**Batch: A1**

**Roll No.: 1911004, 1911005,1911012**

**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

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**CO2: Analyse 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 l. 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](http://en.wikipedia.org/wiki/Software_engineering) and [systems engineering](http://en.wikipedia.org/wiki/Systems_engineering), a **use case** is a list of steps, typically defining interactions between a role (known in [Unified Modeling Language](http://en.wikipedia.org/wiki/Unified_Modeling_Language) (UML) as an "[actor](http://en.wikipedia.org/wiki/Actor_(UML))") 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](http://en.wikipedia.org/wiki/Project_stakeholder) goals. The detailed requirements may then be captured in [Systems Modeling Language](http://en.wikipedia.org/wiki/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 characteristic of process models and frameworks like [Unified Process](http://en.wikipedia.org/wiki/Unified_Process) (UP), [Rational Unified Process](http://en.wikipedia.org/wiki/Rational_Unified_Process) (RUP), [Oracle Unified Method](http://en.wikipedia.org/wiki/Oracle_Unified_Method) (OUM), etc. With its iterative and evolutionary nature, use case is also a good fit for [agile development](http://en.wikipedia.org/wiki/Agile_software_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.

**Requirement Modeling:**

*Significance of every diagram is to be written*

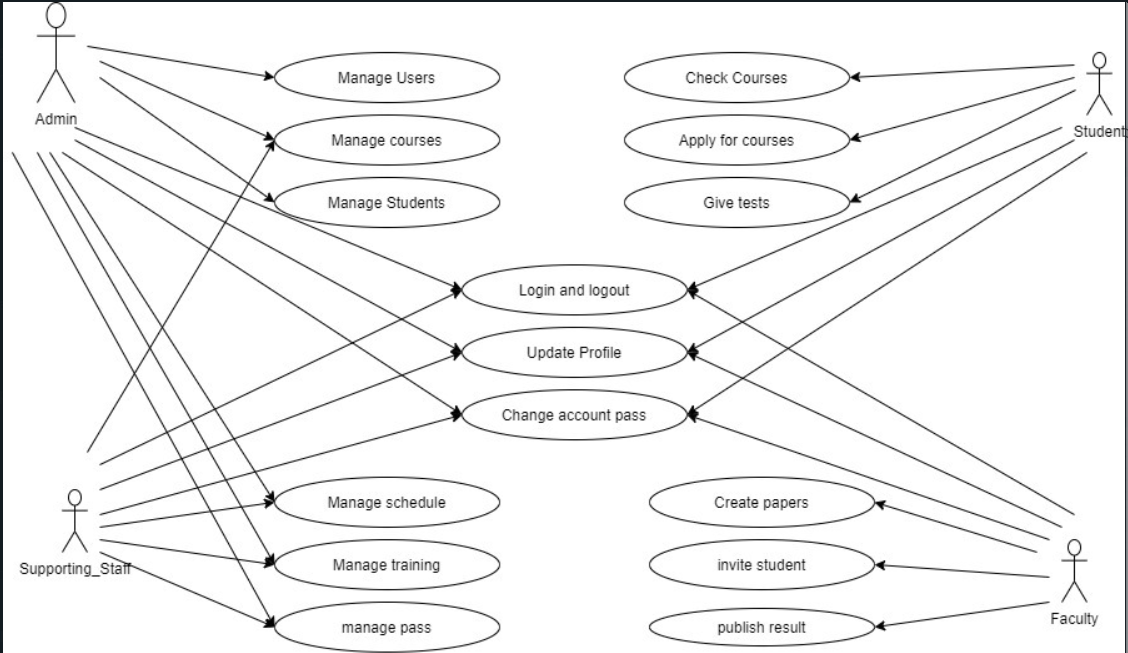
1. **Use Case**

A UML use case diagram is the primary form of system/software requirements for a new software program under development. Use cases specify the expected behavior, and not the exact method of making it happen. A key concept of use case modeling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior. A use case diagram is usually simple. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified. When the initial task is complete, use case diagrams are modelled to present the outside view.

* It only summarizes some of the relationships between use cases, actors, and systems.
* It does not show the order in which steps are performed to achieve the goals of each use case.

In brief, the purposes of use case diagrams can be said to be as follows −

* Used to gather the requirements of a system.
* Used to get an outside view of a system.
* Identify the external and internal factors influencing the system.
* Show the interaction among the requirements of the actors.

