
1. Employee Leave Management System

2. Perform Requirement Gathering Activity.

Functional Requirements:

1. **User Authentication:** Secure login for employees, managers, and administrators.
2. **Leave Request:** Employees can submit leave requests specifying type (sick, vacation, personal, etc.), duration, and reason.
3. **Leave Approval Workflow:** Managers receive leave requests, review, and approve/deny them.
4. **Leave Balances:** Display available leave balances for employees.
5. **Leave Types:** Support for multiple types of leaves with distinct policies and accrual rates.
6. **Calendar View:** Visual representation of leave schedules for better planning.
7. **Notifications:** Automated notifications for leave requests, approvals, and denials.
8. **Reporting:** Generate reports on leave trends, usage, and employee attendance.

Non-Functional Requirements:

1. **Scalability:** Ability to handle a growing number of users and leave requests without performance issues.
2. **Security:** Data encryption, role-based access control, and secure storage of sensitive information.
3. **User-Friendly Interface:** Intuitive design for easy navigation and use by employees and managers.
4. **Accessibility:** Compatibility across various devices and browsers for remote access.
5. **Reliability:** Minimal downtime and robust backup and recovery mechanisms.
6. **Compliance:** Adherence to labor laws, company policies, and regulatory requirements regarding leaves.
7. **Integration:** Compatibility with other HR systems (payroll, attendance) for seamless data exchange.

3. Create System Requirement Specification :-

3.1 Main Objects

The main objective of the project on Employee Leave Management System is to manage the details of Employee, Leave, Company, Payroll, Salary. It manages all the information about Employee, Leave Type, Salary, Employee. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the Employee, Leave, Leave Type, Company. It tracks all the details about the company, payroll, salary.

3.2 Introduction

The Employee Leave Management system in PHP is a comprehensive web-based application designed to streamline and automate the leave request and approval process within an organization. This system allows employees to submit leave requests online, specifying the type of leave, duration, and reason. The application employs PHP for server-side scripting, ensuring efficient data processing and seamless integration with a relational database for secure storage of leave records. The system facilitates real-time communication between employees and supervisors, enabling timely and transparent leave approvals or rejections. Additionally, it provides administrators with tools to generate leave reports, track employee attendance, and maintain accurate records for payroll purposes. Overall, the Employee Leave Management system in PHP enhances organizational efficiency, reduces paperwork, and ensures a systematic approach to managing employee leaves.

3.3 Abstract

This system allows employees to submit leave requests online, specifying the type of leave, duration, and reason. The application employs PHP for server-side scripting, ensuring efficient data processing and seamless integration with a relational database for secure storage of leave records. The system facilitates real-time communication between employees and supervisors, enabling timely and transparent leave approvals or rejections. Additionally, it provides administrators with tools to generate leave reports, track employee attendance, and maintain accurate records for payroll purposes. Overall, the Employee Leave Management system in PHP enhances organizational efficiency, reduces paperwork, and ensures a systematic approach to managing employee leaves.

3.4 Functional & Non Functional Requirements

Functional Requirements:

1. **User Authentication:** Secure login for employees, managers, and administrators.
2. **Leave Request:** Employees can submit leave requests specifying type (sick, vacation, personal, etc.), duration, and reason.
3. **Leave Approval Workflow:** Managers receive leave requests, review, and approve/deny them.
4. **Leave Balances:** Display available leave balances for employees.
5. **Leave Types:** Support for multiple types of leaves with distinct policies and accrual rates.
6. **Calendar View:** Visual representation of leave schedules for better planning.
7. **Notifications:** Automated notifications for leave requests, approvals, and denials.
8. **Reporting:** Generate reports on leave trends, usage, and employee attendance.

Non-Functional Requirements:

1. **Scalability:** Ability to handle a growing number of users and leave requests without performance issues.
2. **Security:** Data encryption, role-based access control, and secure storage of sensitive information.
3. **User-Friendly Interface:** Intuitive design for easy navigation and use by employees and managers.
4. **Accessibility:** Compatibility across various devices and browsers for remote access.
5. **Reliability:** Minimal downtime and robust backup and recovery mechanisms.
6. **Compliance:** Adherence to labor laws, company policies, and regulatory requirements regarding leaves.
7. **Integration:** Compatibility with other HR systems (payroll, attendance) for seamless data exchange.

3.5 Development

The project development approach for a digital platform that helps organizations save time and resources by automating the complex process of leave management can vary depending on the organization's needs and resources. However, a common approach is the Agile methodology, which involves iterative and incremental development, frequent feedback and collaboration with stakeholders, and continuous improvement.

The justification for using a digital platform for leave management is that it can save time and resources for the organization by automating the complex process of leave management. Manual leave management processes can be time-consuming and error-prone, leading to inefficiencies and potential compliance issues. A digital platform can streamline the process, reduce errors, ensure compliance, and provide real-time visibility into employee leave status.

Moreover, a digital platform can enhance employee satisfaction by providing an easy-to-use interface for submitting and tracking leave requests, and enabling employees to view their leave balances and usage history. This can lead to improved employee engagement and retention.

Hardware Requirements

Name of Components	Specification
Processor	i3 4 th generation or above
RAM	4GB RAM
Hard Disk	Minimum 80 GB

Software Requirements

Name of Components	Specification
Operating System	Windows or Linux
Software development Kit	visual studio code
Tools & languages	HTML, CSS, JAVASCRIPT, PHP, MySQL

3.6 Testing

Testing is an essential part of software development and is the process of identifying defects or errors in software products. It is important because it helps to ensure that the software is of high quality, functions as expected, and meets the requirements of the end-users. Testing can help identify bugs, security vulnerabilities, and other issues that may arise during the software development process.

There are different types of testing, and each type serves a unique purpose in the software development lifecycle. Here are some of the most common types of testing:

- **Unit Testing:** This is the testing of individual software components or modules to ensure that each one works correctly and as expected. Unit testing is usually automated, and it helps to identify defects early in the development cycle.

-
- **Integration Testing:** This type of testing is carried out to ensure that individual software components or modules work together as expected when integrated. Integration testing is critical for identifying defects that may arise from the interaction between components.
 - **System Testing:** This is the testing of the entire system or software product to ensure that it meets the requirements of the end-users. System testing can be manual or automated, and it covers a wide range of tests, including functional, performance, and security.

Test Case

A test case is a document that contains a set of test data, preconditions, expected outcomes, and post conditions that have been produced for a specific test scenario in order to check compliance against a given requirement.

The Test Case serves as the starting point for the test execution, and after applying a set of input values, the application has a definitive consequence and exits the system at some end point, which is also known as the execution post condition.

3.7 Deployment

1. Planning and Preparation

- **Define Objectives:** Identify what you want to achieve with the leave management system (e.g., streamline requests, improve tracking, etc.).
- **Select a Deployment Method:** Decide whether to host on-premises, use cloud-based services, or a hybrid approach.

2. System Configuration

- **Set Up User Roles:** Define different roles (e.g., admin, employee, manager) and their permissions.
- **Customize Features:** Tailor the system to your organization's policies (e.g., leave types, approval workflows).
- **Integrate with Existing Systems:** Ensure compatibility with payroll, HR, and calendar systems.

3. Testing

- **Conduct Unit Testing:** Test individual components for functionality.
- **User Acceptance Testing (UAT):** Engage a group of end-users to test the system in real-world scenarios and gather feedback.
- **Load Testing:** Assess how the system performs under high user loads.

4. Training

- **Develop Training Materials:** Create user manuals, video tutorials, and FAQs.
- **Conduct Training Sessions:** Organize workshops or webinars for different user roles to familiarize them with the system.

5. Deployment

- **Data Migration:** Transfer existing leave data to the new system, ensuring accuracy.
- **Go Live:** Launch the system, ensuring all users have access and credentials.
- **Monitor Performance:** Keep an eye on system performance and user engagement.

6. Post-Deployment Support

- **Gather Feedback:** Regularly check in with users for any issues or improvements.
- **Provide Ongoing Support:** Offer helpdesk support for troubleshooting.
- **Iterate and Improve:** Regularly update the system based on user feedback and changing organizational needs.

7. Compliance and Security

- **Ensure Data Compliance:** Make sure the system adheres to regulations (e.g., GDPR, HIPAA).
- **Implement Security Measures:** Protect sensitive employee data with encryption and access controls.

Conclusion

By following these steps, you can effectively deploy an employee leave management system that meets your organization's needs and enhances the leave management process. Regularly revisiting and refining the system will help maintain its effectiveness and user satisfaction.

3.8 Maintenance

1. Regular System Updates

- **Software Updates:** Regularly update the software to patch security vulnerabilities, improve functionality, and add new features.
- **Database Maintenance:** Optimize the database by performing regular backups, indexing, and cleanup of obsolete data.

2. Monitoring and Performance Management

- **System Performance Monitoring:** Use tools to monitor system performance metrics (e.g., response time, uptime).
- **Error Logging and Resolution:** Implement error logging to track and resolve issues promptly.

3. User Support and Training

- **Helpdesk Services:** Provide a dedicated support team to address user queries and issues.
- **Ongoing Training:** Offer refresher training sessions and updated materials whenever there are significant system changes or updates.

