

SOFTWARE REQUIREMENTS SPECIFICATION

Table of Contents.....	i
------------------------	---

1.0 Introduction

- 1.1. Purpose
- 1.2. Scope of Project
- 1.3. Glossary
- 1.4. References
- 1.5. Overview of Document

2.0 Overall description

- 2.1. System Environment
- 2.2. Product functions overview
- 2.3. User characteristics
- 2.4. General constraints

3.0 Functional requirements

4.0 External interface requirements

- 4.1 User interface
- 4.2 Hardware interface
- 4.3 Software interface

5.0 Performance requirements

1. INTRODUCTION

1.1 *Purpose*

The purpose of this document is to describe the details of the online vehicle renting project. It will explain all the details and functions of the system software. The online vehicle renting system helps the customers to search through the company vehicle list and choose which car to rent for short term use and pay them online.

1.2 *Scope of project*

This is an online software that helps in connecting the customers to company vehicle list for renting. The administrators have the maximum authority on the software change as they help in updates when needed or any other necessary changes to database like change status of car or add new cars. The users in the patient category will upload their problems or symptoms in text format and images of the diseased spot on their body and seek remedy. The other category of users known as doctors will look up on the list of patients with their problems and email the patients with their respective medicinal prescriptions that they can print it out and buy the necessary medicines.

The software also allows direct communication between patients and doctors through a chat box to discuss. The main advantage of this software is that the patients can have access to its benefits even in the urgent cases when doctors won't be available in hospital or in remote areas where doctors aren't available.

1.3 *Glossary*

TERM	DEFINITION
Administrator	They exercise most authorization over the software updates.
Customer	They are the users that search for vehicles according to their needs and rent them for use.
Vehicle	They are system products mainly managed by the administrators and rented and used by customers.
Database	Collection of all the information on the doctors and patients.
User	Customer.

1.4 References

[1] www.tutorialspoint.com

[2] www.scribd.com

1.5 Overview of Document

The second chapter, which is the overall description section defines the overview of functionality of the software product to be designed and also describes the informal requirements of the products. The third chapter, which is the requirements specification section is primarily meant for the developers with technical terms and in-depth details of the functionalities. The fourth chapter states all the interfaces for the software system. The fifth chapter gives the limitations for the software to work well.

2. Overall Description

2.1 *System environment*

The software environment is made up of some important active elements which are administrators, customers, vehicles and database. The administrators can access the main database directly and make changes to the system. The customer can only enter his/her details and search for the required vehicles that needs to be rented.

2.2 *Product functions overview*

The below following are the different functionalities that will be used in the product. These functions will be briefly discussed in easy steps in the next section of the document.

- sign in for customers
- login for customers
- add vehicle to the list
- search vehicle to rent
- rent selected vehicle
- after use pay/refund
- buy vehicle for customers

- change vehicle status

2.3 *User characteristics*

There are 3 types of users that use the system and are described below with their characteristics.

(1) Administrators

- High technical expertise
- Well aware of system functionalities
- Experience in database/ software coding

(2) Customers

- Basic knowledge to use computer
- Simple knowledge to use internet

2.4 *General constraints*

The software needs to be within the following constraints to be more user friendly.

[1] software constraints

- must run on windows 7 or above
- must run on any browser

[2] hardware constraints

- may or may not have a camera

- must have connection to internet

- ram of 512mb or above

3. FUNCTIONAL REQUIREMENTS

This section outlines all the different functionalities by the users that can be performed or are possible with the software.

3.1 *sign in for customers*

INTRODUCTION

The customers need to create an account to start using the software. Here they have to give their personal information as inputs to the software which then will be recorded in the database.

INPUT

- username ID

- password

- email

- contact number

- name

- date of birth

- bank account number

PROCESSING

The data that was given by the user is now sent to the database to make a new entry and stored in it for future use.

OUTPUT

If same username is used the account not created and following message shown “username already exists, so please find another one”.

And if no error exists and the account is created then the following message is shown “account successfully created”.

3.2 *add vehicle to the list*

INTRODUCTION

The administrators can add more vehicles to the list or their database by giving following information about vehicles.

INPUT

- car name ID
- car model
- date of manufacture
- cost price
- status of car
- revenue collected

PROCESSING

The data that was given by the user is now sent to the database to make a new entry and stored in it for future use.

OUTPUT

And if no error exists and the account is created then the following message is shown “car successfully added”.

3.3 *login for customers*

INTRODUCTION

The customers need to log in to their account to use the software with their registered account. The logged in tab will consist of all the functions that the customer user has been given access.

INPUT

-username ID

-password

PROCESSING

The data taken as input is then compared with the data stored in the database.

OUTPUT

If the both the data given and stored matches, then the user is logged into that account and the account page is opened with the tabs to perform different functions of the software.

And if the data given and the stored data doesn't match then an error message is shown “wrong username / password”.

3.4 *search vehicle to rent*

INTRODUCTION

The customers need to log in to their account to use the software with their registered account. The logged in tab will have search bar which will search through the entire database for required data the customer is searching for.

INPUT

- Model name of car

PROCESSING

The data taken as input is then compared with the data stored in the database.

OUTPUT

If the both the data given and stored matches, then the user is shown the list of matches found from the database.

3.5 *rent selected vehicle*

INTRODUCTION

When a customer finds the required vehicle he/she can rent the vehicle by clicking on the rent button available beside the car name.

INPUT

- mouse click on rent button

PROCESSING

The information provided by the customer will be recorded by the database and then the car will be made available for the customer.

OUTPUT

After successful entering of the data a message is shown “your car has been booked for you and you can now access it from the company shop”.

3.6 after use pay/refund

INTRODUCTION

When a customer comes back after use of the car he/ she will be either extra charged for extra driving of car or can be refunded on underuse of the vehicle.

INPUT

-car return button after returning the vehicle

PROCESSING

The database records the data and sets the status of car back to use and calculates the customer's bill.

OUTPUT

After successful return of car and payment completion a message will be displayed “thank you for using our services”.

3.7 buy vehicle for customers

INTRODUCTION

The customers can buy the car for full time use directly from the site too by clicking buy button beside the car model and paying online.

INPUT

-clicking the buy button on the screen

PROCESSING

The input is recorded and necessary changes are made in the database.

OUTPUT

A small message is displayed "congratulation son buying your new car".

3.8 *change vehicle status*

INTRODUCTION

The administrators can change the status of the car among three options that is gone for repair or can be rented or can be bought.

INPUT

-select from drop down list

PROCESSING

The button response is recorded by database that change made by the administrators.

OUTPUT

A message is displayed "status of car has changed".

4. EXTERNAL INTERFACE REQUIREMENTS

4.1 *User interfaces*

The user interface for the software shall be compatible to any browser like Mozilla Firefox, google chrome etc.

The user interface shall be implemented using any tool or software package like java, applet etc.

4.2 *Hardware interfaces*

Since the application must run over internet, all hardware shall require to connect internet will be hardware interface for the system. As for e.g. Modem, WAN-LAN, etc.

4.3 *Software interfaces*

As the system connects the browser to the database then the software interface in this case shall be the communication of the software and the database of the server.

5. PERFORMANCE REQUIREMENTS

The following performance advantages should be there in the software for better and smooth use and so that it is liked by the users also.

(1) Response time

The system should respond to the clicking within 3 seconds of the function activation.

(2) Capacity

The system should support at least 100 users at a time.

The system should also support at least 100 vehicles information at a time.

(3) Errors

The system should keep a log of all the errors faced to debug them later by the administrators.

(4) Availability

The server of the software should be active all the time and shouldn't become inactive.

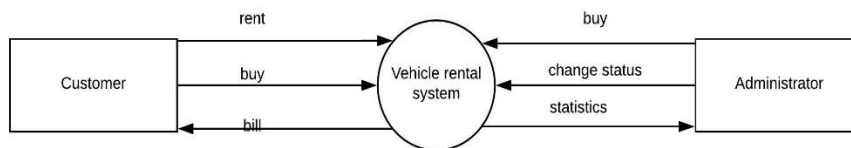
DATA FLOW DIAGRAM (DFD)

A **data flow diagram (DFD)** is a graphical representation of the "flow" of data through an information system modelling its *process* aspects. A data flow diagram shows the way information flows through a process or system. It includes data inputs and outputs, data stores, and the various subprocesses the data moves through.

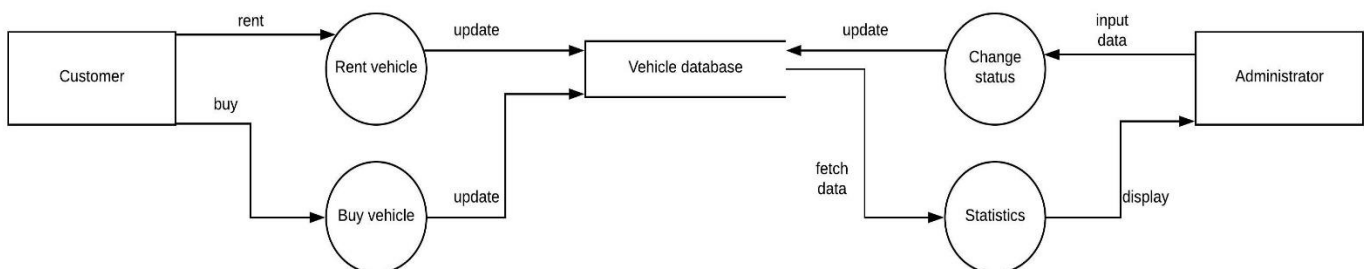
Data flow diagrams are also categorized by level. Starting with the most basic, level 0, DFDs get increasingly complex as the level increases.

Level 0 DFDs, also known as context diagrams, are the most basic data flow diagrams. They provide a broad view that is easily digestible but offers little detail. Level 0 data flow diagrams show a single process node and its connections to external entities.

Level 1 DFDs are still a general overview, but they go into more detail than a context diagram. In a level 1 data flow diagram, the single process node from the context diagram is broken down into subprocesses. As these processes are added, the diagram will need additional data flows and data stores to link them together.



Level - 0 / Context



Level - 1

ENTITY RELATIONSHIP DIAGRAM

ER diagram definition :-

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.

