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- 5

Aim→ Creating a State Diagram To Implement the Employee Management System

---

Theory→

Unified Modeling Language (UML) | State Diagrams

A state diagram is used to represent the condition of the system or part of the system at finite instances of time. It's a behavioral diagram and it represents the behavior using finite state transitions. State diagrams are also referred to as State machines and State-chart Diagrams.

These terms are often used interchangeably. So simply, a state diagram is used to model the dynamic behavior of a class in response to time and changing external stimuli. We can say that each and every class has a state but we don't model every class using State diagrams. We prefer to model the states with three or more states.

Uses of statechart diagram →

1. We use it to state the events responsible for change in state (we do not show what processes cause those events.
2. We use it to model the dynamic behavior of the system .
3. To understand the reaction of objects/classes to internal or external stimuli.

Firstly let us understand what are Behavior diagrams? There are two types of diagrams in UML

:

1. Structure Diagrams → Used to model the static structure of a system, for example- class diagram, package diagram, object diagram, deployment diagram etc.
2. Behavior diagram → Used to model the dynamic change in the system over time. They are used to model and construct the functionality of a system. So, a behavior diagram simply guides us through the functionality of the system using Use case diagrams, Interaction diagrams, Activity diagrams and State diagrams.

Firstly let us understand what are Behavior diagrams? There are two types of diagrams in UML→

1. **Structure Diagrams** – Used to model the static structure of a system, for example- class diagram, package diagram, object diagram, deployment diagram etc.
2. **Behavior diagram** → Used to model the dynamic change in the system over time. They are used to model and construct the functionality of a system. So, a behavior diagram simply guides us through the functionality of the system using Use case diagrams, Interaction diagrams, Activity diagrams and State diagrams.

There are a total of two types of state machine diagrams:

1. Behavioral state machine

It captures the behavior of an entity present in the system.

It is used to represent the specific implementation of an element.

The behavior of a system can be modeled using behavioral state machine diagrams.

2. Protocol state machine

- These diagrams are used to capture the behavior of a protocol.
- It represents how the state of protocol changes concerning the event. It also represents corresponding changes in the system. They do not represent the specific implementation of an element.
- A state machine is any device that stores the status of an object at a given time and can change status or cause other actions based on the input it receives. States refer to the different combinations of information that an object can hold, not how the object behaves. In order to understand the different states of an object, you might want to visualize all of the possible states and show how an object gets to each state, and you can do so with a UML state diagram.
- Each state diagram typically begins with a dark circle that indicates the initial state and ends with a bordered circle that denotes the final state. However, despite having clear start and end points, state diagrams are not necessarily the best tool for capturing an overall progression of events.
- Rather, they illustrate specific kinds of behavior—in particular, shift from one state to another. State diagrams mainly depict states and transitions. States are represented with rectangles with rounded corners that are labeled with the name of the state. Transitions are marked with arrows that flow from one state to another, showing how the states change. Below, you can see both these elements at work in a basic diagram for student life.

Why State Machine Diagram?

- Statechart diagram is used to capture the dynamic aspect of a system. State machine diagrams are used to represent the behavior of an application. An object goes through various states during its lifespan. The lifespan of an object remains until the program is terminated. The object goes from multiple states depending upon the event that occurs within the object. Each state represents

- Some unique information about the object.
- Notation and Symbol for State Machine
- Following are the various notations that are used throughout the state chart diagram. All these notations, when combined, make up a single diagram.

Unified Modeling Language defines three types of states→

Simple state

- They do not have any substrate. Composite state
- These types of states can have one or more than one substrate.
- A composite state with two or more sub states is called an orthogonal state.

Submachine state

- These states are semantically equal to the composite states.
- Unlike the composite state, we can reuse the submachine states.

State diagram applications→

- Like most UML diagrams, state diagrams have several uses. The main applications are as follows:
- Depicting event-driven objects in a reactive system.
- Illustrating use case scenarios in a business context.
- Describing how an object moves through various states within its lifetime.

- Showing the overall behavior of a state machine or the behavior of a related set of state machines.

Why State Machine Diagrams?

State machine diagram typically are used to describe state-dependent behavior for an object. An object responds differently to the same event depending on what state it is in. State machine diagrams are usually applied to objects but can be applied to any element that has behavior to other entities such as: actors, use cases, methods, subsystems systems and etc. and they are typically used in conjunction with interaction diagrams (usually sequence diagrams).