4. Feedback Mechanism

- **User Surveys:** Regularly solicit feedback from users to identify pain points and areas for improvement.
- **Suggestion Box:** Implement a mechanism for users to propose new features or report issues.

5. Compliance and Security Audits

- **Regular Compliance Checks:** Ensure the system adheres to relevant regulations (e.g., GDPR, labor laws).
- **Security Audits:** Conduct periodic security audits to identify and mitigate vulnerabilities.

6. Feature Enhancements

- **Roadmap for New Features:** Develop a plan for future features based on user feedback and technological advancements.
- **Pilot Testing:** Before rolling out new features, conduct pilot tests to gauge usability and effectiveness.

7. Data Management

- **Regular Data Audits:** Review and clean up data to ensure accuracy and relevance.
- **Archiving Policies:** Implement policies for archiving old data to maintain system performance.

8. Documentation

- **Maintain Up-to-Date Documentation:** Ensure user manuals, technical documents, and troubleshooting guides are current.
- **Change Logs:** Keep detailed logs of updates, changes, and enhancements for reference.

9. Scalability Planning

- **Assess Future Needs:** Regularly evaluate the system's capacity and scalability to accommodate growth.
- **Load Testing:** Periodically perform load testing to ensure the system can handle increased usage.

10. Community and Networking

- **Engage with User Communities:** Participate in forums and groups for feedback and shared experiences.
- **Stay Informed on Best Practices:** Keep abreast of industry standards and best practices for leave management systems.

3.9 Feasibility

Technical Feasibility: The proposed Employee Leave Management system demonstrates strong technical feasibility. The technology required for its implementation is readily available and compatible with existing HR infrastructure. The system's architecture allows for seamless integration with other HR systems, ensuring a cohesive and efficient operation. The scalability and adaptability of the technology also make it suitable for future enhancements and updates. Additionally, the user-friendly interfaces and accessibility features contribute to a smooth transition and widespread adoption by both employees and managers.

Economic Feasibility: From an economic perspective, the implementation of the Employee Leave Management system is cost-effective in the long run. While there may be initial investment costs for development and integration, the automated processes reduce manual labor expenses associated with leave management. The reduction in errors and increased efficiency contribute to operational savings over time. The scalability of the system further ensures that it can grow with the organization without incurring significant additional costs, making it a financially sustainable solution.

Operational Feasibility: The operational feasibility of the Employee Leave Management system is evident in its potential to streamline and enhance existing HR processes. The automated workflow for leave requests and approvals minimizes administrative burdens, allowing HR personnel to focus on more strategic tasks. The system's real-time notifications and transparent communication channels improve overall operational efficiency by keeping all stakeholders informed about leave statuses. The integration with payroll and attendance tracking systems ensures a holistic approach to human resource management. Furthermore, the system's commitment to legal compliance ensures that the organization operates within the framework of relevant regulations, reducing the risk of legal issues. Overall, the operational feasibility of the proposed system contributes to a more efficient, transparent, and compliant leave management process.

3.10 Content

1. Overview

- **Introduction to the System:** Briefly explain the purpose and benefits of the leave management system.
- **Objectives:** Outline what the system aims to achieve (e.g., streamline leave requests, improve tracking).

2. User Roles and Access

- **Employee Role:** Describe the functionalities available to regular employees (e.g., request leave, view leave balance).
- **Manager Role:** Detail the additional responsibilities of managers (e.g., approve/reject leave requests, manage team leave schedules).
- **Administrator Role:** Explain the administrative tasks (e.g., configuring leave types, generating reports).

3. Leave Types

- **Types of Leave Available:** List and define various leave categories (e.g., vacation, sick leave, parental leave, unpaid leave).
- **Policies:** Summarize company policies related to each leave type, including accrual rates and limits.

4. Requesting Leave

- **Step-by-Step Guide:** Provide detailed instructions on how to request leave, including screenshots if possible.
- **Required Information:** List information needed for leave requests (e.g., leave type, duration, reason).
- **Submission Process:** Explain how and when requests will be reviewed and responded to.

5. Managing Leave Requests

- **For Employees:** Guide on how to view the status of leave requests and modify or cancel them.
- **For Managers:** Instructions on how to review, approve, or deny requests, including timelines for responses.

6. Leave Balances

- **Checking Leave Balance:** Explain how employees can view their available leave balances.
- **Accrual Details:** Describe how leave is accrued and any relevant calculations or policies

7. Notifications and Alerts

- **Email Notifications:** Outline the types of notifications users will receive (e.g., request confirmations, approval alerts).
- **Reminders:** Explain any reminder systems in place for upcoming leaves or deadlines.

8. Reporting and Analytics

- **Available Reports:** List the types of reports that can be generated (e.g., leave usage reports, trends).
- **Accessing Reports:** Instructions for how users can access and interpret these reports.

9. Frequently Asked Questions (FAQ)

- **Common Queries:** Address common questions regarding leave policies, the system's functionality, and troubleshooting tips.

10. Support and Resources

- **Helpdesk Information:** Provide contact details for technical support and HR assistance.
- **User Guides and Manuals:** Link to any additional training materials or user manuals.

11. Security and Privacy

- **Data Protection Policies:** Briefly explain how employee data is protected and the measures in place to ensure confidentiality.

12. Feedback Mechanism

- **How to Provide Feedback:** Describe how users can share their experiences or suggest improvements for the system.

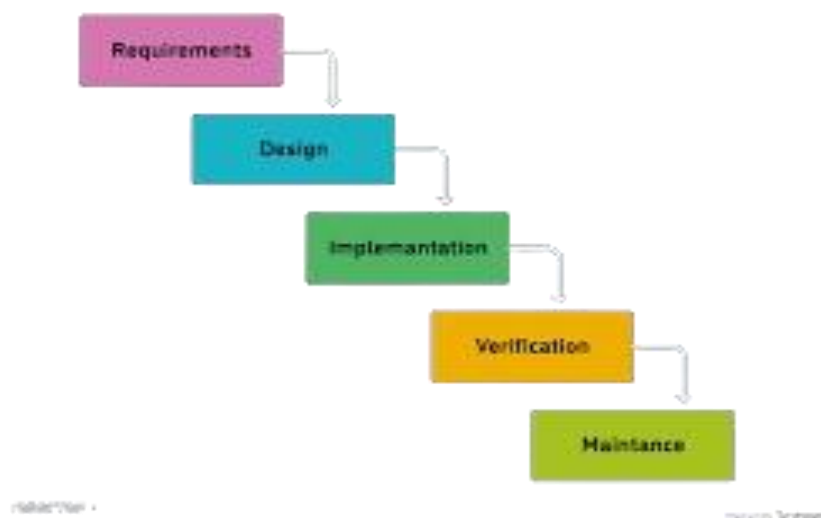
4. Select and Finalize Software Development Process Model and Prepare Report on Process Model. Also prepare Project Plan with Time-line Chart.

4.1 Process Model

Waterfall Model

The waterfall model is a cascading software development life cycle (SDLC) that shows the steps of analysis, planning, realization, testing, implementation, and support as a flow in the development process. Every step of this SDLC process must be carried out step by step. The waterfall technique requires thorough documentation. This SDLC model contains defined properties for each phase.

Many people consider the waterfall life cycle model to be among the best techniques for handling challenging projects. This approach makes it possible to prevent a great deal of mistakes that could arise from having inadequate project management. On the other hand, extensive documentation development is the outcome. The product is better off with more documentation for potential future developers, yet the documentation process is time-consuming.



You will discover the benefits and drawbacks of the Waterfall SDLC methodology.

Benefits:

- Simple to use and understand
- The process's rigor simplifies management because every stage has a defined goal and a process evaluation.

The stages of development are sequential and best suited for small to medium-sized projects with clear requirements.

- It is easy to pinpoint crucial elements within the development cycle.
- Basic job prioritization and classification

Drawbacks:

It is not advised for intricate, object-oriented applications. It is challenging to assess the stage's development while it is still in production. Integration happens in the end, therefore there's no way to know about the problem beforehand.

4.2 Spiral Model

The Spiral Model is a software development process that combines iterative development with the systematic aspects of project management. It emphasizes risk assessment and allows for incremental releases of a system. Here's how you can apply the Spiral Model to develop an Employee Leave Management System:

Spiral Model Phases

1. Planning Phase

- **Objectives:**
 - Define project goals, scope, and deliverables.
 - Identify key stakeholders (HR, employees, managers).
- **Activities:**
 - Gather initial requirements.
 - Determine constraints (budget, timeline).
 - Establish a preliminary risk assessment.

2. Risk Analysis Phase

- **Objectives:**
 - ❖ Identify potential risks and their impacts on the project.
- **Activities:**
 - ❖ Conduct risk analysis workshops with stakeholders.
 - ❖ Develop mitigation strategies for identified risks (e.g., technology risks, user resistance).
 - ❖ Document findings and review with the team.

3. Engineering Phase

- **Objectives:**
 - ❖ Develop prototypes or initial versions of the system.
- **Activities:**
 - ❖ **Iteration 1:**
 - Design the architecture.
 - Develop the basic functionalities (e.g., leave request submission).
 - Create a user interface prototype.

- ❖ **Iteration 2:**

- Implement additional features (e.g., leave approval workflow).
- Refine user interface based on feedback.

- ❖ **Iteration 3:**

- Add reporting and analytics functionalities.
- Improve system performance and security measures.