Uses of ER diagrams :-

- **Database design:** ER diagrams are used to model and design relational databases, in terms of logic and business rules (in a logical data model) and in terms of the specific technology to be implemented (in a physical data model.) In software engineering, an ER diagram is often an initial step in determining requirements for an information systems project. It's also later used to model a particular database or databases. A relational database has an equivalent relational table and can potentially be expressed that way as needed.
- **Database troubleshooting:** ER diagrams are used to analyze existing databases to find and resolve problems in logic or deployment. Drawing the diagram should reveal where it's going wrong.

- **Business information systems:** The diagrams are used to design or analyze relational databases used in business processes. Any business process that uses fielded data involving entities, actions and interplay can potentially benefit from a relational database. It can streamline processes, uncover information more easily and improve results.
- **Business process re-engineering (BPR):** ER diagrams help in analyzing databases used in business process re-engineering and in modeling a new database setup.
- **Education:** Databases are today's method of storing relational information for educational purposes and later retrieval, so ER Diagrams can be valuable in planning those data structures.
- **Research:** Since so much research focuses on structured data, ER diagrams can play a key role in setting up useful databases to analyze the data.

Features of ER diagram :-

ER Diagrams are composed of entities, relationships and attributes. They also depict cardinality, which defines relationships in terms of numbers. Here's a glossary:

Entity

A definable thing—such as a person, object, concept or event—that can have data stored about it. Think of entities as nouns. Examples: a customer, student, car or product. Typically shown as a rectangle.

Entity type: A group of definable things, such as students or athletes, whereas the entity would be the specific student or athlete. Other examples: customers, cars or products.

Entity set: Same as an entity type, but defined at a particular point in time, such as students enrolled in a class on the first day. Other examples: Customers who purchased last month, cars currently registered in Florida. A related term is instance, in which the specific person or car would be an instance of the entity set.

Entity categories: Entities are categorized as strong, weak or associative. A **strong entity** can be defined solely by its own attributes, while a **weak entity** cannot. An associative entity associates entities (or elements) within an entity set.

Entity keys: Refers to an attribute that uniquely defines an entity in an entity set. Entity keys can be super, candidate or primary. **Super key:** A set of attributes (one or more) that together define an entity in an entity set. **Candidate key:** A minimal super key, meaning it has the least possible number of attributes to still be a super key. An entity set may have more than one candidate key. **Primary key:** A candidate key chosen by the database designer to uniquely identify the entity set. **Foreign key:** Identifies the relationship between entities.

Relationship

How entities act upon each other or are associated with each other. Think of relationships as verbs. For example, the named student might register for a course. The two entities would be the student and the course, and the relationship depicted is the act of enrolling, connecting the two entities in that way. Relationships are typically shown as diamonds or labels directly on the connecting lines.

Recursive relationship: The same entity participates more than once in the relationship.

Attribute

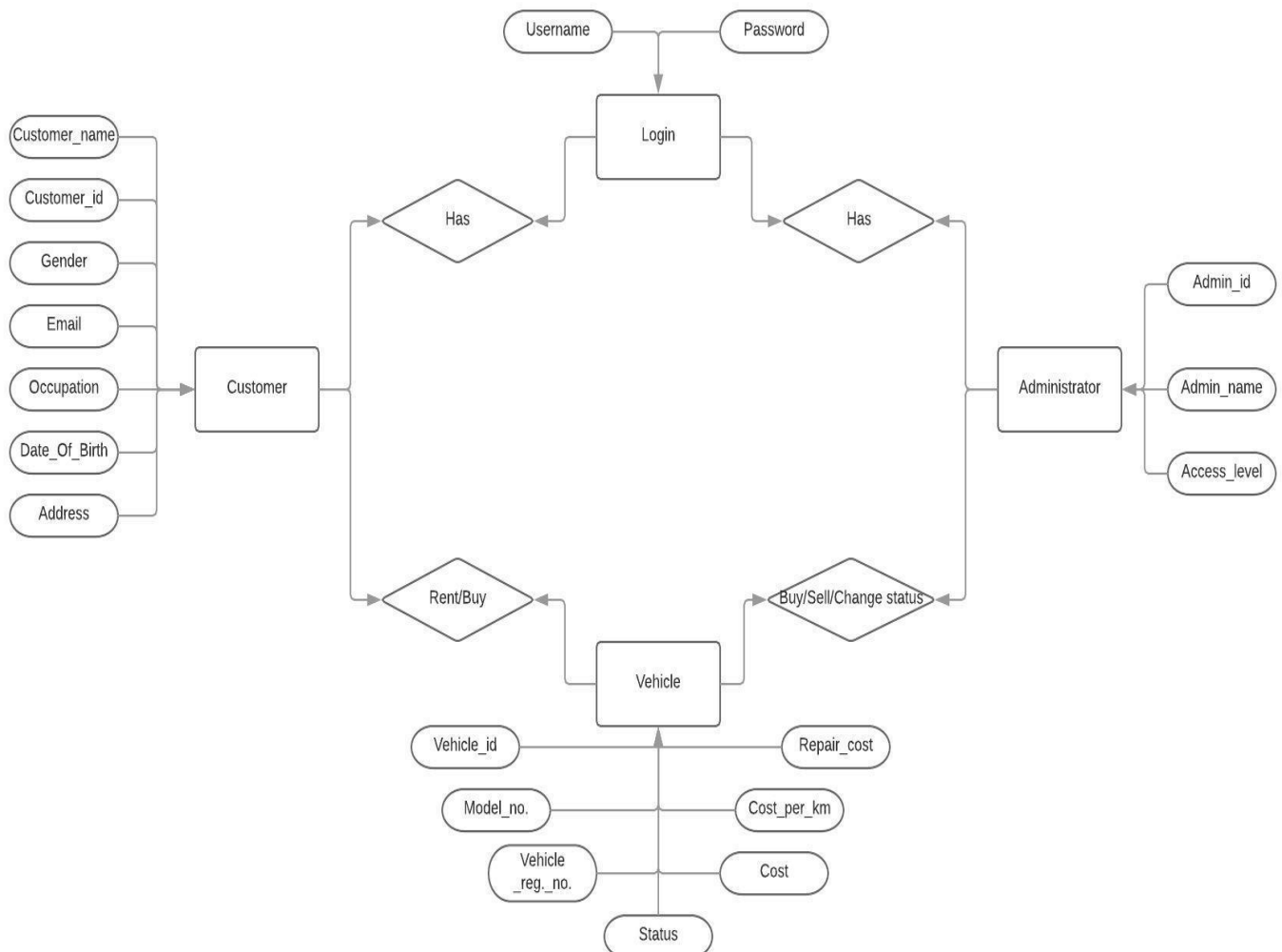
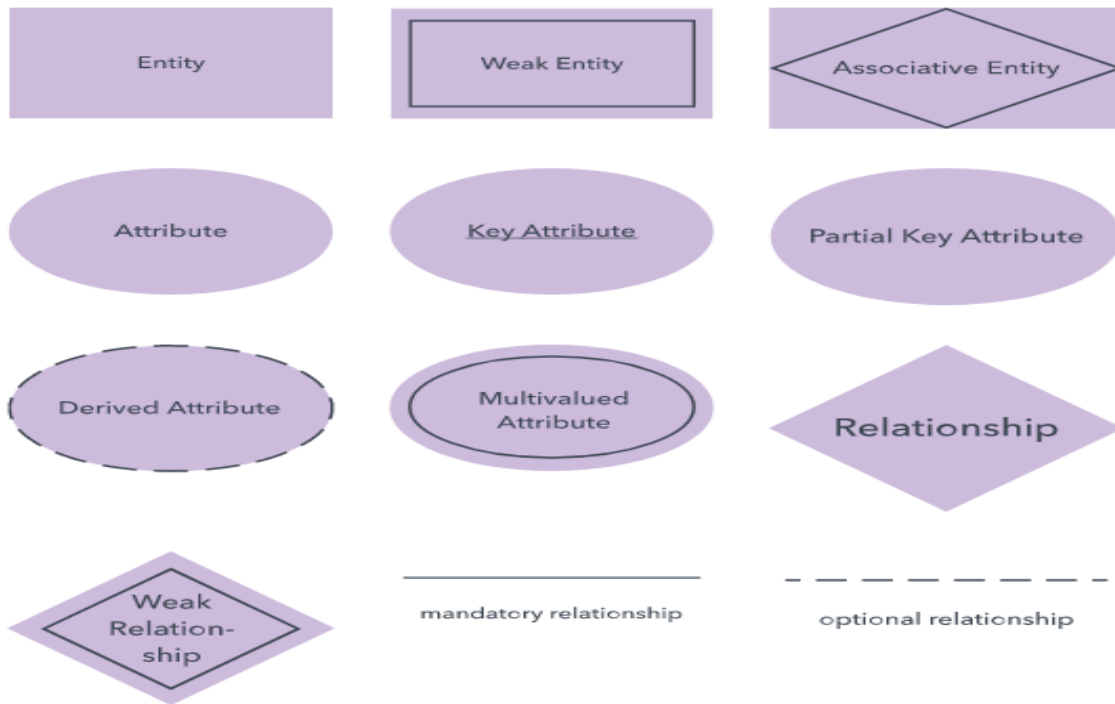
A property or characteristic of an entity. Often shown as an oval or circle.

Descriptive attribute: A property or characteristic of a relationship (versus of an entity.)

Attribute categories: Attributes are categorized as simple, composite, derived, as well as single-value or multi-value. **Simple:** Means the attribute value is atomic and can't be further divided, such as a phone number. **Composite:** Sub-attributes spring from an attribute. **Derived:** Attributed is calculated or otherwise derived from another attribute, such as age from a birthdate.

Multi-value: More than one attribute value is denoted, such as multiple phone numbers for a person.

Single-value: Just one attribute value. The types can be combined, such as: simple single-value attributes or composite multi-value attributes.



UML DIAGRAMS

UML is an acronym that stands for Unified Modeling Language. Simply put, UML is a modern approach to modeling and documenting software. In fact, it's one of the most popular business process modeling techniques. It is based on diagrammatic representations of software components. By using visual representations, we are able to better understand possible flaws or errors in software or business processes.

USES OF UML

Mainly, UML has been used as a general-purpose modeling language in the field of software engineering. UML itself finds different uses in software development and business process documentation:

Sketch

UML diagrams, in this case, are used to communicate different aspects and characteristics of a system. However, this is only a top-level view of the system and will most probably not include all the necessary details to execute the project until the very end.

