***UNIFIED MODELING LANGUAGE***

* ***Introduction***

UML (Unified Modeling Language) is a general-purpose, graphical modeling language in the field of Software Engineering. UML is used to specify, visualize, construct, and document the artifacts (major elements) of the software system. It was initially developed by Grady Booch, Ivar Jacobson, and James Rumbaugh in 1994-95 at Rational software, and its further development was carried out through 1996. In 1997, it got adopted as a standard by the Object Management Group.

UML (Unified Modeling Language) is a general-purpose, graphical modeling language in the field of Software Engineering. UML is used to specify, visualize, construct, and document the artifacts (major elements) of the software system. It was initially developed by Grady Booch, Ivar Jacobson, and James Rumbaugh in 1994-95 at Rational software, and its further development was carried out through 1996. In 1997, it got adopted as a standard by the Object Management Group.

* ***What is UML?***

The UML stands for Unified modeling language, is a standardized general-purpose visual modeling language in the field of Software Engineering. It is used for specifying, visualizing, constructing, and documenting the primary artifacts of the software system. It helps in designing and characterizing, especially those software systems that incorporate the concept of Object orientation. It describes the working of both the software and hardware systems.

The UML was developed in 1994-95 by Grady Booch, Ivar Jacobson, and James Rumbaugh at the Rational Software. In 1997, it got adopted as a standard by the Object Management Group (OMG).

The Object Management Group (OMG) is an association of several companies that controls the open standard UML. The OMG was established to build an open standard that mainly supports the interoperability of object-oriented systems. It is not restricted within the boundaries, but it can also be utilized for modeling the non-software systems. The OMG is best recognized for the Common Object Request Broker Architecture (CORBA) standards.

**NOTE: -**Diagrams are based on example of CAR RENTAL SERVICE MANAGEMENT.

* ***Introduction***
  1. **Class: -**

In object-oriented programming, a class is an extensible program-code-template for creating objects, providing initial values for state (member variables) and implementations of behaviour (member functions or methods). In many languages, the class name is used as the name for the class (the template itself), the name for the default constructor of the class (a subroutine that creates objects), and as the type of objects generated by instantiating the class; these distinct concepts are easily conflated. Although, to the point of conflation, one could argue that is a feature inherent in a language because of its polymorphic nature and why these languages are so powerful, dynamic, and adaptable for use compared to languages without polymorphism present. Thus, they can model dynamic systems (i.e., the real world, machine learning, AI) more easily.

* 1. **Object: -**

In the object-oriented programming paradigm, object can be a combination of variables, functions, and data structures; in particular in class-based variations of the paradigm it refers to a particular instance of a class. An Object is an identifiable entity with some characteristics and behaviour. An Object is an instance of a Class.

When a class is defined, no memory is allocated but when it is instantiated (i.e., an object is created) memory is allocated. Each object contains data and code to manipulate the data. Objects can interact without having to know details of each other’s data or code, it is sufficient to know the type of message accepted and type of response returned by the objects.

* 1. **Attribute/Field: -**

Attributes can be defined by each class. When we represent them as objects, we assign values to those attributes. For e.g., even every ‘file’ object has a path. Attributes are also known as properties.

