



K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Batch: A2

Roll No.: 1911027

Experiment / assignment / tutorial No. 4

Title: To develop UML diagrams for selected project

Aim: To learn and understand the way of creating various UML diagrams for requirement analysis

CO: Analyze the software requirements and Model the defined problem with the help of UML diagram.

Books/ Journals/ Websites referred:

1. Roger Pressman, "Software Engineering", sixth edition, Tata McGraw Hill.
2. System Analysis & Design by Satzinger, Jackson and Burd, Cengage Learning, 2007
3. System Analysis and Design Methods by Jeffery I. Whitten, Lonnie D Bentley, McGraw Hill, 7th edition.
4. System Analysis and Design by Alan Dennis, Barbara H. Wixom, Roberta M. Roth, Wiley India 4th edition
5. http://en.wikipedia.org/wiki/Software_requirements_specification
6. http://en.wikipedia.org/wiki/Use_case

Pre Lab/ Prior Concepts:

In software and systems engineering, a **use case** is a list of steps, typically defining interactions between a role (known in Unified Modeling Language (UML) as an "actor") and a system, to achieve a goal. The actor can be a human or an external system.

In systems engineering, use cases are used at a higher level than within software engineering, often representing missions or stakeholder goals. The detailed requirements may then be captured in Systems Modeling Language (SysML) or as contractual statements.

As an important requirement technique, use cases have been widely used in modern software engineering over the last two decades. Use case driven development is a key



K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

characteristic of process models and frameworks like Unified Process (UP), Rational Unified Process (RUP), Oracle Unified Method (OUM), etc. With its iterative and evolutionary nature, use case is also a good fit for agile development.

A sequence diagram is a graphical view of a scenario that shows object interaction in a time-based sequence of what happens first, what happens next.

Sequence diagrams establish the roles of objects and help provide essential information to determine class responsibilities and interfaces.

In UML, class diagrams are one of six types of structural diagram. Class diagrams are fundamental to the object modelling process and model the static structure of a system. Depending on the complexity of a system, you can use a single class diagram to model an entire system, or you can use several class diagrams to model the components of a system.

Class diagrams are the blueprints of your system or subsystem. You can use class diagrams to model the objects that make up the system, to display the relationships between the objects, and to describe what those objects do and the services that they provide.

In its basic form, an activity diagram is a simple and intuitive illustration of what happens in a workflow, what activities can be done in parallel, and whether there are alternative paths through the workflow. Activity diagrams as defined in the Unified Modeling Language are derived from various techniques to visually illustrate workflows. Activity diagrams are used to visualize the workflow of a business use case. A complete workflow description will have a basic flow, and one or several alternative flows. This workflow has a structure that we can define textually, using informal if, if-then-else, or does-until statements of various kinds. For a simple workflow with a simple structure such textual definitions may be quite sufficient, but in the case of more complex structures, activity diagrams help to clarify and make more apparent what the workflow is. Historically, activity diagramming techniques have mostly been used in the business process modeling domain, but this article will also briefly discuss how you can use it in the system modeling domain.



K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Requirement Modeling:

1. Use Case:

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system. The main purpose of a use case diagram is to portray the dynamic aspect of a system. It accumulates the system's requirement, which includes both internal as well as external influences. It invokes persons, use cases, and several things that invoke the actors and elements accountable for the implementation of use case diagrams. It represents how an entity from the external environment can interact with a part of the system.

Following are the purposes of a use case diagram given below:

1. It gathers the system's needs.
2. It depicts the external view of the system.
3. It recognizes the internal as well as external factors that influence the system.
4. It represents the interaction between the actors.