- **Forward Design** – The design of the sketch is done before coding the application. This is done to get a better view of the system or workflow that you are trying to create. Many design issues or flaws can be revealed, thus improving the overall project health and well-being.
- **Backward Design** – After writing the code, the UML diagrams are drawn as a form of documentation for the different activities, roles, actors, and workflows.

Blueprint

In such a case, the UML diagram serves as a complete design that requires solely the actual implementation of the system or software. Often, this is done by using CASE tools (Computer Aided Software Engineering Tools). The main drawback of using CASE tools is that they require a certain level of expertise, user training as well as management and staff commitment.

Pseudo Programming Language

UML is not a stand-alone programming language like Java, C++ or Python, however, with the right tools, it can turn into a pseudo programming language. In order to achieve this, the whole system needs to be documented in different UML diagrams and, by using the right software, the diagrams can be directly translated into code. This method can only be beneficial if the time it takes to draw the diagrams would take less time than writing the actual code.

Despite UML having been created for modeling software systems, it has found several adoptions in business fields or non-software systems.

Types of UML Diagrams

There are several types of UML diagrams and each one of them serves a different purpose regardless of whether it is being designed before the implementation or after (as part of documentation).

The two most broad categories that encompass all other types are Behavioral UML diagram and Structural UML diagram. As the name suggests, some UML diagrams try to analyze and depict the structure of a system or process, whereas others describe the behavior of the system, its actors, and its building components. The different types are broken down as follows:

Behavioral UML Diagram

- Activity Diagram
- Use Case Diagram
- Interaction (Sequence and collaboration) diagram
- Timing Diagram
- State Machine Diagram
- Communication Diagram
- Sequence Diagram

Structural UML Diagram

- Class Diagram
- Object Diagram
- Component Diagram
- Composite Structure Diagram
- Deployment Diagram
- Package Diagram
- Profile Diagram

Not all of the 14 different types of UML diagrams are used on a regular basis when documenting systems and/or architectures. And in this document we are going to focus on few of them.

• USE CASE DIAGRAM

To model a system, the most important aspect is to capture the dynamic behavior. Dynamic behavior means the behavior of the system when it is running/operating.

Only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior.

In UML, the use case diagram is dynamic in nature, there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. Use case diagrams consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

In brief, the purposes of use case diagrams can be said to be as follows –

- Used to gather the requirements of a system.
- Used to get an outside view of a system.
- Identify the external and internal factors influencing the system.
- Show the interaction among the requirements are actors.

• **CLASS DIAGRAM**

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

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of objectoriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

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

The purpose of the class diagram can be summarized as –

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

• **OBJECT DIAGRAM**

Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams.

Object diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams. Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment.

Object diagrams are used to render a set of objects and their relationships as an instance.

The purpose of the object diagram can be summarized as –

- Forward and reverse engineering.
- Object relationships of a system
- Static view of an interaction.
- Understand object behaviour and their relationship from practical perspective

• COLLABORATION DIAGRAM

The next diagram is the collaboration diagram. It shows the object organization as seen in the following diagram. In the collaboration diagram, the method call sequence is indicated by some numbering technique. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram.

Method calls are similar to that of a sequence diagram. However, difference being the sequence diagram does not describe the object organization, whereas the collaboration diagram shows the object organization.

To choose between these two diagrams, emphasis is placed on the type of requirement. If the time sequence is important, then the sequence diagram is used. If organization is required, then collaboration diagram is used.

Following things are to be identified clearly before drawing the collaboration diagram -

- Objects taking part in the interaction.
- Message flows among the objects.
- The sequence in which the messages are flowing.
- Object organization.

• SEQUENCE DIAGRAM

The sequence diagram has four objects (Customer, Order, SpecialOrder and NormalOrder).

The following diagram shows the message sequence for *SpecialOrder* object and the same can be used in case of *NormalOrder* object. It is important to understand the time sequence of message flows. The message flow is nothing but a method call of an object.

Sequence diagrams can be used –

- To model the flow of control by time sequence.
- To model the flow of control by structural organizations.
- For forward engineering.
- For reverse engineering.

- **ACTIVITY DIAGRAM**

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc

The purpose of an activity diagram can be described as –

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

- **STATE CHART DIAGRAM**

The name of the diagram itself clarifies the purpose of the diagram and other details. It describes different states of a component in a system. The states are specific to a component/object of a system.

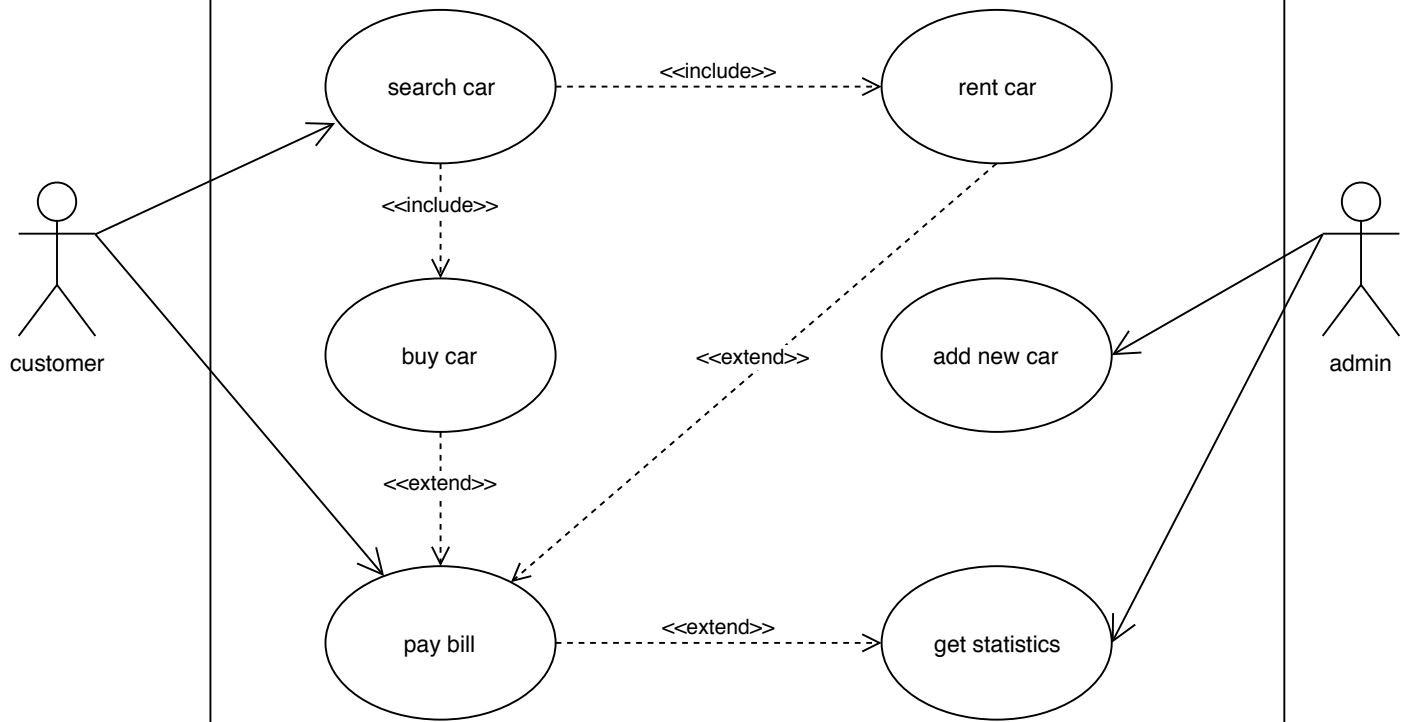
A Statechart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.

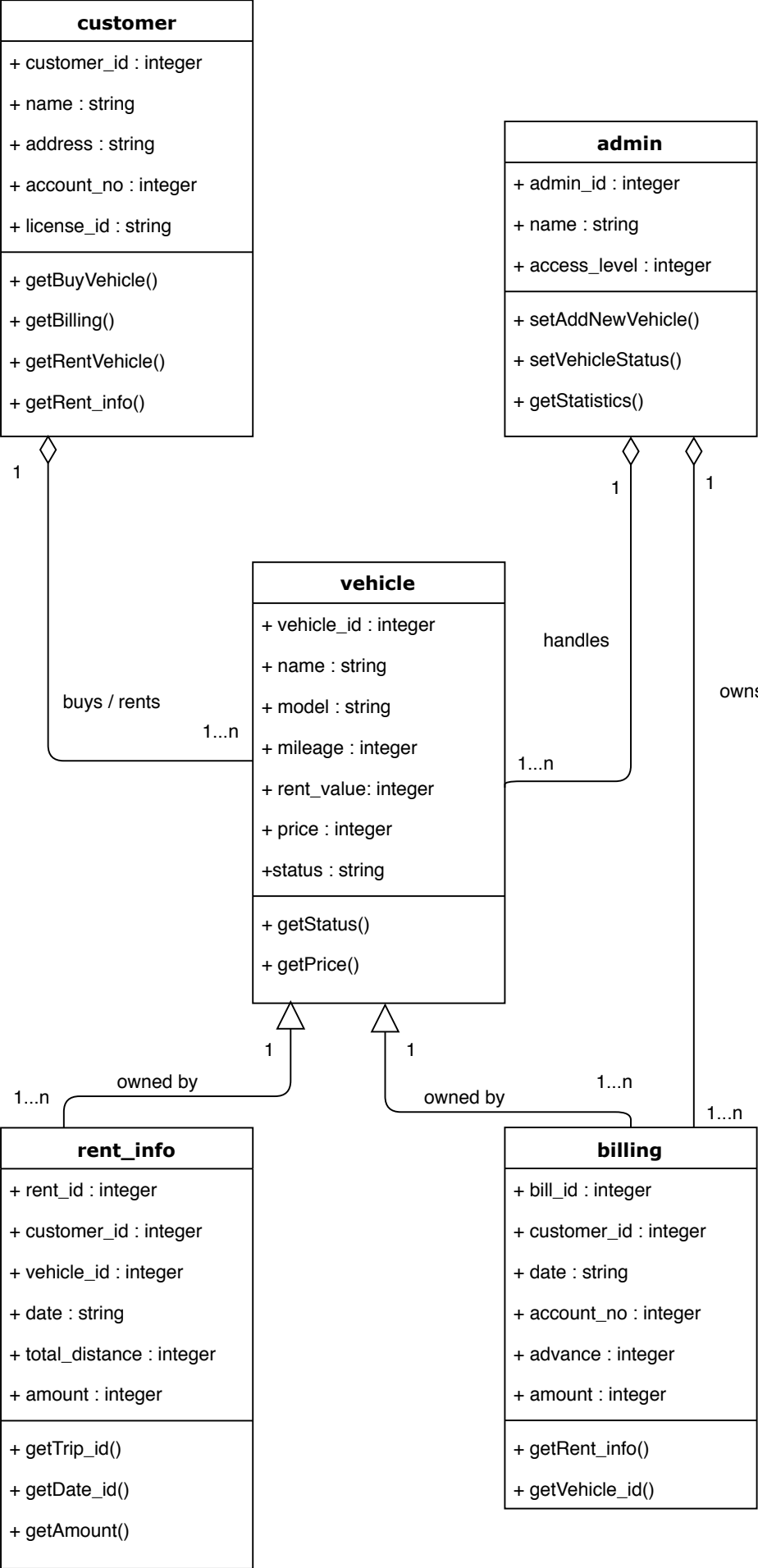
Activity diagram explained in the next chapter, is a special kind of a Statechart diagram. As Statechart diagram defines the states, it is used to model the lifetime of an object.