4. Evaluation Phase

- **Objectives:**

- ❖ Validate and evaluate the system against requirements.

- **Activities:**

- ❖ Conduct user acceptance testing (UAT) for each iteration.
- ❖ Gather feedback from stakeholders.
- ❖ Identify areas for improvement.
- ❖ Document lessons learned and adjust the plan for the next iteration.

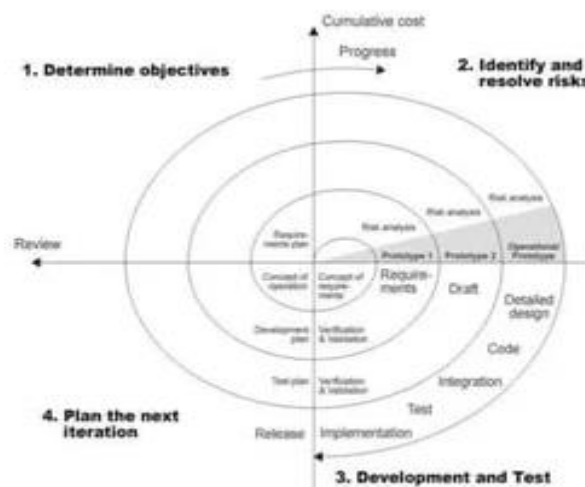
5. Delivery Phase

- **Objectives:**

- ❖ Prepare for deployment of the system.

- **Activities:**

- ❖ Finalize documentation (user manuals, technical specs).
- ❖ Train users (employees and managers).
- ❖ Prepare the system for deployment.
- ❖ Launch the system for all users.



4.3 Project Plan

1. Project Overview

- **Project Name:** Employee Leave Management System
- **Project Manager:** [Name]
- **Start Date:** [Start Date]
- **End Date:** [End Date]
- **Budget:** [Budget]

2. Project Objectives

- Streamline the leave request and approval process.
- Improve tracking of employee leave balances.
- Ensure compliance with leave policies and regulations.
- Enhance user experience for employees and managers.

3. Scope of Work

- **Inclusions:**
 - ❖ User role management (employee, manager, admin).
 - ❖ Leave request submission and approval workflows.
 - ❖ Leave balance tracking and reporting.
 - ❖ Notifications and alerts.
- **Exclusions:**
 - ❖ Integration with payroll systems (if not required).
 - ❖ Mobile application development (if not in the initial phase).

4. Stakeholders

- **Internal:**
 - ❖ HR Department
 - ❖ IT Department
 - ❖ Employees
 - ❖ Management
- **External** (if applicable):
 - ❖ Software Vendors
 - ❖ Consultants

5. Resource Allocation

- **Team Members:**
 - ❖ Project Manager: [Name]
 - ❖ Developers: [Names]
 - ❖ QA Testers: [Names]
 - ❖ HR Representatives: [Names]
 - ❖ IT Support: [Names]
- **Tools and Technologies:**
 - ❖ Development Platform: [e.g., .NET, Java]
 - ❖ Database: [e.g., MySQL, PostgreSQL]
 - ❖ Project Management: [e.g., Jira, Trello]

6. Risk Management

- **Potential Risks:**
 - ❖ Delays in requirements gathering.
 - ❖ Integration issues with existing systems.
 - ❖ User resistance to adopting the new system.
- **Mitigation Strategies:**
 - ❖ Regular check-ins with stakeholders.
 - ❖ Clear communication and training.
 - ❖ Pilot testing with a small user group before full deployment.

7. Monitoring and Evaluation

- **Key Performance Indicators (KPIs):**
 - ❖ User satisfaction ratings post-deployment.
 - ❖ Reduction in time taken for leave requests to be processed.
 - ❖ Number of leave-related errors reported.

4.4 Time-line Chart

4.4 Time-line Chart

5. Draw System Flow Chart, Decision Tree And Decision Table

5.1 What is Decision Tree?

A Decision Tree is a type of supervised machine learning used to categorize or make predictions based on how a previous set of questions were answered.

The model is a form of supervised learning, meaning that the model is trained and tested on a set of data that contains the desired categorization.

1.Start

- Receive Leave Request
- Check Employee Eligibility

2.Employee Eligibility

- Is the employee eligible for leave?
 - Yes: Proceed to Leave Type
 - No: Reject Leave Request (Notify Employee)

3. Leave Type

- What type of leave is requested?
 - Annual Leave: Proceed to Annual Leave Policy
 - Sick Leave: Proceed to Sick Leave Policy
 - Unpaid Leave: Proceed to Unpaid Leave Policy
 - Other: Review and assess

4. Annual Leave Policy

- Is the leave within the allowed limit?
 - Yes: Approve Leave
 - No: Reject Leave Request (Notify Employee)

5. Sick Leave Policy

- Is medical documentation provided?
 - Yes: Approve Leave
 - No: Request Documentation (Hold Request)

6. Unpaid Leave Policy

- Is the reason valid?
 - Yes: Approve Leave
 - No: Reject Leave Request (Notify Employee)


7.End

5.2 Decision Table

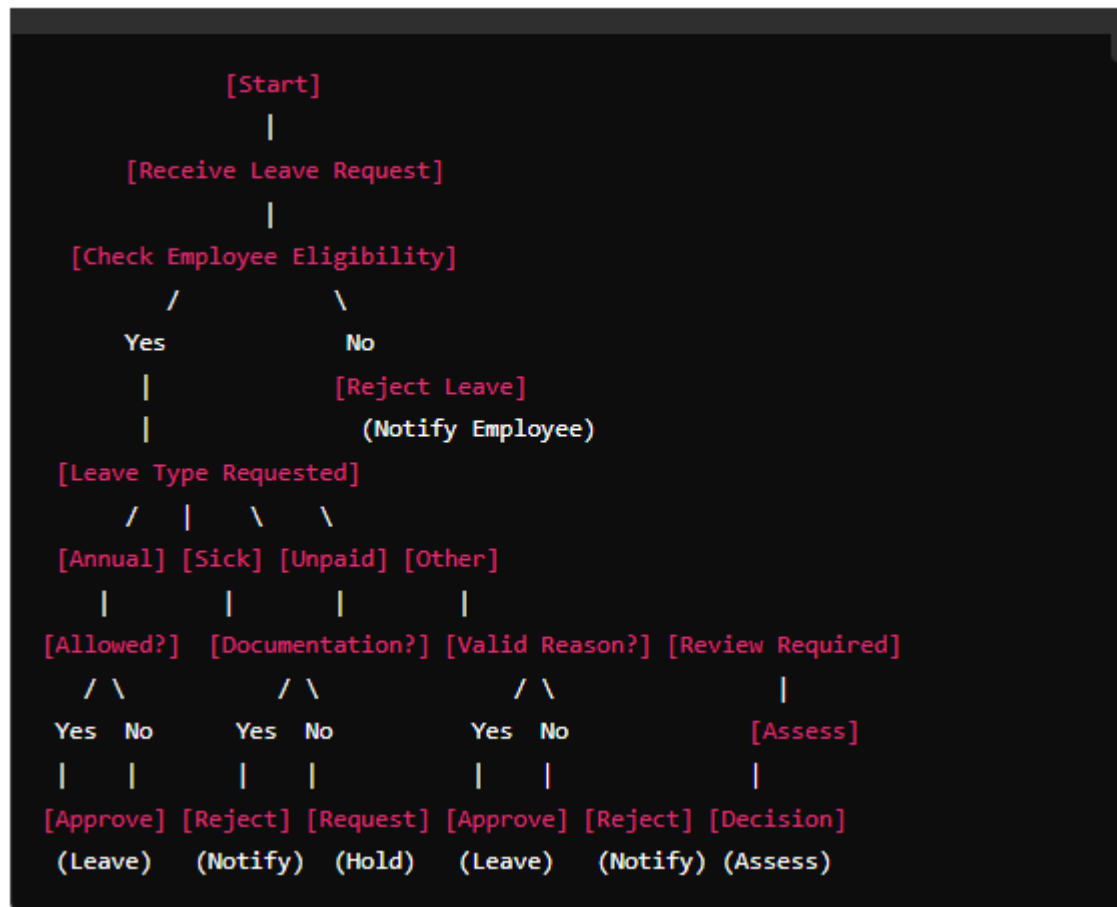
A decision table is the tabular representation of several input values, cases, rules, and test condition.

Decision Table Testing is a software testing methodology used to test system behavior for various input combinations.

The Decision table is also called a Cause-effect table, as the causes and effects for comprehensive test coverage are captured in this table.

Condition	Annual Leave	Sick Leave	Casual Leave	Balance Check	Approval Status	Action
1. Employee Status	Active	Active	Active	Sufficient	Approved	Leave granted
2. Employee Status	Active	Active	Active	Insufficient	Denied	Leave request denied
3. Employee Status	Inactive	Inactive	Inactive	Any	Denied	Leave request denied
4. Employee Status	Active	Active	Active	Sufficient	Pending	Awaiting manager approval
5. Employee Status	Active	Active	Active	Insufficient	Pending	Awaiting manager approval
6. Leave Type	Annual Leave	Sick Leave	Casual Leave	Any	Approved	Leave granted
7. Leave Type	Annual Leave	Sick Leave	Casual Leave	Any	Denied	Leave request denied
8. Leave Type	Annual Leave	Sick Leave	Casual Leave	Sufficient	Pending	Awaiting manager approval
9. Notice Period	> 3 days	Any	Any	Any	Approved	Leave granted
10. Notice Period	≤ 3 days	Any	Any 	Any	Denied	Leave request denied

5.3 Figure Of Decision Tree



5.4 What Is Flowchart ?

A flowchart is a diagram that depicts a process, system or computer algorithm. They are widely used in multiple fields to document, study, plan, improve and communicate often complex processes in clear, easy-to-understand diagrams.

