displayed to the user through monitor.

### **Dashboard**

We can view the entire admin panel in dashboard once we login.

### **Manage Income vehicle**

We can edit and delete the properties of vehicles.

### **Manage Outgoing Vehicle**

We can view the outgoing vehicle and print receipt through printer.

### **Parking Receipt**

It contains Vehicle information like incoming and outgoing time and payments etc.

## **5. SYSTEM TESTING AND IMPLEMENTATION**

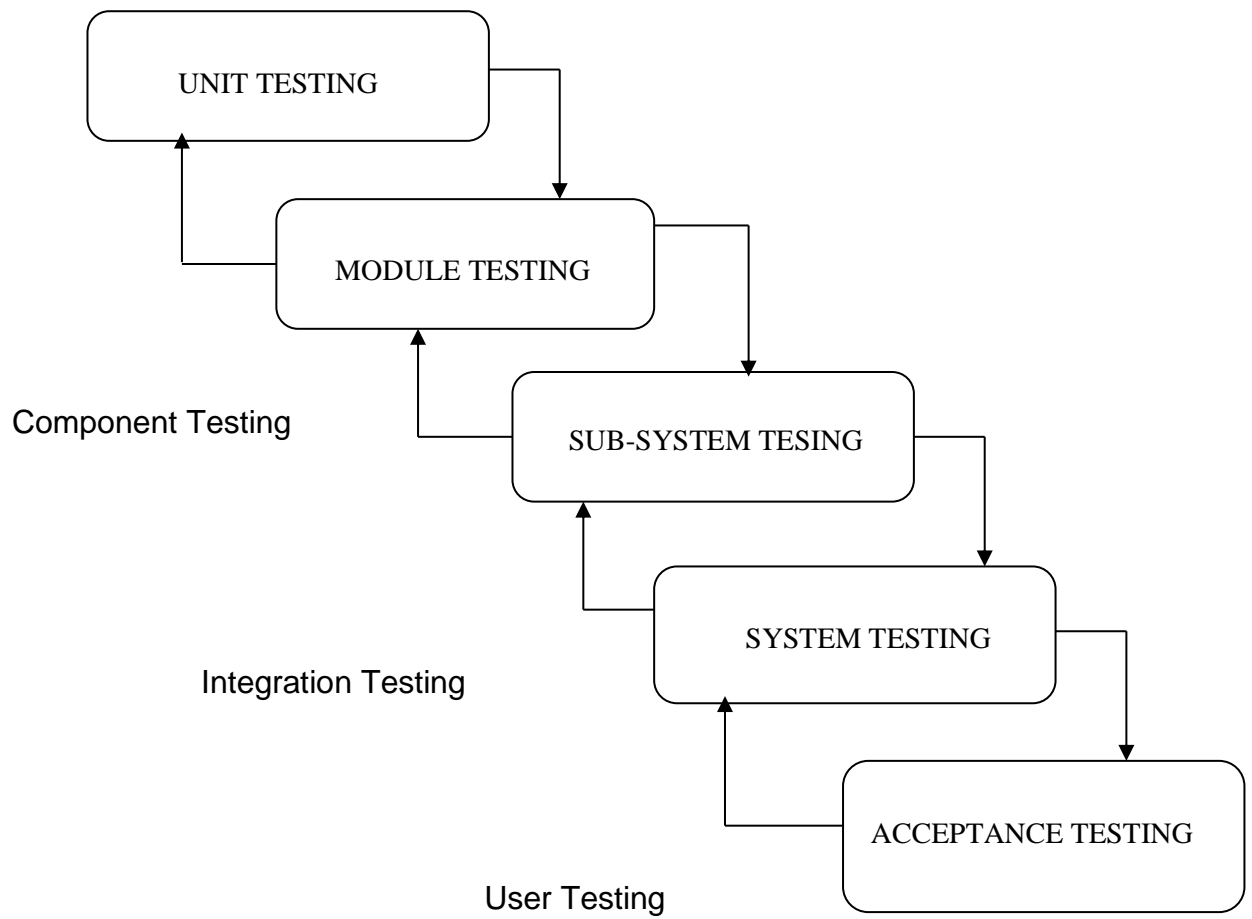
Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

### **STRATEGIC APPROACH TO SOFTWARE TESTING**

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software, we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally, we arrive at system testing, where the software and other system elements are tested as a whole.



## **Testing Methods**

The different types of testing are: -

- ✓ Unit Testing
- ✓ Integration Testing
- ✓ Validation Testing
- ✓ Output Testing
- ✓ User Acceptance Testing

### **5.1 UNIT TESTING**

Unit testing focuses verification efforts on the smallest unit of software design, the module. This is also known as “Module Testing” The modules are tested separately this testing is carried out during programming stage itself. In this step each module is found to be working satisfaction as regard to the expected output from the module.

### **5.2 INTEGRATION TESTING**

Integration testing focuses on the design and construction of the software architecture. Data can be lost across an interface; one module can have adverse effect on another sub functions and show on. Thus, integration testing is a systematic technique for constructing test to uncover errors associated with in the interface. In this project, all the modules are companied and then the entire program is tested as a whole.