Following are the main purposes of using Statechart diagrams –

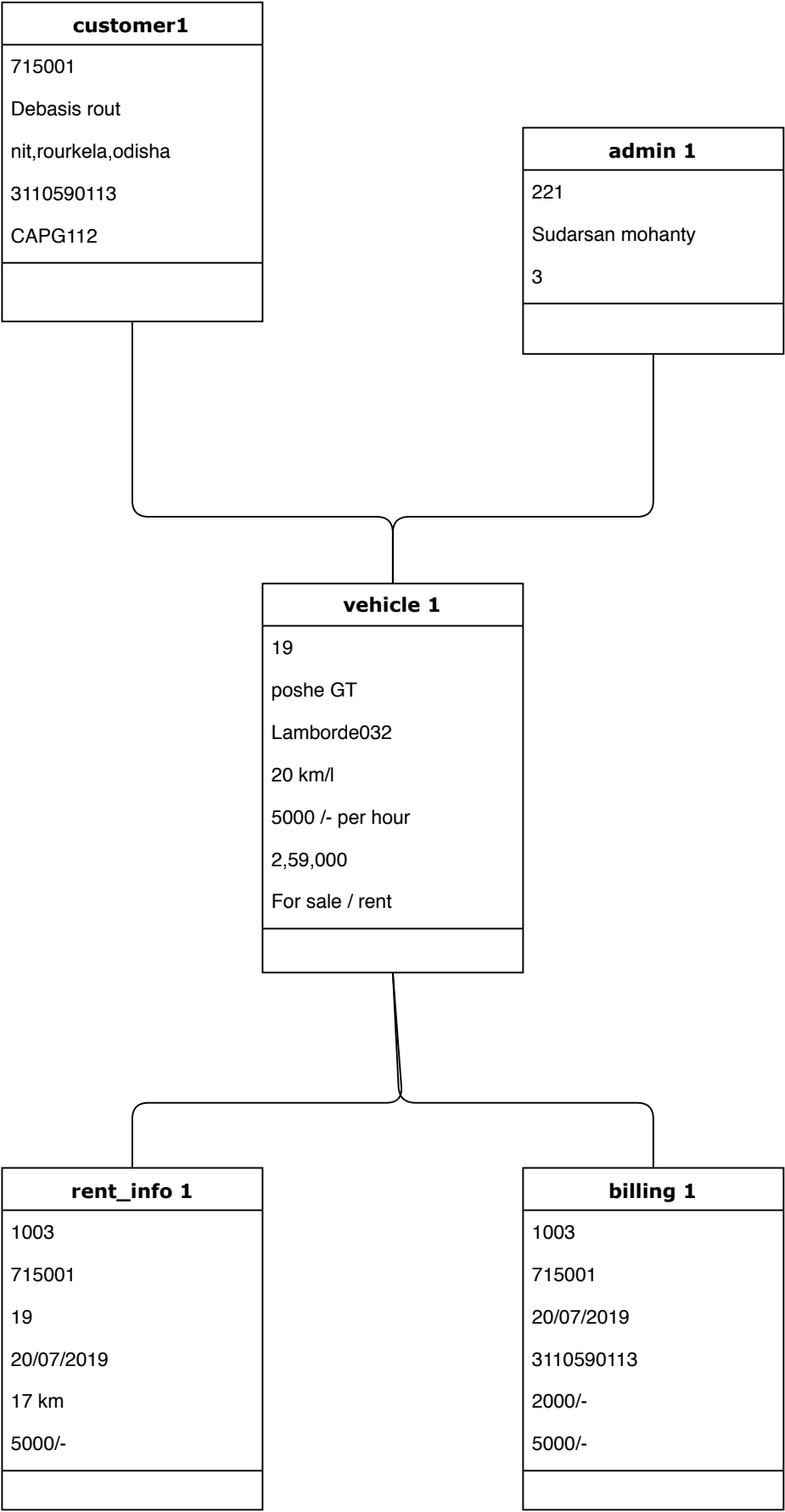
- To model the dynamic aspect of a system.
- To model the life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model the states of an object.