5.5 Diagram:- FlowChart.

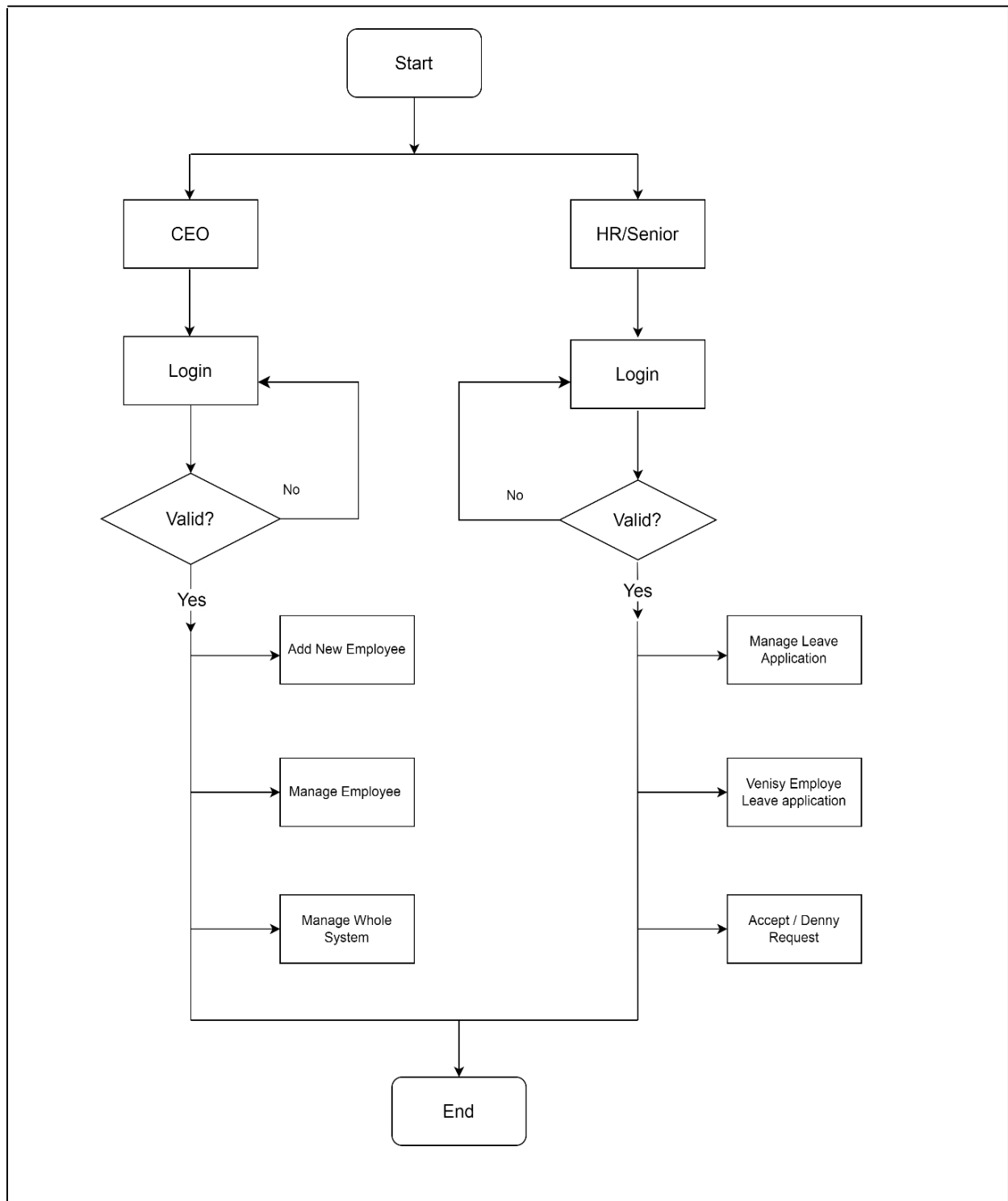
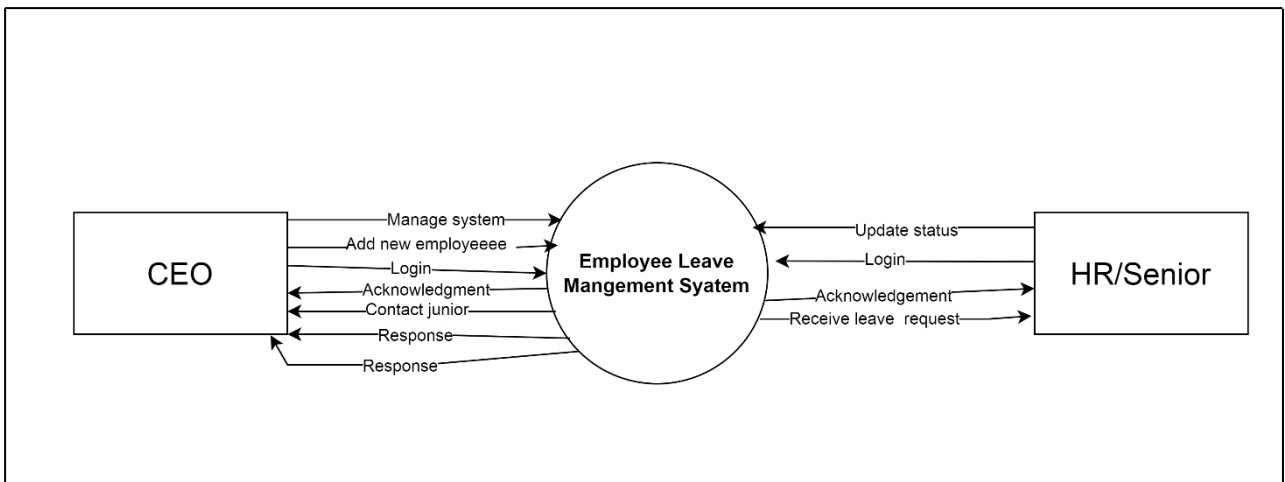


