Name of Student: Pushkar Sane		
Roll Number: 45		**Tutorial Number: 5**
Title of Tutorial: UML Diagrams (Activity Diagram)		
DOP: 25-09-2023		**DOS: 25-09-2023**
CO Mapped: **CO1, CO2, CO3**	**PO Mapped:** **PO1, PO3, PO6**	**Signature:**

Tutorial No. 5

Aim: UML Diagrams (Activity Diagram)

Description:

- **Introduction to UML Activity Diagrams**

  What is UML?

  Unified Modeling Language (UML) is a standardized modeling language used in software engineering to visualize, specify, construct, and document the artifacts of software systems. UML provides a variety of diagram types for different aspects of system modeling, and one of the most important is the Activity Diagram.

- **Purpose of Activity Diagrams**

  Activity Diagrams are used to depict the dynamic aspects of a system, focusing on the flow of activities or actions within the system. They provide a high level view of how different activities interact and the sequence in which they occur. Activity Diagrams are particularly useful for modeling workflows, business processes, and the logic within software systems.

- **Benefits of Activity Diagrams**

  Activity Diagrams offer several advantages in system modeling:

  1. **Clarity:** They provide a clear, visual representation of complex processes and activities.
  2. **Communication**: They facilitate communication between stakeholders by presenting a common visual language.
  3. **Analysis:** They assist in analyzing and understanding the workflow and logic of a system.
  4. **Design:** They serve as a foundation for designing the structure and behavior of software systems.
  5. **Documentation:** They provide a visual reference for system documentation.

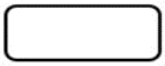
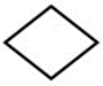
- **Elements of an Activity Diagram**

  A typical Activity Diagram consists of several key elements:

  1. **Activities:** Activities represent high level tasks or processes within the system. They are typically depicted as rounded rectangles and describe a unit of work or action.
  2. **Actions:** Actions represent specific steps or operations within an activity. They are usually depicted as rectangles and represent the lowest level of granularity in the diagram.
  3. **Control Nodes:** Control nodes are used to direct the flow of activities. Key control nodes include initial nodes (starting points), final nodes (ending points), decision nodes (branching points), and merge nodes (joining points).
  4. **Edges and Object Flows:** Edges connect activities, actions, and control nodes to define the flow of control or data between them. Object flows represent the movement of data or objects between elements.

- **Activity Diagram Notations and Symbols**

  Understanding the notations used in Activity Diagrams is essential for creating and interpreting them effectively.

Name	Symbol	Description
Initial State		This shows the starting point or first activity of the flow.
Final State		The end of the Activity diagram, also called as a final activity.
Action		It represents the activity to be performed.
Decision		A logic where a decision is to be made is depicted by a diamond.

Transition		A transition link represents control flow between nodes.

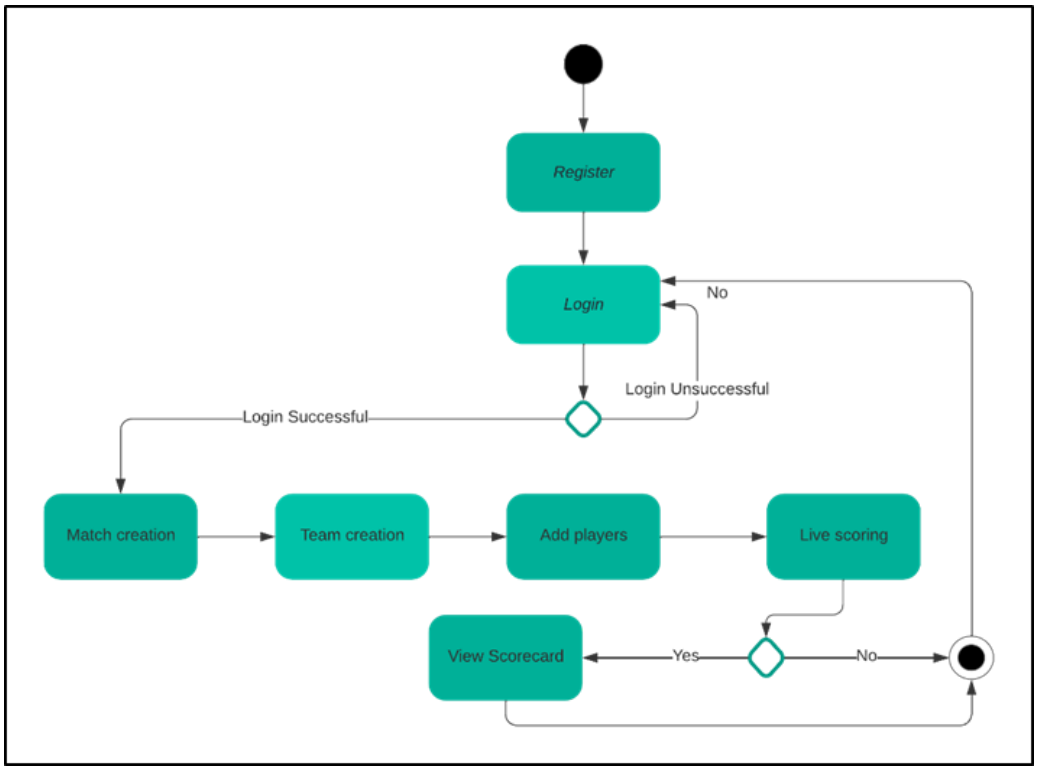
- **Creating Activity Diagrams**

  Creating an Activity Diagram involves several steps:

  1. **Identify Activities:** Identify and define the high level activities or processes that need to be modelled within the system.
  2. **Define Actions:** For each activity, break down the process into specific actions or steps.
  3. **Establish Control Flow:** Use control nodes such as initial, final, decision, merge, fork, and join nodes to define the flow of activities and actions.
  4. **Connect Elements:** Use edges to connect activities, actions, and control nodes, indicating the sequence of execution or data flow.

- **Problem Statement**

  To create a live cricket scoring app. Here, various types of functions are provided such as ball-by-ball scoring, professional scorecard etc.



- **Use Cases and Applications**

  Activity Diagrams find applications in various domains:

  1. **Software Development:** In software engineering, Activity Diagrams help model and visualize the flow of actions and processes within software systems, aiding in design and development.

  2. **Business Process Modelling:** Business analysts use Activity Diagrams to represent and analyze business processes, helping organizations streamline operations and improve efficiency.

  3. **System Analysis and Design:** Systems analysts and designers use Activity Diagrams to understand and model complex system interactions, making them an essential tool in system development.

Conclusion:

In conclusion, Activity Diagrams in UML are a valuable tool for modelling and understanding dynamic processes and workflows. They provide clarity, facilitate communication, aid in analysis, support system design, and serve as documentation references. Whether used in software development, business process improvement, or system analysis, Activity Diagrams play a vital role in creating efficient and well structured systems. Understanding and effectively employing this UML diagram can greatly enhance the quality and success of system development projects.