### **5.3 VALIDATION TESTING**

Validation testing is the requirement established as a part of software requirement analysis is validated against the software that has been constructed. 33 This test provides the final assurance whether the software needs all functional, behavioral and performance requirements. Thus, the proposed system under consideration has been tested by using validation testing and found to be working satisfactory.

### **5.4 OUTPUT TESTING**

After performing the validation testing, the next step is the output testing of the proposed system, since no system could be useful if it does not produce required output in the specific format. Tested asking the users about the format required by them, the output is considered

into two ways: one is on the screen and the other is printed format. The output format on the screen is found to be correct as the format designed according to the user needs, for the hard copy also, the output comes as specified by the user. Hence output testing does not result in correction in the system.

## **5.5 USER ACCEPTANCE TESTING**

User acceptance testing of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keep in touch with the prospective system user at time of developing and making changes wherever required.

## **5.6 WHITEBOX TESTING**

White box Testing is done with the project which drive test cases that do the following

- ✓ Guarantee that all the independent paths with in modules have been exercise at least once.
- ✓ Exercise all logical decision on the true and false side.
- ✓ Execute all loops at the boundaries and within their operation bounds.
- ✓ Exercise internal data structures to ensure the validity It is aimed at ensuring that the system works accurately and efficiently before live operation command.

## **5.7 BLACKBOX TESTING**

Black box System methods focus on the functional requirement of the software. Using the black box testing method, the following errors are identified and rectified in the package.

- ✓ Incorrect or Missing functions
- ✓ Interface Errors
- ✓ Errors in data Structures or external database access.



## **6. CONCLUSION**

This Application provides a computerized version of Car Parking Management System which will benefit the parking premises.

It makes entire process online and can generate reports. It has a facility of staff's login where staff can fill the visitor details and generate report.

The Application was designed in such a way that future changes can be done easily. The following conclusions can be deduced from the development of the project.

Automation of the entire system improves the productivity and provides a friendly graphical user interface which proves to be better when compared to the existing system then it gives appropriate access to the authorized users depending on their permissions. And effectively overcomes the delay in communications, updating of information becomes so easier.

System security, data security and reliability are the striking features.

Thus, the System has adequate scope for modification in future if it is necessary.

## **7. SCOPE OF FUTURE ENHANCEMENT**

Every application has its own merits and demerits. The project has covered almost all the requirements. Further requirements and improvements can easily be done since the coding is mainly structured or modular in nature. Changing the existing modules or adding new modules can append improvements. In future this application will be developed android application with advanced features and upload in playstore and the parking allotments can be sent through SMS.