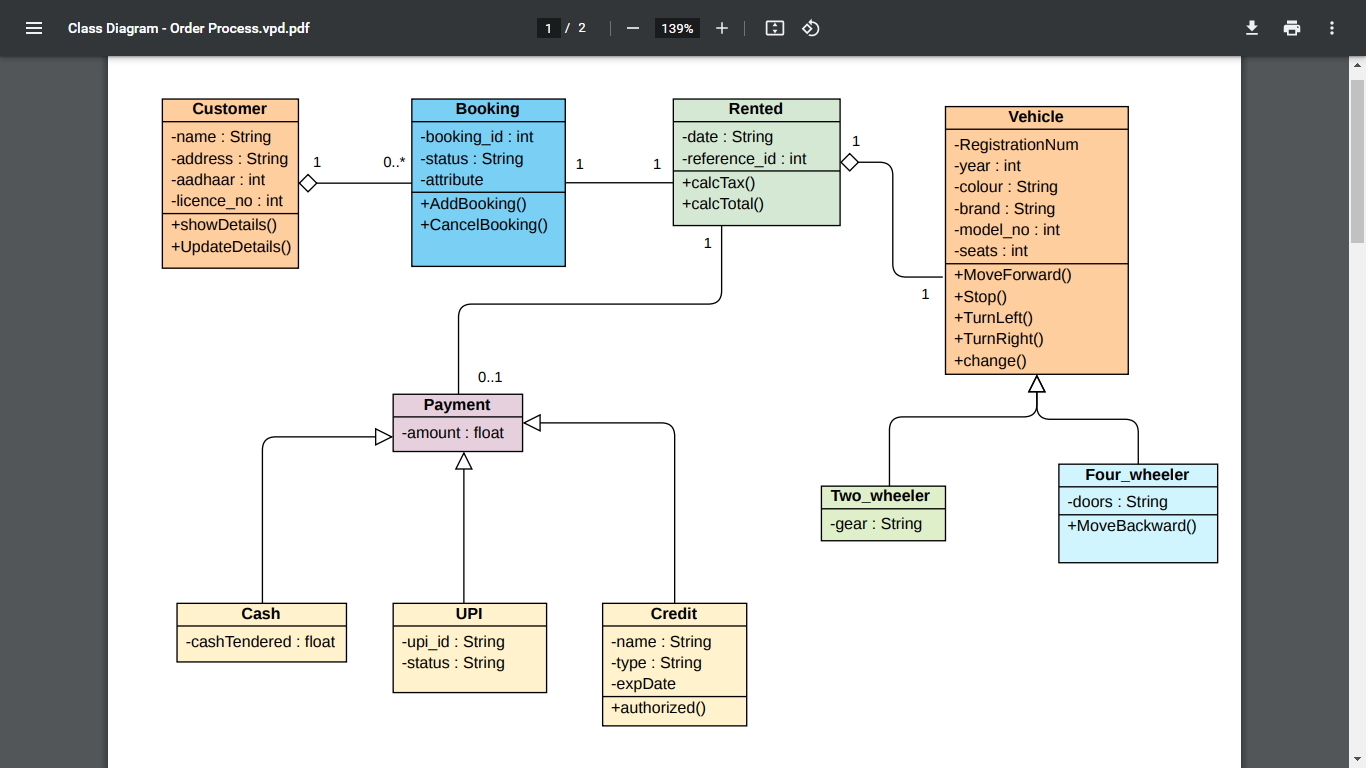
Attributes are typically defined as read-only or read-write. Read-only attributes can be set only when the object is created, while read-write attributes can be altered at any time. The value of an attribute can itself be another object. And it is not necessary for every class to have attributes and methods.