USE CASE DIAGRAM FOR CAR RENTAL SYSTEM

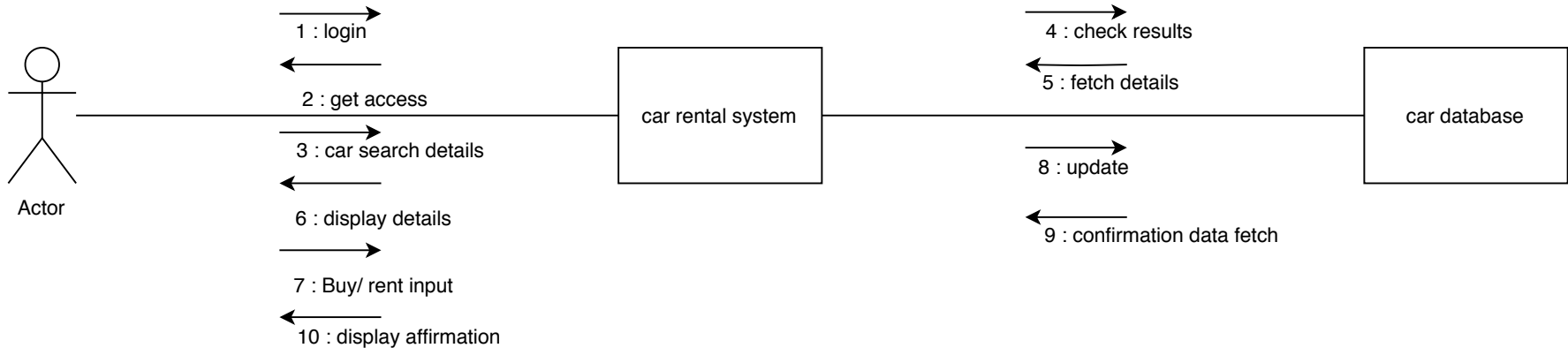




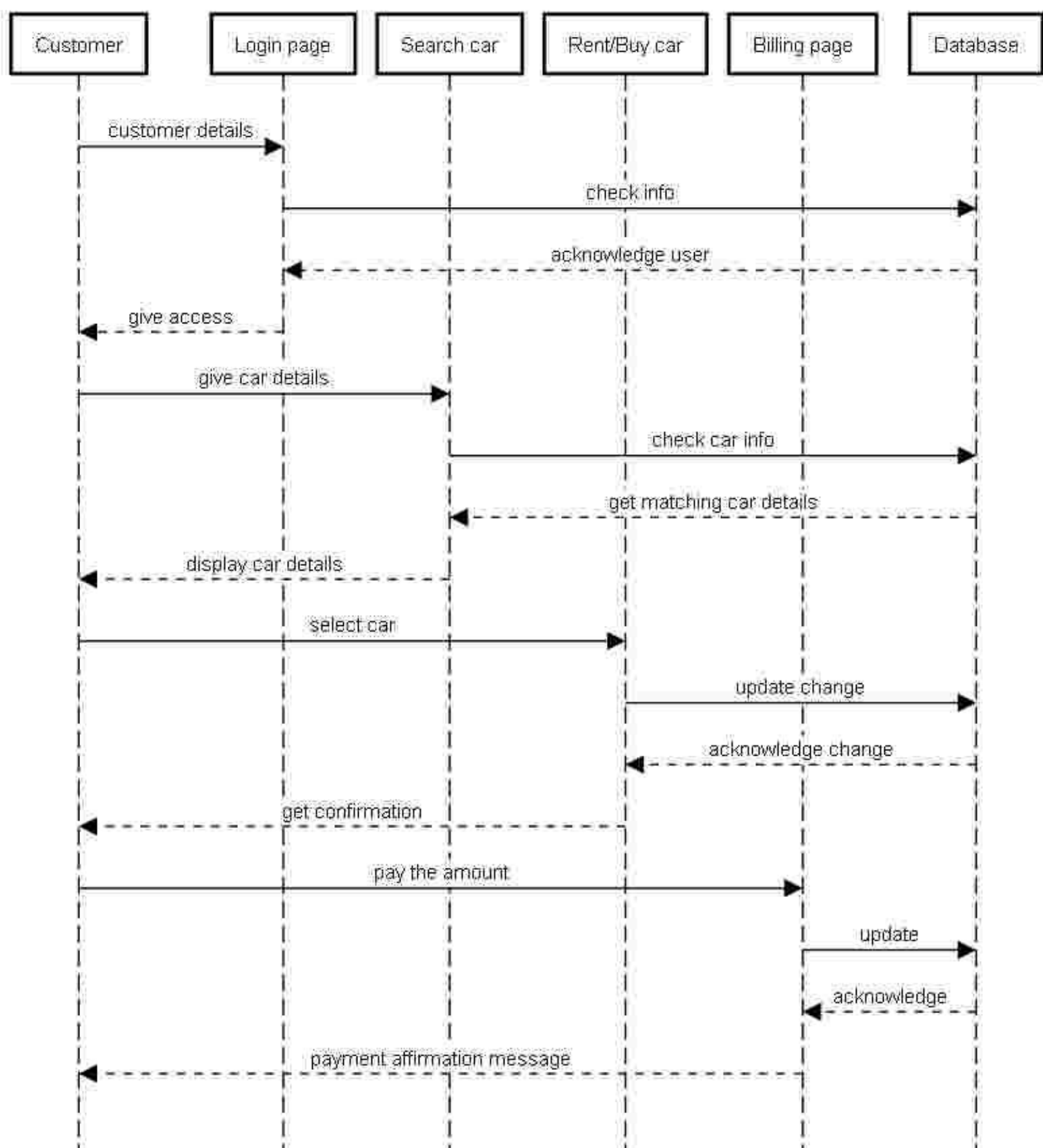
CLASS DIAGRAM



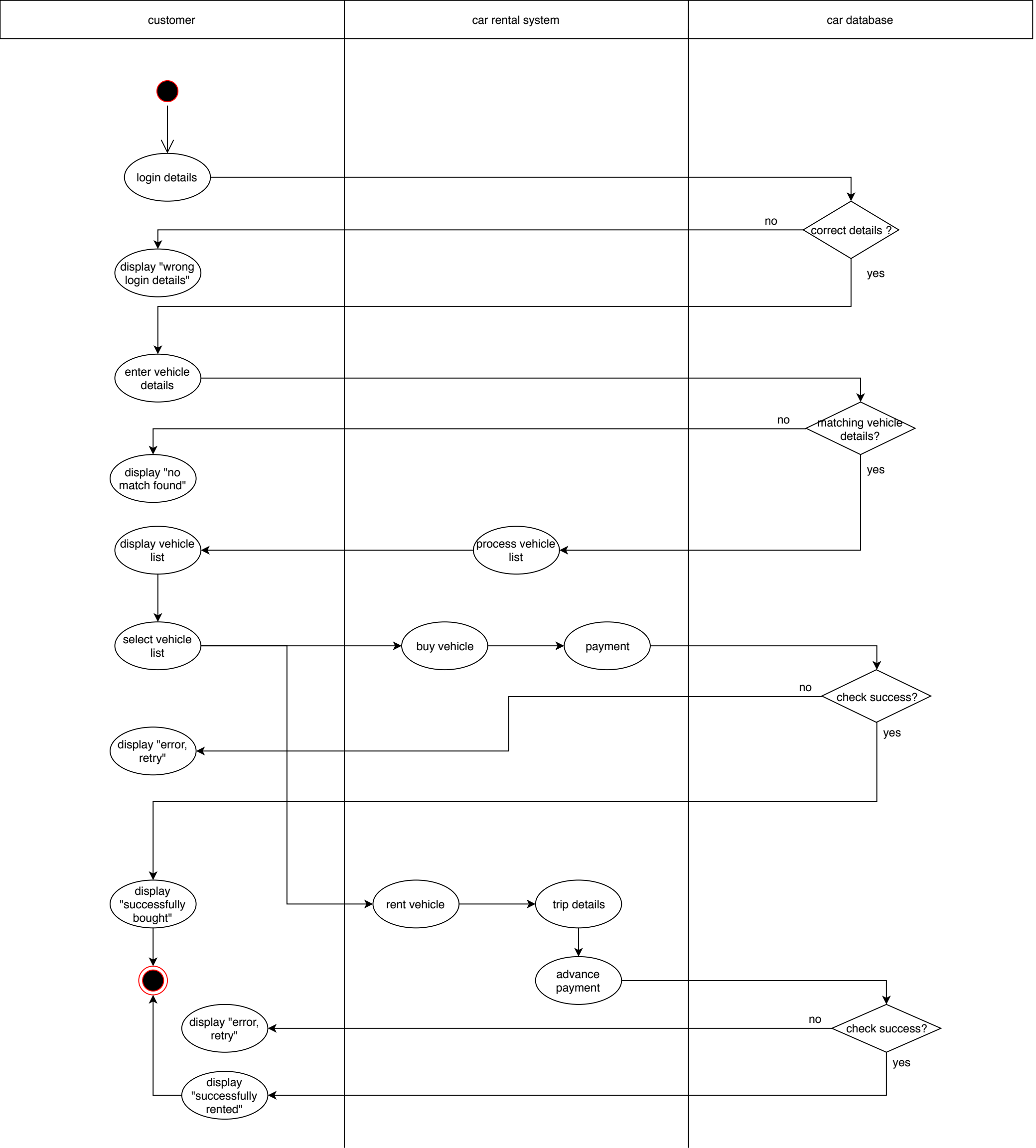
OBJECT DIAGRAM



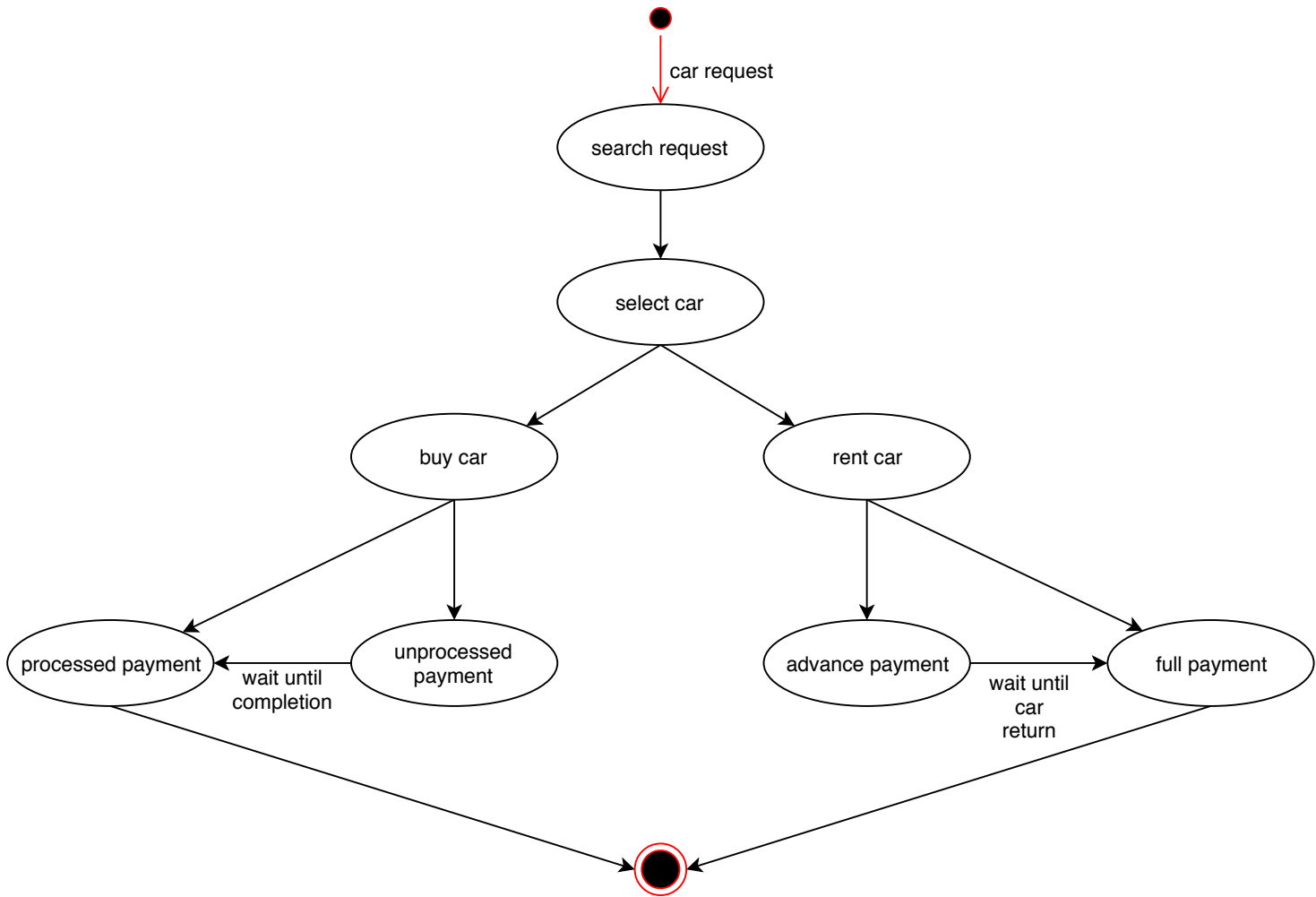
COLLABORATION DIAGRAM



SEQUENCE DIAGRAM



ACTIVITY DIAGRAM



STATE CHART DIAGRAM

CODING

Styles.css

```
* {  
  
    margin:0;  
  
    padding:0;  
  
}  
  
body {  
  
    font-family:Arial, Helvetica, sans-serif;  
  
    font-size:12px;  
  
    background:#000000 ;  
  
    color: #ffffff;  
  
}  
  
a {  
  
    color:#ff9c00;  
  
    text-decoration:underline;  
  
}  
  
a:hover {  
  
    color:#ff9c00;  
  
    text-decoration:none;  
  
}  
  
#container {  
  
    width: 100%;  
  
}  
  
#header {  
  
    width:100%;
```

```
}

.logoBackground {
    background:#000000;
    width:300px;
}

.logo {
    color:#ff9c00;
    font-size:60px;
    padding:10px 0 0 30px;
}

.author {
    padding:0 0 10px 30px;
    color:#707070;
}

.author a {
    color:#707070;
}

.menu {
    background:url(images/menuBackground.png) repeat-x left top;
    height:103px;
    text-align:center;
    margin-top:265px;
}

.menu ul {
    margin:0;
    padding-top:32px;
}
```

```
.menu li {  
    list-style:none;  
    display:inline;  
    padding:20px 15px 0 15px;  
}
```

```
.menu li a {  
    color:#ffffff;  
    font-size:24px;  
    text-decoration:none;  
}
```

```
.menu li a:hover {  
    color:#ff9c00;  
    font-size:24px;  
    text-decoration:none;  
}
```

```
.menu #active a {  
    color:#ff9c00;  
    font-size:24px;  
    text-decoration:none;  
}
```

```
.allContent {  
    width:800px;  
    margin:0 auto;  
    padding:130px 0 0 0;  
}
```

```
.leftContent p, .rightContent p {  
    padding:40px;
```

```
}

.leftContent {

    float:left;

    width:400px;

    background:url(images/leftContent.png) repeat-y left top;

}

.rightContent {

    margin-left:436px;

    width:364px;

    background:url(images/rightContent.png) repeat-y left top;

}

.rightContent ul {

    margin:0;

    padding:0 40px;

}

.rightContent li {

    list-style:none;

    padding:0 0 20px 0;

}

#footer {

    text-align:center;

    color:#707070;

    padding:30px 0;

}

.clearfloat {

    clear:both;

    height:0;
```

```
font-size: 1px;

line-height: 0px;

}
```