## 8. BIBLIOGRAPHY

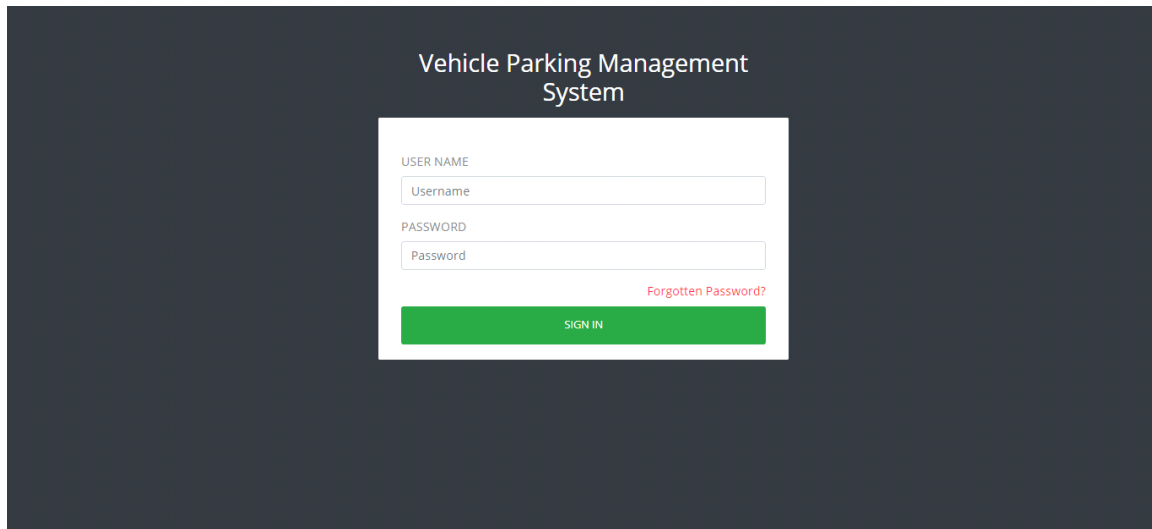
### REFERENCES

- ✓ Australian Computer Society, 2003, ASC Code of Ethics. Retrieved March 15 2007, from <http://www.acs.org.au.htm>
- ✓ Elmira, R. and Nava the, S. 2004. Enhanced Entity Relationship and UML. In Fundamentals of Database Systems, 3rd Edition
- ✓ Out Source 2 India n.d. :Why Do Software Projects Fail? Retrieved 22 March 2007 from <http://www.outsource2india.com/software/SoftwareProjectFailure.as>
- ✓ Six Sigma n.d. : Applying Six Sigma to Software Implementation Projects Retrieved 22March 2007 from <http://software.isixsigma.com/library/content/c040915b.asp>
- ✓ Somerville, Ian 2004. Object Oriented Design Software Engineering, 7thEdition
- ✓ Ramakrishnan, R.and Gehrke, J. 2003. The Relational Model In Database Management Systems, 3rdEdition
- ✓ **Systems Analysis and Design** - Second Edition, 1998; By Elias M. Awad
- ✓ **Software Engineering** - Fifth Edition, 2001; By Roger S. Pressman, Ph.D.
- ✓ **SQL for SQL Server** - First Indian Edition 2003; By Bryan Syverson
- ✓ **Visual Basic 6 Database How – To** - First Indian Edition 1999; By Eric Winemiller, Jason Roff, Bill Heyman, Ryan Groom
- ✓ **MySQL, PL/SQL** - Second Revised Edition; By Ivan Bayross

## 9. APPENDICES

### A. INPUT DESIGN

#### Login Page



The image shows a login page for the 'Vehicle Parking Management System'. The page has a dark blue background. In the center, there is a white rectangular box containing the login form. The form has two input fields: 'USER NAME' with a placeholder 'Username' and 'PASSWORD' with a placeholder 'Password'. Below the password field, there is a red link that says 'Forgotten Password?'. At the bottom of the form, there is a green button labeled 'SIGN IN'.

Vehicle Parking Management  
System

USER NAME  
Username

PASSWORD  
Password

[Forgotten Password?](#)

SIGN IN

# Profile

Admin

Dashboard

Vehicle Category >

Add Vehicle

Manage Vehicle >

Reports >

Search Vehicle

Dashboard

Dashboard / Profile / Admin Profile

Admin Profile

Admin Name

Admin

User Name

admin

Contact Number

7898799798

Email

tester1@gmail.com

Update

Copyright © 2019

Vehicle Parking Management System

## Add Category

Admin

Dashboard

Vehicle Category >

Add Vehicle

Manage Vehicle >

Reports >

Search Vehicle

Dashboard / Category / Add Category

Add Category

Category Name

Vehicle Category

Add

Copyright © 2019

Vehicle Parking Management System

## Update Category

Admin

Dashboard

Vehicle Category >

Add Vehicle

Manage Vehicle >

Reports >

Search Vehicle

Dashboard / Category / Update Category

Update Category

Category Name

Four Wheeler Vehicle

Update

Copyright © 2019

Vehicle Parking Management System

## Add Vehicle

Admin

Dashboard

Vehicle Category >

Add Vehicle

Manage Vehicle >

Reports >

Search Vehicle

Dashboard

Dashboard / Vehicle / Add Vehicle

Add Vehicle

Select

Select Category

Vehicle Company

Vehicle Company

Registration Number

Registration Number

Owner Name

Owner Name

Owner Contact Number

Owner Contact Number

Add

Copyright © 2019

Vehicle Parking Management System

## Between Dates Reports

Admin

Dashboard

Vehicle Category >

Add Vehicle

Manage Vehicle >

Reports >

Search Vehicle

Dashboard

Dashboard / Reports / Between Dates Reports

Between Dates Reports

From Date

dd-mm-yyyy

To Date

dd-mm-yyyy

Submit

Copyright © 2019

Vehicle Parking Management System

# Search Vehicle

Admin

Dashboard

Vehicle Category >

Add Vehicle

Manage Vehicle >

Reports >

Search Vehicle

Dashboard / Search Vehicle / Search Vehicle

Search Vehicle

Search By Parking Number

Search

Result against "665566590" keyword

S.NO	Parking Number	Owner Name	Vehicle Reg. Number	Action
1	665566590	Test	Del-98789789	View