* 1. **Relationship: -**

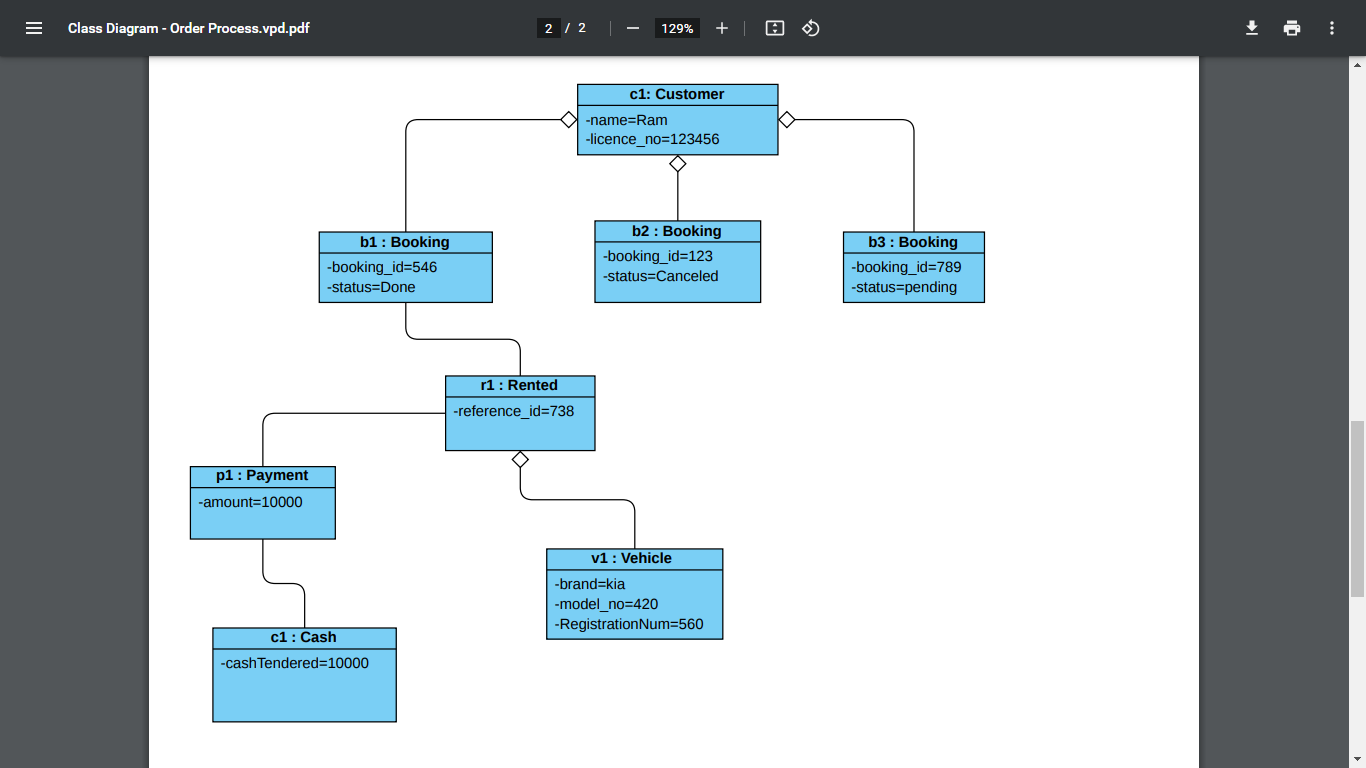
Relationships in UML are used to represent a connection between structural, behavioural, or grouping things. They describe the behaviour that is desired, or can be expected, between elements. It is also called a link that describes how two or more things can relate to each other during the execution of a system.

**Aggregation: -** In some cases, we might come across “whole-part” relationships between two things, in which one thing is a part of another thing. Such relationship is known as an aggregation or also known as “has-a” relationship.