Fig:- Flow Chart

6. Draw Data Flow Diagram Up to 2nd Level

DFD Level-0



Level 1 DFD's

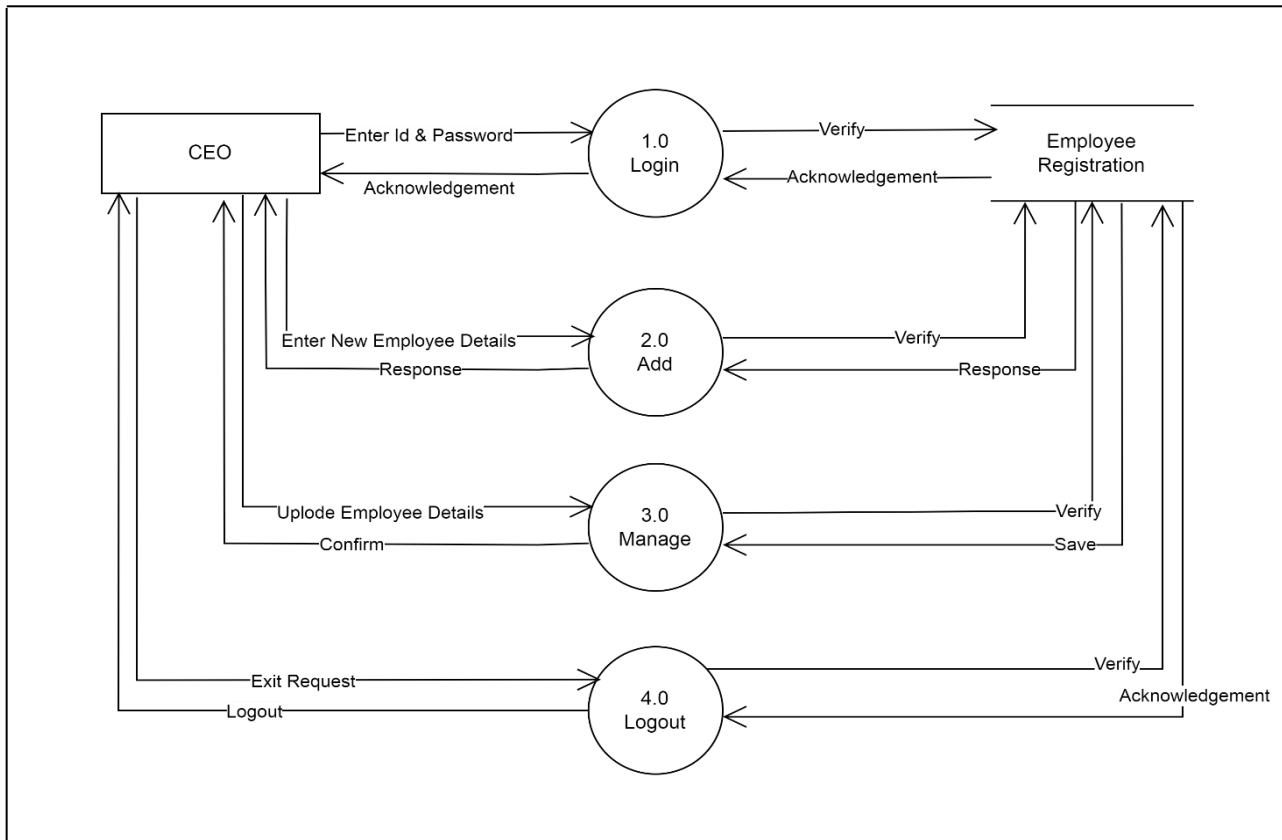


Fig:- Admin & Receiver Data Flow Diagram-1

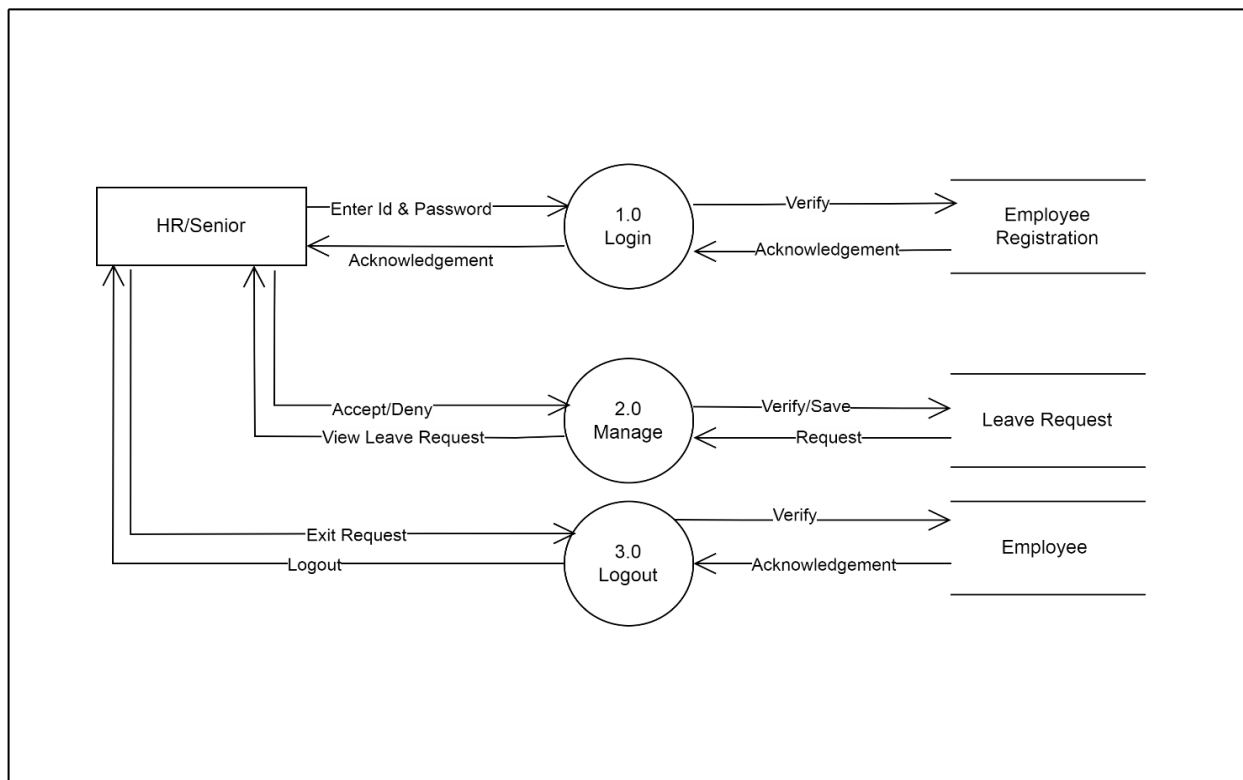





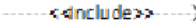

Fig:- User (Citizen) Data Flow Diagram -1

7. Identify And Write Use Case Scenarios . Also Draw The Use Case Diagram.

7.1 What Is Use-Case Diagram?

A Use-Case Diagram is a Type of Behavioral diagram used in software engineering to visually represent the interaction between users and a system. It focuses on the functional requirements of a system and describes the high-level functionality and the relationships between actors and use cases.

7.2 Symbol Of Use-Case Diagram

NO	FIGURE	NAME	EXPLANATION
1		Actor	Functionality provided by the system as units that exchange messages between units or actors, is usually expressed using the verb at the beginning of the phrase name use case.
2		Use Case	People, processes, or other systems that interact with information systems that will be created outside the information system that will be created themselves.
3		Association	Communication between actors and use cases that participate in use cases <u>has interaction with</u> actors.
4		Include	Additional use case relations to a use case and use case that are added require this use case to carry out its functions.
5		Extend	Additional use case relations to a use case and use case that are added can stand alone even without additional use cases.

