

## **4.1 SYSTEM ARCHITECTURE:**

This section consists of the UML diagrams related to the modules developed such as Activity diagram, Sequence diagram and Use case diagram.

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

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

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

### **GOALS:**

The Primary goals in the design of the UML are as follows:

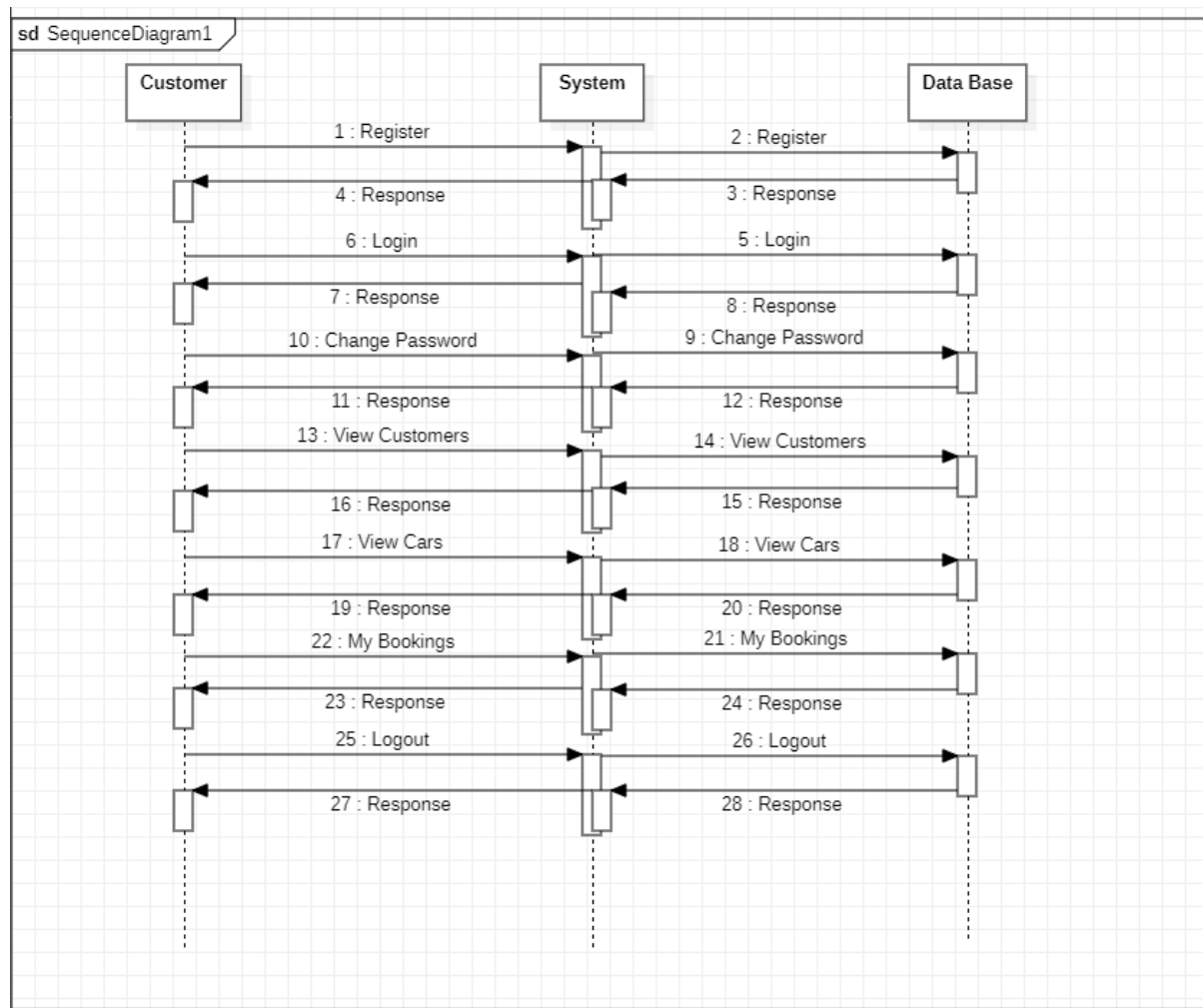
- Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
- Provide extendibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development process.
- Provide a formal basis for understanding the modelling language.
- Encourage the growth of OO tools market.

