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- 2

Aim-To create Use-Case diagram to depict the user's view of the system.

Theory->

A use case is a collection of interactions which delivers a valuable result to a user. A use case should realise a goal or objective of a user or a particular class of users.

A use case captures a contract between the stakeholders of a system about its behaviour. The use case describes the system's behaviour under various conditions as the system responds to a request from one of its stakeholders, called the primary actor.

The purpose of a use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and Statechart) also have the same purpose.

We will look into some specific purpose, which will distinguish it from other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

When the initial task is complete, use case diagrams are modelled to present the outside view.

In brief, the purposes of use case diagrams can be said to be as follows:

- Used to gather the requirements of a system.
- Used to get an outside view of a system.
- Identify the external and internal factors influencing the system.
- Show the interaction among the requirements are actors.

The main applications of use cases are:

- Early development steps;
- Classifying requirements;
- Guiding the rest of the development process;
- Base of (acceptance) test cases.

A UML use case diagram consists of several elements.

These elements are as follows :

- □ Use Case: A specific use case within the scenario.
- □ Actor: An actor within the described scenario.
- □ Relationship: A specific relationship between two elements in the diagram:

A relationship where use case A (source) extends use case B (destination). This extension is optional within use case B. Example: 'Online help' may be triggered when performing an 'ATM transaction'. A relationship where use case A (source) includes use case B (destination). The inclusion is not optional within use case B. Example: 'Card identification' is triggered when 'withdrawing' money.

Association: An associative relationship between an actor and a use case (e.g. 'Customer' may make use of 'ATM transaction').

Generalization: A generalization between two actors or two use cases (e.g. 'Customer' is a 'User').

A requirements analysis concept

- A case of a use of the system/product
- Describes the system's actions from a the point of view of a user
- Tells a story
- A sequence of events involving
- Interactions of a user with the system
- Specifies one aspect of the behavior of a system, without specifying the structure of the system
- Is oriented toward satisfying a user's goal

The notation for graphical UML Use Case diagrams is quite simple and makes use of standard UML elements. All elements are shown in figure 1. Most elements are annotated, with the exception of the association element between the "Generalized actor" and "Use Case", and the generalization between"Actor `` and"Generalised actor". In literature, some elements may sometimes be visualized in a slightly different way (e.g. different arrows or line types) according to the author's preference.

As with most UML methods and techniques, there is no fixed procedure for creating Use Case models or diagrams. The notation and technique are provided mainly as a guideline. The next section provides an example of the use case diagramming process. When creating UML Use Case diagrams, it is important to apply a suitable level of abstraction. There shouldn't be too many use cases and associations within the diagram to maintain readability and usefulness.

Use Case Descriptions

- actors - something with a behavior or role, e.g., a person, another system, organization.
- scenario - a specific sequence of actions and interactions between actors and the system, a.k.a. use case instance
- use case - a collection of related success and failure scenarios, describing actors using the system to support a goal.

In UML modeling, a relationship is a connection between two or more UML model elements that adds semantic information to a model.
In the product, you can use several UML relationships to define the structure between model elements. Examples of relationships include associations, dependencies, generalizations, realizations, and transitions.

Use Case Diagram Objective

- Built in early stages of development
- Purpose
- Specify the context of a system
- Capture the requirements of a system
- Validate a systems architecture
- Drive implementation and generate test cases
- Developed by analysts and domain experts

Elements of use case diagram: Actor

- Actor is someone interacting with a use case (system function). Named by noun.

• Similar to the concept of user, but a user can play different roles; (example: a prof. can be instructor and researcher – plays 2 roles with two systems).

• Actor triggers a use case.
• Actor has responsibility toward the system (inputs), and Actors have expectations from the system (outputs).

Use Case

• System function (process – automated or manual).
• Named by verb -Do something
Each Actor must be linked to a use case, while some use cases may not be linked to actors.

Linking Use Cases

• Association relationships
• Generalization relationships
• One element (child) "is based on" another element (parent)

• Include relationships
• One use case (base) includes the functionality of another (inclusion case)
• Supports re-use of functionality
• Extend relationships
• One use case (extension) extends the behavior of another (base)

Actors may be connected to use cases by associations, indicating that the actor and the use case communicate with one another using messages.