7.3 Use Case Diagram

Admin Side

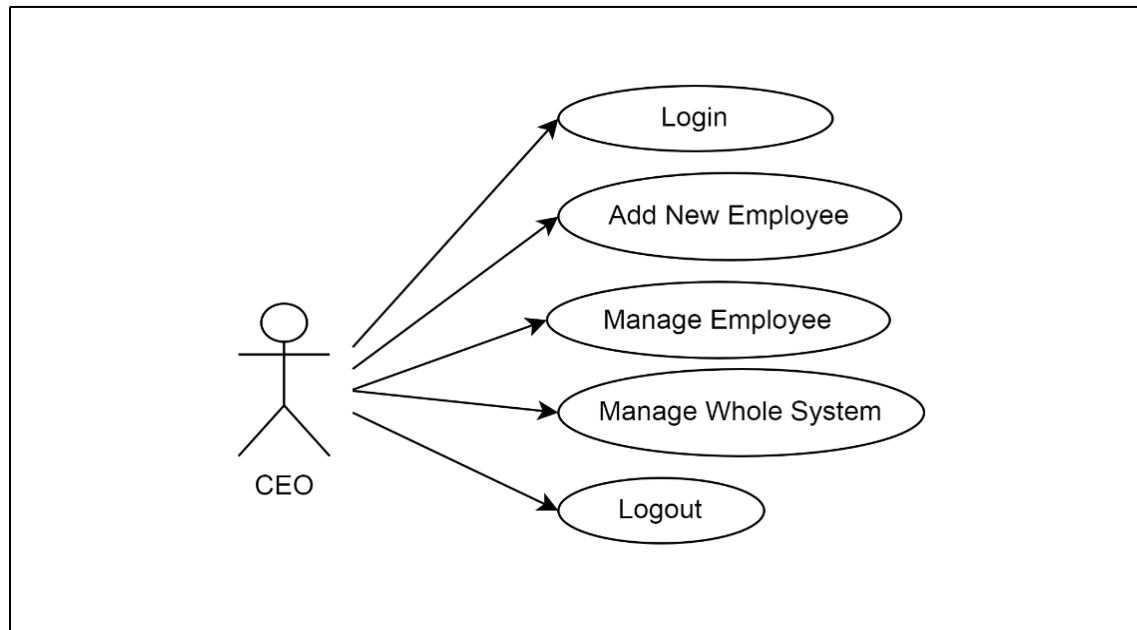


Fig: Admin Use case Diagram

User Side

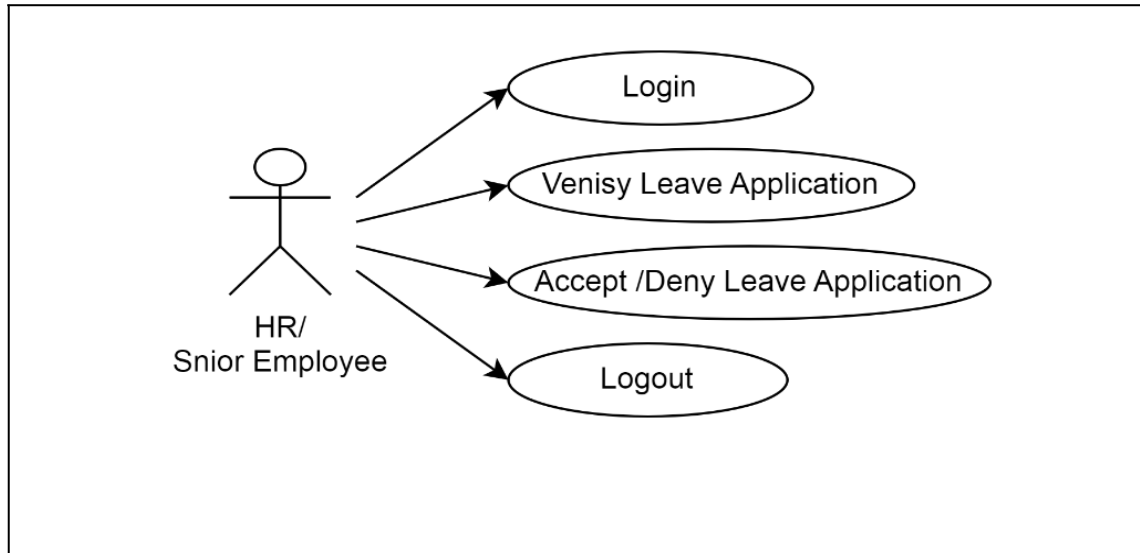


Fig :-User (Citizen) Use case Diagram

8. Create UML Diagram:-

7.1 Class Diagram

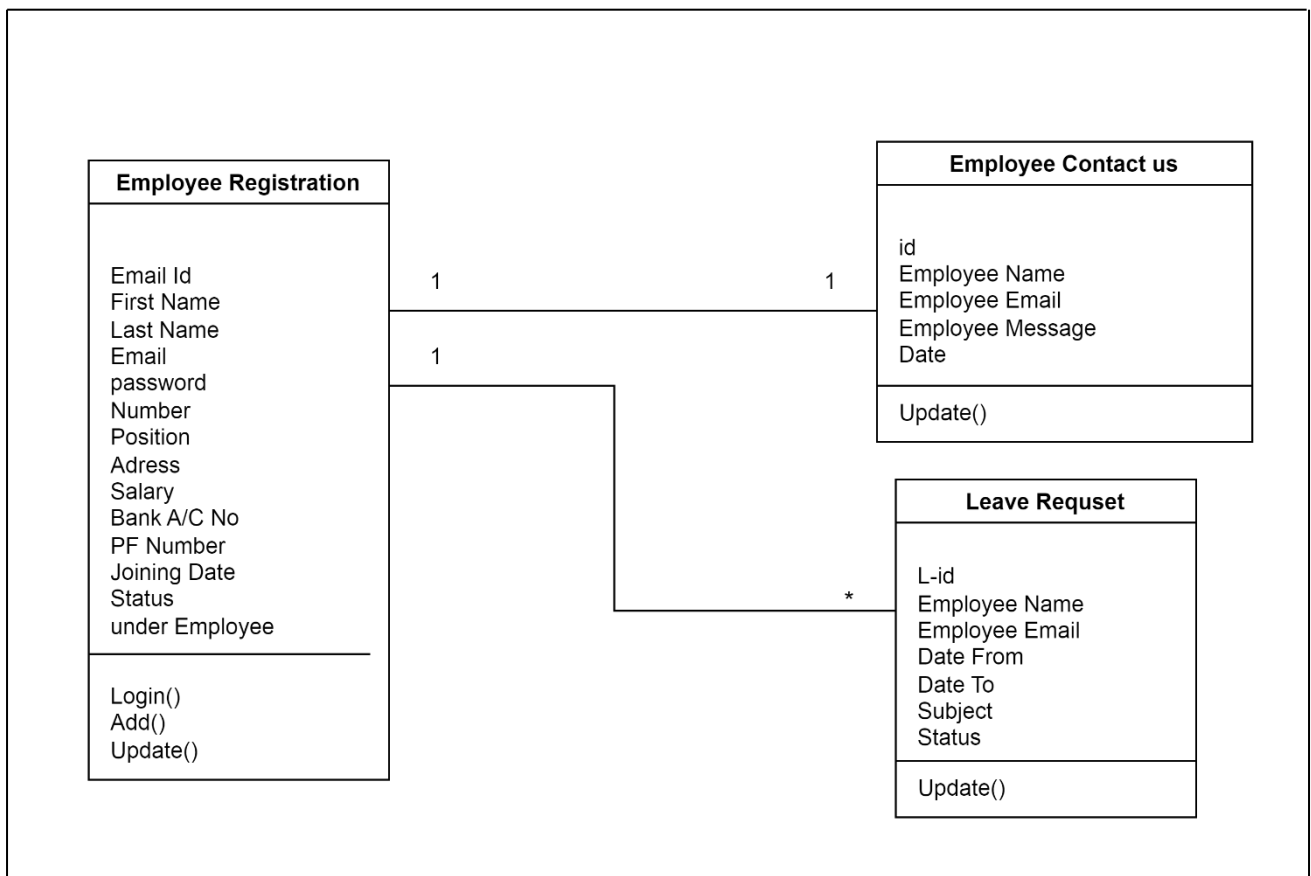
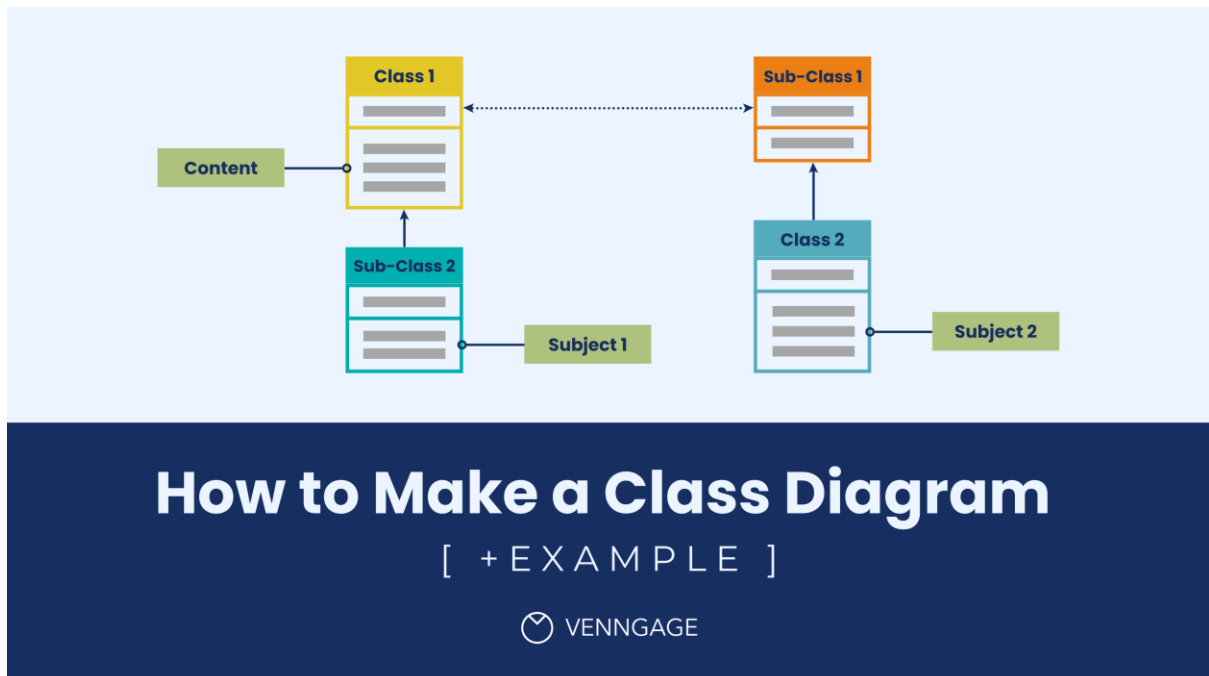