home.php

```
<html>

  <head>

    <title>Car Rental </title>

    <link href="styles.css" rel="stylesheet" type="text/css" />

  </head>

  <body>

    <div id="container">

      <br class="clearfloat" />

      <div id="header">

        <div class="logoBackground">

          <div class="logo">CAR RENTALS</div>

          <div class="author">Quality Vehicles</div>

        </div>

        <div class="menu">

          <ul>

            <li id="active"><a href="">Home</a></li>

            <li><a href="customer.php">Customers</a></li>
```

```
<li><a href="cars.php">Cars</a></li>
```

```
<li><a href="book.php">Book Car</a></li>
```

```
<li><a href="return.php">Return Car</a></li>
```

```
<li><a href="report.php">Report</a></li>
```

```
</ul>
```

```
</div>
```

```
</div>
```

```
<br class="clearfloat" />
```

```
<div id="footer">
```

```
<p>
```

```
</p>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

Customer.php

```
<html>
```

```
<head>
```

```
<title>Car Rental </title>
```

```
<link href="styles.css" rel="stylesheet" type="text/css" />
```

```
</head>
```

```
<body>
```

```
<div class="logoBackground">
```

```
<div class="logo"><h6>CUSTOMERS</h6></div>
```

```
</div>
```

```
<div class="menu">
```

```
<ul>
```

```
<li><a href="home.php">Home</a></li>
```

```
<li id="active"><a href="">Customers</a></li>
```

```
<li><a href="cars.php">Cars</a></li>
```

```
<li><a href="book.php">Book Car</a></li>
```

```
<li><a href="return.php">Return Car</a></li>
```

```
<li><a href="report.php">Report</a></li>
```

```
</ul>
```

```
</div>
```

```
<div class="leftContent">
```

```
<b><form action="regcustomer.php" method="post">
```

```
<h1> ADD CUSTOMER </h1>
```

```
Cid : <input type="text" name="Cid"/><br><br>
```

```
First Name: <input type="text" name="fname"/><br><br>
```

```
Last Name: <input type="text" name="lname"/><br><br>
```

```
Mobile: <input type="text" name="mobile"/><br><br>
```

```
Driving License No: <input type="text" name="dlno"/><br><br>
```

```
Insurance No:<input type="text" name="ino"/><br><br>
<input type="submit" name="submit" value="Register">
</form></b>

</div>
```

```
<div class="rightContent">

<b><form action="viewcustomer.php" method="post" >

<h1> VIEW CUSTOMERS </h1>

<input type="submit" value="View"/>

</form></b>

</div>

</body>

</html>
```

Regcustomer.php

```
<html>

<body>

<?php

    $database_host = "localhost";

    $database_user = "root";

    $database_pass = "";

    $database_name = "car_rental";

    $connection = mysqli_connect($database_host, $database_user, $database_pass,
    $database_name);

    if(mysqli_connect_errno()){
```



```
        die("Failed connecting to MySQL database. Invalid credentials" .
mysqli_connect_error(). "(" .mysqli_connect_errno(). ")" ); }

$Cid=$_POST["Cid"];
$Fname=$_POST["fname"];
$Lname=$_POST["lname"];
$Mobile=$_POST["mobile"];
$Dlno=$_POST["dlno"];
$Insno=$_POST["ino"];

$result="INSERT INTO customer(Cid,Fname,Lname,Mobile,Dno,Vehicle_id)
VALUES('$Cid','$Fname','$Lname','$Mobile','$Dlno','$Insno')";

mysqli_query($connection,$result) or die(mysqli_error($connection));

echo "<h3>New customer has been successfully added</h3><br><br>";

?>

<?php

$result1="SELECT Cid FROM customer WHERE Dlno='$Dlno'";

$result2=mysqli_query($connection,$result1);

while($row = mysqli_fetch_assoc($result2))
{

    echo "<h3>Customer ID is :</h3>".$row["Cid"];

}

?>

</body>

</html>
```

Viewcustomer.php

```
<html>

<body>

<?php

    $database_host = "localhost";

    $database_user = "root";

    $database_pass = "";

    $database_name = "car_rental";

    $connection = mysqli_connect($database_host, $database_user, $database_pass,
    $database_name);

    if(mysqli_connect_errno()){

        die("Failed connecting to MySQL database. Invalid credentials" .
    mysqli_connect_error(). "(" .mysqli_connect_errno(). ")" ); }

    $res="SELECT * FROM car_rental.customer";

    $result=mysqli_query($connection,$res);

    echo "<h1><center>Customers</h1><br><br>";

?>

<center>

<table border='1'>

<tr>

<th>Cid</th>

<th>FName</th>

<th>Lname</th>

<th>Mobile</th>

<th>DI No.</th>

<th>Insurance No.</th>
```

```
</tr>

<?php
if (mysqli_num_rows($result) > 0) {
while($row = mysqli_fetch_assoc($result))
{
echo "<tr>";
echo "<td>" . $row["Cid"] . "</td>";
echo "<td>" . $row["Fname"] . "</td>";
echo "<td>" . $row["Lname"] . "</td>";
echo "<td>" . $row["Mobile"] . "</td>";
echo "<td>" . $row["Dno"] . "</td>";
echo "<td>" . $row["Vehicle_id"] . "</td>";
echo "</tr>";

}
}

?>

</table>

</body>

</html>
```

Cars.php

```
<html>

<head>

<title>Car Rental </title>

<link href="styles.css" rel="stylesheet" type="text/css" />

</head>
```

<body>

<div class="logoBackground">

<div class="logo"><h6>CARS</h6></div>

</div>

<div class="menu">

Home

Customers

<li id="active">Cars

Book Car

Return Car

Report

</div>

<div class="leftContent">

<form action="registercar.php" method="post" >

<h1> ADD CAR </h1>

License No: <input type="text" name="lno"/>

Model: <input type="text" name="model"/>

Year : <input type="text" name="year"/>

Car type: <select name = "Cartype">

<option>Compact</option>

<option>Medium</option>

<option>Large</option>

<option>SUV</option>

<option>Van</option>

<option>Truck</option>

</select>

Car Owner: <select name = "Carown">

<option>Individual</option>

<option>Bank</option>

<option>Company</option>

</select>

<h2>Enter Owner Details : </h2>

Unique ID: <input type="text" name="uid"/>

Name: <input type="text" name="oname"/>

City : <input type="text" name="city"/>

<input type="submit" name="submit" value="Register">

</form>

</div>

<div class="rightContent">

<h1> VIEW CARS & RATES</h1>

<form action="viewcar.php" method="post">

Select Car type to view: <select name="Ctype">

<option>Compact</option>

<option>Medium</option>

<option>Large</option>

<option>SUV</option>

<option>Van</option>

<option>Truck</option>

</select>

<input type="submit" name="submit1" value="View">

</form>

<h1> UPDATE RENTAL RATES </h1>

<form action="update.php" method="post">

Select Car type to update: <select name = "Ctype">

<option>Compact</option>

<option>Medium</option>

<option>Large</option>

<option>SUV</option>

<option>Van</option>

<option>Truck</option>

</select>

Daily Rate: <input type="text" name="udrate"/>

Weekly Rate: <input type="text" name="uwrate"/>

<input type="submit" name="submit1" value="Update">

</form>

</div>

</body>

</html>

Registercar.php

```
<html>
```

```
<body>
```

```
<?php
```

```
    $database_host = "localhost";
```

```
    $database_user = "root";
```

```
    $database_pass = "";
```

```
    $database_name = "car_rental";
```

```
    $connection = mysqli_connect($database_host, $database_user, $database_pass,  
$database_name);
```

```
    if(mysqli_connect_errno()){
```

```
        die("Failed connecting to MySQL database. Invalid credentials" .
```

```
mysqli_connect_error(). "(" .mysqli_connect_errno(). ")" ); }
```

```
$License_no=$_POST["lno"];
```

```
$Model=$_POST["model"];
```

```
$Year=$_POST["year"];
```

```
$Ctype=$_POST["Cartype"];
```

```
$Name1=$_POST["Carown"];
```

```
$Ssn=$_POST["uid"];
```

```
$Bid=$_POST["uid"];
```

```
$Compid=$_POST["uid"];
```

```
$Name=$_POST["oname"];
```

```
$Bname=$_POST["oname"];
```

```
$Cname=$_POST["oname"];
```

```
$City=$_POST["city"];
```

```

$result="INSERT INTO car(License_no,Model,Year,Ctype)
VALUES('$License_no','$Model','$Year','$Ctype)";

mysqli_query($connection,$result) or die(mysqli_error($connection));

echo "<h3>New car has been successfully added</h3>";


if($Name1=="Individual") {

$result="INSERT INTO individual(Ssn,Name,City) VALUES('$Ssn','$Name','$City)";

}

else if($Name1=="Bank") {

$result="INSERT INTO bank(Bid,Bname,City) VALUES('$Bid','$Bname','$City)";

}

else {

$result="INSERT INTO company(Compid,Cname,City)
VALUES('$Compid','$Cname','$City)";

}

mysqli_query($connection,$result) or die(mysqli_error($connection));

echo "<h3>New owner has been successfully added</h3>";

?>

```

<?php

```

$res="SELECT Vehicle_id from car where License_no='$License_no'";

$result2=mysqli_query($connection,$res);

while($row1 = mysqli_fetch_assoc($result2)) {

echo "<h3>Vehicle ID is :</h3>".$row1["Vehicle_id"];

$Vehicle_id=$row1["Vehicle_id"]; }


if($Name1=="Individual") {

```



```

$res1="SELECT Owner_id from individual where Ssn='$Ssn';

