Shri Ramdeobaba College of Engineering and Management Nagpur

Name- Harsh Agrawal CSE SEM-6 Roll no-43 Section-B

Subject-Software Engineering LAB.

Practical no- 3

Aim- Creating a Activity Diagram To Implement the Employee Management System

---

Theory->

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 the 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 modeling where their primary use is to depict the dynamic aspects of a system. An activity diagram is very similar to a flowchart. So let us understand if an
- activity diagrams or flowcharts are any different.

- An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram. Activities modeled can be sequential and concurrent. In both cases the activity diagram will have a beginning (an initial state) and an end (a final state). In between there are ways to depict activities, flows, decisions, guards, merge and time events and more.
- The basic usage of the activity diagram is similar to the other four UML diagrams. The specific usage is to model the control flow from one activity to another. This control flow does not include messages.
- Activity diagram is suitable for modeling the activity flow of the system. An application can have multiple systems. Activity diagram also captures these systems and describes the flow from one

system to another. This specific usage is not available in other diagrams. These systems can be databases, external queues, or any other system.

- We will now look into the practical applications of the activity diagram. From the above discussion, it is clear that an activity diagram is drawn from a very high level. So it gives a high level view of a system. This high level view is mainly for business users or any other person who is not a technical person.
- This diagram is used to model the activities which are nothing but business requirements. The diagram has more impact on business understanding rather than on implementation details.

Activity diagram can be used for −

- Modeling workflow by using activities.
- Modeling business requirements.
- High level understanding of the system's functionalities.
- Investigating business requirements at a later stage.

Activity Diagram Notations –

Initial State – The starting state before an activity takes place is depicted using the initial state.

UML-State-Diagram

- A process can have only one initial state unless we are depicting nested activities. We use a black filled circle to depict the initial state of a system. For objects, this is the state when they are instantiated. The Initial State from the UML Activity Diagram marks the entry point and the initial Activity State.
- 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. Basically any action or event that takes place is represented using an activity.
- For example – Consider the previous example of opening an application opening the application is an activity state in the activity diagram.
- Action Flow or Control flows – 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.
- 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.
- Guards – A Guard refers to a statement written next to a decision node on an arrow sometimes within square brackets.

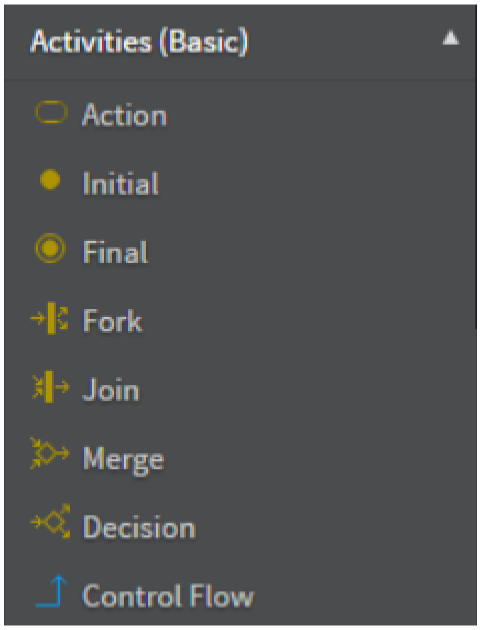
- The statement must be true for the control to shift along a particular direction. Guards help us know the constraints and conditions which determine the flow of a process.

- 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 arrow from the parent activity state and outgoing arrows towards the newly created activities.
- For example: In the example below, the activity of making coffee can be split into two concurrent activities and hence we use the fork notation.

- 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.

- For example – When both activities i.e. steaming the milk and adding coffee get completed, we converge them into one final activity.