- Support higher level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices.

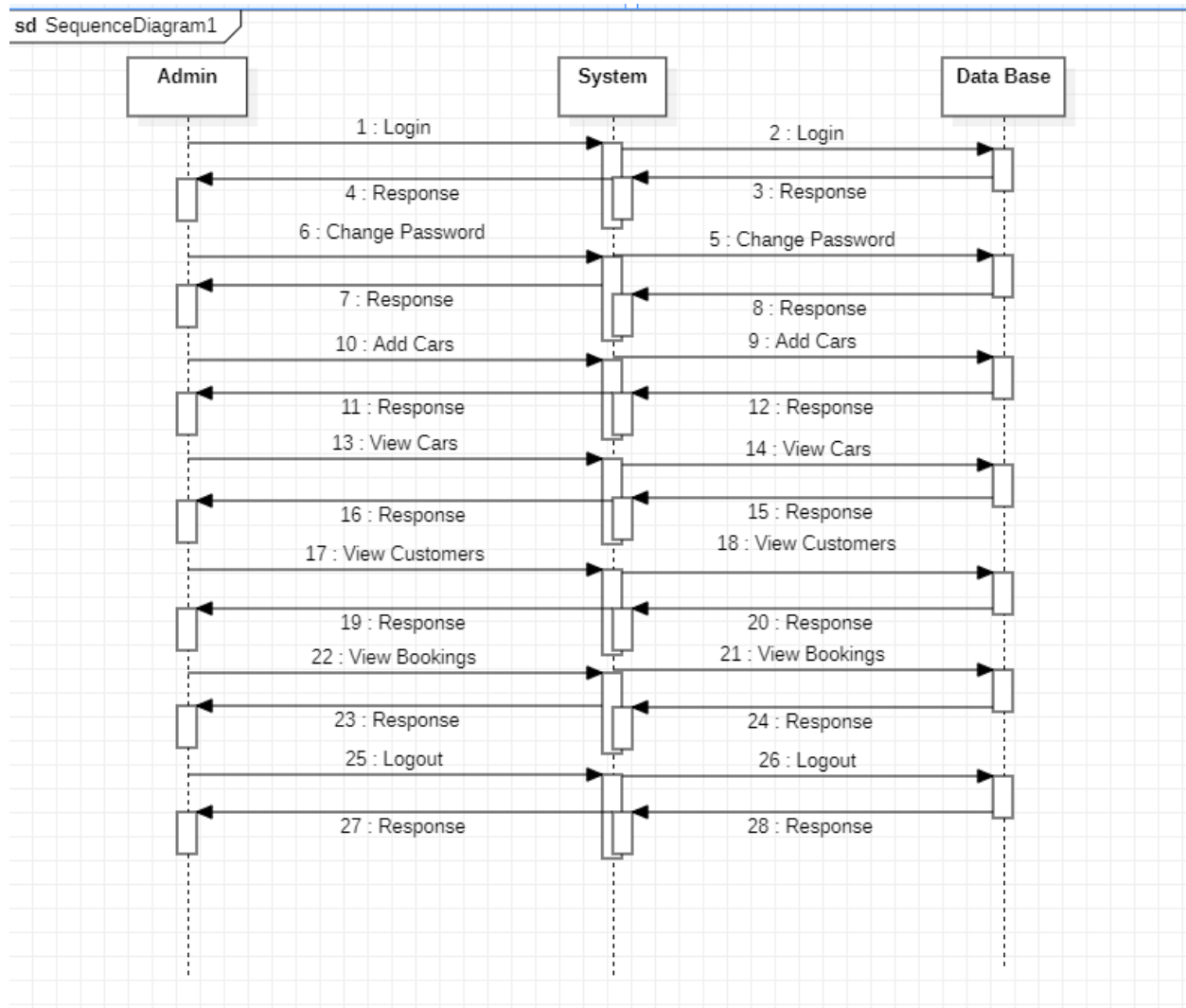
## Sequence Diagram

A grouping outline in UML is a sort of association chart that shows how procedures work with each other and in what request. “It is a build of a Message Sequence Chart. Succession outlines are now and again called occasion charts, occasion situations, and timing graphs”.