Fig:- Class Diagram

7.1.1 Symbols of Class Diagram



7.2 Sequence Diagram

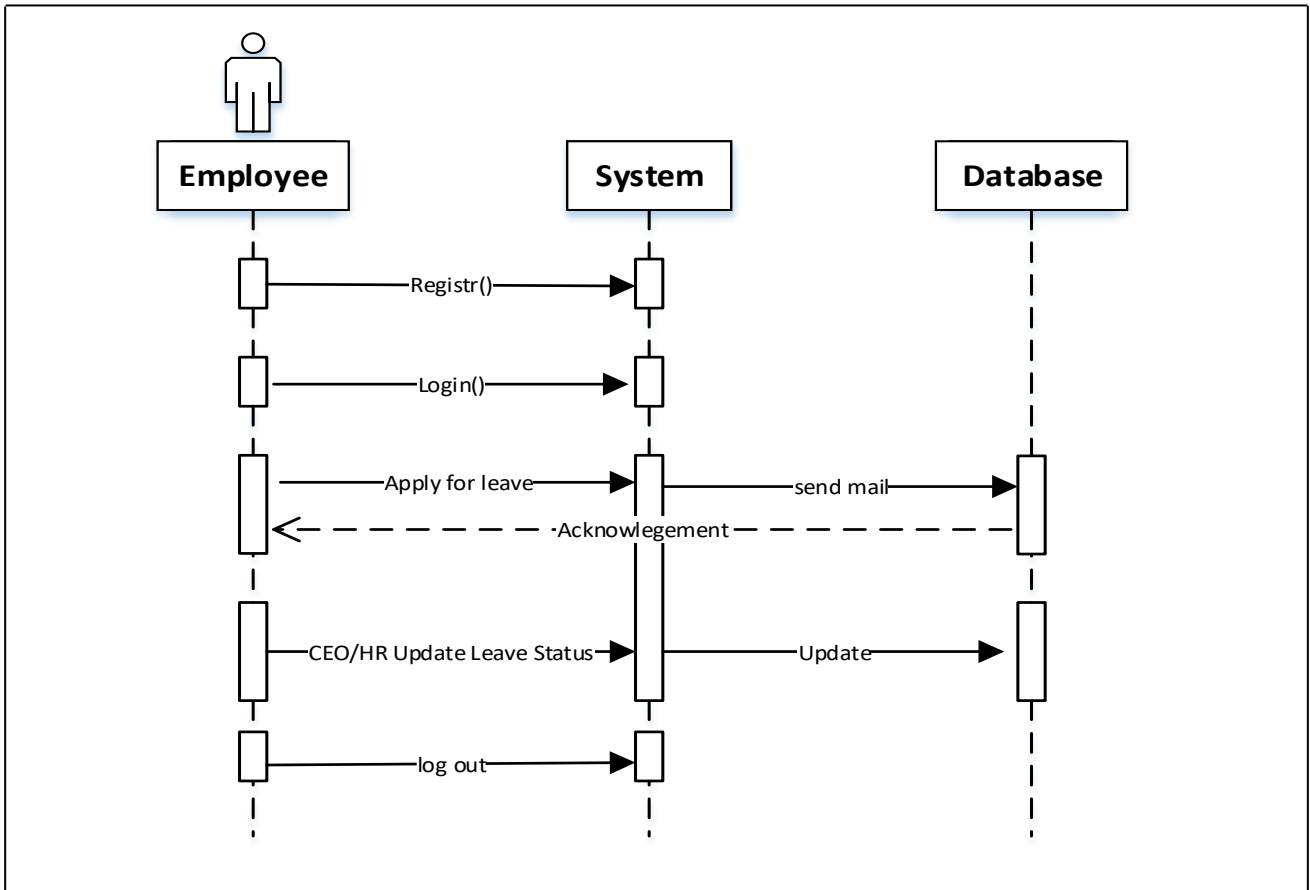
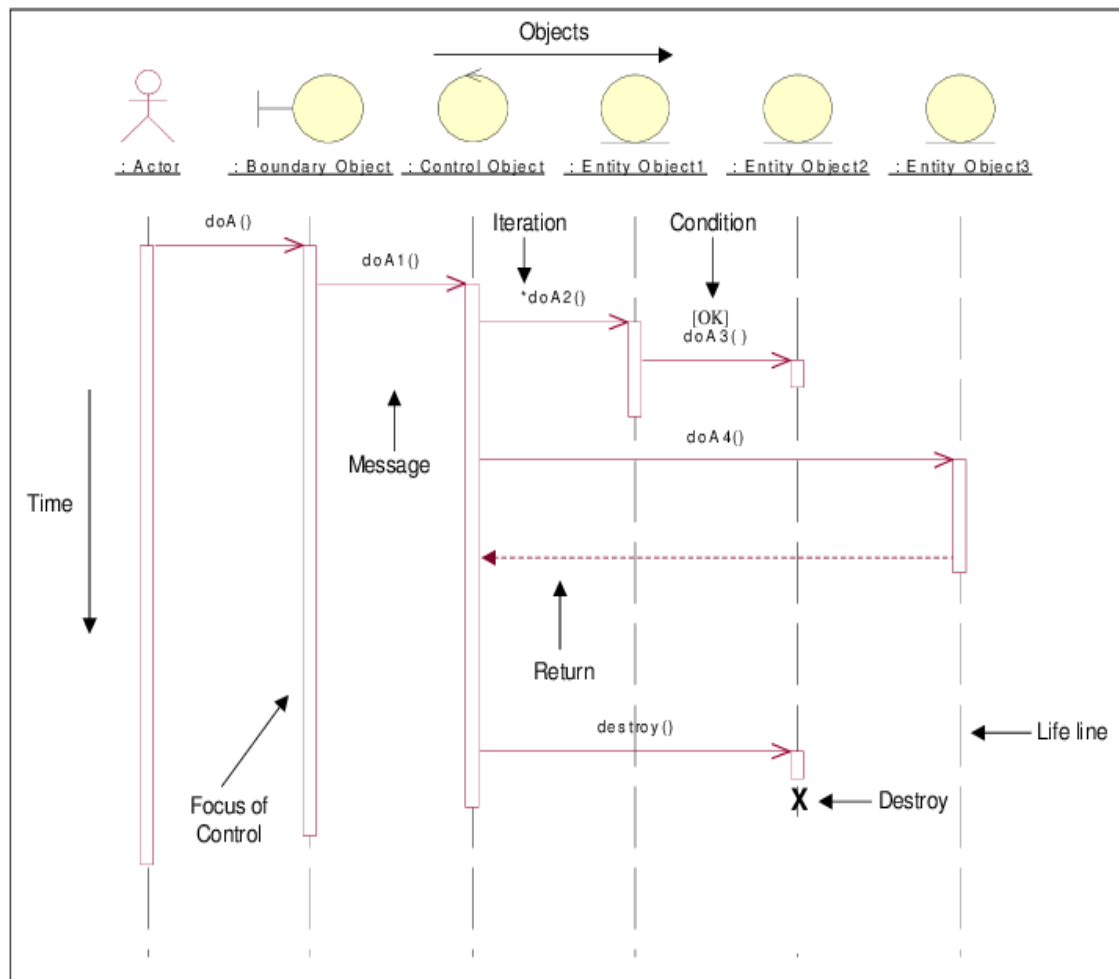


