Assignment 3
Prepare activity model for system to connect users with similar interests

OBJECTIVE:

- To Identify activities involved within the College Event Planning.
- Design the Activity Diagram.

THEORY:

Activity Diagram

Activity Diagrams illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. An activity diagram provides a view of the behavior of a system by describing the sequence of actions in a process. Activity diagrams are similar to flowcharts because they show the flow between the actions in an activity; however, activity diagrams can also show parallel or concurrent flows and alternate flows.In activity diagrams, you use activity nodes and activity edges to model the flow of control and data between actions.

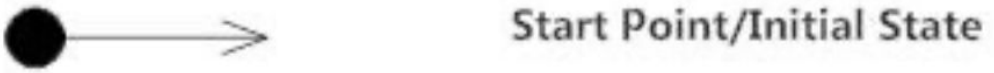
Activity diagrams are helpful in the following phases of a project:

- Before starting a project, you can create activity diagrams to model the most important workflows.
- During the requirements phase, you can create activity diagrams to illustrate the flow of events that the use cases describe.
- During the analysis and design phases, you can use activity diagrams to help define the behavior of operations.

The various components used in the diagram and the standard notations are explained below.

Activity Diagram Notations

1) Initial State – The starting state before an activity takes place is depicted using the initial state. In all of the below activity diagrams of the system initial state is mentioned in it.
e.gHere the initial state is the state of the system before the home page is viewed.

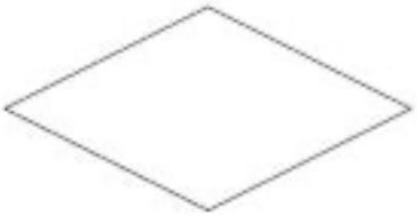


2) Action or Activity State – An activity represents execution of an action on objects or by objects. We represent an activity using a rectangle with rounded corners.

3) Action edges – Action flows or Control flows are also referred to as paths and edges. They are used to show the transition from one activity state to another. An activity state can have multiple incoming and outgoing action flows. We use a line with an arrow head to depict a Control Flow. If there is a constraint to be adhered to while making the transition it is mentioned on the arrow.

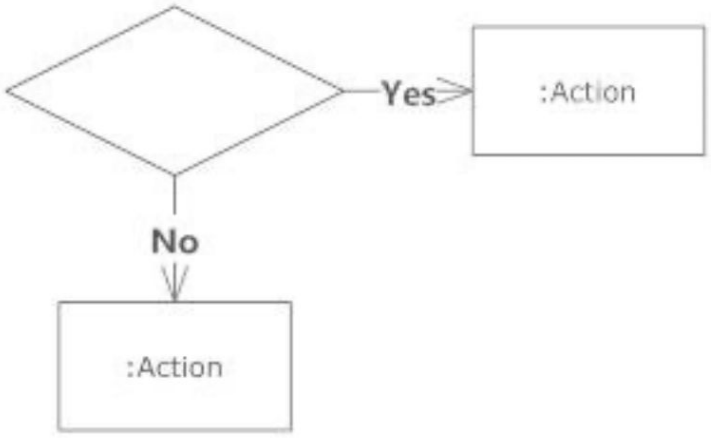
Action Flow

4) Decision node and Branching – When we need to make a decision before deciding the flow of control, we use the decision node. The outgoing arrows from the decision node can be labelled with conditions or guard expressions.It always includes two or more output arrows.