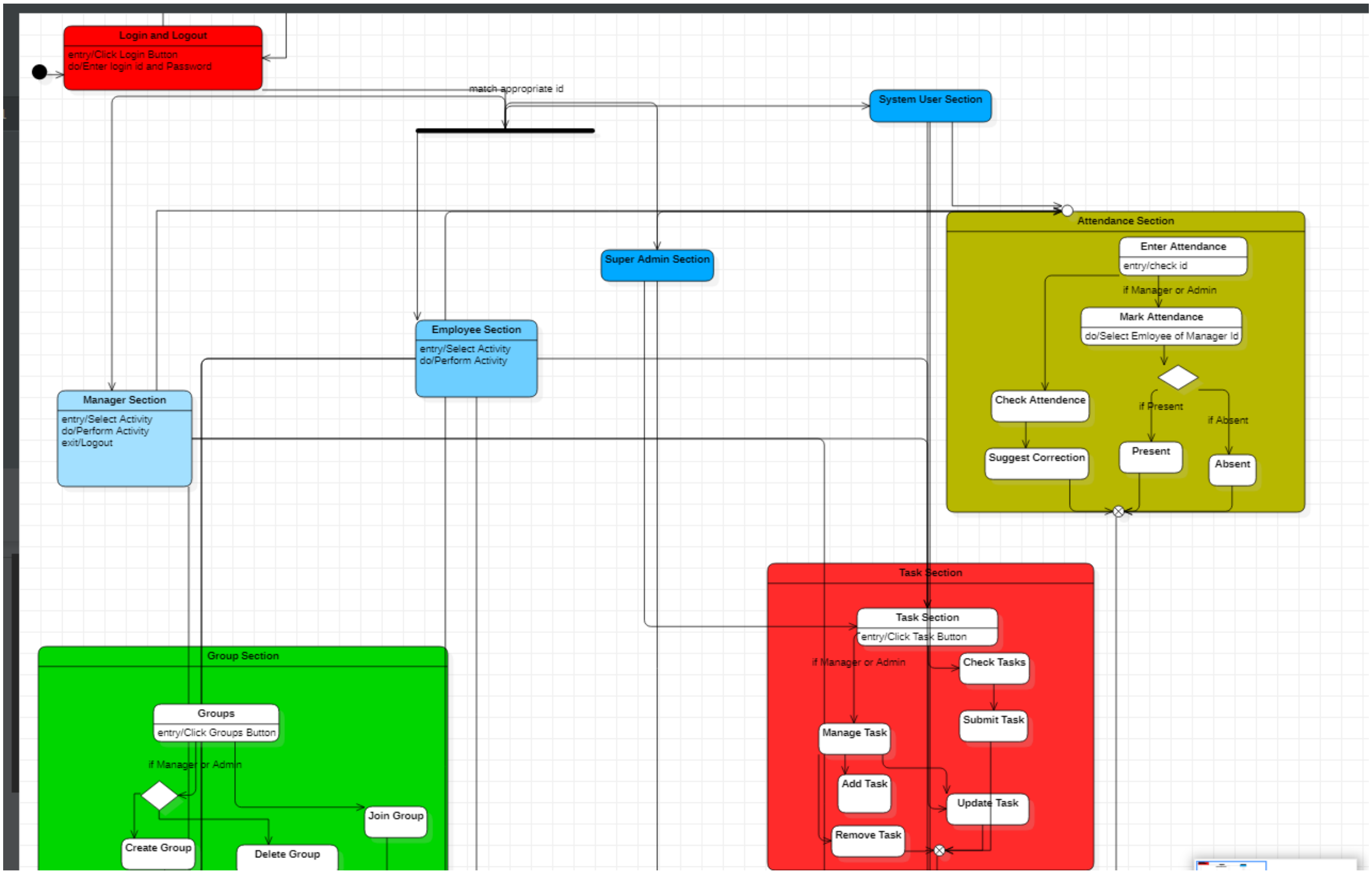
—---

State UML Diagram for Employee Management System.

This is the State chart Diagram created for the employee Management System. The Application has four types of users i.e Super Admin, Manager, Employee and System User. There are various types of states in the state diagram.