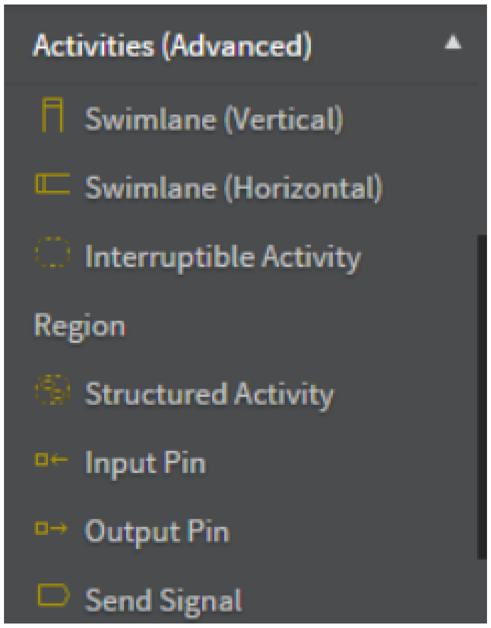
- Merge or Merge Event – Scenarios arise when activities which are not being executed concurrently have to be merged. We use the merge notation for such scenarios. We can merge two or more activities into one if the control proceeds onto the next activity irrespective of the path chosen.
- For example – In the diagram below: we can't have both sides executing concurrently, but they finally merge into one. A number can't be both odd and even at the same time.
- Swimlanes – We use swimlanes for grouping related activities in one column. Swimlanes group related activities into one column or one row. Swimlanes can be vertical and horizontal. Swimlanes are used to add modularity to the activity diagram. It is not mandatory to use swimlanes. They usually give more clarity to the activity diagram. It's similar to creating a function in a program. It's not mandatory to do so, but it is a recommended practice.
- We use a rectangular column to represent a swimlane as shown in the figure above.
- For example – Here different sets of activities are executed based on if the number is odd or even. These activities are grouped into a swimlane.
- We can have a scenario where an event takes some time to complete. We use an hourglass to represent a time event.
- For example – Let us assume that the processing of an image takes a lot of time. Then it can be represented as shown below.
- 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.

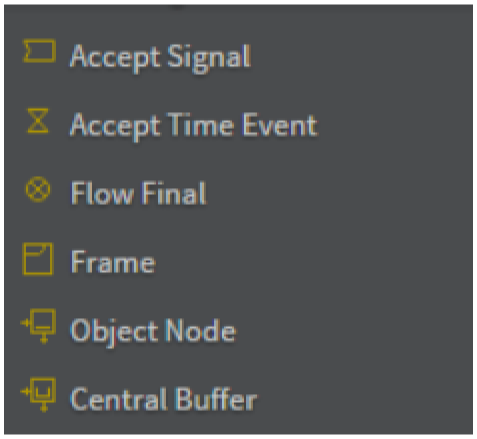


How to Draw an activity diagram –

- Identify the initial state and the final states.
- Identify the intermediate activities needed to reach the final state from the initial state.
- Identify the conditions or constraints which cause the system to change control flow.
- Draw the diagram with appropriate notations.
- UML-Activity-Diagram

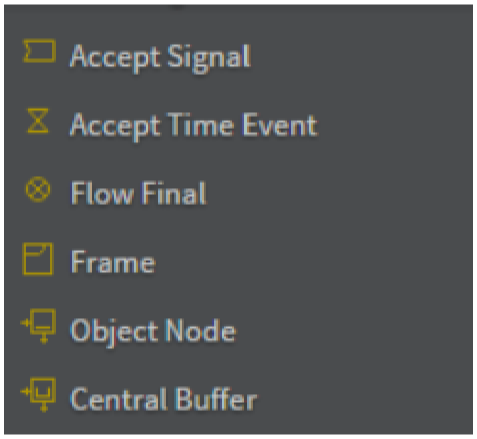
The above diagram prints the number if it is odd otherwise it subtracts one from the number and displays it.

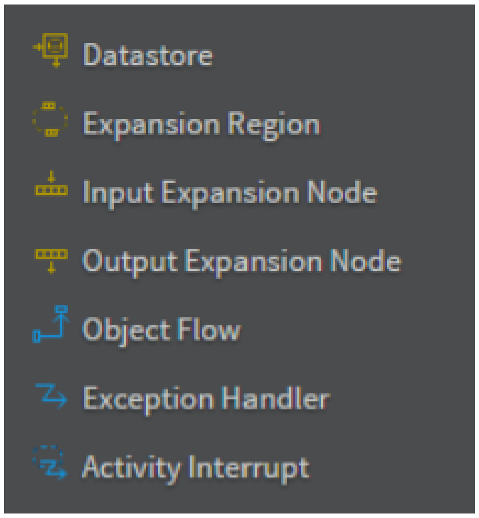




Uses of an Activity Diagram –

- Dynamic modeling of the system or a process.
- Illustrate the various steps involved in a UML use case.
- Model software elements like methods,operations and functions.
- We can use Activity diagrams to depict concurrent activities easily.
- Show the constraints, conditions and logic behind algorithms.





Activity Diagram for Employee Management System

There are various types of Activities that are present in the activity diagram of the Employee management system.

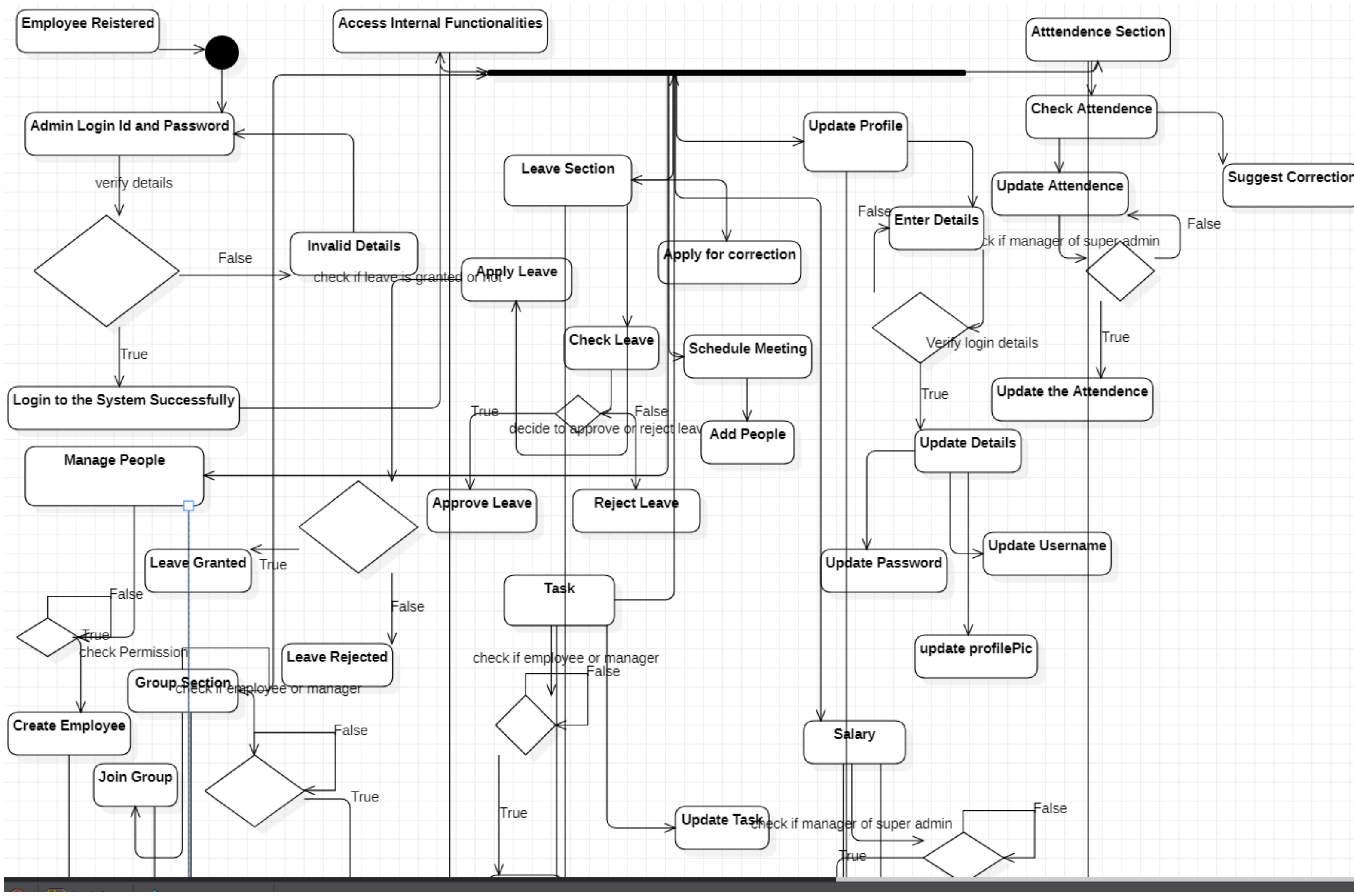
The first activity in the Employee management system is login into the system.

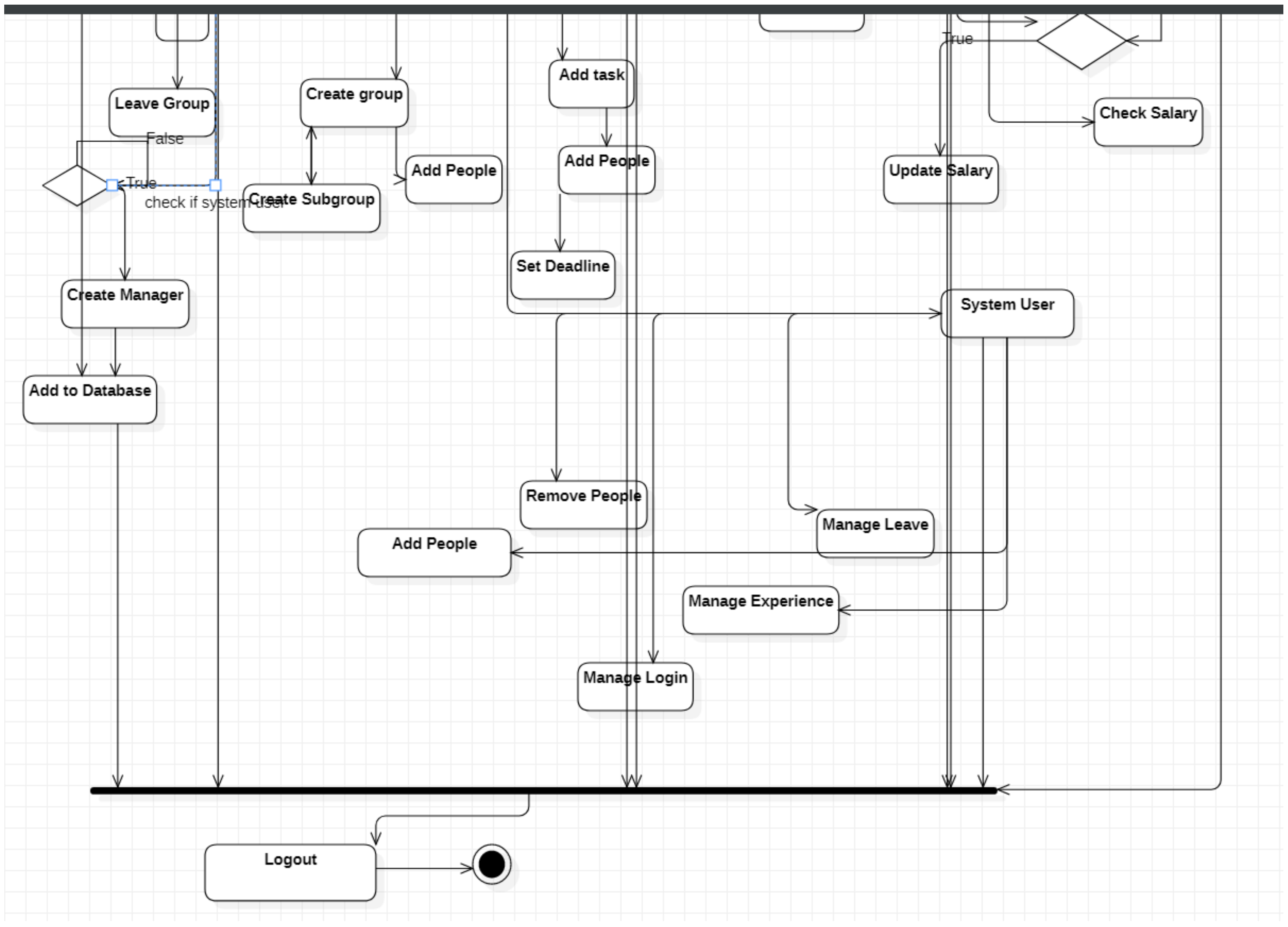
We have to first login to the system then id and Password is verified if the enter details are correct then we can access to the internal functionalities of the system and if details are incorrect then we cannot proceed.

- The next activity is the **leave section**.Employee can apply for leave which is to be checked by the manager if the leave is approved the granted else rejected. Employees can check the leave and apply for corrections if any.

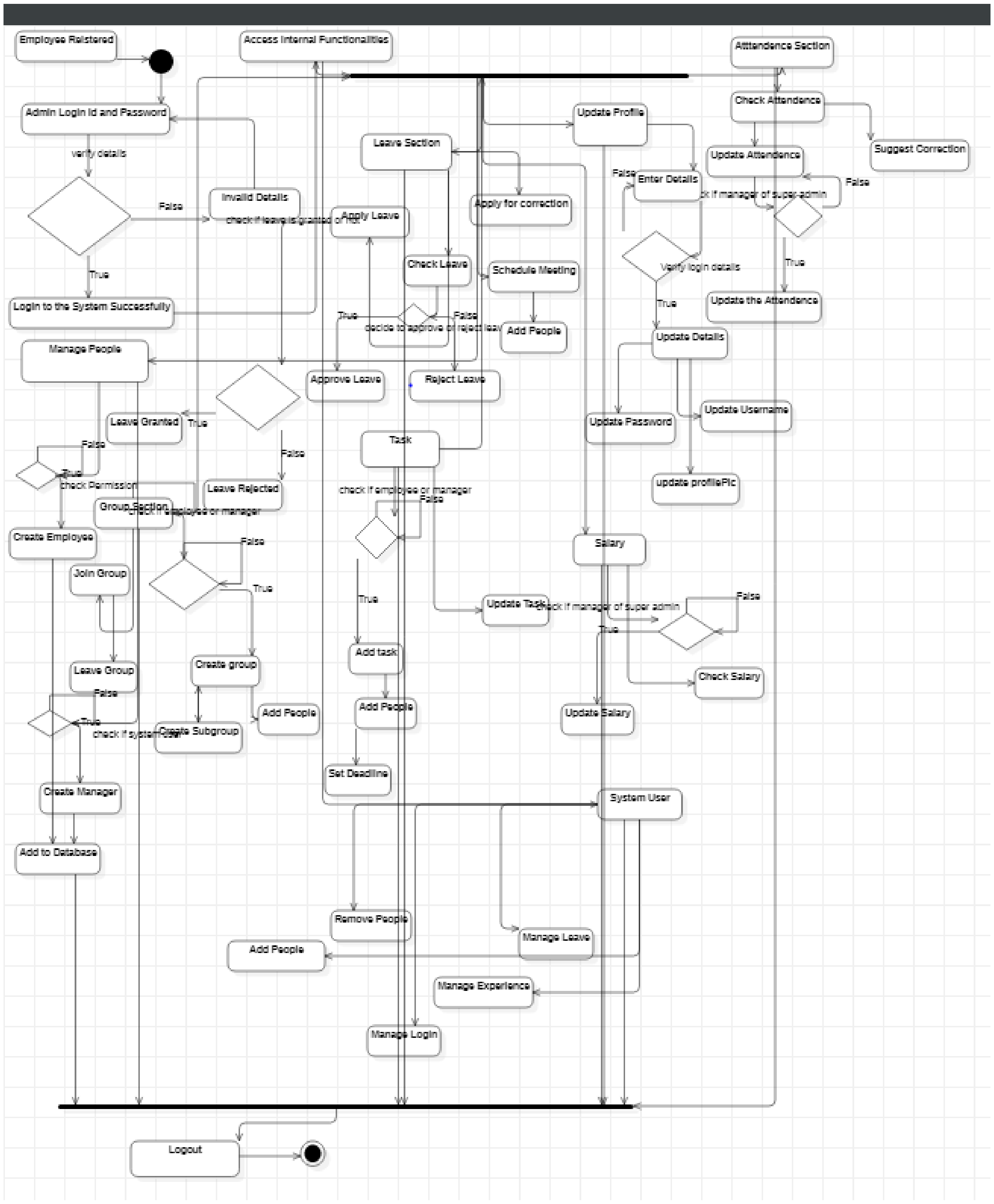
- Next activity is the **attendance activity** where managers can add or update the attendance and employees can just check the attendance and apply for corrections if any.
- Next Activity is the **update details** activity. All the users who have an account in the system can update his/her own profile. Before updating the profile the user has to first verify the account details if correct he can update username, password and profile picture.
- Next Activity is the **salary** where employees can add and update the salary of the Employees . employees can only view the status of their salaries.
- Next Activity is the **task section** where the Managers can add the task with task details with task deadline and people who are assigned the task. Employees can only update the details of their performance.
- Next Activity is **Group Section** activity. Managers can create groups and people according to their choice. They can also create subgroups as well. Employees can leave the group if they want.
- Next section is the **System user** section where the system user is responsible for managing several functionalities of the system. They can add or remove people in the system. They can Manage leave and experience of all the users. They are responsible for smooth login and logout in the system.
- Every Activity has a **check permission** feature where every time the permissions are verified if available for a particular feature.

Hence these are all the activities in the Activity diagram of the Employee Management System.





Complete Diagram->



Result-> The Activity diagram for the Employee management system is studied and verified successfully.