****

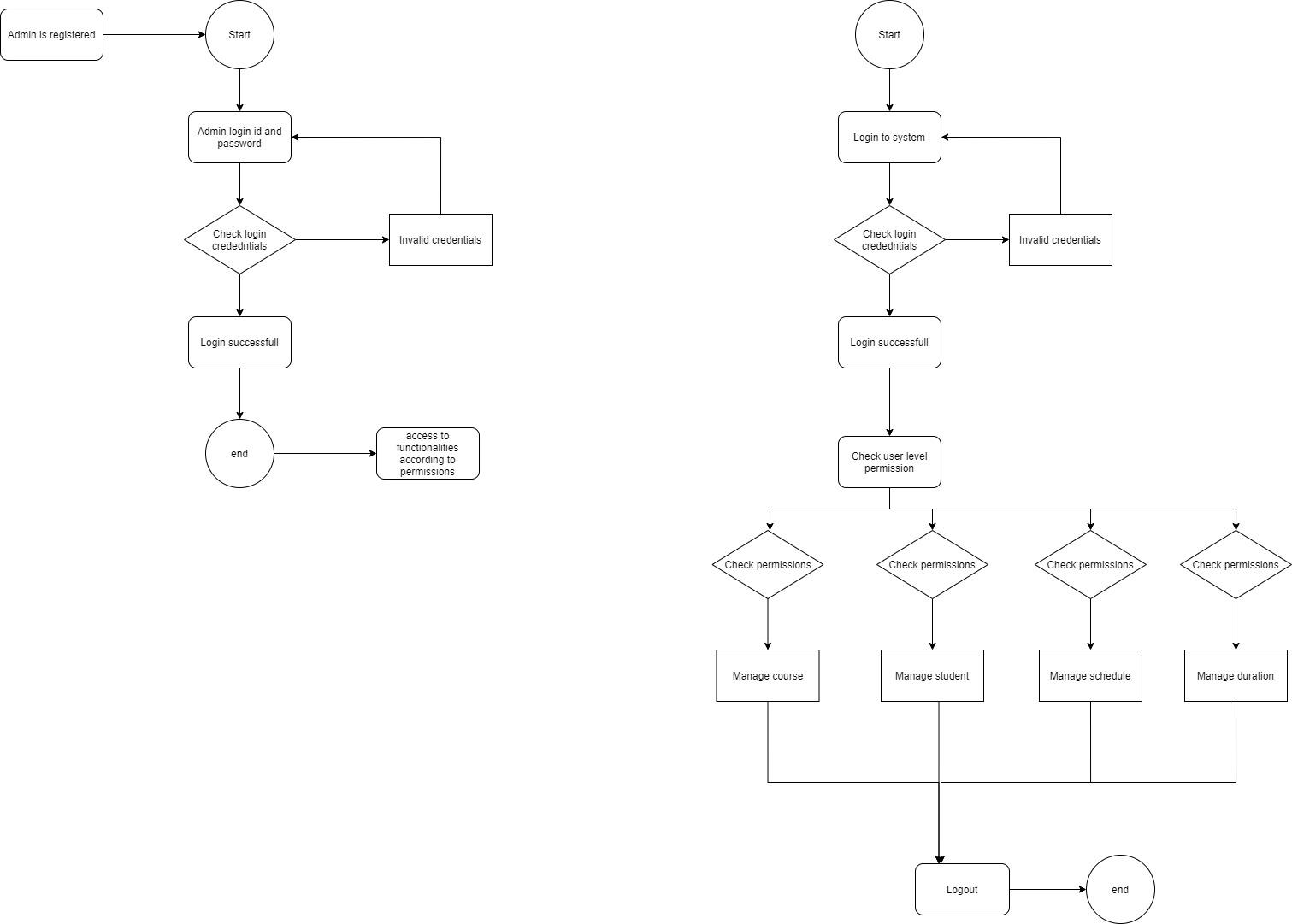
1. **Activity Diagram**

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 their primary use is to depict the dynamic aspects of a system. 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 significance of an activity diagram can be described as −

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

****

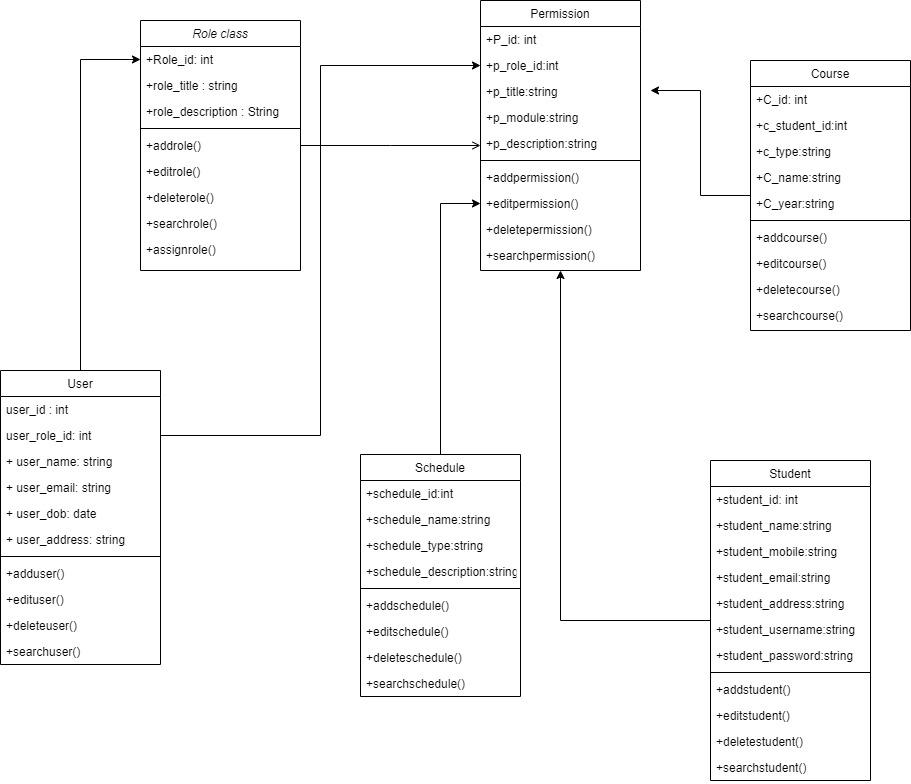
1. **Class Diagram**

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. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

The purpose of class diagrams 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.