Decision Symbol

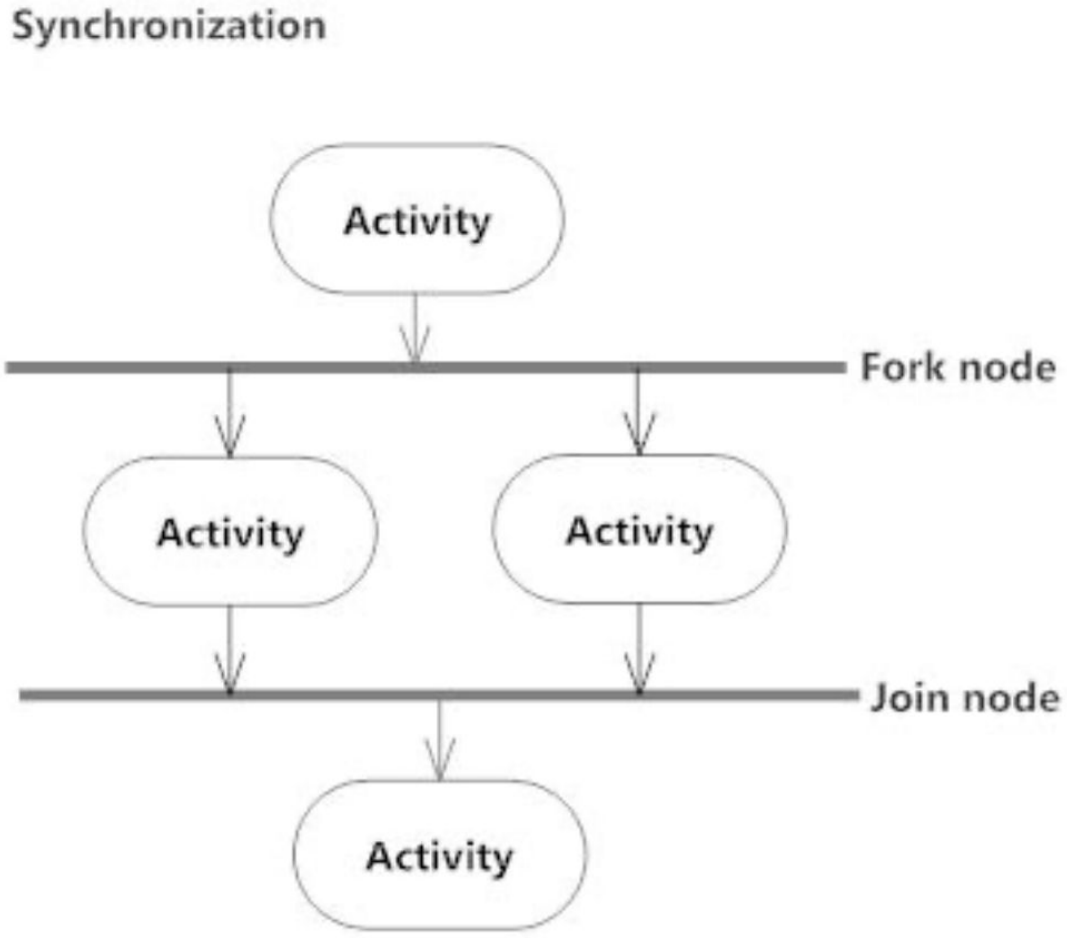
5) Guards – A Guard refers to a statement written next to a decision node on an arrow sometimes within square brackets. The statement must be true for the control to shift along a particular direction. Guards help us know the constraints and conditions which determine the flow of a process.



