## 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 industryof 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 thesethings. 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 theyare capable of and ought to execute duties. Interacting with other objects, sending data fromone 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 include the following diagrams:

* + - Class diagram
    - Object diagram
    - Use case diagram
    - Sequence diagram
    - Activity diagram
    - Statechart diagram
    - Deployment diagram
    - Component diagram

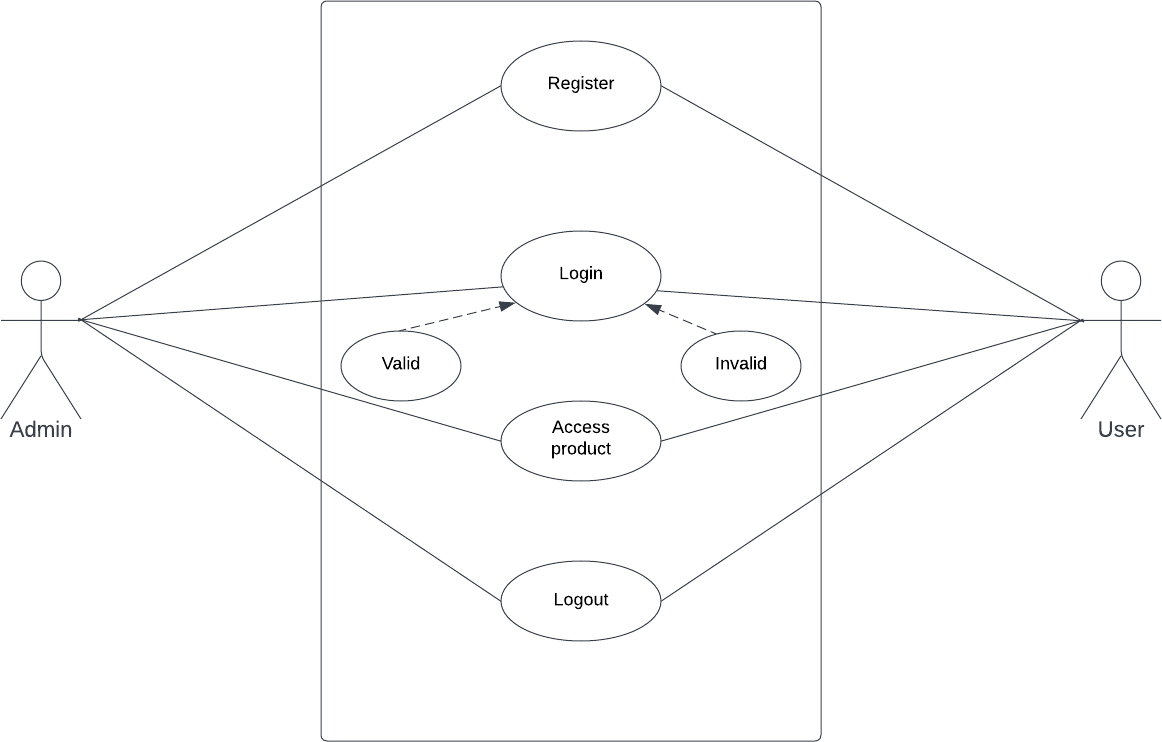
## USE CASE DIAGRAM

A use case diagram is a visual representation of the interactions between system components. A approach for identifying, outlining, and organizing system requirements is called a use case. The word "system" here refers to a thing that is being created or run, like a website for mail-order product sales and services. UML (Unified Modeling Language), a standard language for the modelling of real-world objects and systems, uses use case diagrams.

The planning of general requirements, the validation of a hardware design, the testing and debugging of a software product in development, the creation of an online help reference, or the completion of a job focused on customer support are all examples of system objectives. For instance, use cases in a product sales context can involve customer service, item ordering, catalogue updating, and payment processing. There are four elements in a use case diagram.