The significance of the class diagram can be summarized as −

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



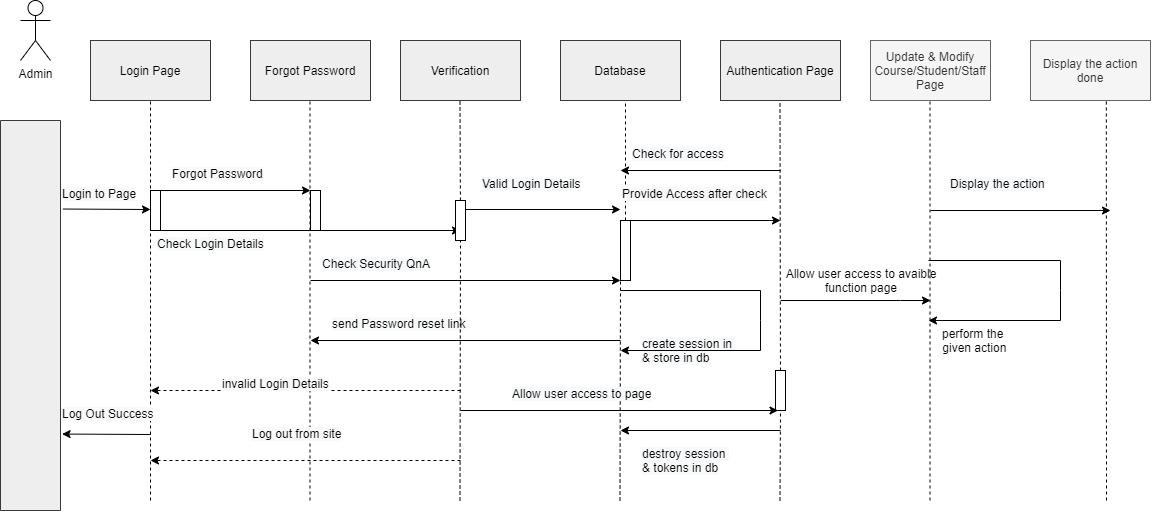
1. **Sequence**

Sequence diagrams, commonly used by developers, model the interactions between objects in a single use case. They illustrate how the different parts of a system interact with each other to carry out a function, and the order in which the interactions occur when a particular use case is executed.

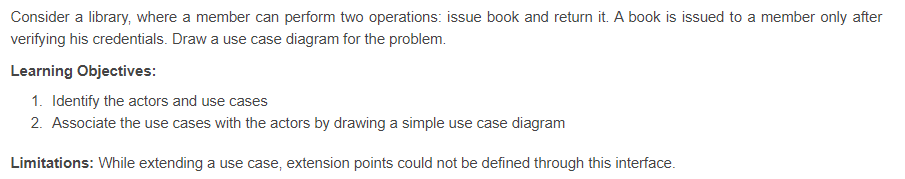
Sequence diagrams describe how and in what order the objects in a system function. document and understand requirements for new and existing systems.

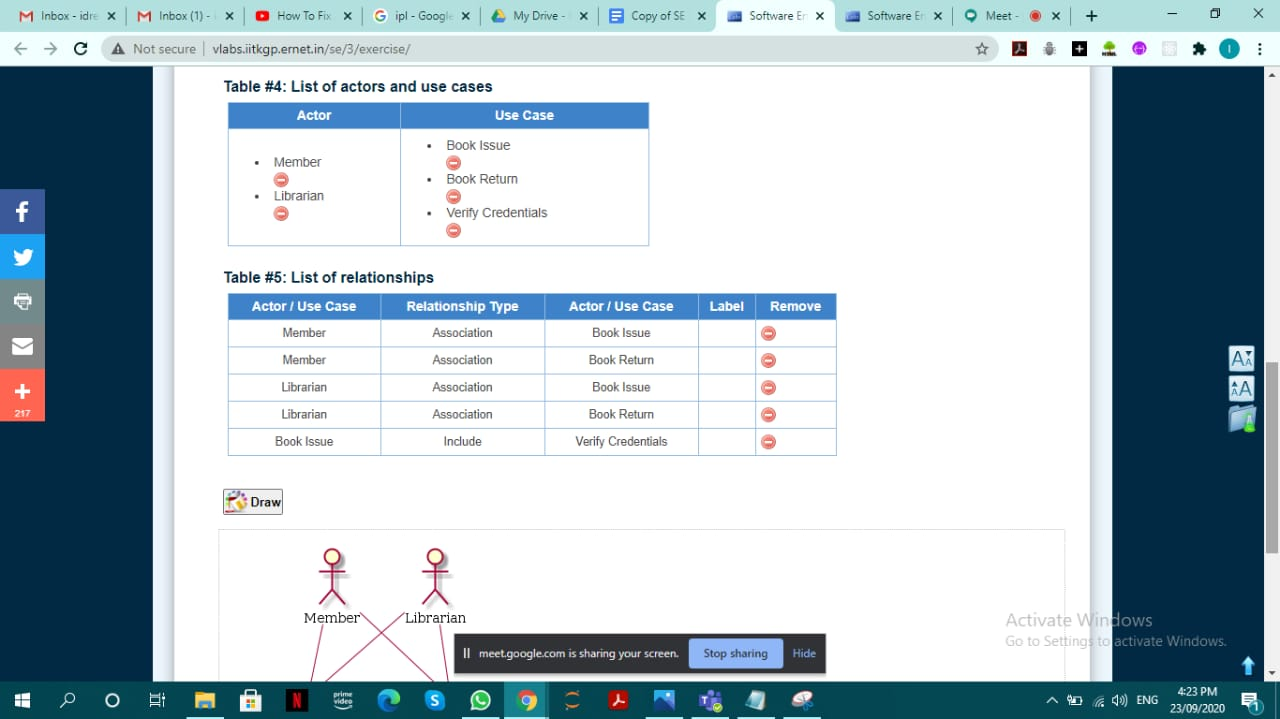
Significance of the sequence diagram can be summarized as follows:

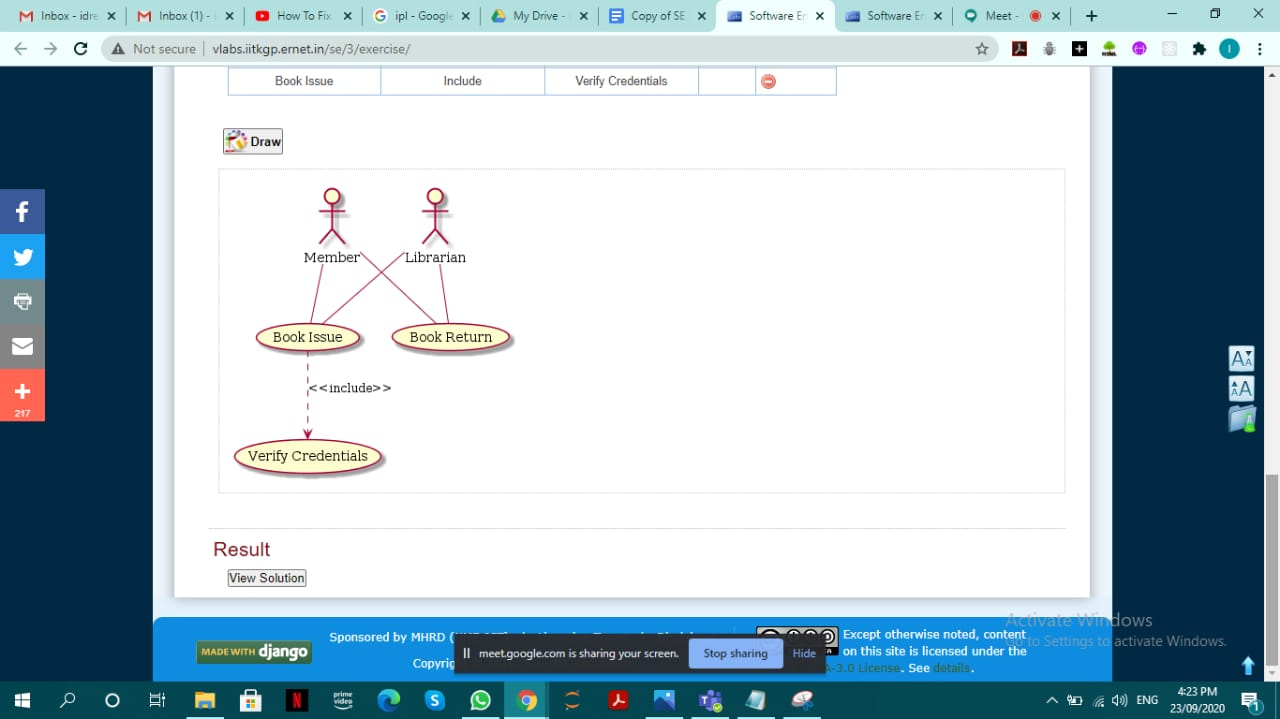
* Used to model and visualise the logic behind a sophisticated function, operation or procedure.
* They are also used to show details of UML use case diagrams.
* Used to understand the detailed functionality of current or future systems.
* Visualise how messages and tasks move between objects or components in a system.