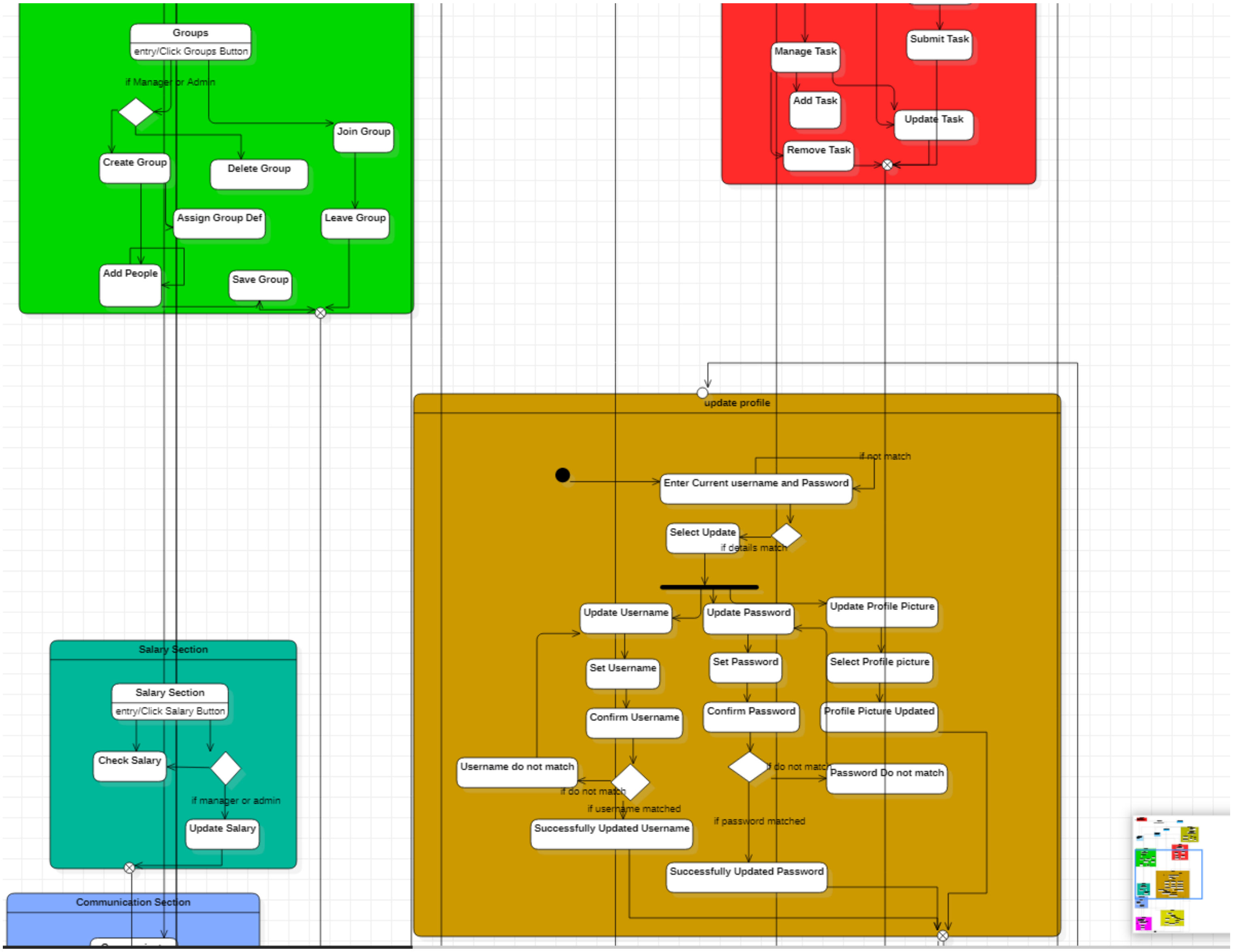
Attendance section→Everyone can view their own Attendance but only the Managers and Super Admin can mark anybody's attendance. The Attendance can be marked as present or absent.

Next is the Task Section→Everybody can check the task scheduled for them. They can put the updates for the tasks and mark the task completed if they have done it. Managers and super Admin can schedule tasks for anybody. They can add people to the task and can remove the task if required.



Next is the group Section→ Managers and Super Admin can create or delete a group to add to this they can give a definition to the group they can add people in the group. Employees can join a created group and can leave it as well if required.

Next is the update profile section→ This Section is common for Everybody. Before doing anything the user has to first validate the username and password if correct then only further changes could be done. For setting up the username or password the user has to confirm the details two times if the details are correct and matched then only it would be saved. Profile pictures can be directly updated in one go.