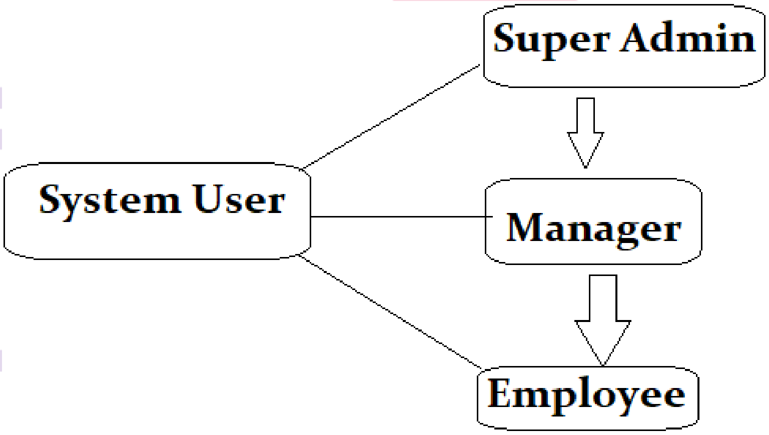
Employee Management System-**An employee management system is a software that helps your employees to give their best efforts every day to achieve the goals of your organization. It guides and manages employees' efforts in the right direction. It also securely stores and manages personal and other work-related details for your employees. That makes it easier to store and access the data when there is a need.**
Our employee management system offers an intelligent module to keep you organized. With the software, you can get all the data of your employees at your fingertips. The advantage of employee management software is that it is cloud-based software. Giving you access wherever you are at. You don't have to hold on to the decisions for lack of information.

With employee management software, you can utilize the standard metrics for employee management. You can also customize or create your own metrics for the needs of the staff management in your company with ease. The pocket HRMS has a very easy to handle user interface and is seamless to help you with the performance management of the employees.

Actors

- ❖ Super Admin
- ❖ Manager
- ❖ Employee
- ❖ System User

Control Structure->



★ Employee->

All the Employees are Managed by the Managers and Super Admin.

Various features of the Employees are-

- They can create their profile on the Application and can login to their account successfully. They can also change their profile settings such as changing the profile photo, username and Password.
- They can Check their Attendance and can put a remark or note if there is a mistake.
- They can Apply for leave which would be checked by the manager.

- They can check for their salary as well
- They can join or leave any of the project groups which are created by the managers or super admin
- They can receive and submit their tasks so as to be checked by the managers.
- They can communicate with anybody in the organization and can share the data.

❖ Manager->

  - They can create their profile on the Application and can login to their account successfully. They can also change their profile settings such as changing the profile photo, username and Password.
  - They can control all the Employees. They can Manage the attendance of the Employees.
  - They can grant leave to the Employees as required.
  - They can check and Schedule tasks for the Employees and can create groups according to the Projects.
  - They can schedule meetings and can Apply for leaves and can check their attendance and salary.
  - They can Communicate and share data with anybody.

❖ Super Admin->

  - They can create their profile on the Application and can login to their account successfully. They can also change their profile settings such as changing the profile photo, username and Password.
  - They can Control everybody in the organization, Manage users and full application.

- They can Communicate and share data with anybody. They can attend and schedule meetings.
- They can Manage and assign tasks for managers.
- Manage Salary for managers.

❖ System User->

- They can add and remove people from the Application.
- They Ensure Smooth Working of the Application.
- They can create their profile on the Application and can login to their account successfully. They can also change their profile settings such as changing the profile photo, username and Password.
- They can Communicate and share data with anybody.
- Has Access to Every Setting and Feature of the Application.

Activities and Functionalities->

★ **Employee-**

- Create Account
- Update Account(change Username, Password and Profile photo etc)
- Apply for leave
- Check Attendance
- Check tasks/Assignments
- Check for Meetings
- Check Salary Status
- Join or leave Group
- Communicate and share data

★ **Manager**

- Create account

- Update Account(change Username, Password and Profile photo etc)
- Apply for leave, Grant leave to Employee, Manage Employee Attendance
- Schedule Meetings
- Assign tasks for Employees
- Create Employee Groups
- Check Status
- Communicate and share data