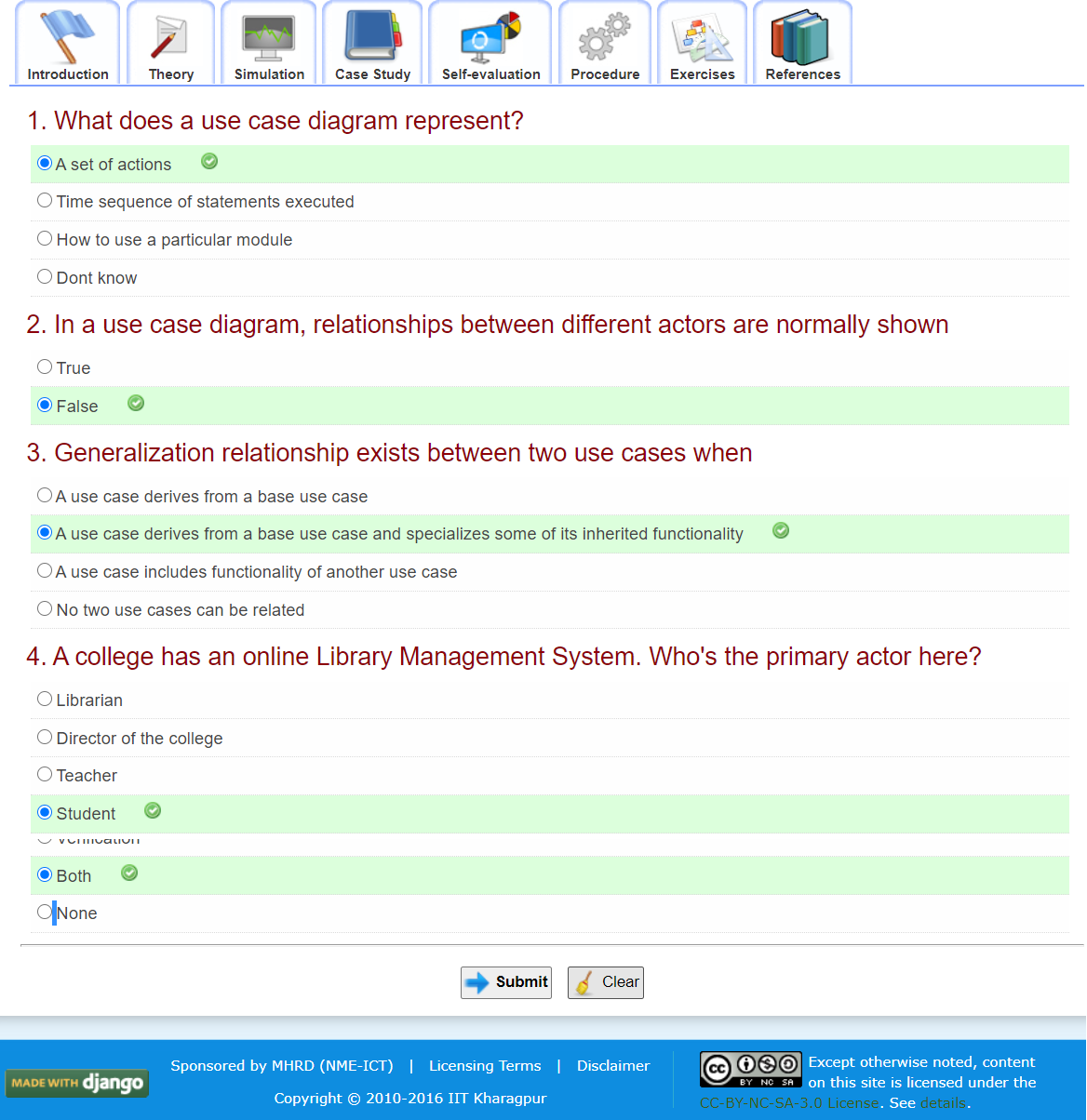
****

**Virtual Lab Link:** [**http://vlabs.iitkgp.ernet.in/se/3/exercise/**](http://vlabs.iitkgp.ernet.in/se/3/exercise/)

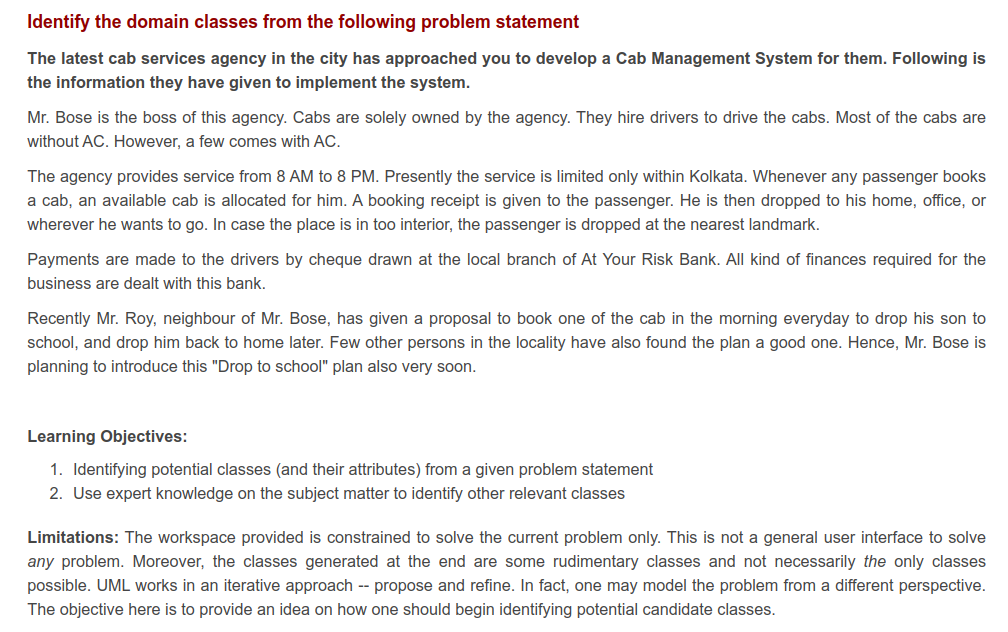
****

****

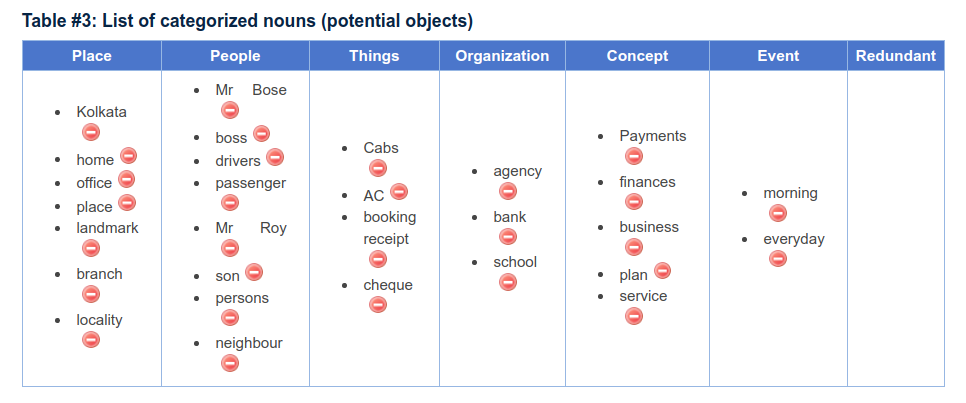
****

****

**Virtual Lab Link:** [**http://vlabs.iitkgp.ernet.in/se/5/exercise/**](http://vlabs.iitkgp.ernet.in/se/5/exercise/)





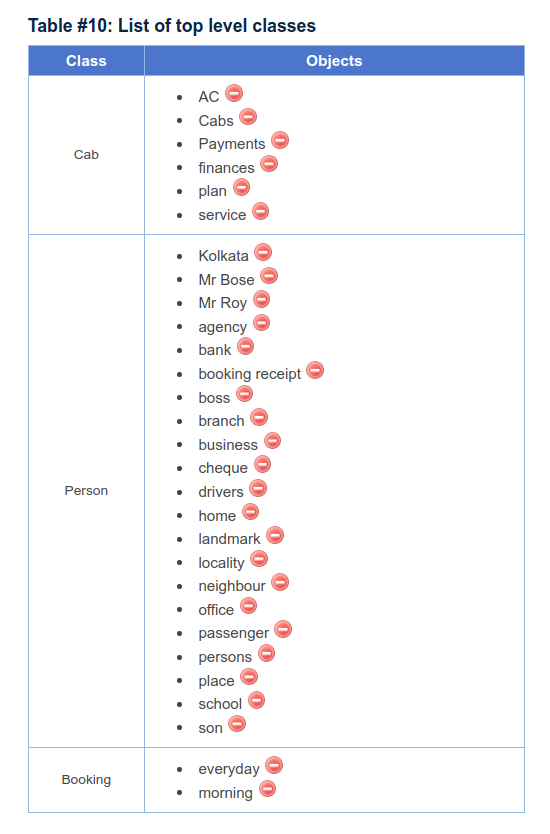