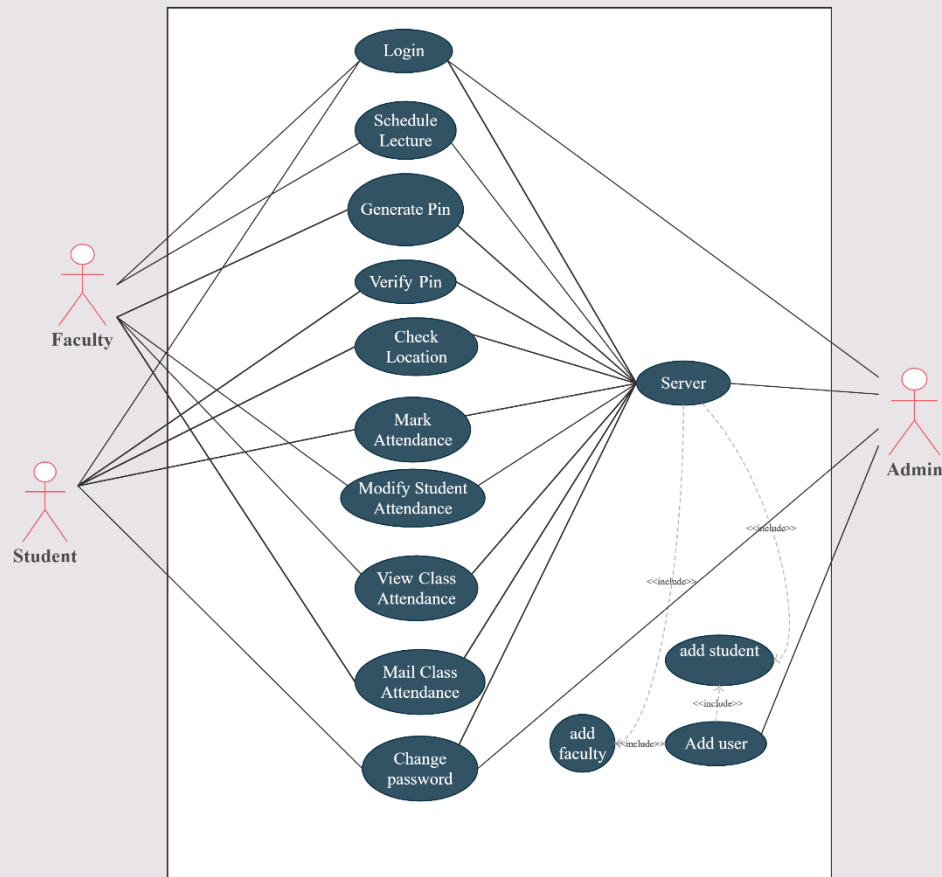
It is essential to analyse the whole system before starting with drawing a use case diagram, and then the system's functionalities are found. And once every single functionality is identified, they are then transformed into the use cases to be used in the use case diagram. After that, we will enlist the actors that will interact with the system. The actors are the person or a thing that invokes the functionality of a system. It may be a system or a private entity, such that it requires an entity to be pertinent to the functionalities of the system to which it is going to interact. Once both the actors and use cases are enlisted, the relation between the actor and use case/ system is inspected. It identifies the no of times an actor communicates with the system. Basically, an actor can interact multiple times with a use case or system at a particular instance of time. Following are some rules that must be followed while drawing a use case diagram:

1. A pertinent and meaningful name should be assigned to the actor or a use case of a system.
2. The communication of an actor with a use case must be defined in an understandable way.
3. Specified notations to be used as and when required.
4. The most significant interactions should be represented among the multiple no of interactions between the use case and actors.



K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)



2. Activity Diagram:

We use Activity Diagrams to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram. An activity diagram focuses on condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram. UML models basically three types of diagrams, namely, structure diagrams, interaction diagrams, and behavior diagrams. 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. We can depict both sequential processing and concurrent processing of activities using an activity diagram. They are used in business and process modelling where