### Customer -Sequence Diagram:



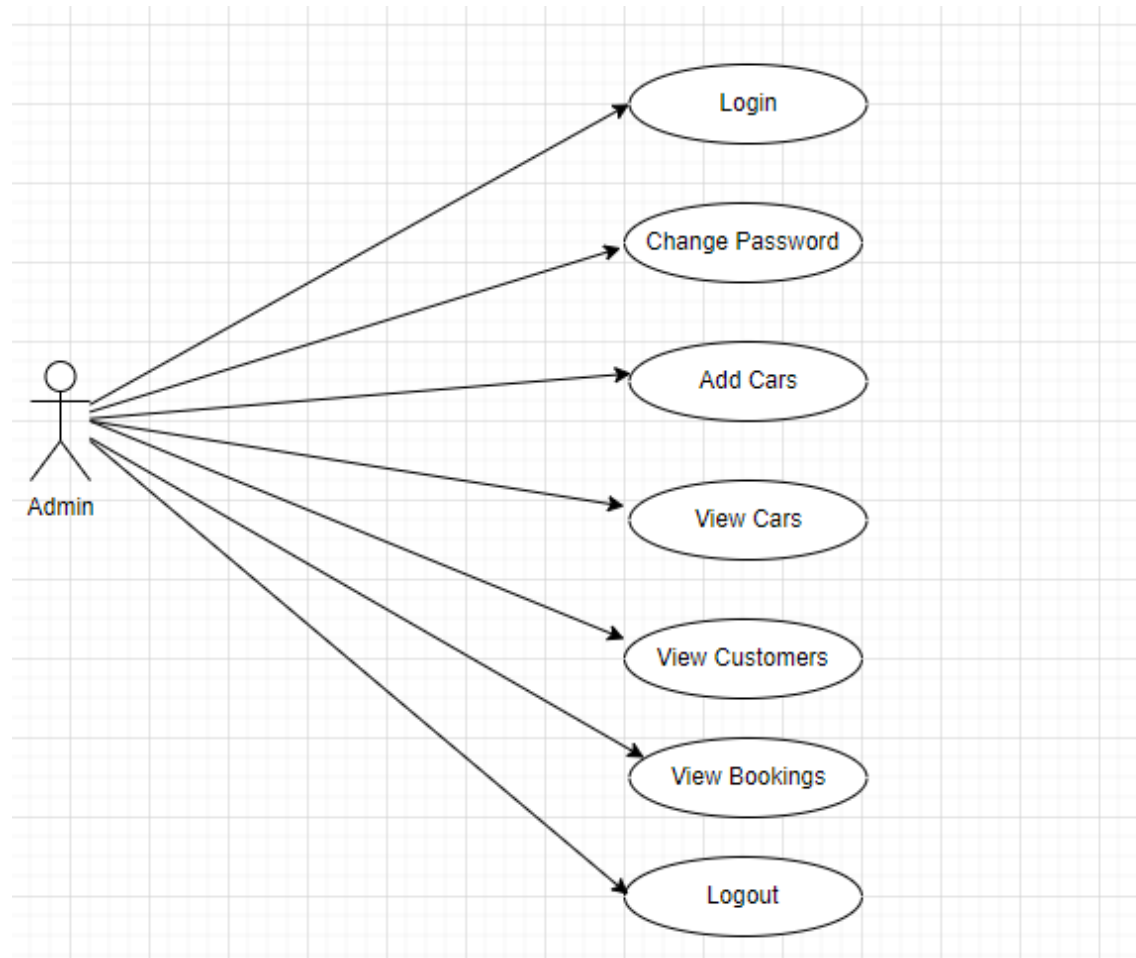
## Admin-Sequence Diagram:



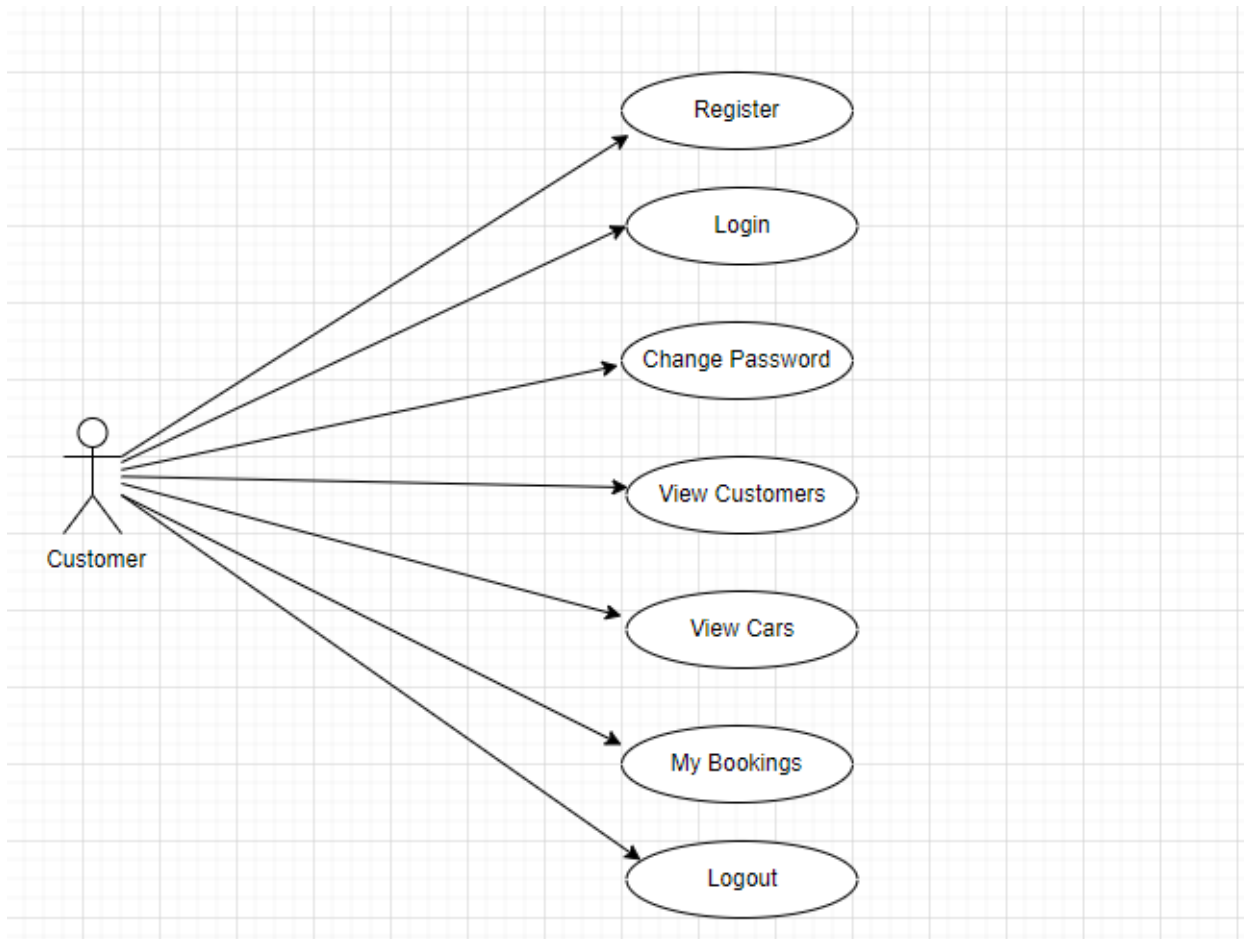
## Use Case Diagram

An utilization case outline in the UML is a sort of conduct graph characterized by and made from a Use-case examination. Its motivation is to introduce a graphical diagram of the usefulness given by a framework regarding on-screen characters, their objectives (spoke to as use cases), and any conditions between those utilization cases. The fundamental motivation behind an utilization case outline is to indicate what framework capacities are performed for which on-screen character. Jobs of the on-screen characters in the framework can be delineated”.