$result3=mysqli_query($connection,$res1);

while($row2 = mysqli_fetch_assoc($result3)) {

    echo "<h3>Owner ID is :</h3>".$row2["Owner_id"];

    $Owner_id=$row2["Owner_id"];}

}

else if($Name1=="Bank") {

    $res1="SELECT Owner_id from bank where Bid='$Bid';

    $result3=mysqli_query($connection,$res1);

    while($row2 = mysqli_fetch_assoc($result3)) {

        echo "<h3>Vehicle ID is :</h3>".$row2["Owner_id"];

        $Owner_id=$row2["Owner_id"];}

    }

else {

    $res1="SELECT Owner_id from company where Compid='$Compid';

    $result3=mysqli_query($connection,$res1);

    while($row2 = mysqli_fetch_assoc($result3)) {

        echo "<h3>Vehicle ID is :</h3>".$row2["Owner_id"];

        $Owner_id=$row2["Owner_id"];}

    }

    $result1="INSERT INTO owns(Vehicle_id,Owner_id,Ctype)
VALUES('$Vehicle_id','$Owner_id','$Ctype)";

    mysqli_query($connection,$result1) or die(mysqli_error($connection));

?>

</body>

</html>

```

Viewcar.php

```
<html>

<body>

<?php

    $database_host = "localhost";

    $database_user = "root";

    $database_pass = "";

    $database_name = "car_rental";

    $connection = mysqli_connect($database_host, $database_user, $database_pass,
    $database_name);

    if(mysqli_connect_errno()){

        die("Failed connecting to MySQL database. Invalid credentials" .
    mysqli_connect_error(). "(" .mysqli_connect_errno(). ")" ); }

    $Ctype=$_POST["Ctype"];

    $res="SELECT Vehicle_id,License_no,Model,Year,Drate AS Daily_Rate,
    Wrate AS Weekly_Rate FROM car AS C JOIN rates AS R
    WHERE C.Ctype=R.Ctype AND C.Ctype='$Ctype'";

    $result=mysqli_query($connection,$res);

    echo "<h1><center>".$Ctype."&nbsp;Cars</h1><br><br>";

?>

<center>

<table border='1'>

<tr>

<th>Vehicle ID</th>

<th>License No</th>
```

```
<th>Model</th>
```

```
<th>Year</th>
```

```
<th>Daily Rate</th>
```

```
<th>Weekly Rate</th>
```

```
</tr>
```

```
<?php
```

```
if (mysqli_num_rows($result) > 0) {
```

```
while($row = mysqli_fetch_assoc($result))
```

```
{
```

```
echo "<tr>";
```

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

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

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

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

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

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

```
echo "</tr>";
```

```
}
```

```
}
```

```
?>
```

```
</table>
```

```
</body>
```

```
</html>
```

Update.php

```
<html>

<body>

<?php

    $database_host = "localhost";

    $database_user = "root";

    $database_pass = "";

    $database_name = "car_rental";

    $connection = mysqli_connect($database_host, $database_user, $database_pass,
$database_name);

    if(mysqli_connect_errno()){

        die("Failed connecting to MySQL database. Invalid credentials" .
mysqli_connect_error(). "(" . mysqli_connect_errno(). ")"); }

    $Drate=$_POST["udrate"];

    $Wrate=$_POST["uwrate"];

    $Ctype=$_POST["Ctype"];

    if(isset($_POST["udrate"]) AND isset($_POST["uwrate"])) {

        $res="UPDATE rates SET Drate=$Drate,Wrate=$Wrate WHERE
Ctype='$Ctype'";

    }

    $result=mysqli_query($connection,$res);

    echo "<h1><center>".$Ctype."&nbsp;Rates updated</h1><br><br>";

?>

</table>

</body>

</html>
```

Book.php

```
<html>

  <head>

    <title>Car Rental </title>

    <link href="styles.css" rel="stylesheet" type="text/css" />

  </head>

  <body>

    <div class="logoBackground">

      <div class="logo"><h6>BOOKINGS</h6></div>

    </div>

    <div class="menu">

      <ul>

        <li><a href="home.php">Home</a></li>

        <li><a href="customer.php">Customers</a></li>

        <li><a href="cars.php">Cars</a></li>

        <li id="active"><a href="">Book Car</a></li>

        <li><a href="return.php">Return Car</a></li>

        <li><a href="report.php">Report</a></li>

      </ul>

    </div>

    <div class="leftContent">
```

<form action="available.php" method="post" >

**<h1> BOOK A CAR </h1>

**

Enter Customer ID: <input type="text" name="cbid"/>

Enter date yyyy-mm-dd: <input type="text" name="Sdate"/>

Select Car type: <select name = "Ctype">

<option>Compact</option>

<option>Medium</option>

<option>Large</option>

<option>SUV</option>

<option>Van</option>

<option>Truck</option>

</select>

Rent type: <select name = "Rtype">

<option>Daily</option>

<option>Weekly</option>

</select>

No of days : <select name = "Days">

<option>0</option><option>1</option><option>2</option><option>3</option><option>4</option>

<option>5</option><option>6</option><option>7</option>

</select>

No of weeks : <select name = "Weeks">

<option>0</option><option>1</option><option>2</option><option>3</option>

</select>

<input type="submit" name="check" value="Check Availability">

Enter Vehicle id: <input type="text" name="Vehicleid"/>

<input type="submit" name="book" value="Book">

```
</form></b>
```

```
<br>
```

```
<b><form action="viewrental.php" method="post" >
```

```
    <h2> VIEW ACTIVE & SCHEDULED RENTALS </h2>
```

```
    <input type="submit" name="submit1" value="View">
```

```
</form></b>
```

```
</div>
```

```
<div class="rightContent">
```

```
<SCRIPT LANGUAGE="JavaScript">
```

```
var num=1
```

```
img1 = new Image ()
```

```
img1.src = "images/1.jpg"
```

```
img2 = new Image ()
```

```
img2.src = "images/2.jpg"
```

```
img3 = new Image ()
```

```
img3.src = "images/3.jpg"
```

```
img4 = new Image ()
```

```
img4.src = "images/4.jpg"
```

```
img5 = new Image ()
```

```
img5.src = "images/5.jpeg"
```

```
img6 = new Image ()
```

```
img6.src = "images/6.jpg"
```

```
text1 = "MEDIUM"
```

```
text2 = "S.U.V"
```

```
text3 = "COMPACT"
```

```
text4 = "LARGE"
```

```
text5 = "TRUCK"
```

```
text6 = "VAN"
```

```
function slideshowUp()
```

```
{
```

```
num=num+1
```

```
if (num==7)
```

```
{num=1}
```

```
document.mypic.src=eval("img"+num+".src")
```

```
document.raj.iname.value=eval("text"+num)
```

```
}
```

```
function slideshowBack()
```

```
{
```

```
num=num-1
```

```
if (num==0)
```

```
{num=6}
```

```
document.mypic.src=eval("img"+num+".src")
```

```
document.raj.iname.value=eval("text"+num)
```

```
}
```

```
</SCRIPT>
```



```
<CENTER>

<IMG SRC="images/1.jpg" NAME="mypic" BORDER=0 HEIGHT="380" WIDTH="500">

<FORM NAME="raj">

<INPUT TYPE="text" WIDTH="100" NAME="iname" VALUE="Medium">

</FORM><A HREF="JavaScript:slideshowBack()"> Back</A> <A
HREF="JavaScript:slideshowUp()"> Next</A>


</div>

</body>

</html>
```

Available.php

```
<html>

<body>

<?php

    $database_host = "localhost";

    $database_user = "root";

    $database_pass = "";

    $database_name = "car_rental";

    $connection = mysqli_connect($database_host, $database_user, $database_pass,
    $database_name);

    if(mysqli_connect_errno()){

        die("Failed connecting to MySQL database. Invalid credentials" .
        mysqli_connect_error(). "(" .mysqli_connect_errno(). ")" ); }


```

```

$Cid=$_POST["cbid"];

$Sdate=$_POST["Sdate"];

$Ctype=$_POST["Ctype"];

$Rtype=$_POST["Rtype"];

$Nodays=$_POST["Days"];

$Noweeks=$_POST["Weeks"];

$Vehicle_id=$_POST["Vehicleid"];

?>

<table border='1'><tr><th>Vehicle ID</th></tr>

<?php

$res="select vehicle_id from car where ctype='$Ctype' and Vehicle_id not in
(SELECT Vehicle_id FROM rental WHERE rental.Ctype='$Ctype' and Sdate='$Sdate')";

$result=mysqli_query($connection,$res);

if ($Noweeks>=1) {

    $num=$Noweeks*7;

    $D2=date('Y-m-d', strtotime($Sdate. ' + $num days'));

}

if ((mysqli_num_rows($result) > 0) || (strtotime($Sdate) > strtotime($D2))) {

    echo "<br><h2>Congrats Vehicle is available</h2><br>";

    echo "<h3>List of Available vehicles</h3><br>";

    while($row = mysqli_fetch_assoc($result))

    {

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

        echo "</tr>";

    }

}

```

```

else

    echo "Car is not available";

?>

</table>

<?php

if($Vehicle_id!=null) {

    $res="INSERT into rental(Cid,Vehicle_id,Ctype,Rtype,Sdate,Nodays,Noweeks)
values('$Cid','$Vehicle_id','$Ctype','$Rtype','$Sdate','$Nodays','$Noweeks')";

    $result=mysqli_query($connection,$res);

    echo "Rental has been added";

}

?>

<?php

$res2="SELECT Rid,Cid,Vehicle_id,Ctype,Rtype,Sdate,Nodays,Noweeks FROM rental";

$result2=mysqli_query($connection,$res2);

echo "<h1><center>Active & Scheduled Rentals</h1><br><br>";

?>

<center>

<table border='1'>

<tr>

<th>RID</th>

<th>Customer ID</th>

<th>Vehicle id</th>

<th>Car type</th>

```

```

<th>Rent type</th>

<th>Start Date</th>

<th>No of days</th>

<th>No of weeks</th>

</tr>

<?php
if (mysqli_num_rows($result2) > 0) {
while($row2 = mysqli_fetch_assoc($result2))
{
echo "<tr>";

echo "<td>" . $row2["Rid"] . "</td>";

echo "<td>" . $row2["Cid"] . "</td>";

echo "<td>" . $row2["Vehicle_id"] . "</td>";

echo "<td>" . $row2["Ctype"] . "</td>";

echo "<td>" . $row2["Rtype"] . "</td>";

echo "<td>" . $row2["Sdate"] . "</td>";

echo "<td>" . $row2["Nodays"] . "</td>";

echo "<td>" . $row2["Noweeks"] . "</td>";

echo "</tr>";