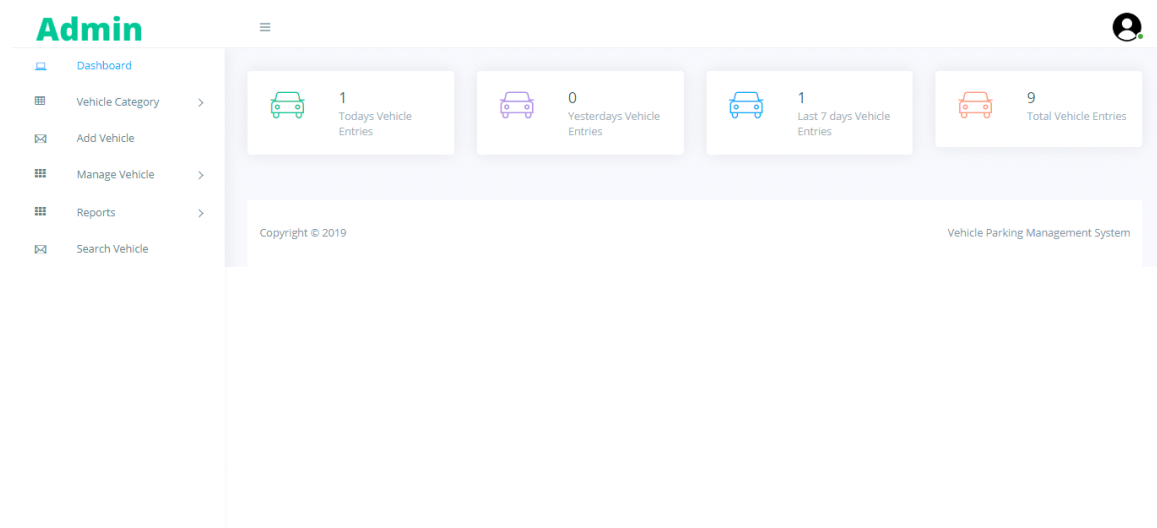
Copyright © 2019

Vehicle Parking Management System

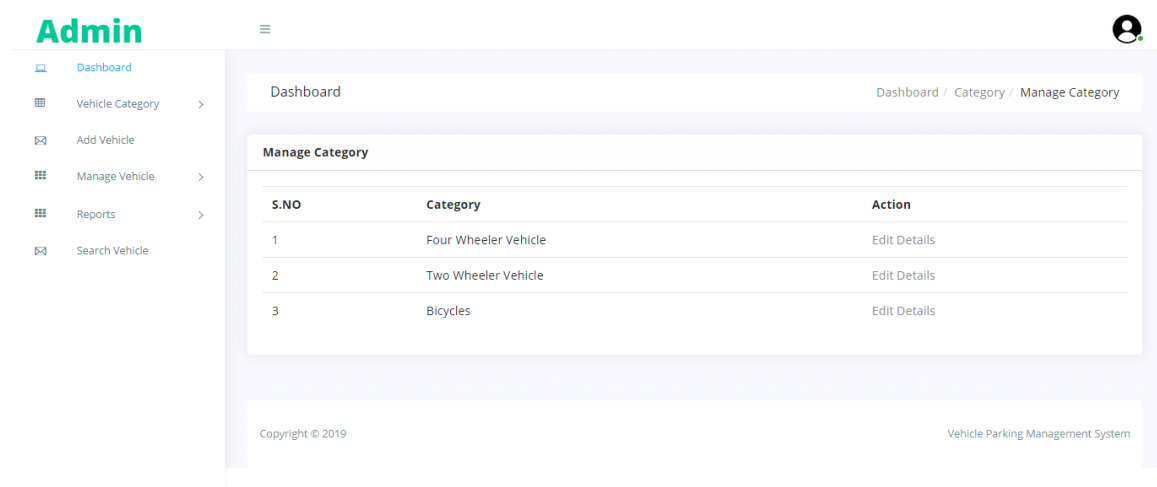


## B) OUTPUT REPORT

### Dashboard



### Manage Category



# Manage Incoming Vehicle

Admin

Dashboard

Vehicle Category >

Add Vehicle

Manage Vehicle 

Manage In Vehicle

Manage Out Vehicle

Reports >

Search Vehicle

Dashboard / Manage Vehicle / Manage Incoming Vehicle

Manage Incoming Vehicle

S.NO	Parking Number	Owner Name	Vehicle Reg Number	Action
1	323009894	Abhi	DEL-55776	<a href="#">View</a>   <a href="#">Print</a>
2	522578915	Mahesh	DEL-895623	<a href="#">View</a>   <a href="#">Print</a>
3	917725207	ABC	DL 1c RT2323	<a href="#">View</a>   <a href="#">Print</a>
4	665566590	Test	Del-98789789	<a href="#">View</a>   <a href="#">Print</a>

Copyright © 2019

Vehicle Parking Management System

# View Incoming Vehicle

Admin

Dashboard

Vehicle Category >

Add Vehicle

Manage Vehicle >

Reports >

Search Vehicle

Dashboard / View Vehicle / Incoming Vehicle

View Incoming Vehicle

Parking Number	665566590
Vehicle Category	Two Wheeler Vehicle
Vehicle Company Name	Honda
Registration Number	Del-98789789
Owner Name	Test
Owner Contact Number	8989898898
In Time	2019-11-21 10:48:38
Status	Vehicle In

Remark :

Parking Charge:

Status :


Outgoing Vehicle

Update

Copyright © 2019

Vehicle Parking Management System

## Parking Receipt

Vehicle Parking receipt			
Parking Number	665566590	Vehicle Category	Two Wheeler Vehicle
Vehicle Company Name	Honda	Registration Number	Del-98789789
Owner Name	Test	Owner Contact Number	8989898898
In Time	2019-11-21 10:48:38	Status	Incoming Vehicle
			

## Manage Outgoing Vehicle

Admin

Dashboard

Vehicle Category >

Add Vehicle

Manage Vehicle >

Reports >

Search Vehicle

Dashboard / Manage Vehicle / Manage Outgoing Vehicle

Manage Outgoing Vehicle

S.NO	Parking Number	Owner Name	Vehicle Reg Number	Action
1	521796069	Rakesh Chandra	DEL-678787	View   Print
2	469052796	Pankaj	DEL-895623	View   Print
3	734465023	Avinash	DEL-562389	View   Print
4	432190880	Harish	DEL-451236	View   Print
5	486258836	Test User	DL 1C TY2322	View   Print

Copyright © 2019

Vehicle Parking Management System

# View Outgoing Vehicle

Admin

Dashboard

Vehicle Category >

Add Vehicle

Manage Vehicle >

Reports >

Search Vehicle

Dashboard

Dashboard / View Vehicle / Incoming Vehicle

View Outgoing Vehicle

Parking Number	486258836
Vehicle Category	Two Wheeler Vehicle
Vehicle Company Name	Honda Activa
Registration Number	DL 1C TY2322
Owner Name	Test User
Owner Contact Number	1234567890
In Time	2019-07-07 17:02:02
Out Time	2019-07-07 17:02:42
Remark	Vehicle Out
Status	Out
Parking Fee	40

Copyright © 2019

Vehicle Parking Management System

# View Report

Admin

Dashboard

Vehicle Category >

Add Vehicle

Manage Vehicle >

Reports >

Search Vehicle

Dashboard

Dashboard / Between Date Reports / Between Date Reports

Between Date Reports

Report from 2019-11-01 to 2019-11-21

S.NO	Parking Number	Owner Name	Vehicle Reg Number	Action
1	665566590	Test	Del-98789789	View

Copyright © 2019

Vehicle Parking Management System

## C. SAMPLE CODING

### index.php

```
<?php
session_start();
error_reporting(0);
include('includes/dbconnection.php');

if(isset($_POST['login']))
{
    $adminuser=$_POST['username'];
    $password=md5($_POST['password']);
    $query=mysqli_query($con,"select ID from tbladmin where  UserName='$adminuser' &&
Password='$password' ");
    $ret=mysqli_fetch_array($query);
    if($ret>0){
        $_SESSION['vpmsaid']=$ret['ID'];
        header('location:dashboard.php');
    }
    else{
        $msg="Invalid Details.";
    }
}
?>
<!doctype html>
<html class="no-js" lang="">
<head>
    <title>VPMS-Login Page</title>
    <link rel="apple-touch-icon" href="https://i.imgur.com/QRAUqs9.png">
    <link rel="shortcut icon" href="https://i.imgur.com/QRAUqs9.png">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/normalize.css@8.0.0/normalize.min.css">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/css/bootstrap.min.css">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/font-awesome@4.7.0/css/font-
awesome.min.css">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/gh/lykmapipo/themify-
icons@0.1.2/css/themify-icons.css">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/pixeden-stroke-7-icon@1.2.3/pe-
icon-7-stroke/dist/pe-icon-7-stroke.min.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/flag-icon-
css/3.2.0/css/flag-icon.min.css">
    <link rel="stylesheet" href="assets/css/cs-skin-elastic.css">
    <link rel="stylesheet" href="assets/css/style.css">
    <link href='https://fonts.googleapis.com/css?family=Open+Sans:400,600,700,800'
rel='stylesheet' type='text/css'>
    <!-- <script type="text/javascript"
src="https://cdn.jsdelivr.net/html5shiv/3.7.3/html5shiv.min.js"></script> -->
</head>
```

```

<body class="bg-dark">
  <div class="sufee-login d-flex align-content-center flex-wrap">
    <div class="container">
      <div class="login-content">
        <div class="login-logo">
          <a href="index.php">
            <h2 style="color: #fff">Vehicle Parking Management System</h2>
          </a>
        </div>
        <div class="login-form">
          <form method="post">
            <p style="font-size:16px; color:red" align="center"> <?php if($msg){
echo $msg;
} ?> </p>
            <div class="form-group">
              <label>User Name</label>
              <input class="form-control" type="text" placeholder="Username"
required="true" name="username">
            </div>
            <div class="form-group">
              <label>Password</label>
              <input type="password" class="form-control" name="password"
placeholder="Password" required="true">
            </div>
            <div class="checkbox">
              <label class="pull-right">
                <a href="forgot-password.php">Forgotten Password?</a>
              </label>
            </div>
            <button type="submit" name="login" class="btn btn-success btn-flat m-b-30
m-t-30">Sign in</button>

          </form>
        </div>
      </div>
    </div>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/jquery@2.2.4/dist/jquery.min.js"></script>
  <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.14.4/dist/umd/popper.min.js"></script>
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/js/bootstrap.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/jquery-match-
height@0.7.2/dist/jquery.matchHeight.min.js"></script>
  <script src="assets/js/main.js"></script>
</body>
</html>