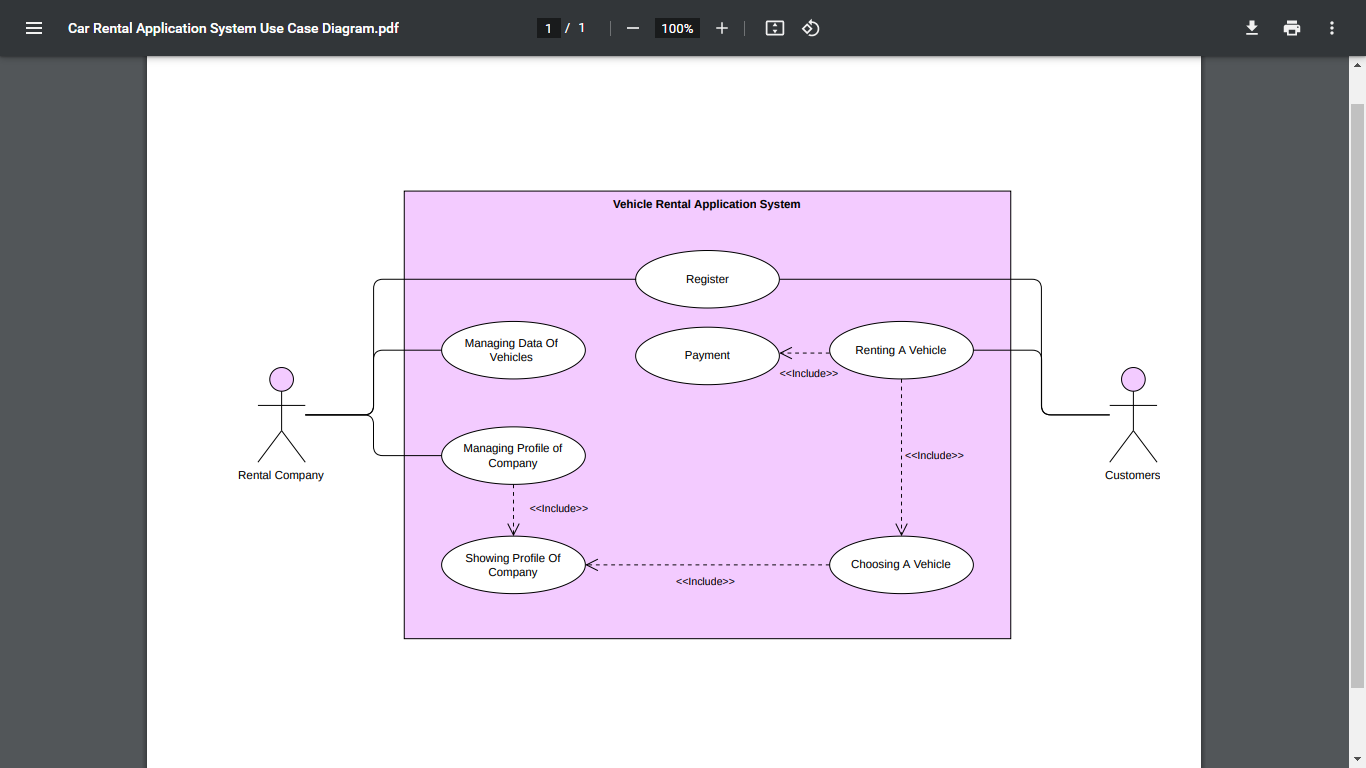
* **Class Diagram: -**

****

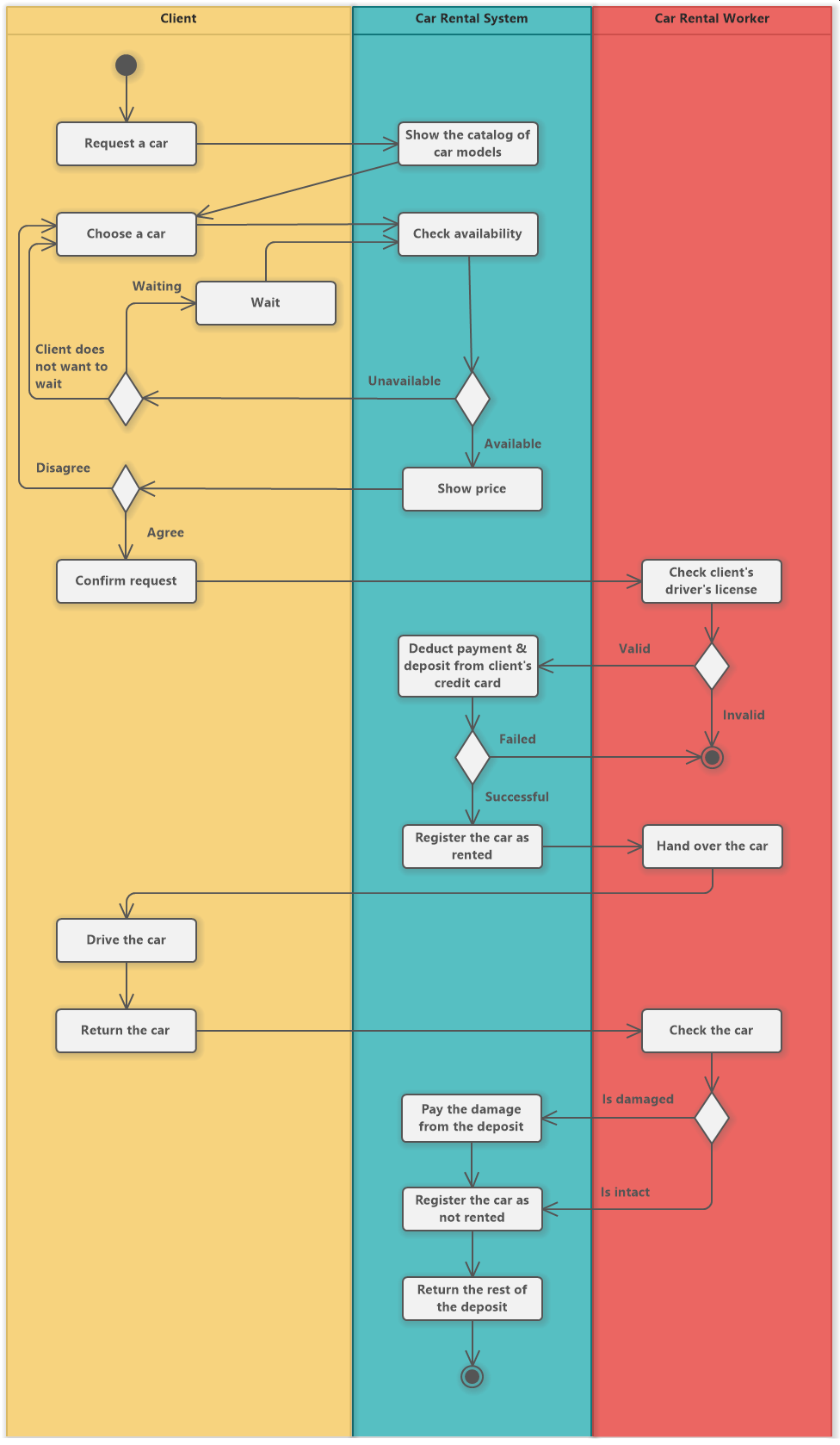
* **Object Diagram: -**

****

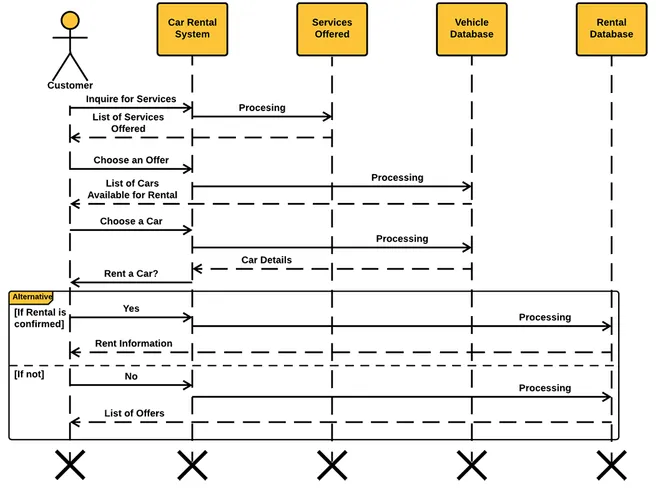
* ***Types of UML Diagram***
  1. **Use case Diagram: -**