Fig:- Sequence Diagram

7.2.1 Symbols of Sequence Diagram



7.3Activity Diagram

Admin :-

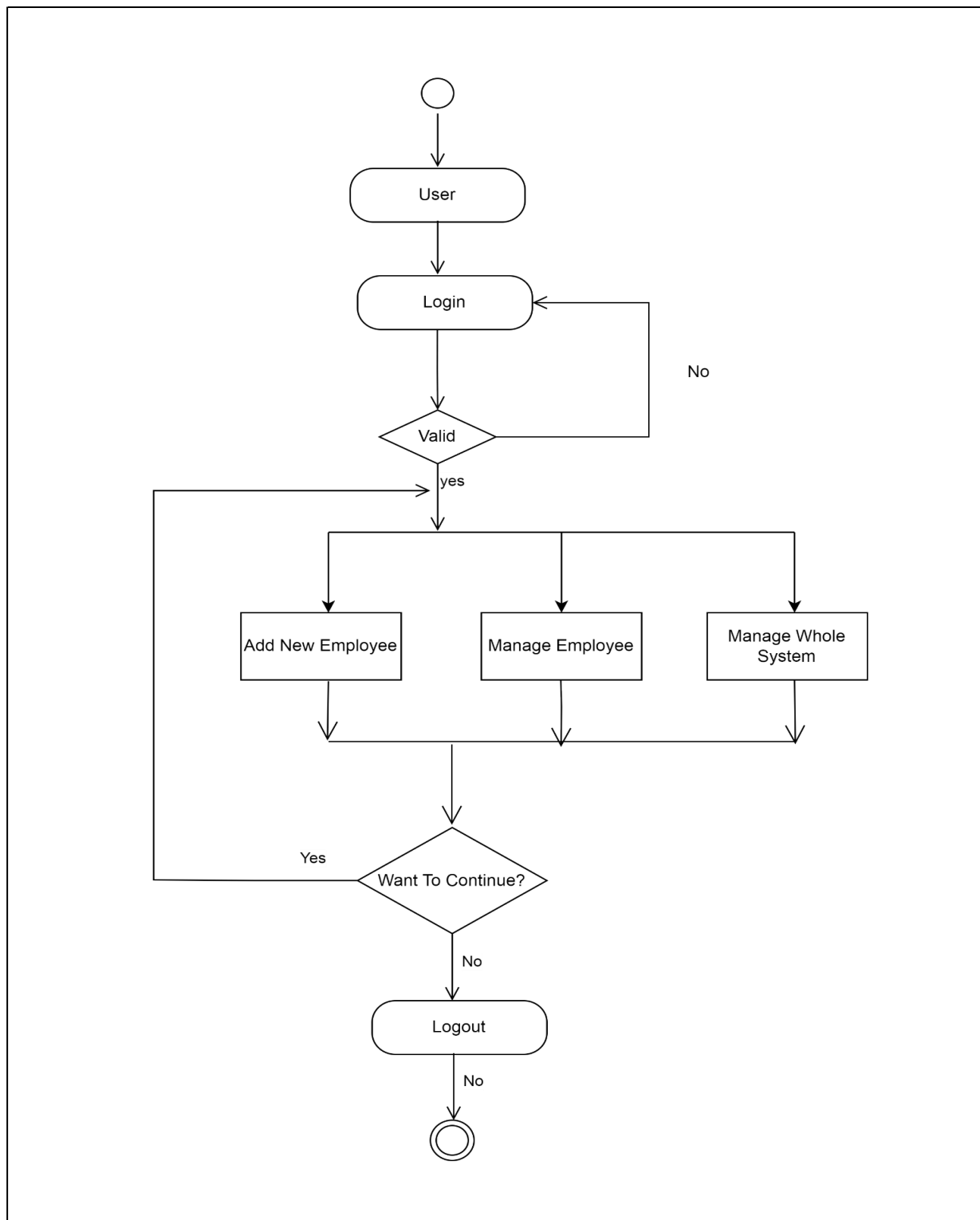


Fig:- Admin Activity Diagram

User:-

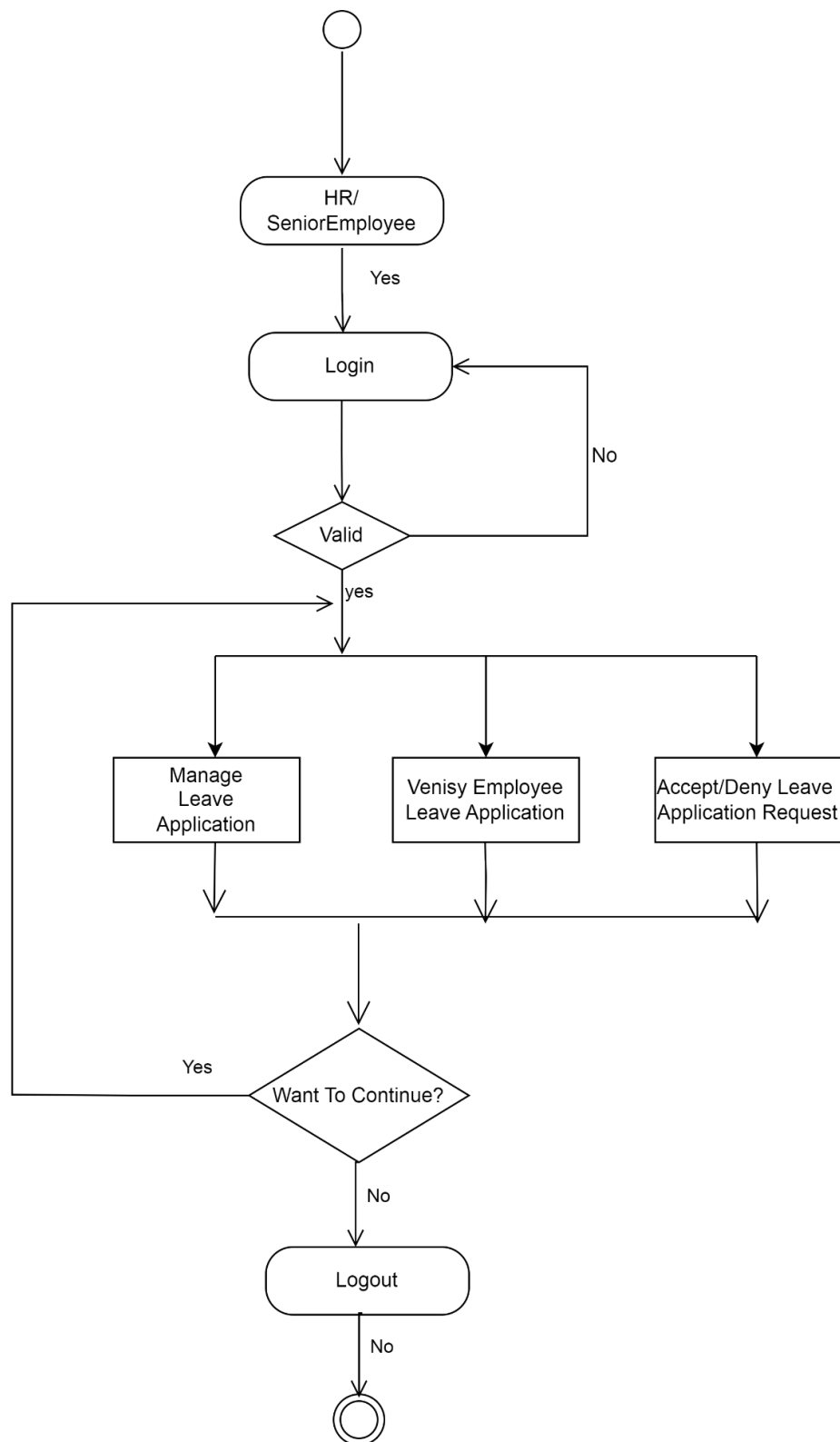




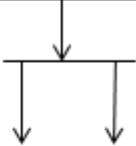
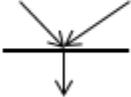



Fig:- User Activity Diagram

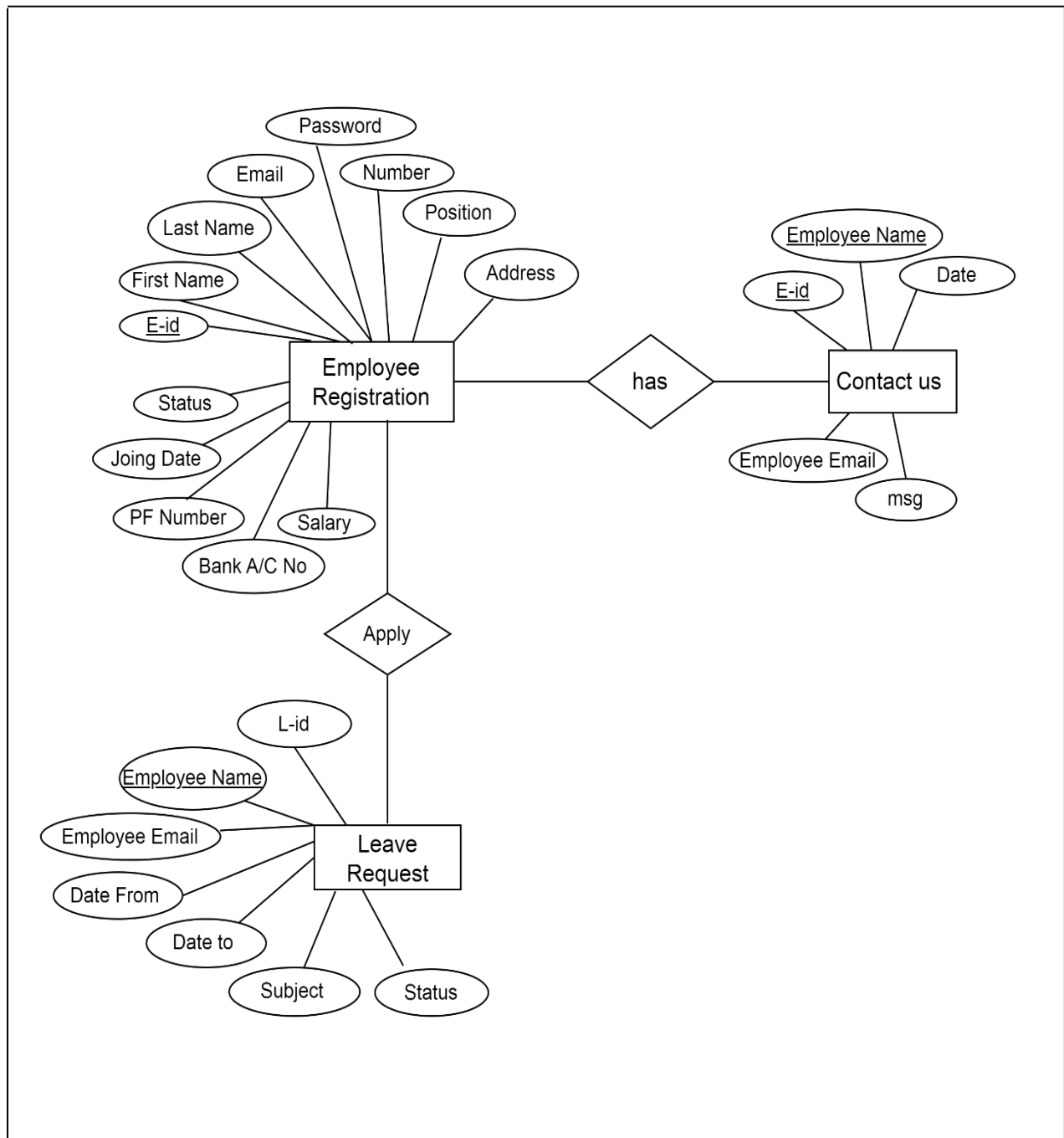
7.3.1 Symbols of Activity Diagram

Sr. No	Name	Symbol
1.	Start Node	
2.	Action State	
3.	Control Flow	
4.	Decision Node	
5.	Fork	
6.	Join	
7.	End State	

7.4 Package Diagram

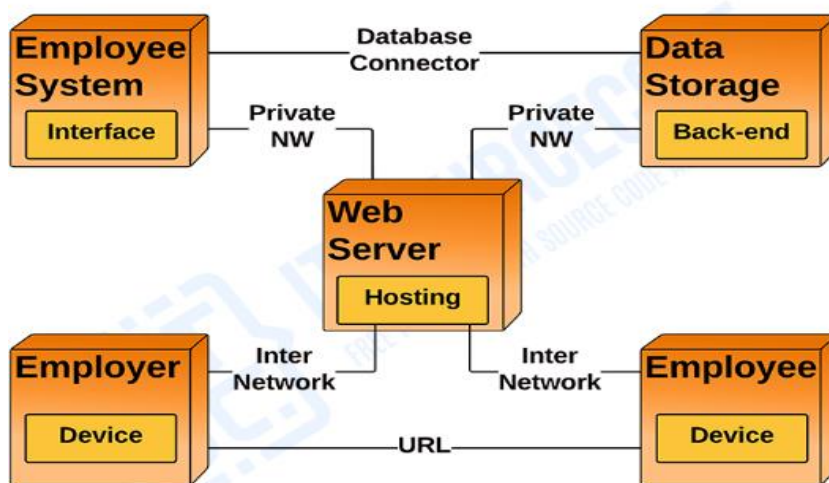
7.5 E-R Diagram

Entity Relationship Diagram



7.6 Deployment Diagram

EMPLOYEE MANAGEMENT SYSTEM



DEPLOYMENT DIAGRAM