```

## dashboard.php

```
<?php
session_start();
error_reporting(0);
include('includes/dbconnection.php');
error_reporting(0);
if (strlen($_SESSION['vpmsaid']==0)) {
    header('location:logout.php');
} else{ ?>
<!doctype html>
<html class="no-js" lang="">
<head>
    <title>VPMS - Admin Dashboard</title>
    <link rel="apple-touch-icon" href="https://i.imgur.com/QRAUqs9.png">
    <link rel="shortcut icon" href="https://i.imgur.com/QRAUqs9.png">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/normalize.css@8.0.0/normalize.min.css">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/css/bootstrap.min.css">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/font-awesome@4.7.0/css/font-
awesome.min.css">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/gh/lykmapipo/themify-
icons@0.1.2/css/themify-icons.css">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/pixeden-stroke-7-icon@1.2.3/pe-
icon-7-stroke/dist/pe-icon-7-stroke.min.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/flag-icon-
css/3.2.0/css/flag-icon.min.css">
    <link rel="stylesheet" href="assets/css/cs-skin-elastic.css">
    <link rel="stylesheet" href="assets/css/style.css">
    <!-- <script type="text/javascript"
src="https://cdn.jsdelivr.net/html5shiv/3.7.3/html5shiv.min.js"></script> -->
    <link href="https://cdn.jsdelivr.net/npm/chartist@0.11.0/dist/chartist.min.css"
rel="stylesheet">
    <link href="https://cdn.jsdelivr.net/npm/jqvmap@1.5.1/dist/jqvmap.min.css"
rel="stylesheet">
    <link href="https://cdn.jsdelivr.net/npm/weathericons@2.1.0/css/weather-icons.css"
rel="stylesheet" />
    <link href="https://cdn.jsdelivr.net/npm/fullcalendar@3.9.0/dist/fullcalendar.min.css"
rel="stylesheet" />
    <style>
    #weatherWidget .currentDesc {
        color: #ffffff!important;
    }
    .traffic-chart {
        min-height: 335px;
    }
    #flotPie1 {
        height: 150px;
    }
    #flotPie1 td {
```



```

        padding:3px;
    }
    #flotPie1 table {
        top: 20px!important;
        right: -10px!important;
    }
    .chart-container {
        display: table;
        min-width: 270px ;
        text-align: left;
        padding-top: 10px;
        padding-bottom: 10px;
    }
    #flotLine5 {
        height: 105px;
    }

    #flotBarChart {
        height: 150px;
    }
    #cellPaiChart{
        height: 160px;
    }

</style>
</head>
<body>
<?php include_once('includes/sidebar.php');?>
<?php include_once('includes/header.php');?>
<!-- Content -->
<div class="content">
    <!-- Animated -->
    <div class="animated fadeIn">
        <!-- Widgets -->
        <div class="row">
            <?php
//todays Vehicle Entries
$query=mysqli_query($con,"select ID from tblvehicle where date(InTime)=CURDATE());";
$count_today_vehentries=mysqli_num_rows($query);
?>

            <div class="col-lg-3 col-md-6">
                <div class="card">
                    <div class="card-body">
                        <div class="stat-widget-five">
                            <div class="stat-icon dib flat-color-1">
                                <i class="pe-7s-car"></i>
                            </div>
                            <div class="stat-content">
                                <div class="text-left dib">

```

```

        <div class="stat-text"><span class="count"><?php echo
$count_today_vehentries?></span></div>
        <div class="stat-heading">Todays Vehicle Entries</div>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="col-lg-3 col-md-6">
    <?php
//Yesterdays Vehicle Entrie
$query1=mysqli_query($con,"select ID from tblvehicle where date(InTime)=CURDATE()-
1;");
$count_yesterday_vehentries=mysqli_num_rows($query1);
?>
        <div class="card">
            <div class="card-body">
                <div class="stat-widget-five">
                    <div class="stat-icon dib flat-color-2">
                        <i class="pe-7s-car"></i>
                    </div>
                    <div class="stat-content">
                        <div class="text-left dib">
                            <div class="stat-text"><span class="count"><?php echo
$count_yesterday_vehentries?></span></div>
                            <div class="stat-heading">Yesterdays Vehicle Entries</div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
</div>
<div class="col-lg-3 col-md-6">
    <?php
//Last Sevendays Vehicle Entries
$query2=mysqli_query($con,"select ID from tblvehicle where
date(InTime)>=(DATE(NOW()) - INTERVAL 7 DAY);");
$count_lastsevendays_vehentries=mysqli_num_rows($query2);
?>
        <div class="card">
            <div class="card-body">
                <div class="stat-widget-five">
                    <div class="stat-icon dib flat-color-3">
                        <i class="pe-7s-car"></i>
                    </div>
                    <div class="stat-content">
                        <div class="text-left dib">
                            <div class="stat-text"><span class="count"><?php echo
$count_lastsevendays_vehentries?></span></div>

```

```

        <div class="stat-heading">Last 7 days Vehicle Entries</div>
    </div>
</div>
</div>
</div>
</div>
</div>
<div class="col-lg-3 col-md-6">
    <?php
//Total Vehicle Entries
$query3=mysqli_query($con,"select ID from tblvehicle");
$count_total_vehentries=mysqli_num_rows($query3);
?>
        <div class="card">
            <div class="card-body">
                <div class="stat-widget-five">
                    <div class="stat-icon dib flat-color-4">
                        <i class="pe-7s-car"></i>
                    </div>
                    <div class="stat-content">
                        <div class="text-left dib">
                            <div class="stat-text"><span class="count"><?php echo
$count_total_vehentries?></span></div>
                            <div class="stat-heading">Total Vehicle Entries</div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
<!-- /Widgets -->
</div>
<!-- .animated -->
</div>
<!-- /.content -->
<div class="clearfix"></div>
<!-- Footer -->
<?php include_once('includes/footer.php');?>
<!-- /#right-panel -->
<!-- Scripts -->
<script src="https://cdn.jsdelivr.net/npm/jquery@2.2.4/dist/jquery.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.14.4/dist/umd/popper.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/js/bootstrap.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/jquery-match-
height@0.7.2/dist/jquery.matchHeight.min.js"></script>
<script src="assets/js/main.js"></script>
<!-- Chart js -->