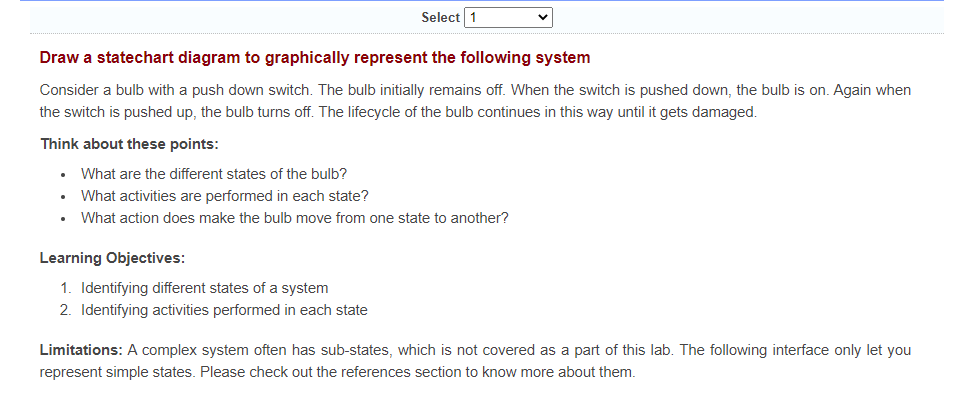


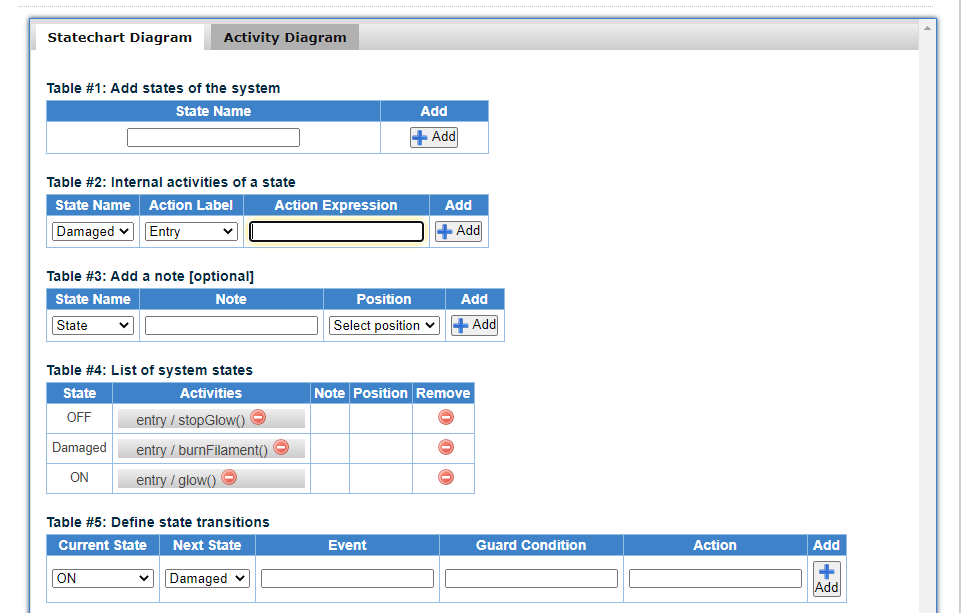


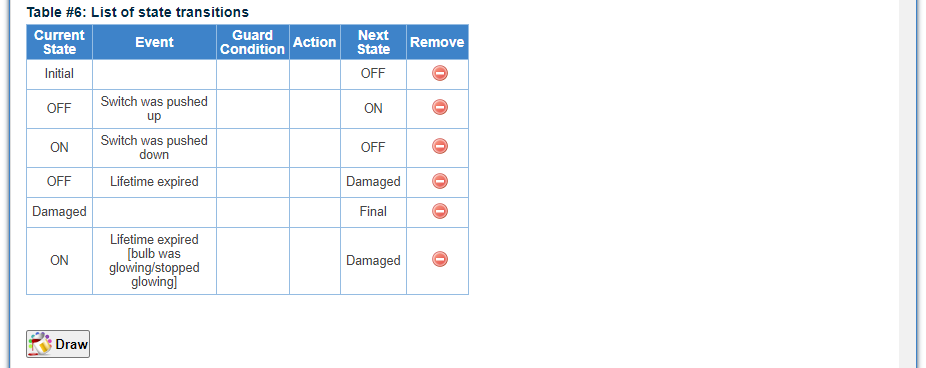


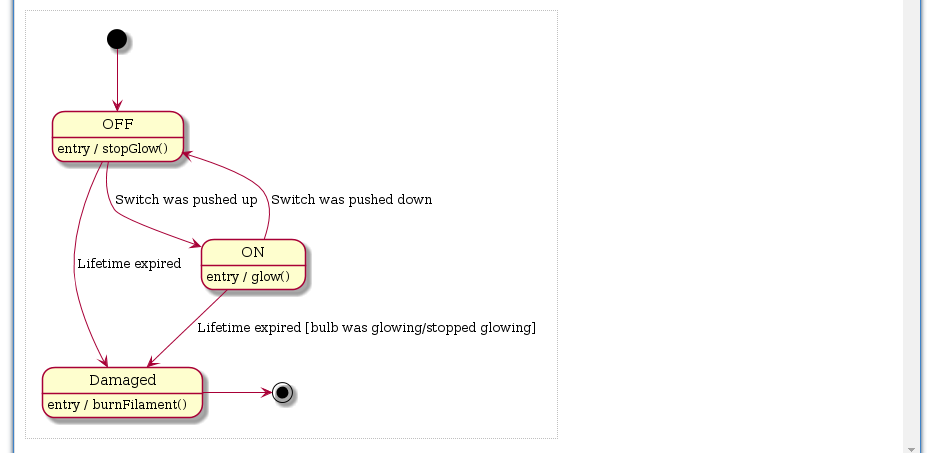


**Virtual Lab Link:** [**http://vlabs.iitkgp.ernet.in/se/6/**](http://vlabs.iitkgp.ernet.in/se/6/)



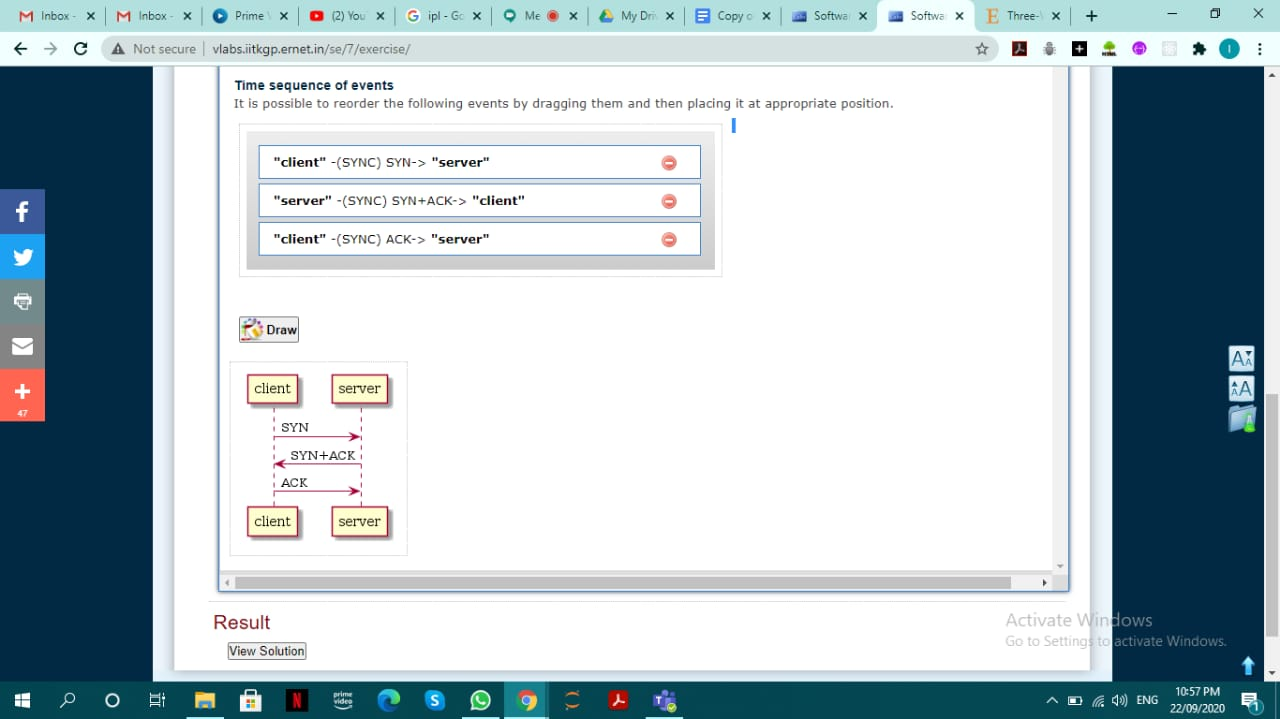


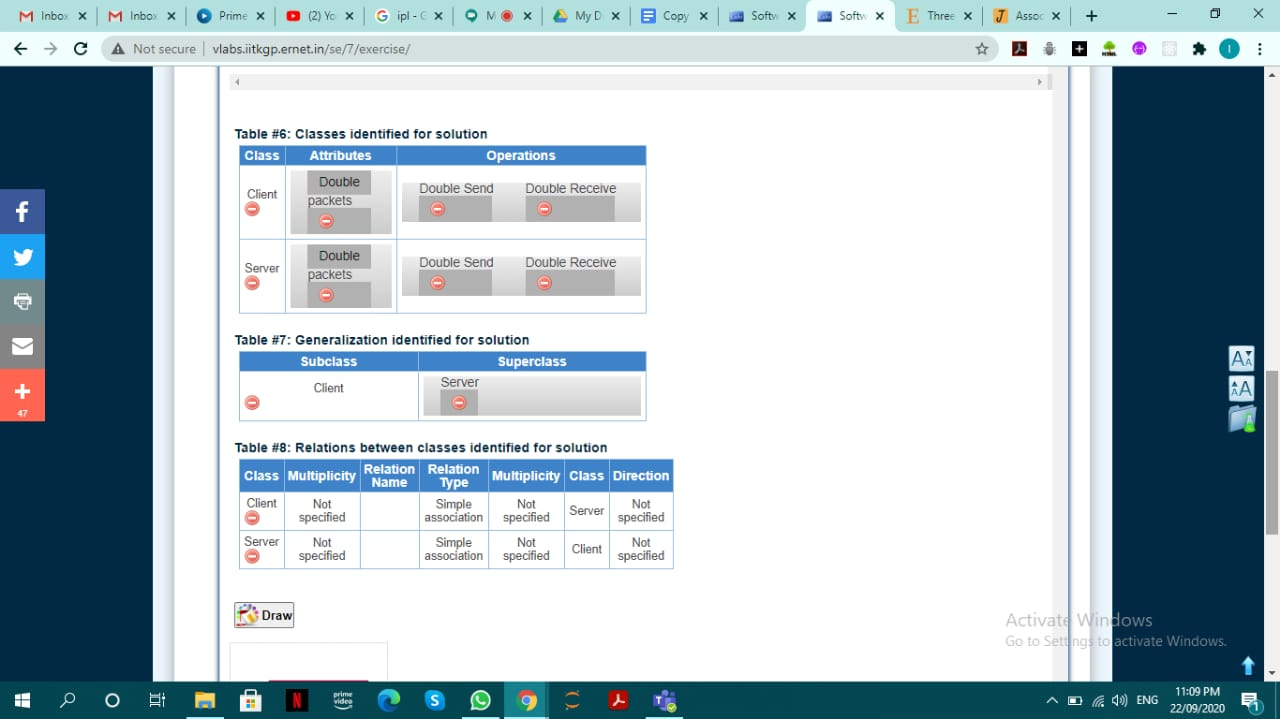