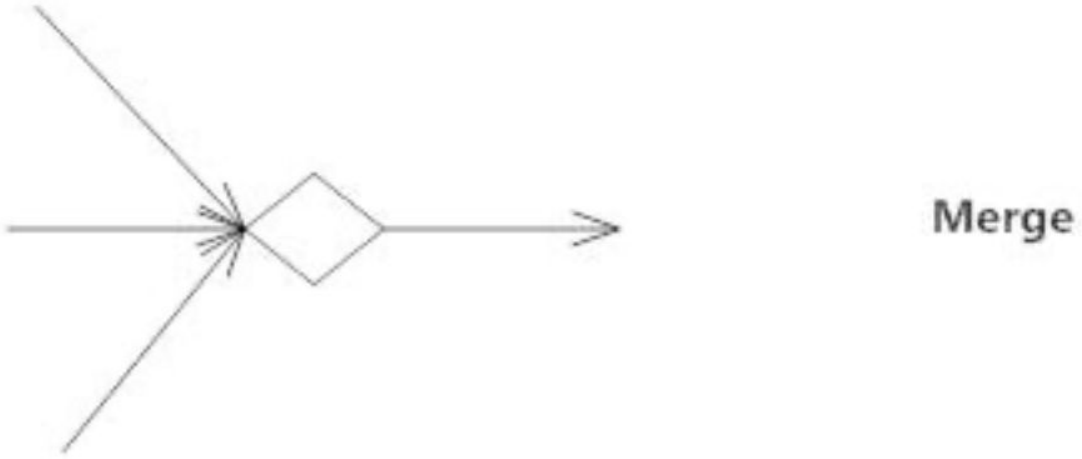
Guard Symbols

6) Fork – Fork nodes are used to support concurrent activities. When we use a fork node when both the activities get executed concurrently i.e. no decision is made before splitting the activity into two parts. Both parts need to be executed in case of a fork statement.
We use a rounded solid rectangular bar to represent a Fork notation with incoming arrows from the parent activity state and outgoing arrows towards the newly created activities.

7) Join – Join nodes are used to support concurrent activities converging into one. For join notations we have two or more incoming edges and one outgoing edge.



8) Merge or Merge Event – Scenarios arise when activities which are not being executed concurrently have to be merged. We use the merge notation for such scenarios. We can merge two or more activities into one if the control proceeds onto the next activity irrespective of the path chosen.

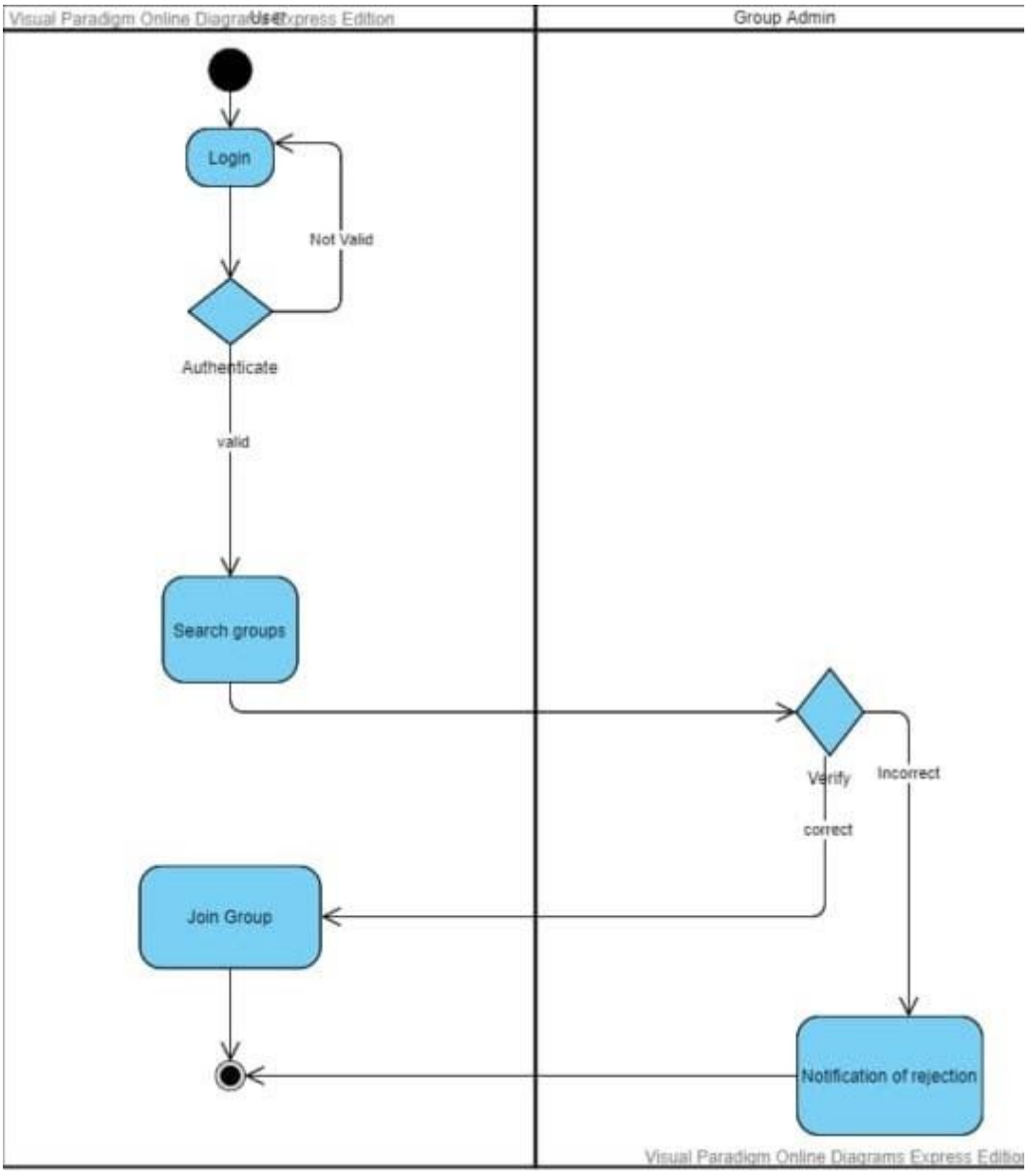


9) Swimlanes - Swimlanes group related activities into one column or one row. In below diagrams we use swim to for grouping related activities in one column such as user , system , verifying authority , admin.

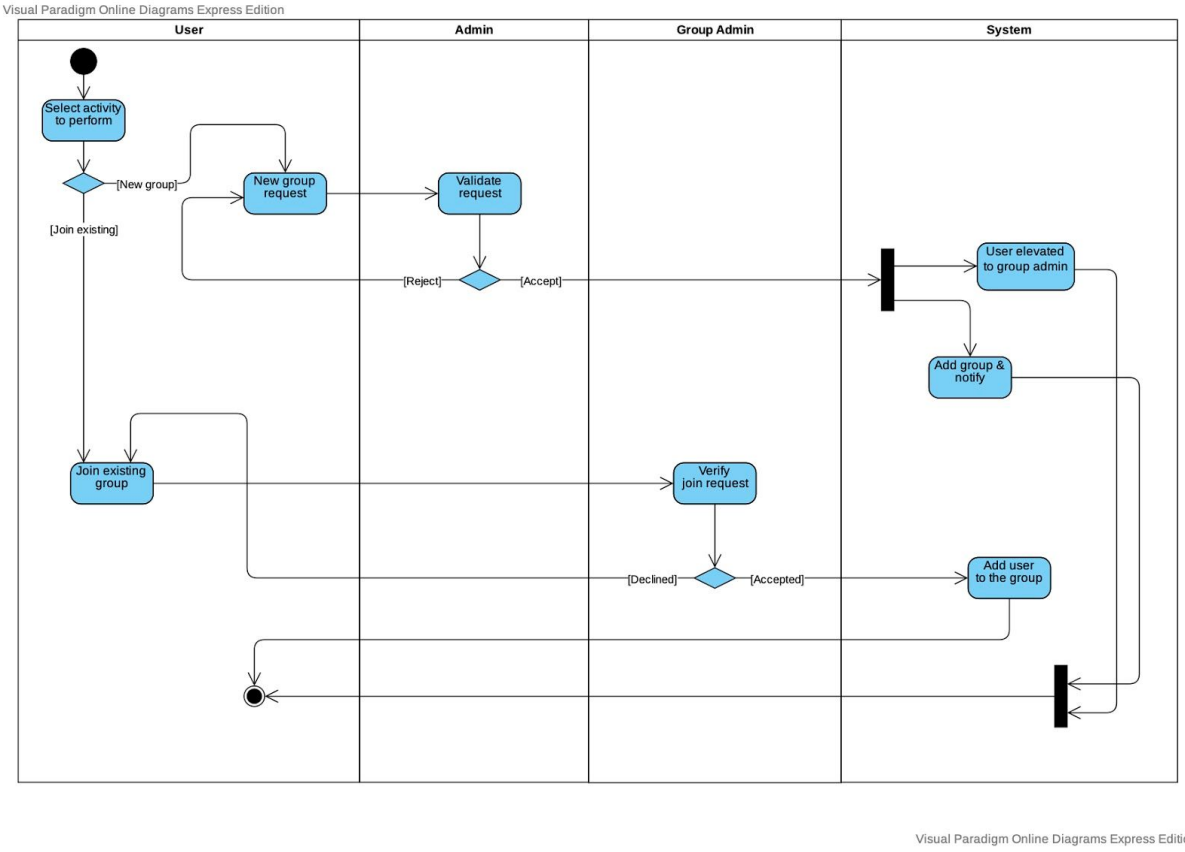
10) Final State or End State – The state which the system reaches when a particular process or activity ends is known as a Final State or End State. We use a filled circle within a circle notation to represent the final state in a state machine diagram. A system or a process can have multiple final states.

ACTIVITY DIAGRAMS:

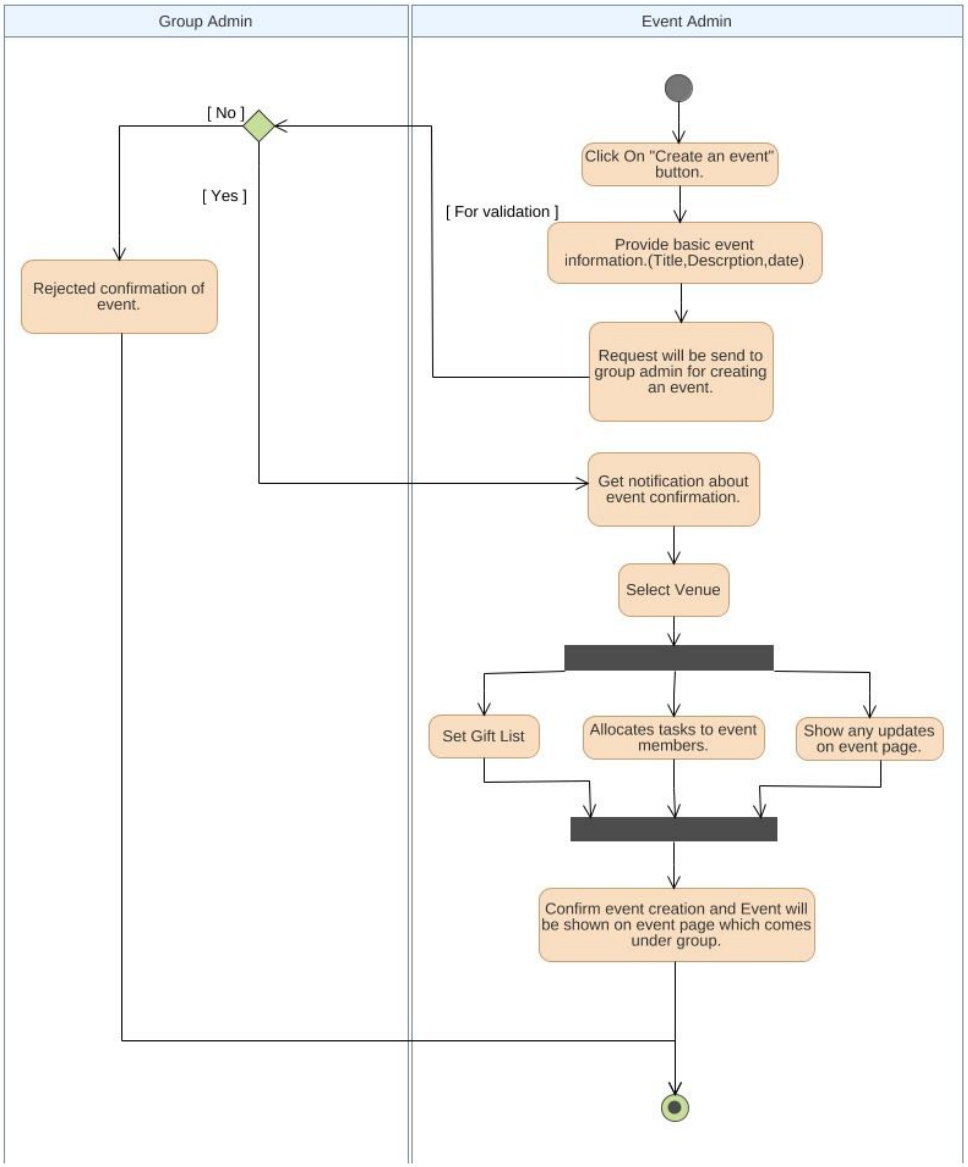
1) Sign up/ Login



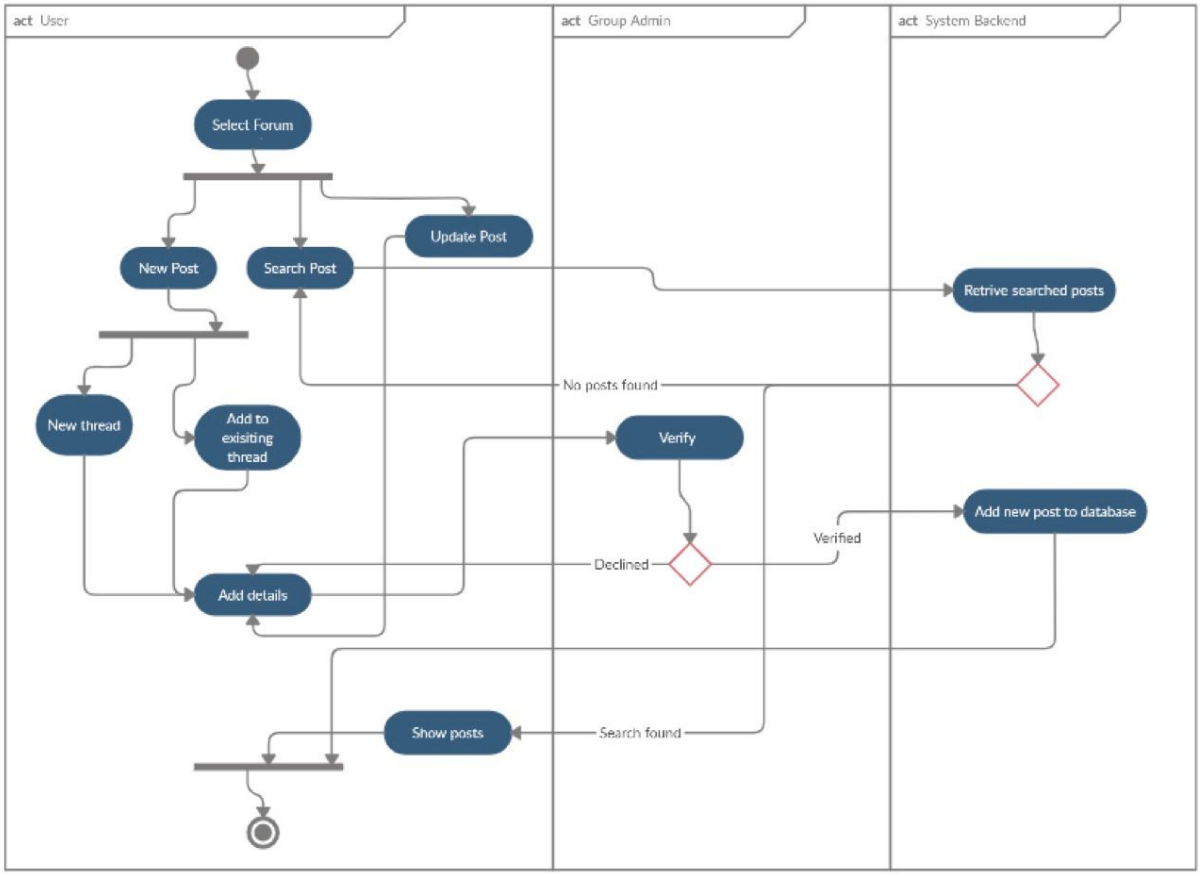
2) Create a new group/ Group Request



3) Event Request



4) Forums



CONCLUSION:

Thus, identified the activity and action states and implemented an activity diagram with swimlanes for all the major use cases of our similar interest based program.