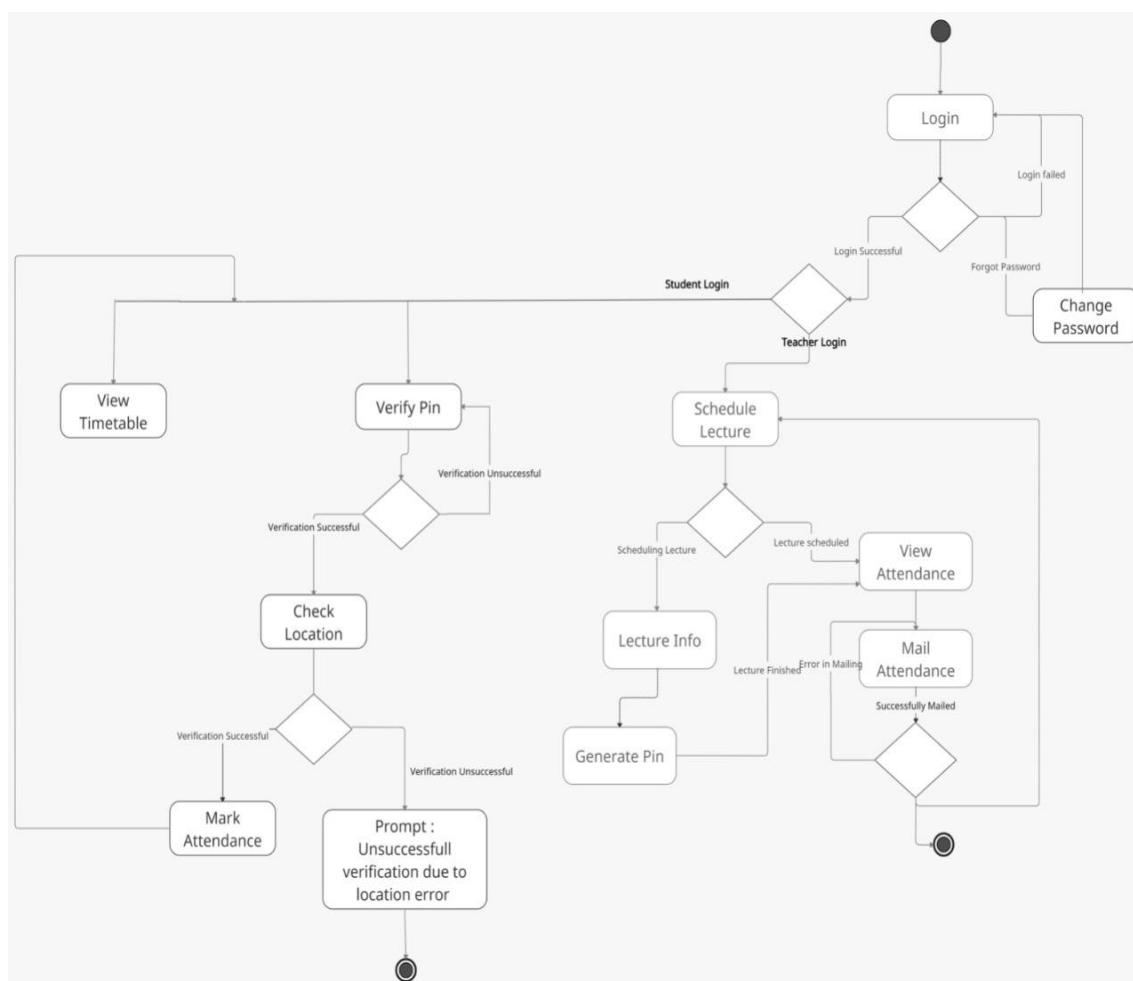
K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

their primary use is to depict the dynamic aspects of a system. The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another. Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part. It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

The purpose of an activity diagram can be described as –

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





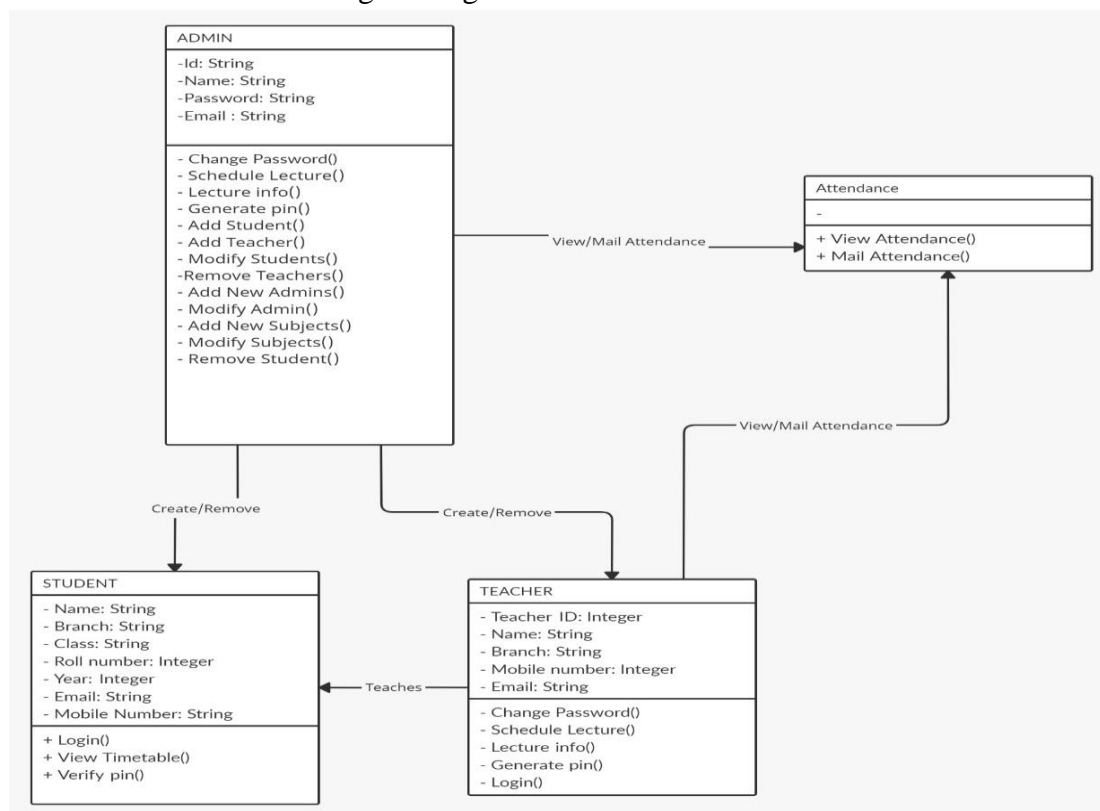
K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

3. 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 modelling of object oriented 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 class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction. UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application, however class diagram is a bit different. It is the most popular UML diagram in the coder community. The purpose of the class diagram can be summarized as –

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





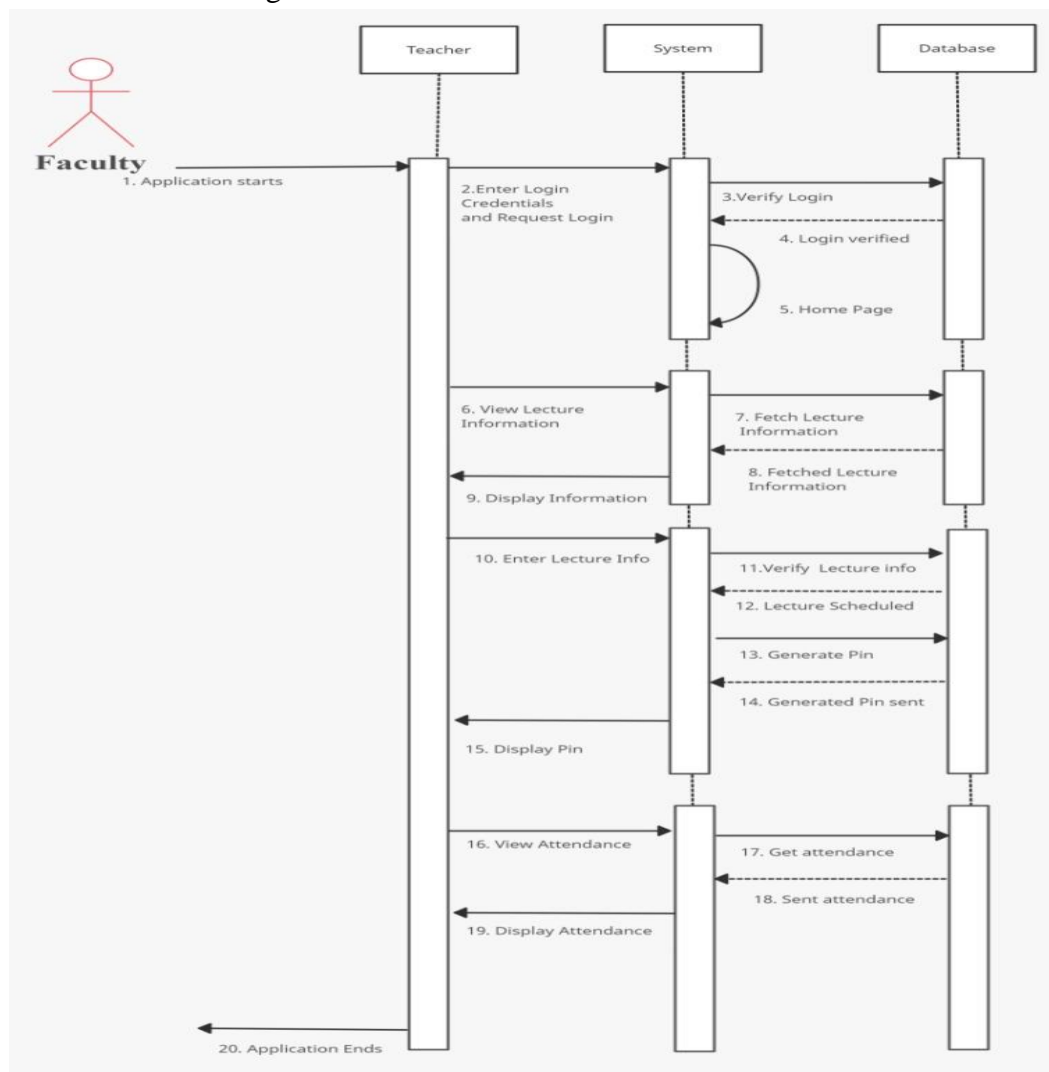
K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

4. Sequence:

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching. Purpose of sequence diagrams:

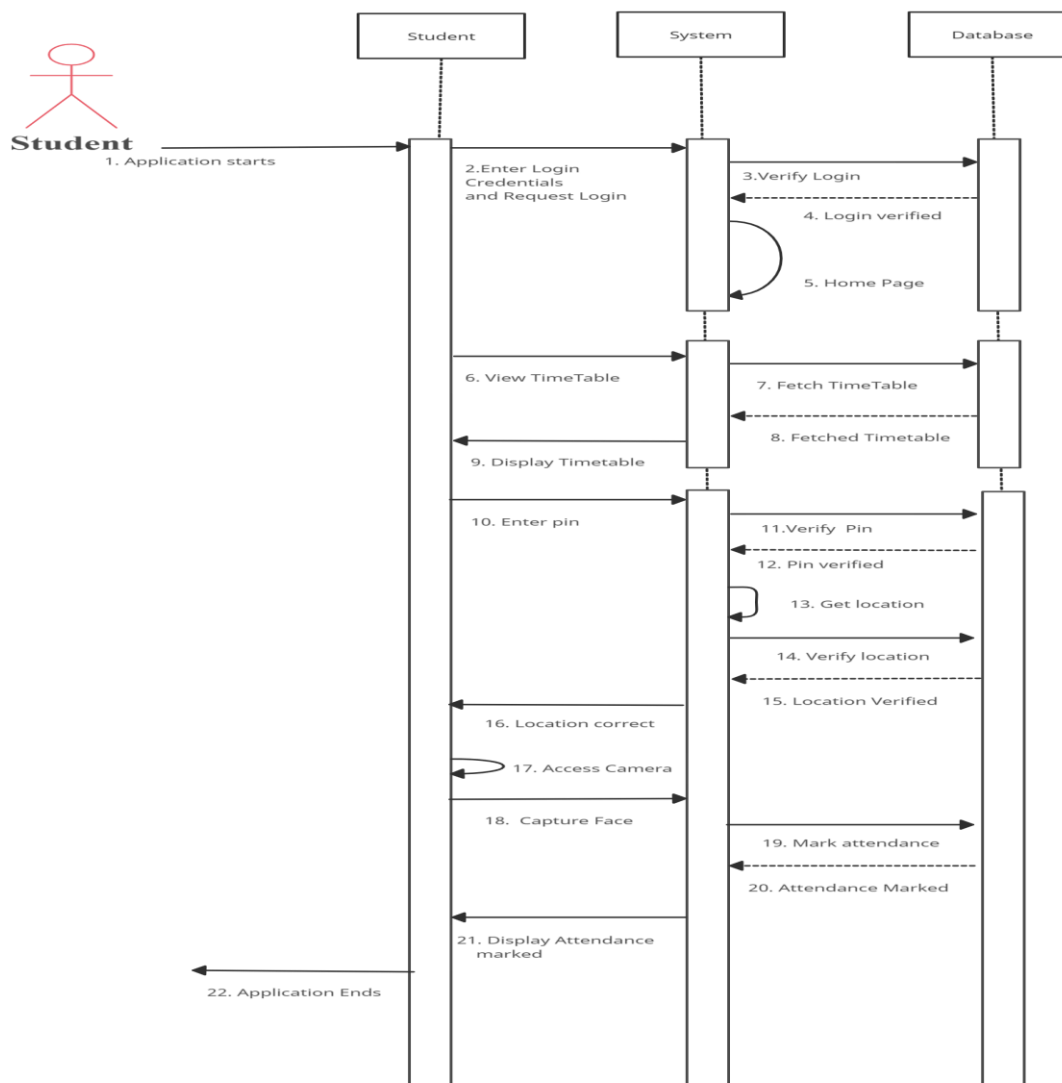
1. To model high-level interaction among active objects within a system.
2. To model interaction among objects inside a collaboration realizing a use case.
3. It either models generic interactions or some certain instances of interaction.





K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)



Conclusion: By performing this experiment understood how to draw UML diagrams for requirements analysis. Understood the concept of use case diagrams, activity diagrams, class diagrams and sequence diagrams and also drawn all four diagrams with respect to our project.

Advantages of Use case diagrams:

- Use case help to capture the functional requirements of a system.
- Use cases are traceable.
- Use cases can serve as the basis for the estimating, scheduling, and validating effort.



K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

- Use case can evolve at each iteration from a method of capturing requirements, to development guidelines to programmers, to a test case and finally into user documentation.
- Use case alternative paths capture additional behavior that can improve system robustness.
- Use cases have proved to be easily understandable by business users, and so have proven an excellent bridge between software developers and end users.

Advantages of Activity diagrams:

- Demonstrate the logic of an algorithm.
- Describe the steps performed in a UML use case.
- Illustrate a business process or workflow between users and the system.
- Simplify and improve any process by clarifying complicated use cases.
- Model software architecture elements, such as method, function, and operation.

Advantages of Class diagrams:

- Helps in better and accurate illustration of the data models, irrespective of the level of complexity involved with the classes and the data.
- It helps with easier and clear understanding of the overview and the schematics involved with the overall system or the process.
- Allows the person to go forth in preparing a visual representation of the overall needs and requirements involved in a system that further helps to disseminate any kind of information involved with the business.
- Helps in rendering the creation of better detailing with the charts for better results and outcomes in the understanding of the prospects involved in the system.
- Allows the individuals comes to a conclusion of the kind of implantation procedures required in the system for better outcomes.

Advantages of Sequence diagrams:

- Represent the details of a UML use case.
- Model the logic of a sophisticated procedure, function, or operation.
- See how objects and components interact with each other to complete a process.
- Plan and understand the detailed functionality of an existing or future scenario.



K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Post Lab Descriptive Questions:

1. Where do use cases fit in the software development life cycle?

ANS)

Ans) 1) Use case is a methodology used in system analysis to identify, clarify and organize system requirements.

2) The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal.

3) The method creates a document that describes all the steps taken by a user to complete an activity.

4) Use cases are typically written by business analysts and can be employed during several stages of software development, such as planning system requirements, validating design, testing software and creating an outline for online help and user manuals.

5) Use case document can help the development team identify and understand where errors may occur during a transaction so they can resolve them.

6) In the project management discipline, use cases are used as a basis for planning and tracking the progress of the development project.

7) This is particularly true for iterative development where use cases are often the primary planning mechanism.

8) Every use case contains 3 essential elements:-

a) The actor:- This can be a single person or a group of people interacting with the process.



K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

b) The goal:- The final successful outcome that completes the process.

c) The system:- The process and steps taken to reach the end goal, including the necessary ~~information~~ functional requirements and their anticipated behaviours.

2. Compare sequence diagram with collaboration diagram. Explain pros and cons of each.

ANS)

Ans)	sequence diagrams	collaboration diagrams
1)	The sequence diagram represents the UML which is used to visualize the sequence of calls in a system that is used to perform a specific functionality.	1) The collaboration diagram also comes under the UML representation which is used to visualize the organization of the objects and their interaction.
2)	Used to represent the sequence of messages that are following from one object to another.	2) - used to represent the structural organization of the system and the messages that are sent and received.
3)	It is used when time sequence is main focus.	3) It is used when object organization is main focus.
4)	These diagrams are better suited for analysis activities.	4) These diagrams are better suited for depicting simpler interactions of the smaller number of objects.



K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

* Advantages of sequence diagrams :-

- 1) Simple notation
- 2) Clearly shows sequence or time ordering of messages.
- 3) Large set of detailed notation options.

* Disadvantages of sequence diagrams :-

- 1) Forced to extend to the right when adding new objects ~~consumes horizontal space~~
- 2) Consumes horizontal space.

* Advantages of collaboration diagrams :-

- 1) Space - economical :- flexible to add new objects in two dimensions.
- 2) Better to illustrate complex branching, iteration and concurrent behaviour.

* Disadvantages of collaboration diagrams :-

- 1) Complex notation.
- 2) ~~It~~ difficult to see sequence of messages.



K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

3. List different notations used in Class diagram with example.

ANS)

Ans 1) The UML Class diagram is a graphical notation used to construct and visualize object oriented systems.

2) A class diagram in the UML is a type of static structure diagram that describes the structure of a system by showing the systems:- classes, their attributes, operations (or methods) and the relationships among objects.

3) class:- a) A class is a blueprint for an object.

b) A class describes what an object will be, but it isn't the object itself.

c) In fact classes describe the type of objects, while objects are usable instances of classes.

d) objects have state and behaviour.

e) A class represents a concept which encapsulate state and behaviour.

name	→ class name.
attributes	→ properties.
methods	→ operations.

Fig. A animal class will have attributes as type, breed, etc and operations like can eat, can fly, etc.

Animal
type.
breed.
can bark()
can fly()



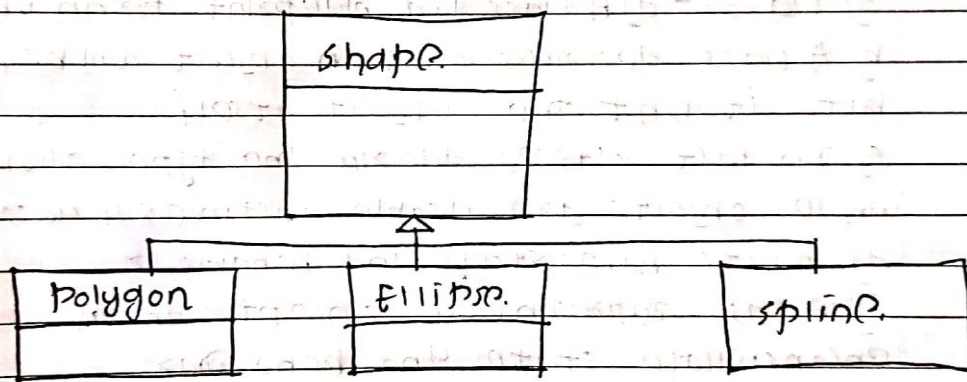
K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

3) Inheritance (or Generalization) :- a) A generalization is a taxonomic relationship between a more general classifier and a more specific classifier. b) Each instance of a the specific classifier is also an indirect instance of the general classifier.

c) It represents a is-a relationship.

d) Example :- polygon, ellipse and spline are generalised to shape.



e) Association :- a) Associations are relationships between classes in a UML class diagram.

b) A structural link between 2 or more classes.

c) Cardinality :- a) It is expressed in terms of

a) one-to-one b) one-to-many c) many-to-many.

d) Composition :- a) A special type of aggregation where parts are destroyed when the whole is destroyed.

e) base class cannot exist alone.