```

```

<script
src="https://cdn.jsdelivr.net/npm/chart.js@2.7.3/dist/Chart.bundle.min.js"></script>
<!--Chartist Chart-->
<script src="https://cdn.jsdelivr.net/npm/chartist@0.11.0/dist/chartist.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/chartist-plugin-legend@0.6.2/chartist-plugin-
legend.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/jquery.flot@0.8.3/jquery.flot.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/flot-
pie@1.0.0/src/jquery.flot.pie.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/flot-
spline@0.0.1/js/jquery.flot.spline.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/simpleweather@3.1.0/jquery.simpleWeather.min.js"></scri
pt>
<script src="assets/js/init/weather-init.js"></script>
<script src="https://cdn.jsdelivr.net/npm/moment@2.22.2/moment.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/fullcalendar@3.9.0/dist/fullcalendar.min.js"></script>
<script src="assets/js/init/fullcalendar-init.js"></script>
<!--Local Stuff-->
<script>
    jQuery(document).ready(function($) {
        "use strict";

        // Pie chart flotPie1
        var piedata = [
            { label: "Desktop visits", data: [[1,32]], color: '#5c6bc0'},
            { label: "Tab visits", data: [[1,33]], color: '#ef5350'},
            { label: "Mobile visits", data: [[1,35]], color: '#66bb6a'}
        ];
        $.plot('#flotPie1', piedata, {
            series: {
                pie: {
                    show: true,
                    radius: 1,
                    innerRadius: 0.65,
                    label: {
                        show: true,
                        radius: 2/3,
                        threshold: 1
                    },
                    stroke: {
                        width: 0
                    }
                }
            },
            grid: {
                hoverable: true,
                clickable: true
            }
        }
    });

```

```

});
// Pie chart flotPie1 End
// cellPaiChart
var cellPaiChart = [
  { label: "Direct Sell", data: [[1,65]], color: '#5b83de'},
  { label: "Channel Sell", data: [[1,35]], color: '#00bfa5'}
];
$.plot('#cellPaiChart', cellPaiChart, {
  series: {
    pie: {
      show: true,
      stroke: {
        width: 0
      }
    }
  },
  legend: {
    show: false
  },
  grid: {
    hoverable: true,
    clickable: true
  }
});
// cellPaiChart End
// Line Chart #flotLine5
var newCust = [[0, 3], [1, 5], [2,4], [3, 7], [4, 9], [5, 3], [6, 6], [7, 4], [8, 10]];
var plot = $.plot($('#flotLine5'),[{
  data: newCust,
  label: 'New Data Flow',
  color: 'fff'
}],
{
  series: {
    lines: {
      show: true,
      lineColor: 'fff',
      lineWidth: 2
    },
    points: {
      show: true,
      fill: true,
      fillColor: "ffffff",
      symbol: "circle",
      radius: 3
    },
    shadowSize: 0
  },
  points: {
    show: true,
  },

```

```

    legend: {
      show: false
    },
    grid: {
      show: false
    }
  });
// Line Chart #flotLine5 End
// Traffic Chart using chartist
if ($('#traffic-chart').length) {
  var chart = new Chartist.Line('#traffic-chart', {
    labels: ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun'],
    series: [
      [0, 18000, 35000, 25000, 22000, 0],
      [0, 33000, 15000, 20000, 15000, 300],
      [0, 15000, 28000, 15000, 30000, 5000]
    ]
  }, {
    low: 0,
    showArea: true,
    showLine: false,
    showPoint: false,
    fullWidth: true,
    axisX: {
      showGrid: true
    }
  });
  chart.on('draw', function(data) {
    if(data.type === 'line' || data.type === 'area') {
      data.element.animate({
        d: {
          begin: 2000 * data.index,
          dur: 2000,
          from: data.path.clone().scale(1, 0).translate(0,
data.chartRect.height()).stringify(),
          to: data.path.clone().stringify(),
          easing: Chartist.Svg.Easing.easeOutQuint
        }
      });
    }
  });
}
// Traffic Chart using chartist End
//Traffic chart chart-js
if ($('#TrafficChart').length) {
  var ctx = document.getElementById( "TrafficChart" );
  ctx.height = 150;
  var myChart = new Chart( ctx, {
    type: 'line',
    data: {