### Admin-Use Case Diagram:



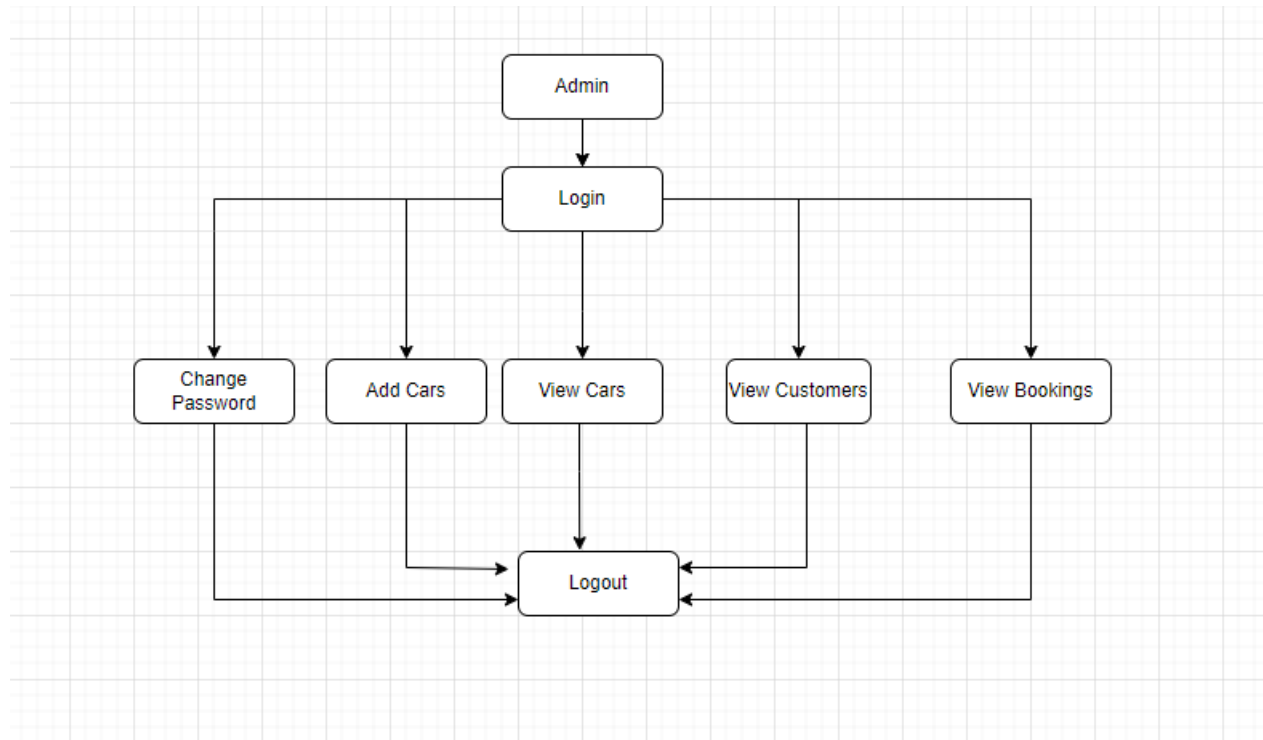
### Customer-Use Case Diagram:



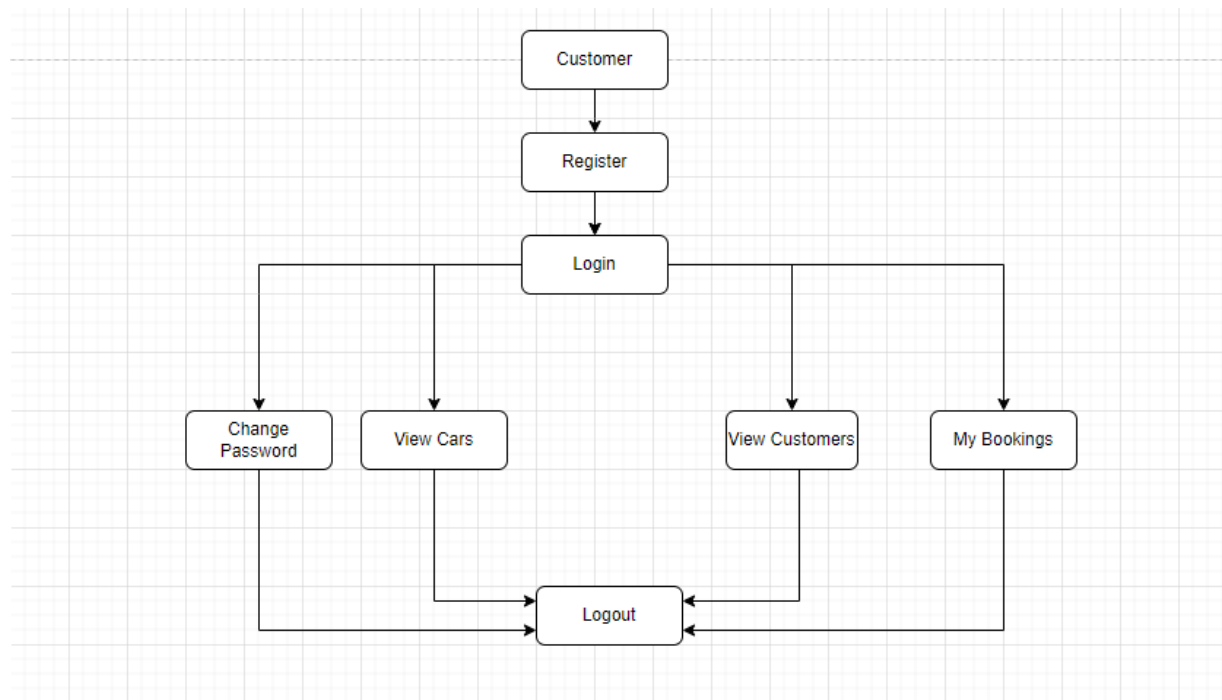
### Activity Diagram

“Movement charts are graphical portrayals of work processes of stepwise exercises and activities with help for decision, emphasis and simultaneousness. In the Unified Modeling Language, movement graphs can be utilized to depict the business and operational bit by bit work processes of parts in a framework. A movement chart demonstrates the general progression of control”.