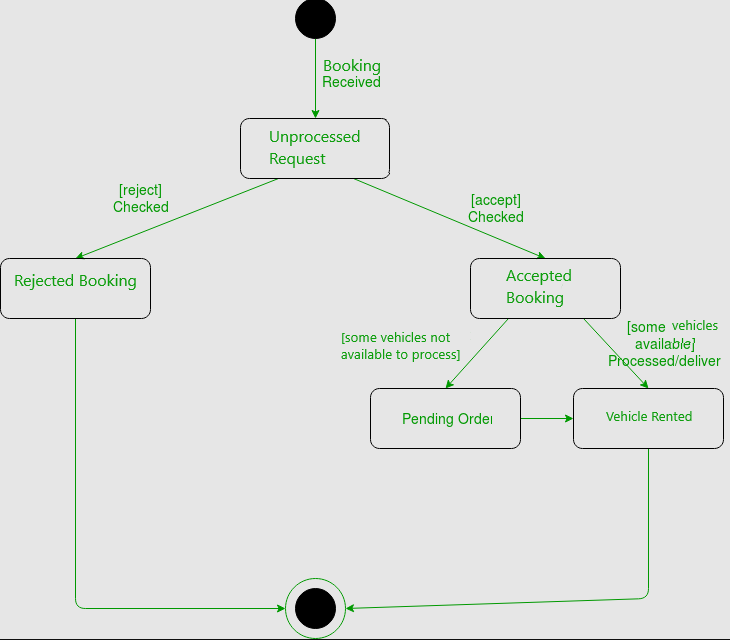
****

* 1. **Activity Diagram: -**

****

* 1. **Sequence Diagram: -**

****

* 1. **State Chart Diagram: -**

**Name: -Abhinay Pawar**

**Roll.No: -BCA21001**

**Class: -SYBCA - A**