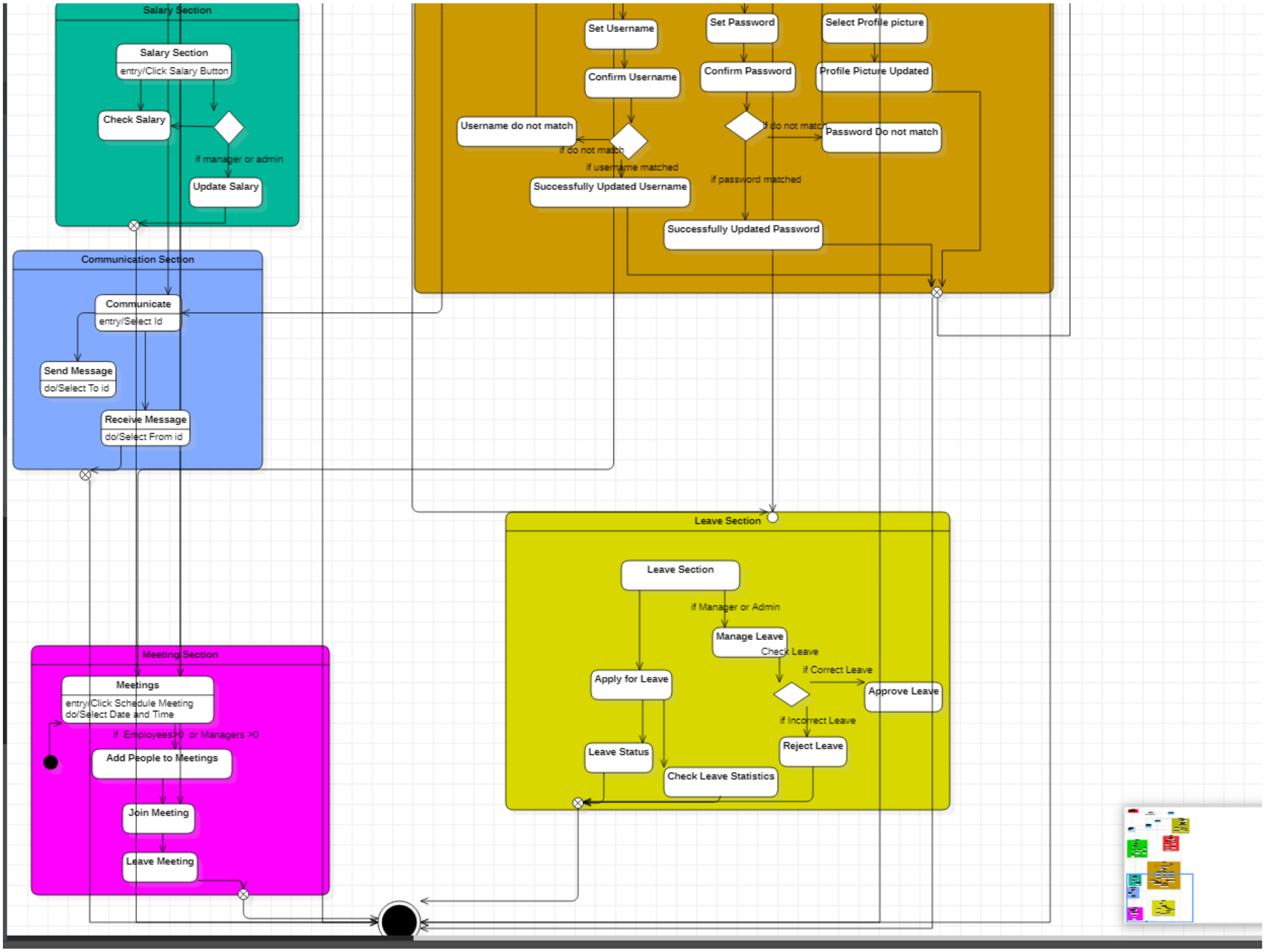


Next is the Salary Section→ Everybody can check his/her salary. Manages and Super Admin can add salary for the junior ones.

Next is Communication Section-> Just enter the id of the person with whom you want to communicate and then share data.

Next is the Meeting Section→ Managers and Super Admin can Schedule meetings for everybody. employees can just join the meetings.

Next is the Leave Section-> Everyone can apply for the leave. They can check the leave status if the leave is granted or not. Managers and super admin can decide whether to permit for the leave or not.



<u>Result →</u> State Chart Diagram has been Studied Successfully. For Employee Management System State Chart Diagram has been created.

The End!!