### Admin-Activity Diagram:

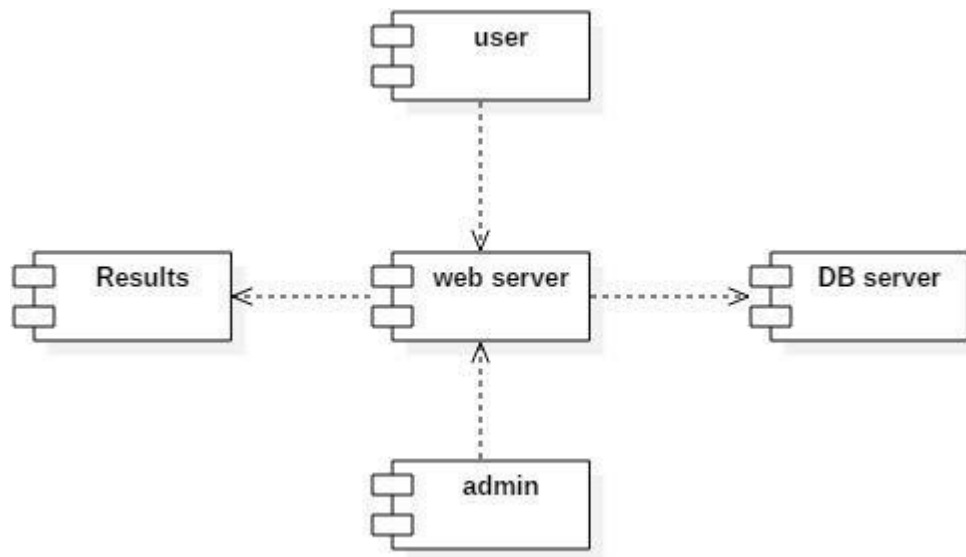


### Customer-Activity Diagram:



## Component Diagram

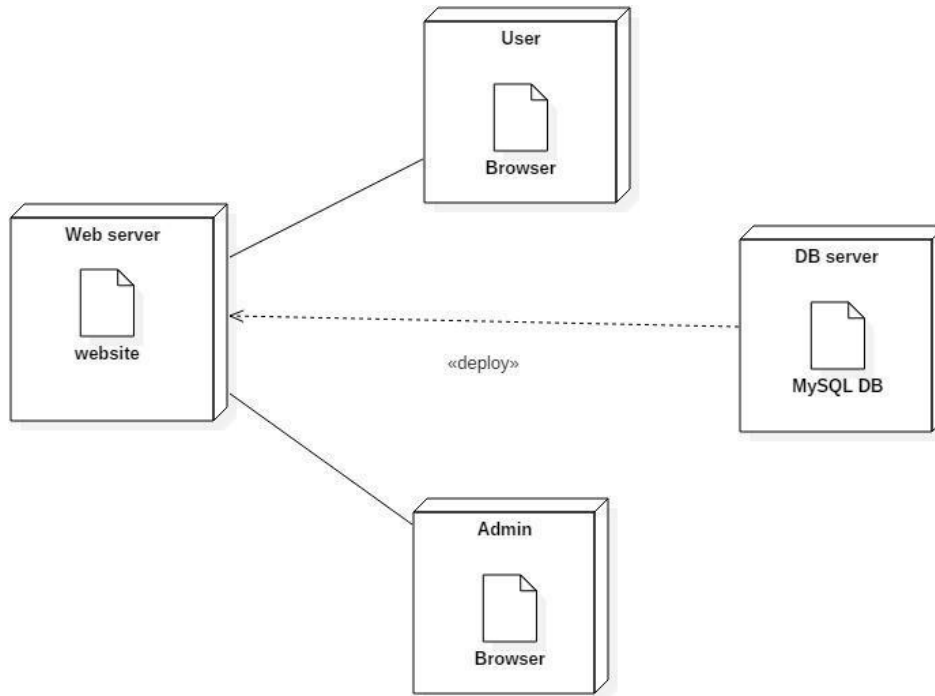
The component diagram is a special purpose diagram, which is used to visualize the static implementation view of a system. It represents the physical components of a system, or we can say it portrays the organization of the components inside a system. The components, such as libraries, files, executables, etc.



executables, etc.

