**HomeSpiration**

**Your Creative Home Improvement Companion**

**UML Diagrams**

**Under the Guidance of : Submitted By :**

Mr Ajith G S Uthraja U

Assistant Professor AJCE Roll No : 48

MCA Department Adm No : 14099

**UML DIAGRAM :**

## The components of the principles of object-oriented programming are represented by the language known as the Unified Modeling Language (UML), which is utilized in the industry of software engineering. It serves as the standard definition of the entire software architecture or structure. Complex algorithms are solved and interacted with in Object-Oriented Programming by treating them as objects or entities. Anything can be one of these things. It could either be a bank manager or the bank itself. The thing can be a machine, an animal, a vehicle, etc. The issue is how we connect with and control them, even though they are capable of and ought to execute duties. Interacting with other objects, sending data from one object to another, manipulating other objects, etc., are examples of tasks. There could be hundreds or even thousands of objects in a single piece of software.

## UML includes the following four diagrams.

## Class Diagram

## Object Diagram

## Use Case Diagram

## Sequence Diagram

## Collaboration Diagram

## Activity Diagram

## State Chart Diagram

## Deployment Diagram

## Component Diagram

## USE CASE DIAGRAM :

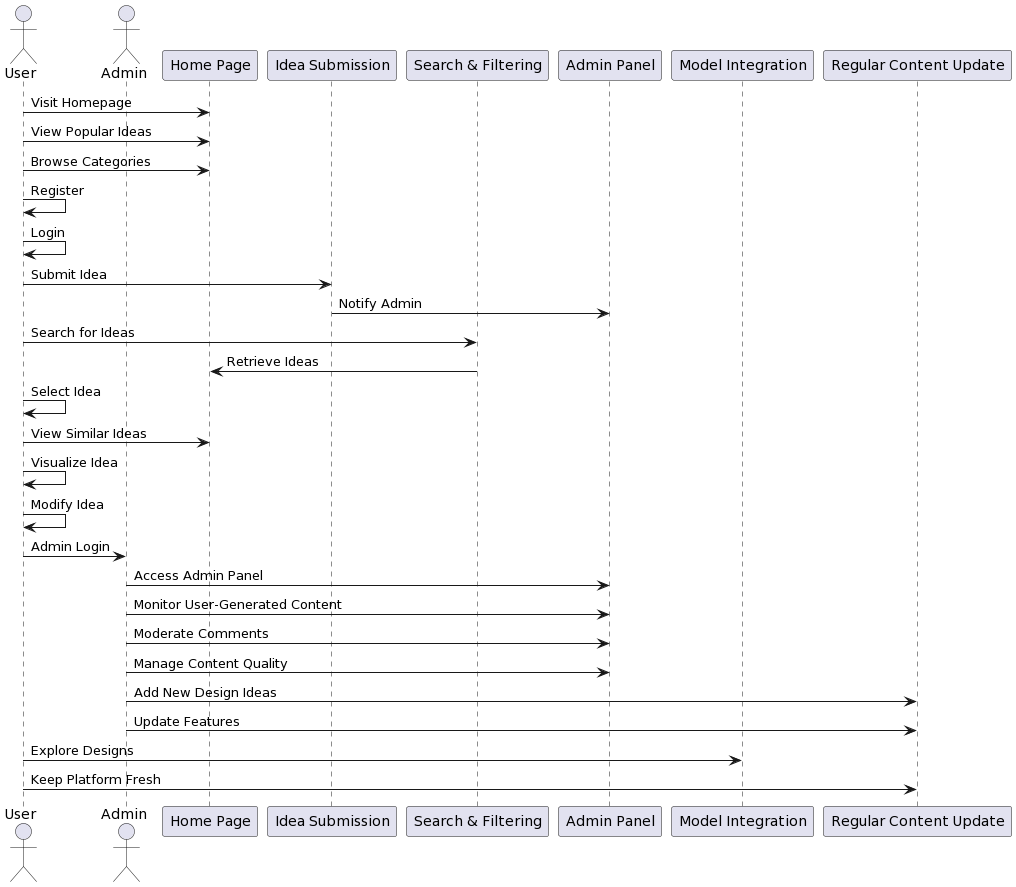
## A use case diagram is a visual representation of the interactions between system components. An approach for identifying, outlining, and organizing system requirements is called a use case. The word "system" in this context refers to a project or business that is under development or operation, such a mail-order goods sales and service web page. The Unified Modeling Language (UML) makes use of use case diagrams. a common notation for simulating systems and things in the actual world. Planning for overall requirements is one of the system objectives. Testing and debugging a software product, and verifying a hardware design Performing a consumer service, developing, writing an online help guide, or focused task Use cases in a product sales context, for instance, would include ordering of goods, catalogue revision, transaction processing, and client.

## Use Case Diagrams are created to depict a system’s functional requirements. To create an effective use case diagram after identifying the aforementioned things, we must adhere to the following rules. The name of use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.

## PlantUML Diagram

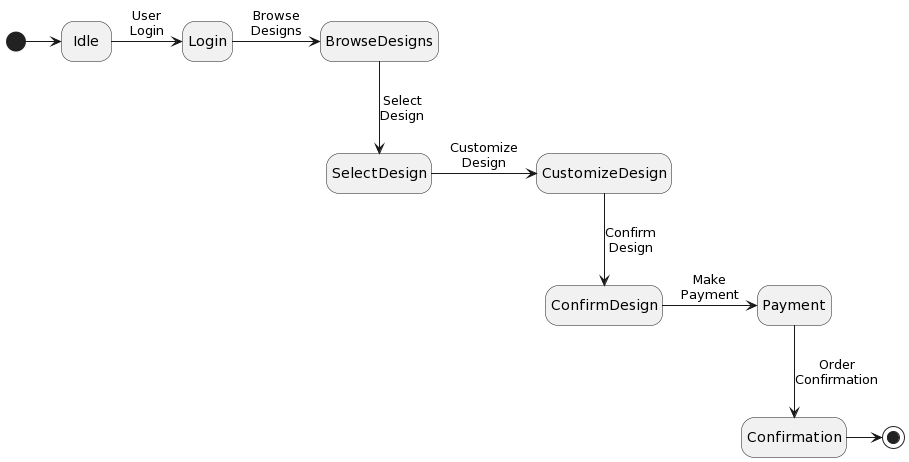
**SEQUENCE DIAGRAM :**

## A sequence diagram essentially shows how things interact with one another sequentially, or the order in which these interactions occur. A sequence diagram can also be referred to as event diagrams or event scenarios. Sequence maps define the actions that the system's components take and in what order. These schematics are Businesspeople and software developers frequently employ documentation and understanding specifications for both current and future systems.



## STATE CHART DIAGRAM :

## Diagrams of state charts are used to show a software system's behavior. A state machine diagram in a UML model can represent the behavior of a class, a subsystem, a package, or even the entire system. It is also known as a state chart or a state transition diagram. We can effectively describe the communications or interactions that occur between external entities and a system using state chart diagrams. These diagrams are used to model the event-based system. An event can be used to control an object's state. State chart diagrams are used in the application system to show the various states of an entity.



## ACTIVITY DIAGRAM :

An activity diagram is a visual representation of how events unfold simultaneously or sequentially. It aids in understanding the flow of activities, emphasizing the progression from one task to the next. Activity diagrams focus on the order in which tasks occur and can depict various types of flows, including sequential, parallel, and alternative paths. To facilitate these flows, activity diagrams incorporate elements such as forks and join nodes, aligning with the concept of illustrating the functioning of a system in a specific manner.

## PlantUML Diagram

## CLASS DIAGRAM :

## Class diagrams are a type of static diagram. It represents the static view of the application. Class diagrams are used for visualizing, describing, and documenting various system components as well as for producing executable code for software applications. A class diagram describes the constraints imposed on the system along with the properties and operations of a class. Class diagrams are widely used I the design of object oriented systems since they are the only UML diagrams that can be directly translated using object-oriented languages. A class diagram shows a collection of classes, interfaces, affiliations, collaborations, and constraints. It is also known as a “structural diagram”

## PlantUML Diagram

## OBJECT DIAGRAM :

Class diagrams are a requirement for object diagrams because they are the source of class diagrams. An object diagram illustrates a specific instance of a class diagram. Diagrams of classes and objects share the same underlying concepts. Object diagrams can also be used to describe a system's static view, which is a brief snapshot of the system. Object diagrams are used to illustrate a collection of items and their connections as an example.

## 

## COMPONENT DIAGRAM :

A component diagram is used to separate the smaller parts of an intricate object-oriented system, making it easier to comprehend. It simulates a system's physical perspective, including all of its internal executables, files, libraries, and other nodes. It shows the relationships and hierarchies between the system's parts. It helps build a functional system. A component is an individual, exchangeable, and executable system unit. Because, a component's implementation details are hidden, an interface is necessary in order to perform a function. With the available and required interfaces explaining its behavior, it operates like a "black box."

## 

## DEPLOYMENT DIAGRAM :

Deployment diagrams show the topology of a system’s physical components, where the software components are installed. Deployment diagrams are used to describe a system’s static deployment view. The key elements of deployment diagrams are nodes and connections between them. The word “deployment” itself explains the diagram’s purpose. Deployment diagrams are used to show the actual hardware that software components are deployed to. Component diagrams and Deployment diagrams share many similarities.

## 