* + - * The boundary, which defines the system of interest in relation to the world around it.
      * The actors, usually individuals involved with the system defined according to theirroles.
      * The use cases, which are the specific roles are played by the actors within and around the system.
      * The relationships between and among the actors and the use cases.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficientuse case 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 diagrams show the actions taken by the components of a system in chronological order. Businesspeople and software engineers frequently use these diagrams to record and comprehend the requirements for new and current systems.

Sequence Diagram Notations –

1. **Actors –** In a UML diagram, an actor represents a particular kind of role that interacts with the system and its objects. An actor is always beyond the purview of the system that we want to use the UML diagram to represent. We employ actors to portray a variety of roles, including those of human users and other outside subjects. In a UML diagram, an actor is represented using a stick personnotation. In a sequence diagram, there might be several actors.
2. **Lifelines –** A lifeline is a named element in a sequence diagram that represents an individual participant. So, in a sequence diagram, each incident is represented by a lifeline. A sequence diagram's lifeline elements are at the top.
3. **Messages –** Messages are used to show how objects communicate with one another.The messages are displayed on the lifeline in chronological sequence. Arrows are how messages are represented. A sequence diagram's main components are lifelines and messages.

Messages can be broadly classified into the following categories:

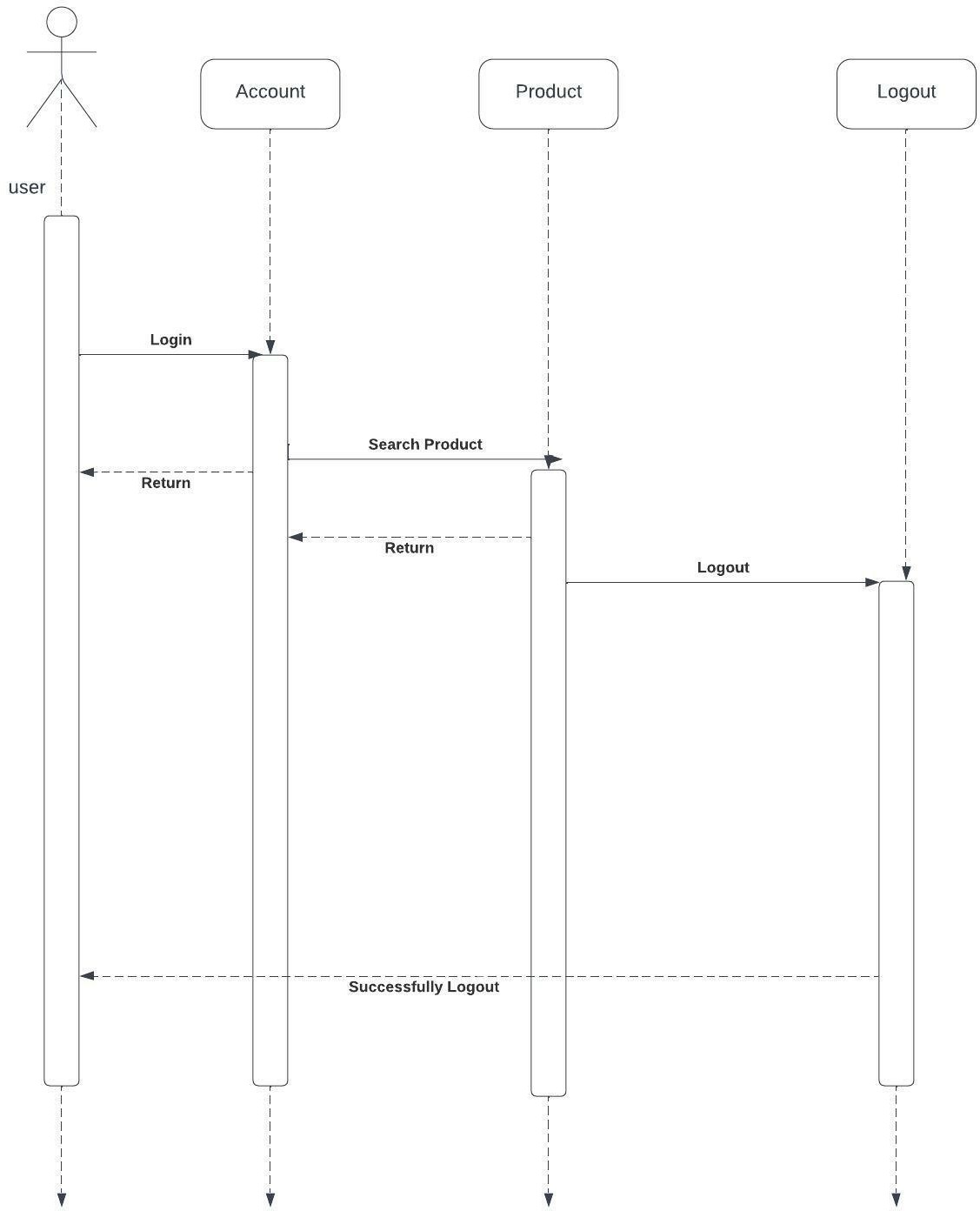
* Synchronous messages
* Asynchronous Messages
* Create message
* Delete Message
* Self-Message
* Reply Message
* Found Message
* Lost Message

1. **Guards –** To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

### Uses of sequence diagrams –

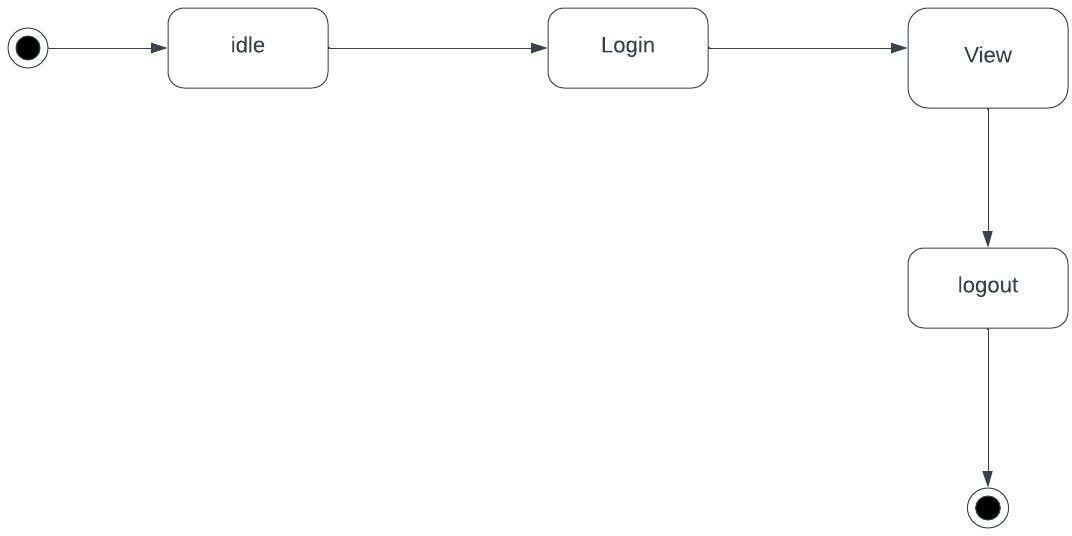
* Used to model and visualize the logic behind a sophisticated function, operation or procedure.
* They are also used to show details of UML use case diagrams.
* Used to understand the detailed functionality of current or future systems.

Visualize how messages and tasks move between objects or components in a system.



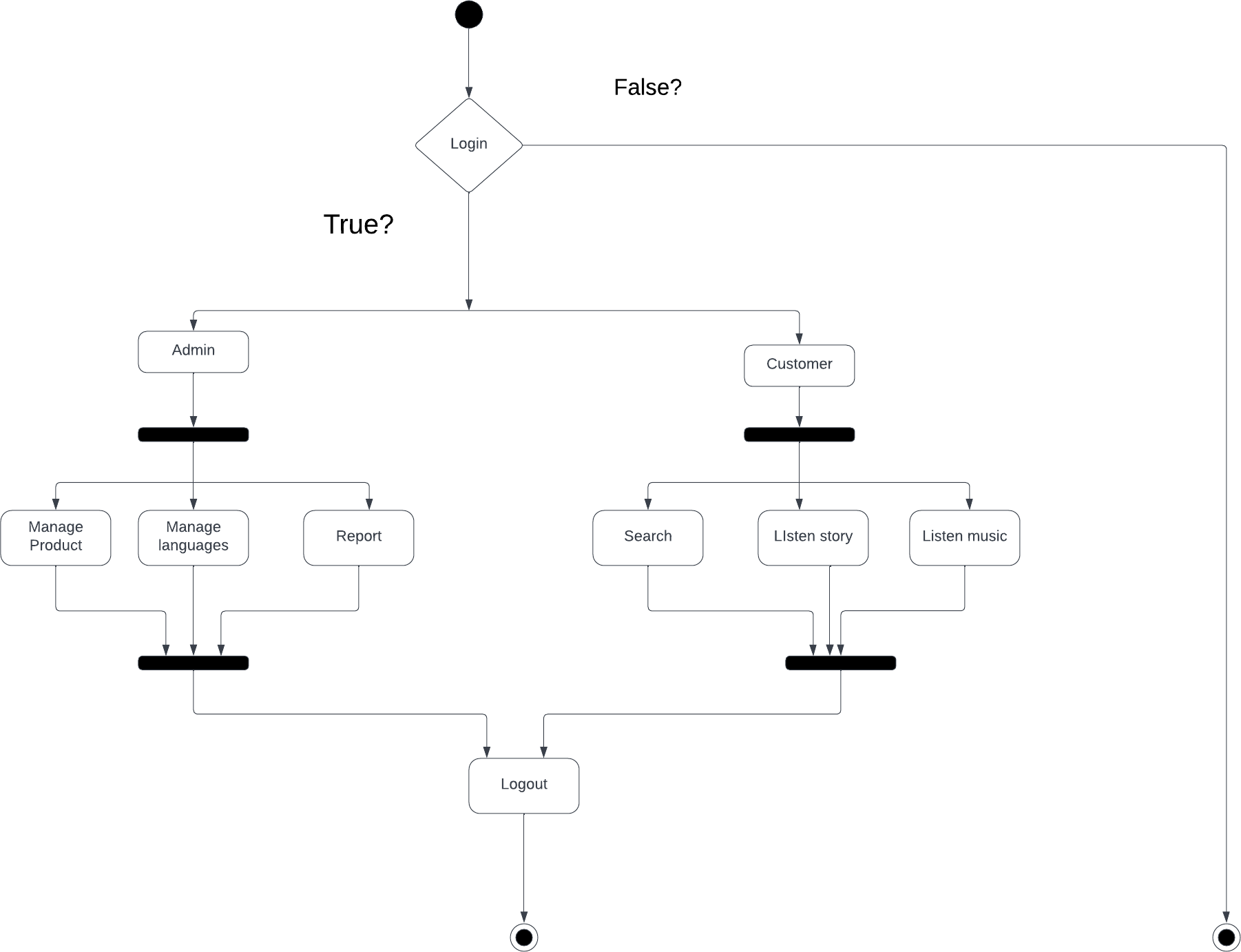
## STATE CHART DIAGRAM

A state chart diagram, also known as a state machine diagram or statechart diagram, is an illustration of the states an object can attain as well as the transitions between those states in the Unified ModelingLanguage (UML).



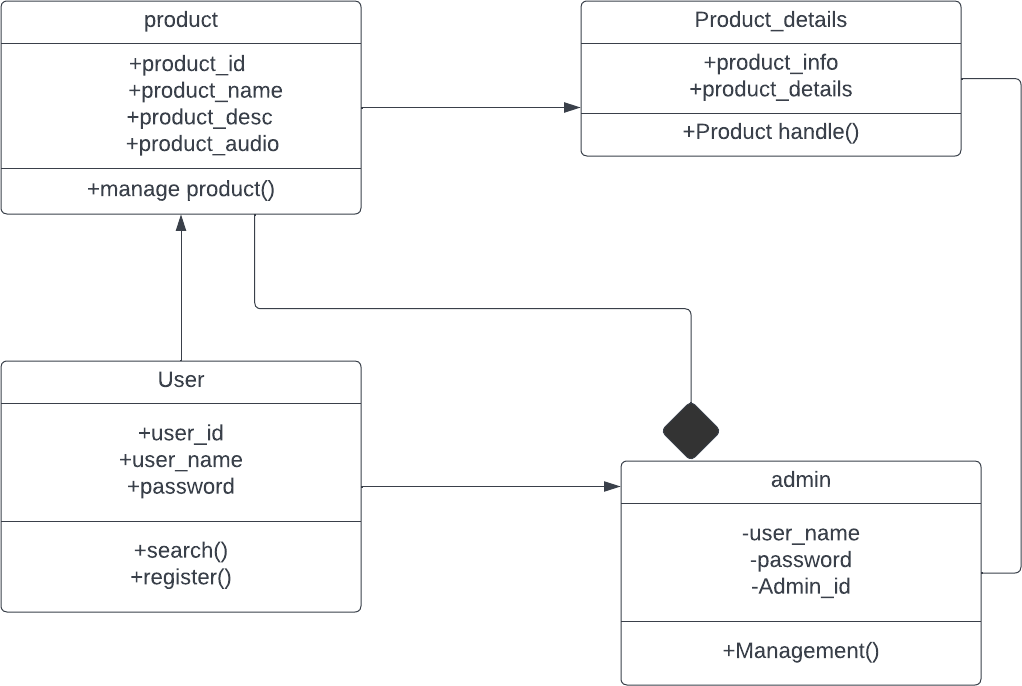
## ACTIVITY DIAGRAM

Another crucial UML diagram for describing the system's dynamic elements is the activity diagram. An activity diagram is essentially a flowchart that shows how one activity leads to another. The action might be referred to as a system operation. One operation leads to the next in the control flow. This flow may be parallel, contemporaneous, or branched. Activity diagrams use many features, such as fork, join, etc., to cope with all types of flow control.An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.



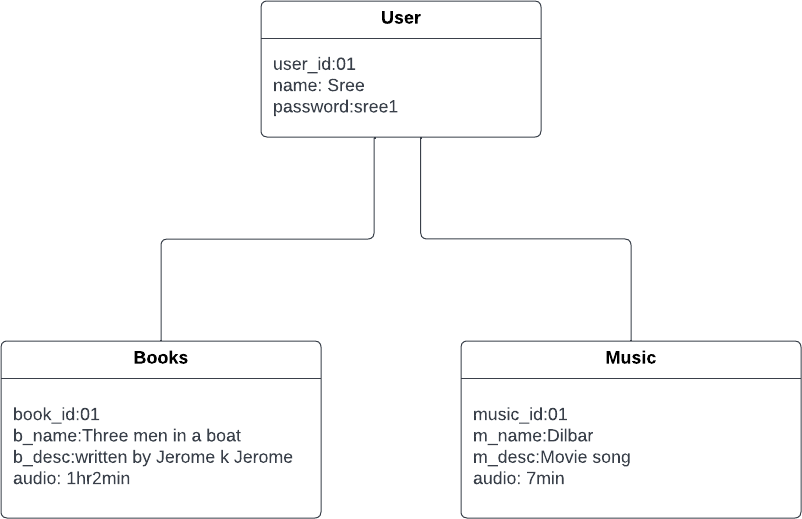
## CLASS DIAGRAM

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling, translating the models into programming code. Class diagrams can also be used for data modeling.



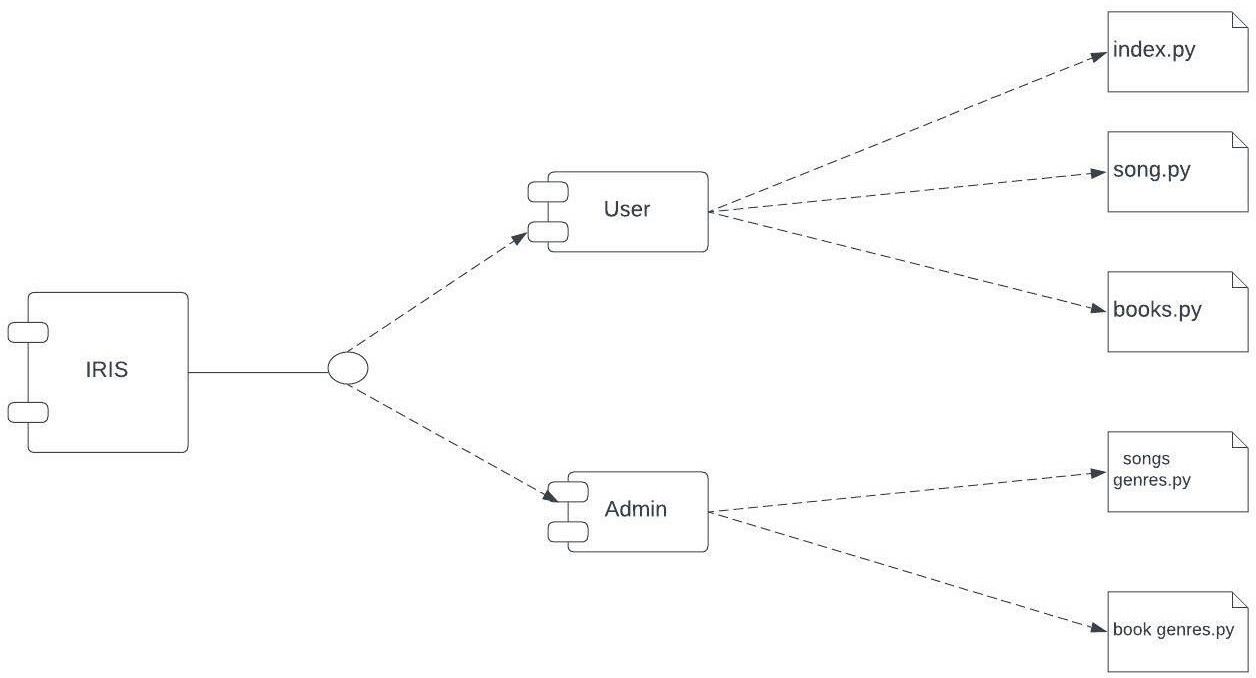
## OBJECT DIAGRAM

Since class diagrams are the source of object diagrams, class diagrams are a prerequisite for object diagrams. An instance of a class diagram is represented by an object diagram. Class and object diagrams both use the same fundamental ideas. The static view of a system is also represented by object diagrams, but this static view represents a momentary snapshot of the system. To represent agroup of items and their connections as an instance, object diagrams are employed.



## COMPONENT DIAGRAM

A component diagram depicts how components are wired together to form larger components or software systems. They are used to illustrate the structure of arbitrarily complex systems.



## DEPLOYMENT DIAGRAM

An execution architecture of a system, containing nodes like hardware or software execution environments, and the middleware linking them, is shown in a deployment diagram, a form of UMLdiagram. Typically, deployment diagrams are used to represent the actual hardware and software of a system. By using it, you can comprehend how the hardware will physically deliver the system. In contrast to other UML diagram types, which primarily depict the logical components of a system, deployment diagrams assist describe the hardware structure of a system.