## Deployment Diagram

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system

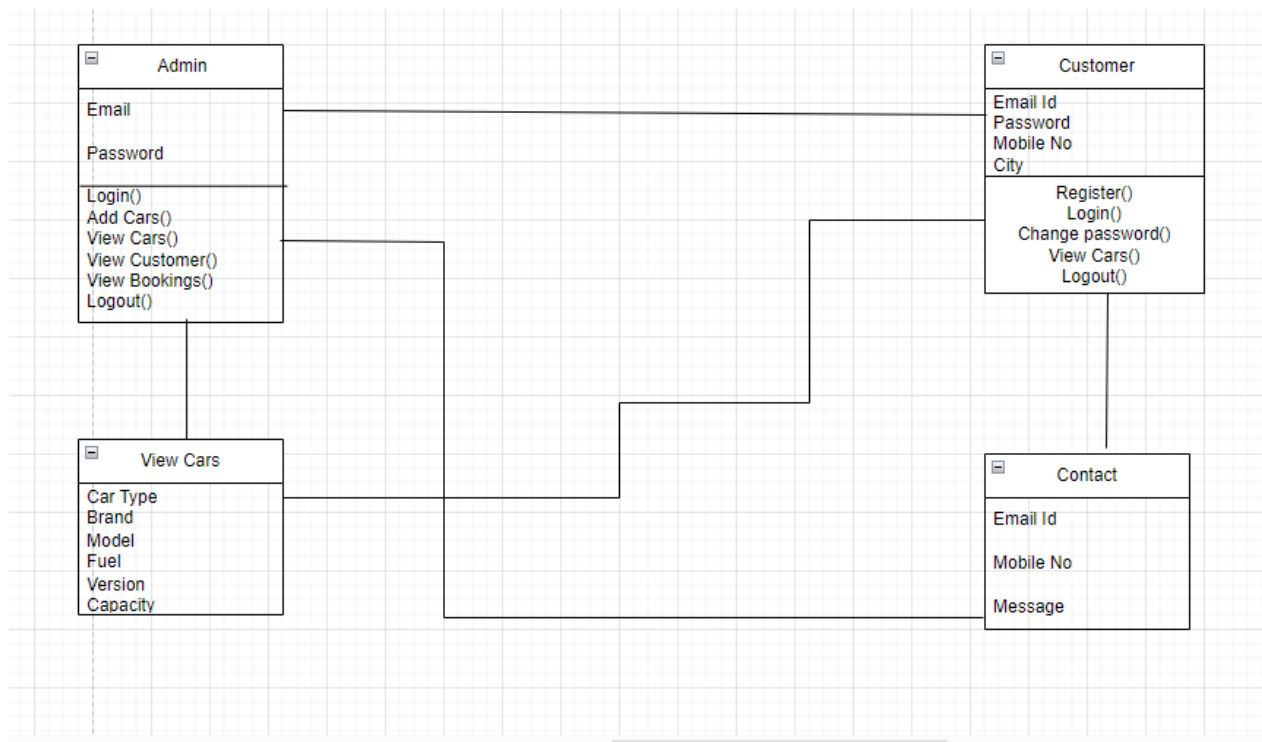


## Class Diagram:

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

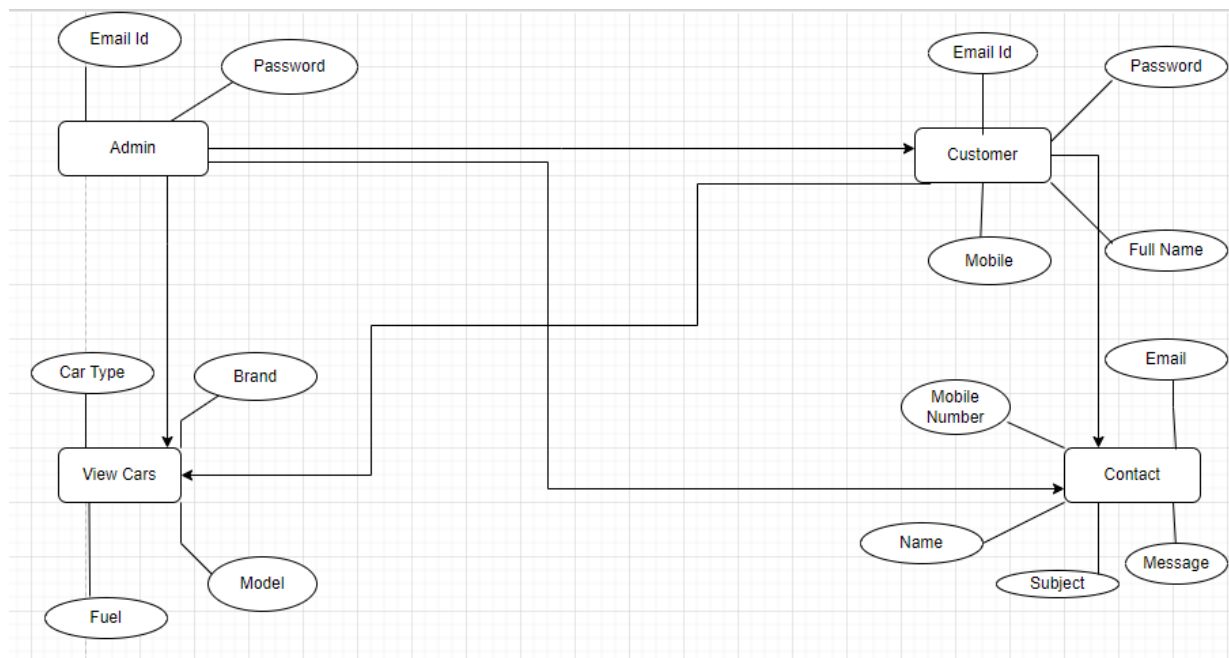
Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.





## ER Diagram

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.



## Data Flow Diagram

A data-flow diagram is a way of representing a flow of data through a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow — there are no decision rules and no loops. A ***data flow diagram*** (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows,