}

}

?>

</table>

</body>

</html>

```

Viewrental.php

```
<html>

<body>

<?php

    $database_host = "localhost";

    $database_user = "root";

    $database_pass = "";

    $database_name = "car_rental";

    $connection = mysqli_connect($database_host, $database_user, $database_pass,
$database_name);

    if(mysqli_connect_errno()){

        die("Failed connecting to MySQL database. Invalid credentials" .
mysqli_connect_error(). "(" .mysqli_connect_errno(). ")" ); }

?>

<?php

$res2="SELECT Rid,Cid,Vehicle_id,Ctype,Rtype,Sdate,Nodays,Noweeks FROM rental";

$result2=mysqli_query($connection,$res2);

echo "<h1><center>Active & Scheduled Rentals</h1><br><br>";

?>

<center>

<table border='1'>

<tr>

<th>RID</th>

<th>Customer ID</th>

<th>Vehicle id</th>
```

```

<th>Car type</th>

<th>Rent type</th>

<th>Start Date</th>

<th>No of days</th>

<th>No of weeks</th>

</tr>

<?php
if (mysqli_num_rows($result2) > 0) {
while($row2 = mysqli_fetch_assoc($result2))
{
echo "<tr>";
echo "<td>" . $row2["Rid"] . "</td>";
echo "<td>" . $row2["Cid"] . "</td>";
echo "<td>" . $row2["Vehicle_id"] . "</td>";
echo "<td>" . $row2["Ctype"] . "</td>";
echo "<td>" . $row2["Rtype"] . "</td>";
echo "<td>" . $row2["Sdate"] . "</td>";
echo "<td>" . $row2["Nodays"] . "</td>";
echo "<td>" . $row2["Noweeks"] . "</td>";
echo "</tr>";
}
}
?>

</table>

```

Return.php

```
<html>
```

```
<head>
```

```
<title>Car Rental </title>
```

```
<link href="styles.css" rel="stylesheet" type="text/css" />
```

```
</head>
```

```
<body>
```

```
<div class="logoBackground">
```

```
<div class="logo"><h6>RETURN CAR </h6></div>
```

```
</div>
```

```
<div class="menu">
```

```
<ul>
```

```
<li><a href="home.php">Home</a></li>
```

```
<li><a href="customer.php">Customers</a></li>
```

```
<li><a href="cars.php">Cars</a></li>
```

```
<li><a href="book.php">Book Car</a></li>
```

```
<li id="active"><a href="">Return Car</a></li>
```

```
<li><a href="report.php">Report</a></li>
```

```
</ul>
```

```
</div>
```

```
<div class="leftContent">
```

```
<b><form action="return1.php" method="post">
```

```
<h1> RENTAL BILL </h1><br><br>
```

```
Enter Rental ID: <input type="text" name="rid"/><br><br>
```

```
<input type="submit" name="enter" value="Enter">
```

```
</form>
```

```
</body>
```

```
</html>
```

Return1.php

```
<html>
```

```
<body>
```

```
<?php
```

```
    $database_host = "localhost";
```

```
    $database_user = "root";
```

```
    $database_pass = "";
```

```
    $database_name = "car_rental";
```

```
    $connection = mysqli_connect($database_host, $database_user, $database_pass,  
$database_name);
```

```
    if(mysqli_connect_errno()){
```

```
        die("Failed connecting to MySQL database. Invalid credentials" .
```

```
mysqli_connect_error(). "(" .mysqli_connect_errno(). ")"); }
```

```
    $Rid=$_POST["rid"];
```

```
    $res="select Noweeks FROM rental,rates where rental.rid='$Rid' and  
rental.Ctype=rates.Ctype";
```

```
    $result=mysqli_query($connection,$res);
```

```
    $row = mysqli_fetch_assoc($result);
```



```

        if ($row["Noweeks"]==0) {

            $res1="select rental.Nodays*rates.Drate AS Amount FROM rental, rates
where rental.rid='$Rid' and rental.Ctype=rates.Ctype";

            $result1=mysqli_query($connection,$res1);

            echo "<h1><center>Amount due</h1><br><br>";

            $row1 = mysqli_fetch_assoc($result1);

            echo "<h1>rupees".$row1["Amount"]."</h1>";

        }

        else{

            $res2="select rental.Noweeks*rates.Wrate AS Amount FROM rental, rates
where rental.rid='$Rid' and rental.Ctype=rates.Ctype";

            $result2=mysqli_query($connection,$res2);

            echo "<h1><center>Amount due</h1><br><br>";

            $row2 = mysqli_fetch_assoc($result2);

            echo "<h1>rupees".$row2["Amount"]."</h1>";

        }

    ?>

</body>

</html>

```

Report.php

```

<html>

<head>

    <title>Car Rental </title>

    <link href="styles.css" rel="stylesheet" type="text/css" />

```

</head>

<body>

<div class="logoBackground">

<div class="logo"><h6>WEEKLY REPORT </h6></div>

</div>

<div class="menu">

Home

Customers

Cars

Book Car

Return Car

<li id="active">Report

</div>

<div class="leftContent">

<form action="report0.php" method="post">

<h2>View profits for owners :</h2>

<input type="submit" name="cartype1" value="View based on Owner id">

</form>

<form action="report1.php" method="post">

<h2>View profits for Car types :</h2>

<input type="submit" name="cartype1" value="View based on Car Types">

```
</form>

<br><br>

<form action="report2.php" method="post">

<h2>View profits for individual car units :</h2> <br>

<input type="submit" name="cartype1" value="View based on Car units">

</form>


</body>

</html>
```

Report0.php

```
<html>

<body>

<?php

    $database_host = "localhost";

    $database_user = "root";

    $database_pass = "";

    $database_name = "car_rental";

    $connection = mysqli_connect($database_host, $database_user, $database_pass,
    $database_name);

    if(mysqli_connect_errno()){

        die("Failed connecting to MySQL database. Invalid credentials" .
        mysqli_connect_error(). "(" .mysqli_connect_errno(). ")" ); }

    $res="select DISTINCT Owner_id,(rental.Nodays*rates.Drate+
    rental.Noweeks*rates.Wrate) AS Amount

    FROM rental, rates ,owns
```

```
where rental.Ctype=rates.Ctype and rental.Vehicle_id=owns.Vehicle_id
```

```
group by Owner_id";
```

```
    $result=mysqli_query($connection,$res);
```

```
    echo "<h1><center>Profits according to owners</h1><br><br>";
```

```
?>
```

```
<center>
```

```
<table border='1'>
```

```
<tr>
```

```
<th>Owner id</th>
```

```
<th>Amount</th>
```

```
</tr>
```

```
<?php
```

```
if (mysqli_num_rows($result) > 0) {
```

```
while($row = mysqli_fetch_assoc($result))
```

```
{
```

```
echo "<tr>";
```

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

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

```
echo "</tr>";
```

```
}
```

```
}
```

```
?>
```

```
</table>
```

```
</body>
```

```
</html>
```

Report1.php

```
<html>

<body>

<?php

    $database_host = "localhost";

    $database_user = "root";

    $database_pass = "";

    $database_name = "car_rental";

    $connection = mysqli_connect($database_host, $database_user, $database_pass,
    $database_name);

    if(mysqli_connect_errno()){

        die("Failed connecting to MySQL database. Invalid credentials" .
mysqli_connect_error(). "(" .mysqli_connect_errno(). ")" ); }

    $res="select Owner_id,rental.Ctype AS Category,SUM(rental.Nodays*rates.Drate+
rental.Noweeks*rates.Wrate) AS Earnings

FROM rental, rates ,owns

where rental.Ctype=rates.Ctype and rental.Vehicle_id=owns.Vehicle_id

group by Owner_id";

    $result=mysqli_query($connection,$res);

    echo "<h1><center>Profits according to car types</h1><br><br>";

?>

<center>

<table border='1'>

<tr>

<th>Owner id</th>

<th>Category</th>

<th>Earnings</th>
```

```
</tr>

<?php
if (mysqli_num_rows($result) > 0) {
while($row = mysqli_fetch_assoc($result))
{
echo "<tr>";
echo "<td>" . $row["Owner_id"] . "</td>";
echo "<td>" . $row["Category"] . "</td>";
echo "<td>" . $row["Earnings"] . "</td>";
echo "</tr>";

}
}

?>

</table>

</body>

</html>
```

Report2.php

```
<html>

<body>

<?php

    $database_host = "localhost";

    $database_user = "root";

    $database_pass = "";

    $database_name = "car_rental";
```

```

    $connection = mysqli_connect($database_host, $database_user, $database_pass,
$database_name);

    if(mysqli_connect_errno()){

        die("Failed connecting to MySQL database. Invalid credentials" .
mysqli_connect_error(). "(" .mysqli_connect_errno(). ")" ); }

    $res="select Owner_id,rental.Vehicle_id AS Vid,

(rental.Nodays*rates.Drate + rental.Noweeks*rates.Wrate) AS Amount,

rental.Ctype AS Category

FROM rental, rates ,owns

where rental.Ctype=rates.Ctype and rental.Vehicle_id=owns.Vehicle_id";

    $result=mysqli_query($connection,$res);

    echo "<h1><center>Profits according to individual cars</h1><br><br>";

?>

<center>

<table border='1'>

<tr>

<th>Owner id</th>

<th>Vehicle id</th>

<th>Amount</th>

<th>Car type</th>

</tr>

<?php

if (mysqli_num_rows($result) > 0) {

while($row = mysqli_fetch_assoc($result))

{

echo "<tr>";

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

```

```
echo "<td>" . $row["Vid"] . "</td>";  
echo "<td>" . $row["Amount"] . "</td>";  
echo "<td>" . $row["Category"] . "</td>";  
echo "</tr>";  
  
}  
  
}  
  
?>  
  
</table>  
  
</body>  
  
</html>
```


OUTPUT PAGES