```

```

labels: [ "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul" ],
datasets: [
  {
    label: "Visit",
    borderColor: "rgba(4, 73, 203,.09)",
    borderWidth: "1",
    backgroundColor: "rgba(4, 73, 203,.5)",
    data: [ 0, 2900, 5000, 3300, 6000, 3250, 0 ]
  },
  {
    label: "Bounce",
    borderColor: "rgba(245, 23, 66, 0.9)",
    borderWidth: "1",
    backgroundColor: "rgba(245, 23, 66,.5)",
    pointHighlightStroke: "rgba(245, 23, 66,.5)",
    data: [ 0, 4200, 4500, 1600, 4200, 1500, 4000 ]
  },
  {
    label: "Targeted",
    borderColor: "rgba(40, 169, 46, 0.9)",
    borderWidth: "1",
    backgroundColor: "rgba(40, 169, 46, .5)",
    pointHighlightStroke: "rgba(40, 169, 46,.5)",
    data: [ 1000, 5200, 3600, 2600, 4200, 5300, 0 ]
  }
]
},
options: {
  responsive: true,
  tooltips: {
    mode: 'index',
    intersect: false
  },
  hover: {
    mode: 'nearest',
    intersect: true
  }
}
} );
}
//Traffic chart chart-js End
// Bar Chart #flotBarChart
$.plot("#flotBarChart", [{
  data: [[0, 18], [2, 8], [4, 5], [6, 13],[8,5], [10,7],[12,4], [14,6],[16,15], [18,
9],[20,17], [22,7],[24,4], [26,9],[28,11]],
  bars: {
    show: true,
    lineWidth: 0,
    fillColor: '#ffffff8a'
  }
}

```

```

    }], {
        grid: {
            show: false
        }
    });
    // Bar Chart #flotBarChart End
});
</script>
</body>
</html>
<?php } ?>

```

## print.php

```

<?php
session_start();
error_reporting(0);
include('includes/dbconnection.php');
if (strlen($_SESSION['vpmsaid']==0)) {
    header('location:logout.php');
    } else{
?>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/normalize.css@8.0.0/normalize.min.css">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/css/bootstrap.min.css">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/font-awesome@4.7.0/css/font-
awesome.min.css">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/gh/lykmapipo/themify-
icons@0.1.2/css/themify-icons.css">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/pixeden-stroke-7-icon@1.2.3/pe-
icon-7-stroke/dist/pe-icon-7-stroke.min.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/flag-icon-
css/3.2.0/css/flag-icon.min.css">
    <link rel="stylesheet" href="assets/css/cs-skin-elastic.css">
    <link rel="stylesheet" href="assets/css/style.css">
<?php
$cid=$_GET['vid'];
$ret=mysqli_query($con,"select * from tblvehicle where ID='$cid'");
$cnt=1;
while ($row=mysqli_fetch_array($ret)) {
    ?>
<div id="exampl">
    <table border="1" class="table table-bordered mg-b-0">
        <tr>
            <th colspan="4" style="text-align: center; font-size:22px;"> Vehicle Parking
receipt</th>
        </tr>
        <tr>
            <th>Parking Number</th>

```



```

        <td><?php echo $row['ParkingNumber'];?></td>
        <th>Vehicle Category</th>
        <td><?php echo $row['VehicleCategory'];?></td>
    </tr>
    <tr>
        <th>Vehicle Company Name</th>
        <td><?php echo $packprice= $row['VehicleCompanyname'];?></td>
        <th>Registration Number</th>
        <td><?php echo $row['RegistrationNumber'];?></td>
    </tr>
    <tr>
        <th>Owner Name</th>
        <td><?php echo $row['OwnerName'];?></td>
        <th>Owner Contact Number</th>
        <td><?php echo $row['OwnerContactNumber'];?></td>
    </tr>
    <tr>
        <th>In Time</th>
        <td><?php echo $row['InTime'];?></td>
        <th>Status</th>
        <td><?php
if($row['Status']== "")
{
    echo "Incoming Vehicle";
}
if($row['Status']=="Out")
{
    echo "Outgoing Vehicle";
}
;?></td>
    </tr>
    <?php if($row['Remark']!=""){ ?>
    <tr>
        <th>Out time</th>
        <td><?php echo $row['OutTime'];?></td>
        <th>Rarking Charge</th>
        <td><?php echo $row['ParkingCharge'];?></td>
    </tr>
    <tr>
        <th>Remark</th>
        <td colspan="3"><?php echo $row['Remark'];?></td>
    </tr>
    <?php } ?>
    <tr>
        <td colspan="4" style="text-align:center; cursor:pointer"><i class="fa fa-print fa-2x" aria-
hidden="true" OnClick="CallPrint(this.value)" ></i></td>
    </tr>
</table>
    <?php } } ?>
</div>

```

```
<script>
function CallPrint(strid) {
var prtContent = document.getElementById("exempl");
var WinPrint = window.open("", "",
'left=0,top=0,width=800,height=900,toolbar=0,scrollbars=0,status=0');
WinPrint.document.write(prtContent.innerHTML);
WinPrint.document.close();
WinPrint.focus();
WinPrint.print();
WinPrint.close();
}
</script>
```

## **logout.php**

```
<?php
session_start();
session_unset();
session_destroy();
header('location:index.php');
?>
```

## **D) ABBREVIATION**

API	: Application programming Interface
ASCII	: American Standard Code for Information Interchange
ASP	: Active Server Pages
DB	: Database
DBMS	: Database Management System
DFD	: Data Flow Diagram
GB	: Gigabyte
HTML	: Hyper Text Markup Language
MB	: Megabyte
PDF	: Portable Document Format
RAM	: Random Access Memory
SGML	: Standard Generalized Markup Language
SQL	: Structured Query Languages
VGA	: Video Graphics Array
XAMPP	: X (cross platform), Apache, MySQL, PHP, Perl