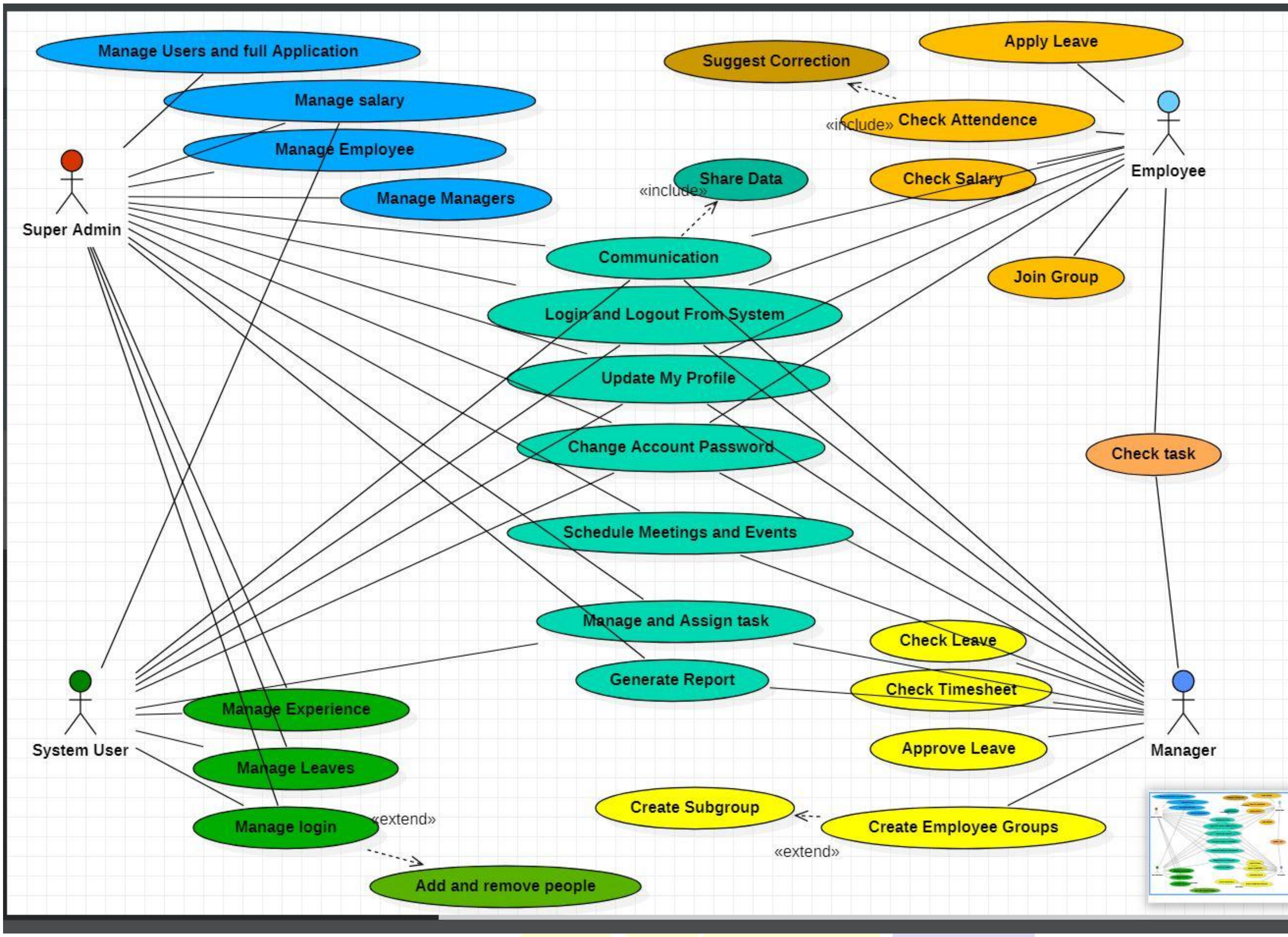
❖ **Super Admin**

- Create Account and Control entire Application
- Update Account(change Username, Password and Profile photo etc)
- Schedule and join Meetings
- Assign tasks for Managers
- Check Status
- Manage leave and Salary
- Communicate and share data

❖ **System User**

- Manage the Complete Application
- Manage Login and logout
- Add and Remove people
- Communicate and share data

Use Case Diagram->



Result: : Use case Diagram for Employe management system has been designed and code conversion has been demonstrated.

Thank you!!