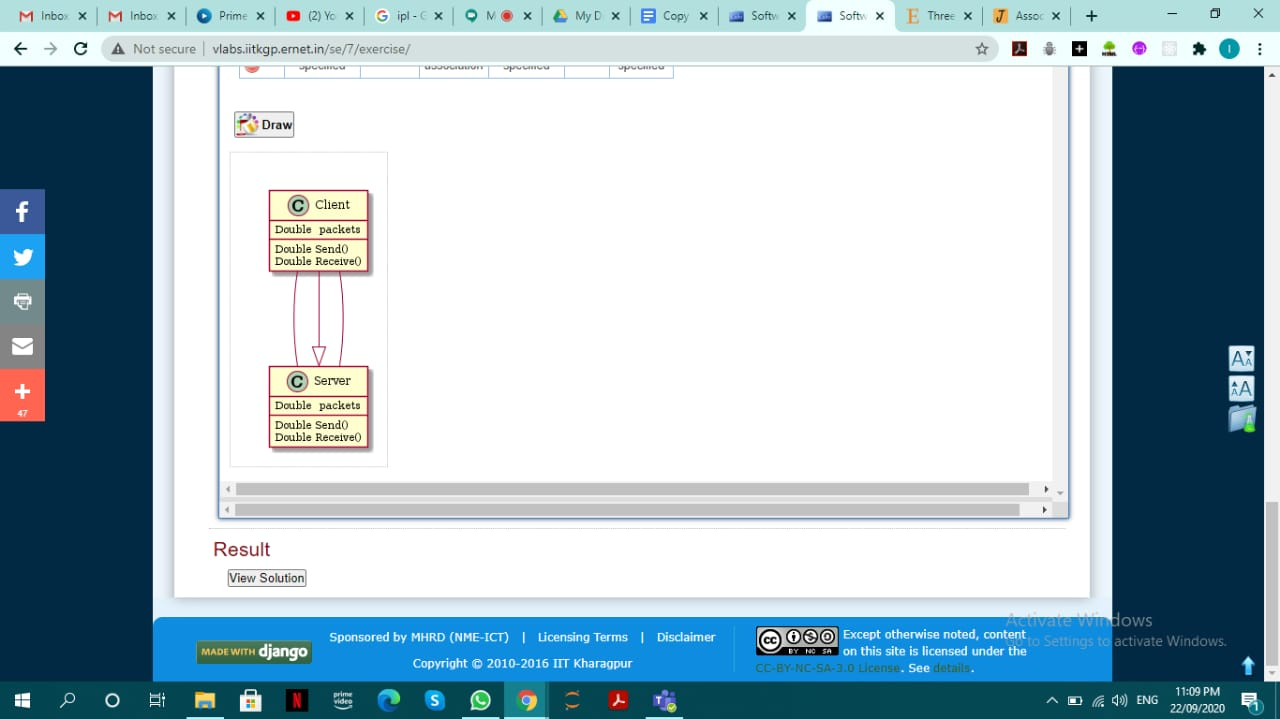




**Virtual Lab Link:**  [**http://vlabs.iitkgp.ernet.in/se/7/**](http://vlabs.iitkgp.ernet.in/se/7/)







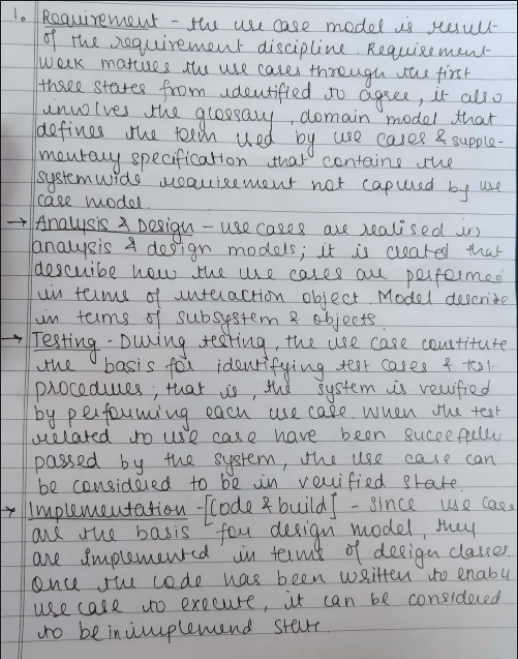


**Conclusion:**

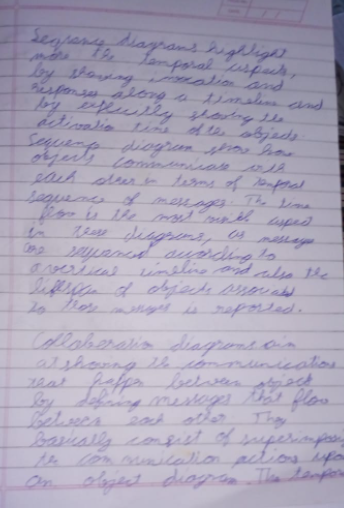
Advantages of UML diagrams such as Use Case, Activity, Class and Sequence have been understood.

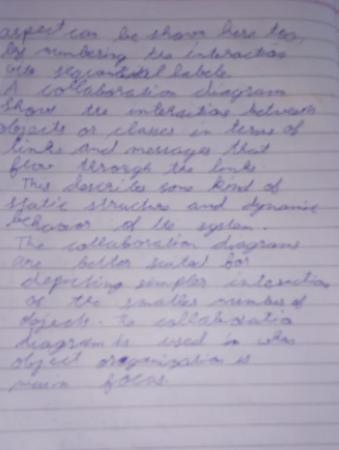
**Post Lab Descriptive Questions:**

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



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





1. List different notations used in Class diagram with example



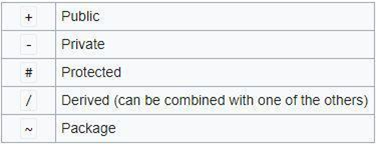
Association:



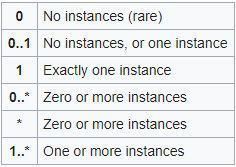
Aggregation:



Visibility:



Instances:



Class:

