Assignment No. 3

Title: Prepare activity model for Feedback System.

Problem Statement :

Prepare Activity Model, Identify Activity states, and Action states. Draw an activity diagram for a feedback system using UML2.0 Notations for major Use Cases.

Objective :

1) To Identify activities involved within the feedback system.
2) Design the Activity Diagram.

Theory :

Activity Diagrams :

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 the 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:

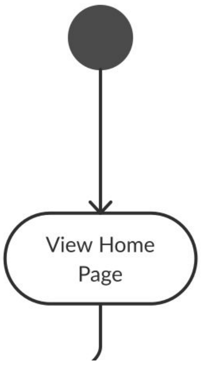
- Before starting a project, you can create activity diagrams to model the most important workflows.
- During the requirements phase, you can create activity diagrams to illustrate the flow of events that the use cases described.
- During the analysis and design phases, you can use activity diagrams 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.g
Here the initial state is the state of the system before the home page is viewed.



2) Action or Activity State –

An activity represents the execution of action on objects or by objects. We represent an activity using a rectangle with rounded corners.

In the real estate trading system, For multiple scenarios, different activity diagrams are drawn. Activities are mentioned in the diagrams at the end.

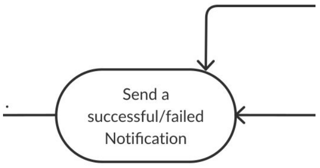
e.g.

Consider the previous example of viewing the home page of an application as an activity state in the activity diagram.

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 arrowhead to depict a Control Flow. If there is a constraint to be adhered to while making the transition it is mentioned on the arrow.

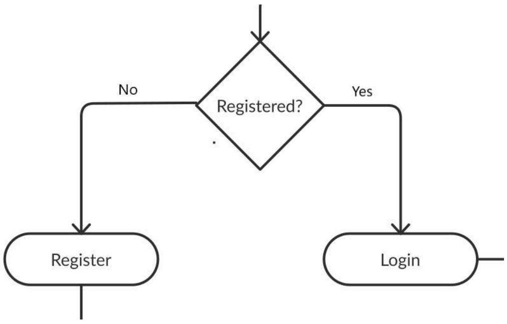
e.g. Here two edges are transiting to one activity send notification and one edge going out from sending notification activity.



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 labeled with conditions or guard expressions. It always includes two or more output arrows.

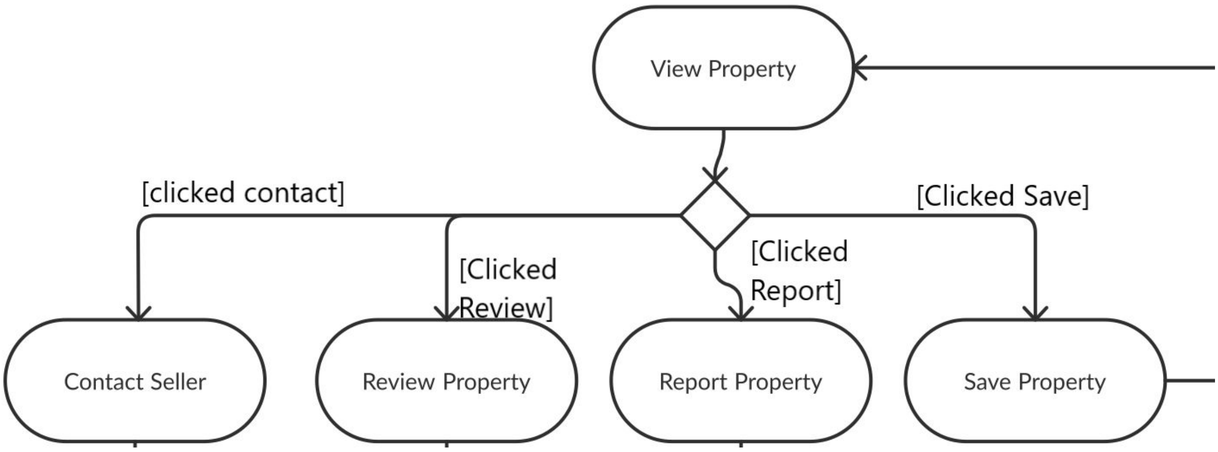
e.g.



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 in a particular direction. Guards help us know the constraints and conditions which determine the flow of a process.

e.g. Here the decision node is taking flow to multiple activities (contact seller, review property, report property, save property), and the guard on the arrow is telling conditions for choosing that activity.

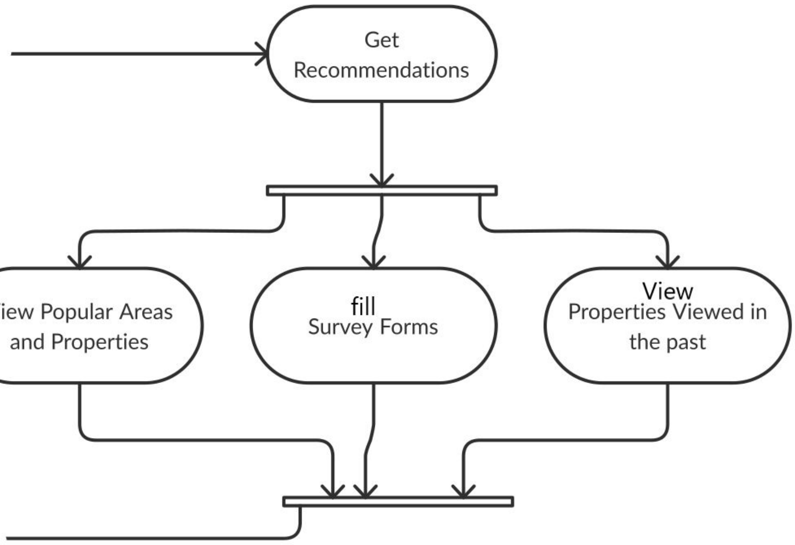


6) Fork –

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.

e.g.
In the example below, the activity of getting a recommendation can be split into three concurrent activities and hence we use the fork notation.


Get Recommendations
View Popular Areas and Properties
fill Survey Forms
View Properties Viewed in the past

7) Join –

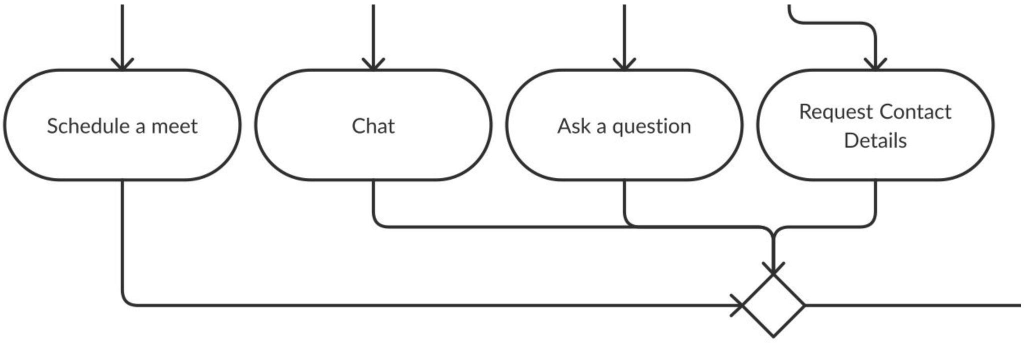
Join nodes are used to support concurrent activities converging into one. For join notations, we have two or more incoming edges and one outgoing edge.

e.g . In the above diagram, the Join node is used to converge 3 activities into one.

8) Merge or Merge Event –

Scenarios arise when activities that 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.

e.g. Hereafter contacting users by different actions, all activities are merging to transit flow to the next activity.

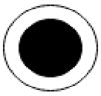


9) Swimlanes -

Swimlanes group related activities into one column or one row. In the 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 –

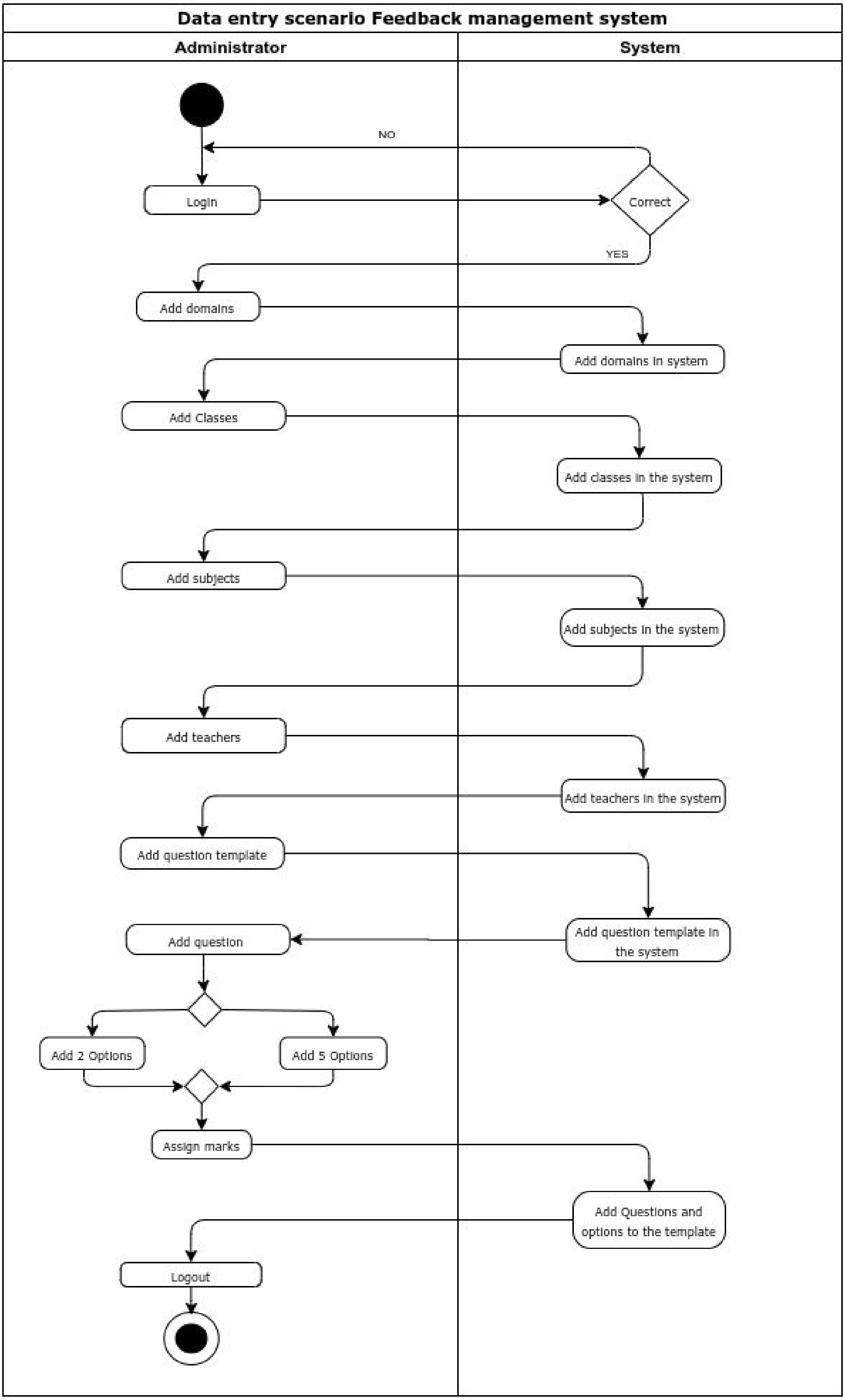
The state which the system reaches when a particular process or activity ends is known as a Final State or End State. We use a filled circle within a circle notation to represent the final state in a state machine diagram. A system or process can have multiple final states.

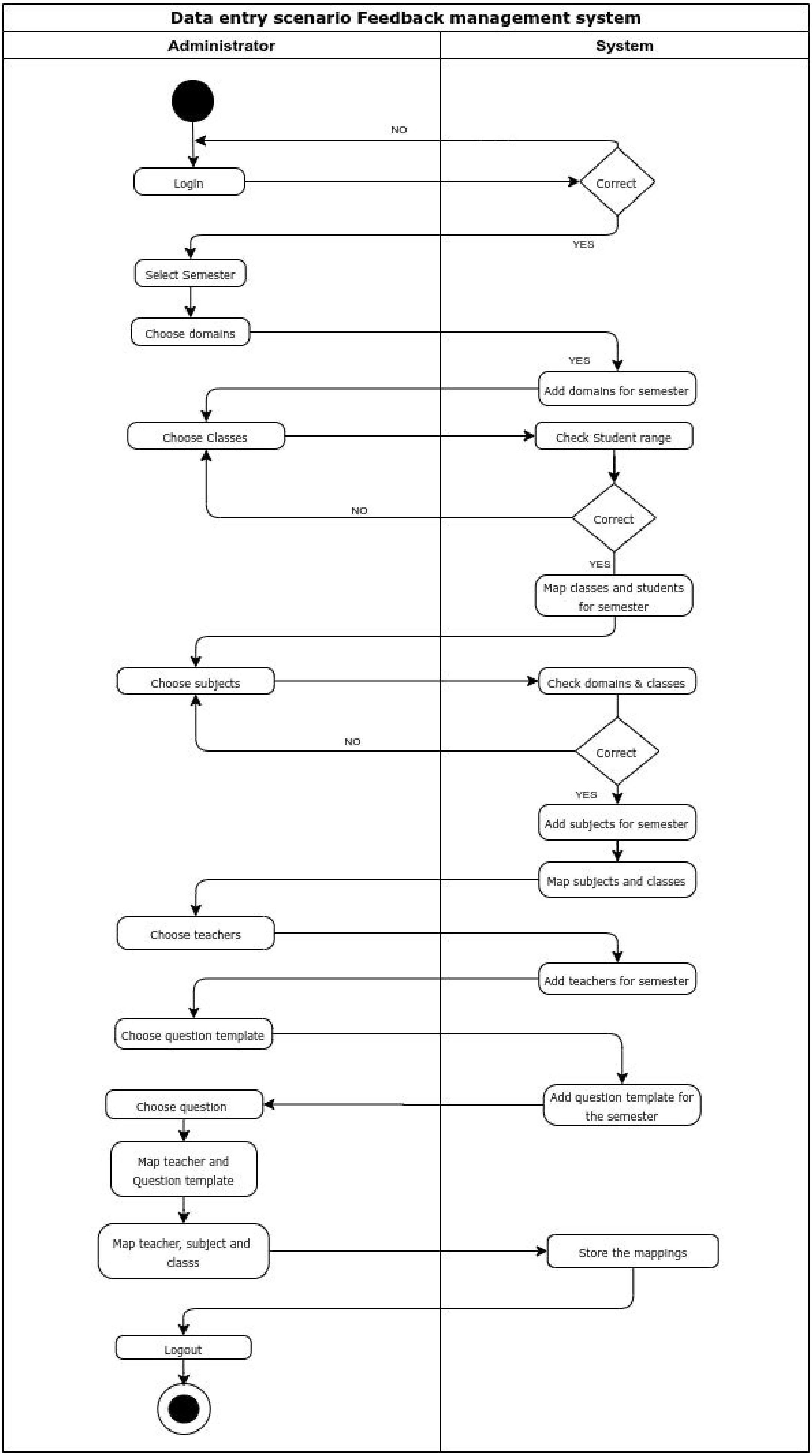


How to Draw an activity diagram –

1. Identify the initial state and the final states. Considering the registration and login diagram we can see that the user enters the home page at the start and gets a successful notification about login or registration at the end.
2. Identify the intermediate activities needed to reach the final state from the initial state. e.g. Enter details, Submit documents, send notification activities are identified before drawing activity diagrams for registration and login.
3. Identify the conditions or constraints which cause the system to change control flow like checking if a user is registered in the system or not in the registration and login activity diagram.
4. Draw the diagram with appropriate notations. In all of the below diagrams, proper notations are used for each situation.

1) **Data Entry Scenario Activity diagram for Feedback System :**


Data entry scenario Feedback management system
Administrator
System
NO
Login
Correct
YES
Add domains
Add domains in system
Add Classes
Add classes in the system
Add subjects
Add subjects in the system
Add teachers
Add teachers in the system
Add question template
Add question
Add question template in the system
Add 2 Options
Add 5 Options
Assign marks
Add Questions and options to the template
Logout


Data entry scenario Feedback management system
Administrator
System
NO
Login
Correct
YES
Select Semester
Choose domains
YES
Add domains for semester
Choose Classes
Check Student range
NO
Correct
YES
Map classes and students for semester
Choose subjects
Check domains & classes
NO
Correct
YES
Add subjects for semester
Map subjects and classes
Choose teachers
Add teachers for semester
Choose question template
Add question template for the semester
Choose question
Map teacher and Question template
Map teacher, subject and classs
Store the mappings
Logout

There are two scenarios for the feedback system activity diagram. First, the admin will create all the entities in the central system. Second, the admin will just map these already created entities in the current semester feedback.

First data entry scenarios consist of only entries of the entities like domain, class, subject, teacher, question, etc. These entities are created one time and can be updated or deleted later.

Now for each semester's feedback, the admin needs to add and map these entities in the currently selected feedback system. Teachers are mapped with questions, classes are mapped with subjects. Note, that these entities are already created at the time of central feedback system entry. In this scenario, these entities are just mapped as per the decisions of the admin.

After this system becomes ready to host feedback for that particular semester.

2) Password Generation and Feedback activity

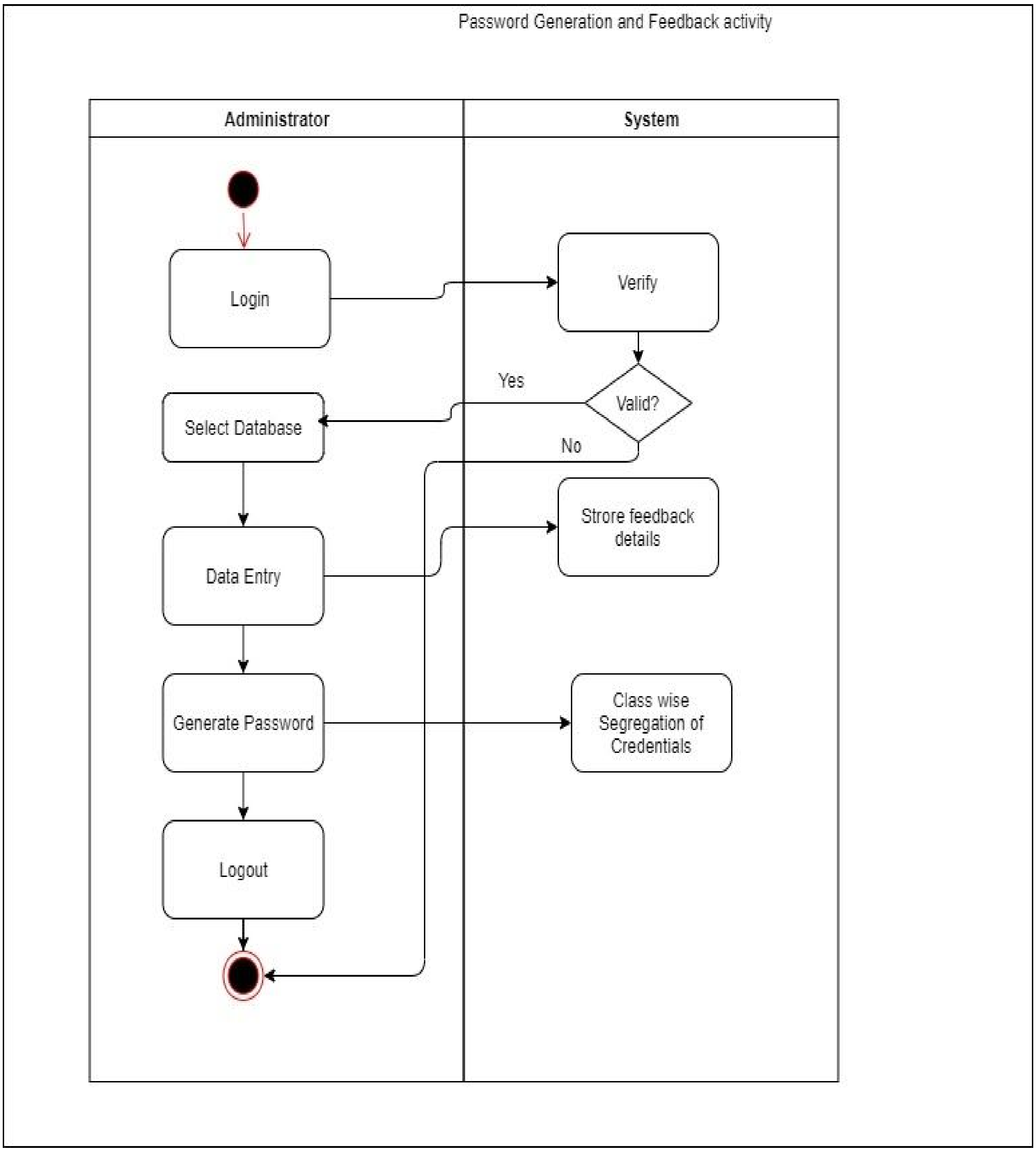


Figure: Generate Feedback and Password Distribution

In this activity, the admin creates the feedback system through the data entry activity which makes the system ready to take the feedback. In this activity admin the generated the password for the class-wise distribution to take the feedback. This is preceded and followed by login and logout activity which consists of verification with the database and validation in case of login activity.

3) Submit Feedback scenario activity diagram for Feedback System :

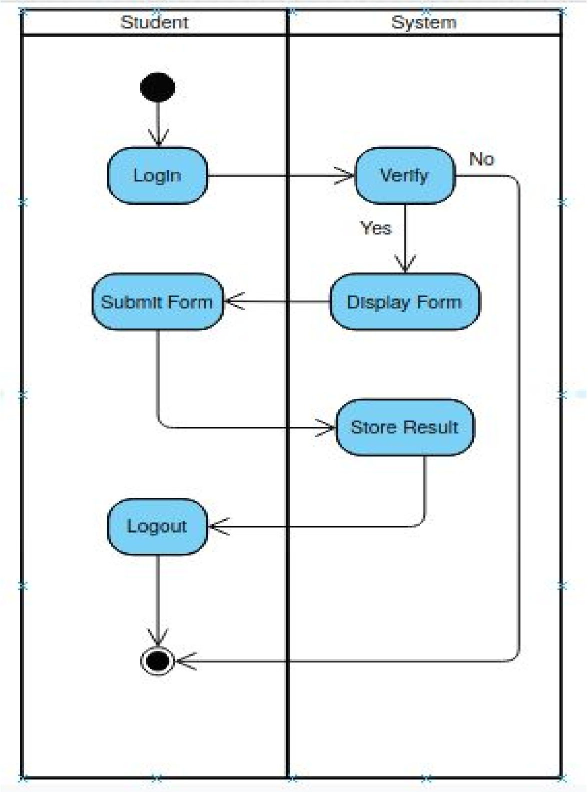


Figure: Submit Feedback Scenario

In the case of submit a feedback scenario, firstly the student needs to login into the feedback system with the Id and passwords provided by the previous scenario. After verification of the login credentials entered by the student, a feedback form will be displayed by the system. Further, after the student submits the feedback form the result will be stored by the system. After successful submission of the feedback form, students can logout from the system.

Conclusion :

Thus, identified the activity and action states and implemented an activity diagram with swimlanes for all the major use cases.